

FY 2012 Draft 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 13, 2012

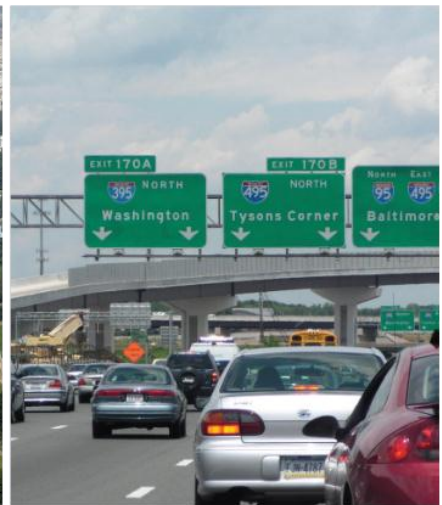


Table of Contents

1	Introduction	1-1
2	Meetings and Technical Assistance (Task Order 1).....	2-1
2.1	Cube Assignments.....	2-1
2.2	WMATA Modeling Process	2-5
2.2.1	Transit Access.....	2-5
2.2.2	Mode Choice	2-6
2.2.3	Points of Interest.....	2-8
3	Improving Mode Choice Modeling (Task Order 2).....	3-1
3.1	Structure of the Mode Choice Model	3-1
3.1.1	Pedestrian Environment Factor	3-4
3.2	Transit Path-Building Software or Procedures.....	3-9
3.3	Transit Assignment.....	3-12
4	Transit Line Summaries and Access Considerations (Task Order 3)	4-1
4.1	LineSum.....	4-1
4.2	Station Access Coding	4-3
5	Model Performance Enhancements (Task Order 4)	5-1
5.1	Model Processes Identified for Distributed Computing	5-1
5.2	Model Results Validation	5-11
5.3	Benefits Quantification	5-13
5.4	Conclusions and Recommendations.....	5-14
6	Begin Converting TRNBUILD to PT (Task Order 5)	6-1
6.1	Displaying Aggregate Ridership Data	6-1
6.2	PT Conversion Analysis	6-3
6.3	Convert TRNBUILD Network to PT	6-3
6.3.1	PT Network.....	6-4
6.3.2	PT Transit Lines	6-5
6.3.3	PT Access Links.....	6-6
6.3.4	PT Path Building	6-11
6.4	Generating PT Network.....	6-15

6.5	Estimated Conversion Effort	6-20
6.5.1	Proposed Work Plan.....	6-21
7	Non-Resident Metrorail Trips (Task Order 6)	7-1
7.1	Base Tables.....	7-1
7.2	Forecasting Theory.....	7-11
7.2.1	Existing Metrorail Stations	7-12
7.2.2	New Stations	7-15
7.3	Forecasting Script.....	7-18
7.4	Test Results	7-22
7.5	Model Integration	7-31
8	Air Passenger Model (Task Order 6)	8-1
8.1	Step One: Estimating Total Air Passengers	8-2
8.2	Step Two: Allocating Ground Side Trip Ends.....	8-5
8.3	Step Three: Mode Choice Model	8-13
8.4	Time of Day	8-19
8.5	Implementing the Model	8-19
8.6	Sources and References	8-22
9	Summary of Recommendations.....	9-1
10	Appendix	10-1
10.1	Script to Forecast Non-Resident Metrorail Trips	10-1
10.2	Airport Passenger Script	10-14

List of Figures

Figure 2-1:	MWCOG Cube Assignment Convergence Test	2-2
Figure 2-2:	Frank Wolfe Relative Gap Results from a TransForM Assignment.....	2-3
Figure 2-3:	Bi-Conjugate Frank Wolfe Relative Gap Results from a TransForM Assignment	2-4
Figure 2-4:	Origin-Based Relative Gap Results from a TransForM Assignment.....	2-4
Figure 2-5:	TransitAccess Inputs and Outputs	2-6
Figure 2-6:	ModeChoice Inputs and Outputs.....	2-7
Figure 3-1:	Mode Choice Nesting Structure.....	3-2
Figure 3-2:	Pedestrian Environment Factors near Metrorail Lines.....	3-7
Figure 3-3:	Pedestrian Environment Factors at a Regional Level	3-8
Figure 4-1:	Current Transit Access Links	4-4
Figure 4-2:	Proposed Change in Transit Access Coding	4-5

Figure 5-1: Model Processes Identified for Further Distributed Computing	5-3
Figure 5-2: Highway Skims Procedure with Distributed Computing (both MDP and IDP)	5-4
Figure 5-3: Transit Skims Procedure with Distributed Computing	5-5
Figure 5-4: Trip Distribution Procedure with Multi-Step Distributed Processing.....	5-7
Figure 5-5: Mode Choice Procedure with Distributed Computing	5-8
Figure 5-6: Highway Assignment Procedure with Distributed Computing	5-9
Figure 5-7: Transit Assignment Procedure with Distributed Computing.....	5-10
Figure 5-8: Comparison of Relative Gap for Parallel and Serial Model Executions (AM Peak)	5-11
Figure 5-9: Comparison of Relative Gap for Parallel and Serial Model Executions (PM Peak).....	5-12
Figure 5-10: Comparison of Relative Gap for Parallel and Serial Model Executions (Mid-day)	5-12
Figure 5-11: Comparison of Relative Gap for Parallel and Serial Model Executions (Night)	5-13
Figure 6-1: Cube Script to Convert Transit-Only Network	6-5
Figure 6-2: Cube Script to Convert TRNBUILD Line Files to PT Format.....	6-6
Figure 6-3: Cube Script to Convert TRNBUILD Access Links to PT Format.....	6-6
Figure 6-4: Sample PT System File	6-12
Figure 6-5: Sample PT Factor File.....	6-12
Figure 6-6: Public Transport Skim Script	6-14
Figure 6-7: Script for Generating PT Walk Access Links.....	6-16
Figure 6-8: Script for Generating PT Drive Access Links	6-17
Figure 6-9: Script for Manually Selecting Access Points	6-18
Figure 6-10: Transit Links Generated by PT	6-19
Figure 7-1: Metrorail Line Segments.....	7-8
Figure 7-2: Non-Resident B+A vs. Employment Density	7-17
Figure 7-3: Boarding Share.....	7-18
Figure 7-4: Model Application Flowchart	7-21
Figure 8-1: Visitor Factor by TAZ.....	8-7
Figure 8-2: BWI Marshall (BWI) Airport Market Factor	8-9
Figure 8-3: Washington Dulles (IAD) Airport Market Factor.....	8-10
Figure 8-4: Reagan National (DCA) Airport Market Factor	8-11
Figure 8-5: Air Passenger Model Structure	8-13
Figure 8-6: Air Passenger Model Flowchart.....	8-21

List of Tables

Table 3-1: Structure of a Drive-Access Transit Path using TRNBUILD.....	3-10
Table 3-2: Structure of a Drive-Access Transit Path using PT	3-10
Table 3-3: Level of Effort Estimates	3-12
Table 3-4: TRNBUILD – PT Output File Field Name Differences	3-13
Table 3-5: PT Stop-to-Stop Output Sample.....	3-13
Table 5-1: Time Savings in Hours for each Feedback Iteration of the Model.....	5-13
Table 5-2: Time Savings in Hours for each Step of the First Model Feedback Iteration.....	5-14
Table 6-1: Level of Effort Estimate.....	6-23

Table 7-1: 2007 Non-Resident Metrorail Trips	7-2
Table 7-2: 2007 Non-Resident Metrorail Trips by Station Jurisdiction (E/I Work, Auto Access)	7-3
Table 7-3: 2007 Non-Resident Metrorail Trips by Station Jurisdiction (E/I Non-Work Auto Access)	7-3
Table 7-4: 2007 Non-Resident Metrorail Trips by Station Jurisdiction (E/I Work, Walk Access).....	7-3
Table 7-5: 2007 Non-Resident Metrorail Trips by Station Jurisdiction (E/I Non-Work, Walk Access).....	7-4
Table 7-6: 2007 Non-Resident Metrorail Trips by Station Jurisdiction (Visitor)	7-4
Table 7-7: 2007 Non-Resident Metrorail Trips by Station Jurisdiction (NHB Business)	7-4
Table 7-8: 2007 Non-Resident Metrorail Trips by Station Jurisdiction (Air Passenger).....	7-5
Table 7-9: 2007 Non-Resident Metrorail Trips by Station Jurisdiction (Total)	7-5
Table 7-10: 2007 Non-Resident Metrorail Trips by Line Segment (E/I Work, Auto Access).....	7-5
Table 7-11: 2007 Non-Resident Metrorail Trips by Line Segment (E/I Non-Work, Auto Access).....	7-6
Table 7-12: 2007 Non-Resident Metrorail Trips by Line Segment (E/I Work, Walk Access)	7-6
Table 7-13: 2007 Non-Resident Metrorail Trips by Line Segment (E/I Non-Work, Walk Access)	7-6
Table 7-14: 2007 Non-Resident Metrorail Trips by Line Segment (Visitor)	7-7
Table 7-15: 2007 Non-Resident Metrorail Trips by Line Segment (NHB Business)	7-7
Table 7-16: 2007 Non-Resident Metrorail Trips by Line Segment (Air Passenger)	7-7
Table 7-17: 2007 Non-Resident Metrorail Trips by Line Segment (Total)	7-8
Table 7-18: 2007 Non-Resident Ridership	7-9
Table 7-19: Metrorail Station Codes	7-12
Table 7-20: Application Growth Factors	7-15
Table 7-21: New Metrorail Stations	7-15
Table 7-22: Estimated New Station Non-Resident Metrorail Boardings+Alightings, 2040	7-22
Table 7-23: Station Jurisdiction 2040 Forecast Summary (E/I Work, Auto Access).....	7-23
Table 7-24: Station Jurisdiction 2040 Forecast Summary (E/I Non-Work, Auto Access).....	7-23
Table 7-25: Station Jurisdiction 2040 Forecast Summary (E/I Work, Walk Access)	7-23
Table 7-26: Station Jurisdiction 2040 Forecast Summary (E/I Non-Work, Walk Access).....	7-24
Table 7-27: Station Jurisdiction 2040 Forecast Summary (Visitor).....	7-24
Table 7-28: Station Jurisdiction 2040 Forecast Summary (NHB Business)	7-24
Table 7-29: Station Jurisdiction 2040 Forecast Summary (Air Passenger).....	7-25
Table 7-30: Station Jurisdiction 2040 Forecast Summary (Total)	7-25
Table 7-31: Line Segment 2040 Forecast Summary (E/I Work, Auto Access).....	7-25
Table 7-32: Line Segment 2040 Forecast Summary (E/I Non-Work, Auto Access).....	7-26
Table 7-33: Line Segment 2040 Forecast Summary (E/I Work, Walk Access)	7-26
Table 7-34: Line Segment 2040 Forecast Summary (E/I Non-Work, Walk Access)	7-26
Table 7-35: Line Segment 2040 Forecast Summary (Visitor).....	7-27
Table 7-36: Line Segment 2040 Forecast Summary (NHB Business)	7-27
Table 7-37: Line Segment 2040 Forecast Summary (Air Passenger)	7-27
Table 7-38: Line Segment 2040 Forecast Summary (Total)	7-28
Table 7-39: 2040-2007 Difference by Station Jurisdiction (E/I Work, Auto Access).....	7-28
Table 7-40: 2040-2007 Difference by Station Jurisdiction (E/I Non-Work, Auto Access).....	7-28
Table 7-41: 2040-2007 Difference by Station Jurisdiction (E/I Work, Walk Access).....	7-29
Table 7-42: 2040-2007 Difference by Station Jurisdiction (E/I Non-Work, Walk Access).....	7-29

Table 7-43: 2040-2007 Difference by Station Jurisdiction (Visitor)	7-29
Table 7-44: 2040-2007 Difference by Station Jurisdiction (NHB Business)	7-30
Table 7-45: 2040-2007 Difference by Station Jurisdiction (Air Passenger).....	7-30
Table 7-46: 2040-2007 Difference by Station Jurisdiction (Total)	7-30
Table 7-47: Trip Tables for Assignment	7-31
Table 8-1: Primary Reason for Selecting Airport Used	8-1
Table 8-2: Annual Air Passenger Enplanements	8-3
Table 8-3: Percent of Air Passengers Connecting from Another Flight	8-4
Table 8-4: Average Daily Local Air Passengers.....	8-4
Table 8-5: 2009 Air Passenger Trip Purpose by Airport (Percentage)	8-5
Table 8-6: Income Weights and Income Distribution by Airport	8-6
Table 8-7: Airport Market Factor by Airport.....	8-8
Table 8-8: 2009 Estimated and Observed Airport Trip Distribution by Airport.....	8-12
Table 8-9: Air Passenger Mode Choice Disutility Equations	8-14
Table 8-10: Bias Coefficients and Nesting Constants by Airport, Mode, and Trip Market	8-16
Table 8-11: Observed and Estimated Air Passenger Trips by Mode	8-18
Table 8-12: Time of Day Percentages for the Air Passenger Model	8-19

1 Introduction

Since FY 2006, the staff of the National Capital Region Transportation Planning Board (TPB) has maintained a consultant-assisted project to evaluate the travel forecasting practices used in the Metropolitan Washington region. 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). During the course of FY 2012, the following task orders were developed collaboratively between TPB staff and AECOM:

1. Attend relevant meetings, such as the Travel Forecasting Subcommittee, and respond to ad hoc requests
2. Consultant recommendations for improving mode choice modeling in the TPB Version 2.3 Travel Model
3. Improving mode choice modeling: Enhancements to the LineSum transit assignment summary program
4. Reducing model run times in the TPB Version 2.3 Travel Model
5. Improving mode choice modeling: Begin conversion from the TRNBUILD transit path building software to the Public Transport (PT) transit path building software
6. Improving mode choice modeling: External/non-resident transit travel and air passenger transit travel

This report documents the work done by AECOM and its subcontractor, Stump/Hausman Partnership, to fulfill these task orders. Chapters 2 through 8 address Task Orders 1 through 6. Chapter 9 presents a summary of consultant recommendations.

2 Meetings and Technical Assistance (Task Order 1)

AECOM participated in each Travel Forecasting Subcommittee meeting and prepared presentations for the November 17, 2011 and May 18, 2012 committee meetings. AECOM also met several times with the MWCOG staff to discuss work tasks, findings and recommendations. In addition, two technical assistance requests were addressed as part of this task order. The first related to MWCOG's discussions with Citilabs regarding convergence issues with the Cube Voyager traffic assignment software. The second involved a briefing on AECOM's efforts to upgrade the WMATA modeling process to be compatible with the MWCOG TPB Version 2.3 Travel Model.

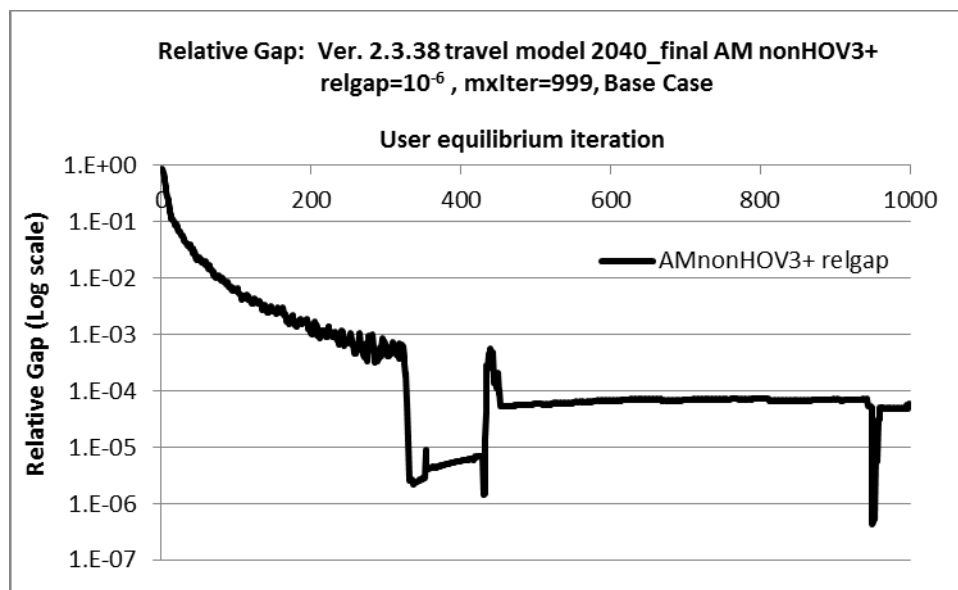
2.1 Cube Assignments

For most applications of the TPB regional travel model (Version 2.3), the model parameters are set to use a traffic assignment closure threshold of a relative gap of 10^{-3} or 300 user equilibrium iterations, whichever comes first. TPB staff has found this to be suitable for most regional analyses, although, in initial tests at the corridor level, TPB staff found that a closure threshold of a 10^{-4} relative gap was needed to eliminate noise in volume difference plots.¹ In later tests at the corridor level, however, TPB staff found that a relative gap threshold of 10^{-4} might still not be enough. For example, in some preliminary project planning work being conducted, TPB staff found that, even with a relative gap threshold of a 10^{-4} , a small network change could result in link volume differences that are well outside of the study corridor. Hence, TPB staff has an interest in relative gap closure thresholds of 10^{-4} and 10^{-5} for some project planning work.

Throughout the year, MWCOG has been in dialogue with Citilabs about the Cube Voyager highway assignment software. MWCOG's attempts to minimize spurious changes in the assigned volumes far from a minor change in the network have led to software applications with extremely tight convergence criteria (i.e., relative gap less than 10^{-4}). Unfortunately, the Cube assignment module generates some dramatic and illogical fluctuations in the relative gap statistic for iterations beyond 10^3 . An example of the convergence profile from one of the MWCOG test runs is shown in Figure 2-1. In theory this should be a uniformly descending pattern that asymptotically approaches zero. It should not have huge jumps from one iteration to the next or "flat line" areas with no change (e.g., the area beyond ca. 300 user equilibrium iterations).

¹ See, for example, Ronald Milone and Mark S. Moran, "TPB Version 2.3 Travel Model on the 3,722-TAZ Area System: Status Report and Sensitivity Tests" (presented at the July 22, 2011 meeting of the Travel Forecasting Subcommittee of the Technical Committee of the National Capital Region Transportation Planning Board, held at the Metropolitan Washington Council of Governments, Washington, D.C., July 22, 2011), 28–29, http://www.mwcog.org/transportation/committee/committee/archives.asp?COMMITTEE_ID=43; Mary Martchouk, "Traffic Assignment Convergence Testing" (presented at the May 18, 2012 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 18, 2012), 3, http://www.mwcog.org/transportation/committee/committee/documents.asp?COMMITTEE_ID=43.

Figure 2-1: MWCOG Cube Assignment Convergence Test



Source: Martchouk, M. (2012, April 20). Citilabs Improvements to Highway Assignment Script. Memorandum.

Citilabs has provided some ideas for getting a better level of convergence and convergence pattern. For example, they suggested adding a COST function to the highway assignment script. This suggestion helped the convergence pattern for four of the six traffic assignments used in the Version 2.3 Travel Model. However, it also increases the run time by 50 percent, so staff has to consider whether the improved convergence is worth the added run time.

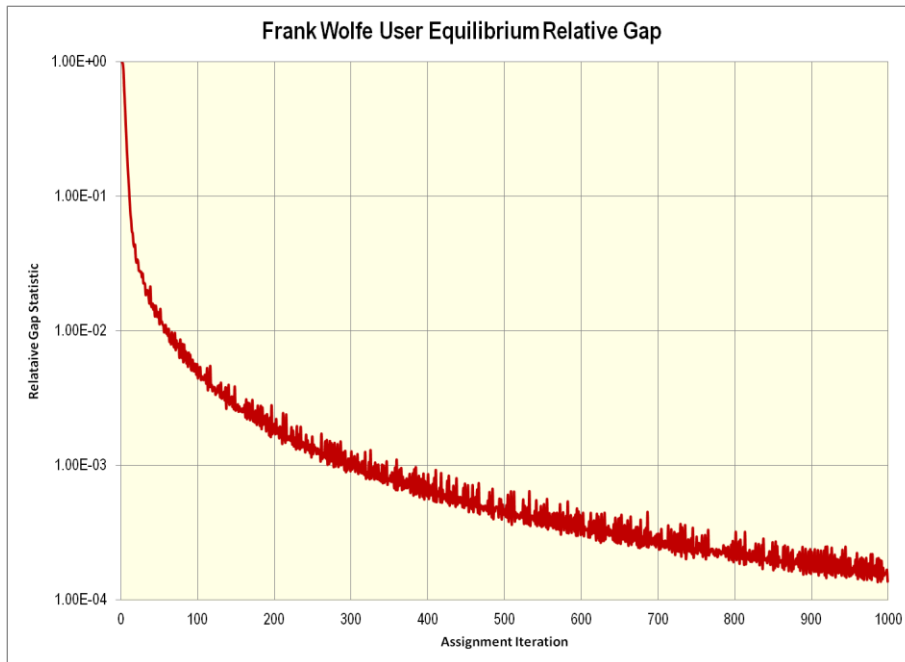
Throughout this process, AECOM has been asked to review materials generated by MWCOG and the responses provided by Citilabs. We believe MWCOG has a legitimate concern. The assignment software does have difficulty producing a theoretically correct convergence descent path for relative gap criteria beyond 0.001. The problem appears to be less pronounced using the Frank Wolfe algorithm rather than the Bi-Conjugate Frank Wolfe algorithm. Unfortunately, the Frank Wolfe algorithm requires much longer run times and does not reach the 10⁻⁴ criteria before the 1000 iteration limit is reached.

There were also concerns about the way tolls impact the convergence algorithm. User equilibrium theory requires a monotonically decreasing volume-delay function for link impedance based on the assigned volume. MWCOG has been using a volume-delay function that is implemented as a lookup function, which also includes tolls in the link impedance calculation. Citilabs suggested both of these “imperfections” could be causing the convergence issues. MWCOG performed tests with a continuous volume-delay function and a variety of different cost functions, but found no significant improvement in the results using the continuous volume-delay function (and the continuous function resulted in slightly longer run times).

The discussion ultimately came down to a question of software. Is this a systemic problem with user equilibrium in general or the Cube Voyager software specifically? To begin to answer this question,

AECOM did several test assignments using the M-NCPPC TransCAD model called TransForM.² This model was originally derived from the MWCOG Version 2.1D model, which was developed on the 2,191-TAZ area system. Figure 2-2, Figure 2-3, and Figure 2-4 show the relative gap statistics from three applications using difference user equilibrium assignment algorithms.

Figure 2-2: Frank Wolfe Relative Gap Results from a TransForM Assignment



² Caliper Corporation, *TransForM – A New Regional Travel Demand Model Developed for Prince George’s County*, Project Final Report (Prince George’s County, Maryland: Maryland-National Capital Park and Planning Commission, August 9, 2006).

Figure 2-3: Bi-Conjugate Frank Wolfe Relative Gap Results from a TransForM Assignment

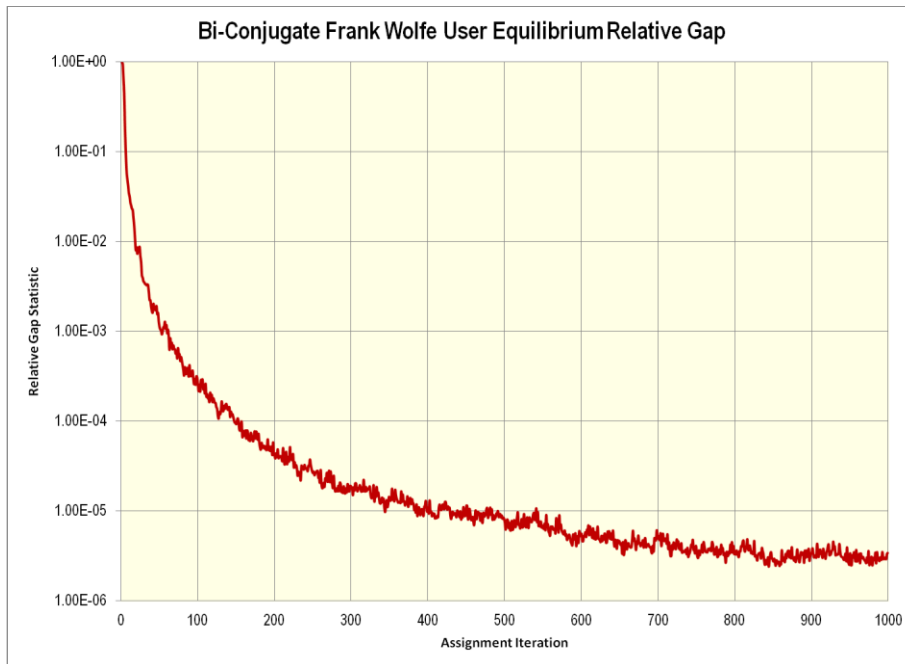
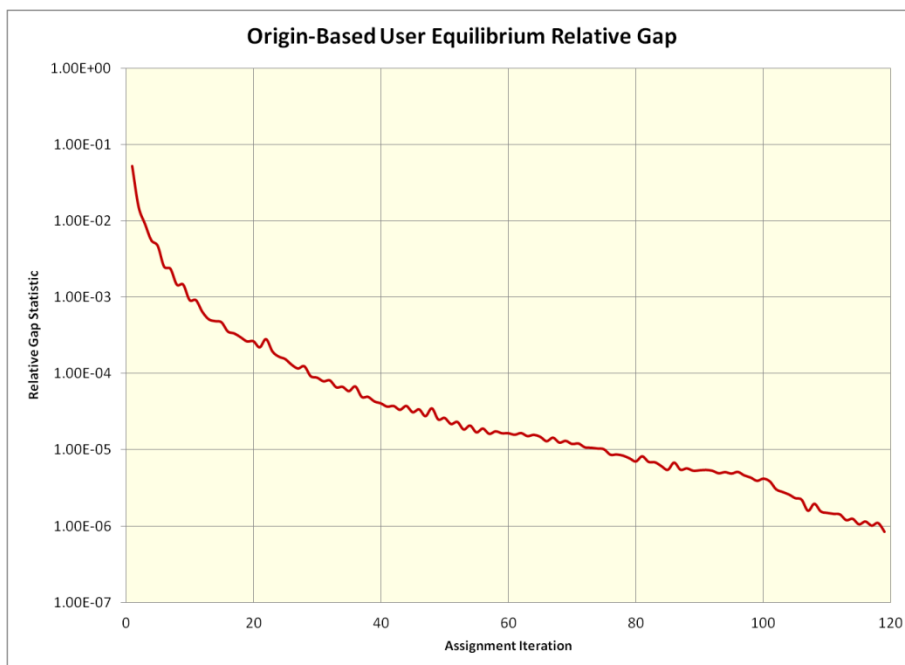


Figure 2-4: Origin-Based Relative Gap Results from a TransForM Assignment



The first point to make is that these assignments are not directly comparable to the MWCOC Version 2.3 Travel Model. The network is less detailed and there are fewer zones in the trip tables (i.e., 2,191 TAZs vs. 3,722 TAZs). These assignments also do not consider tolls. Nonetheless, it is obvious from these results that a logical convergence profile can be achieved using all three algorithms. The Frank Wolfe algorithm (Figure 2-2) does not reach the 10^{-4} convergence criteria in less than 1000 iterations, but the

Bi-Conjugate Frank Wolfe algorithm (Figure 2-3) reaches 10^{-4} at about 150 iterations and 10^{-5} at about 500 iterations. The Origin-Based algorithm (Figure 2-4) reaches 10^{-6} in 120 iterations.

These results suggest that it is theoretically possible for user equilibrium algorithms to achieve very tight convergence criteria in a reasonable amount of time. It does not, however, demonstrate that this can be achieved using the MWCOG Version 2.3 network and zone structure. If MWCOG wishes to pursue further discussions with Citilabs, AECOM would recommend performing a TransCAD assignment using all of the network attributes and trip tables from a congested MWCOG Version 2.3 model application. This would provide a direct software comparison that could help Citilabs focus on any methodological issues that might exist.

2.2 WMATA Modeling Process

During a meeting with MWCOG staff on June 18, 2012, AECOM presented a detailed overview of the updates made to the WMATA transit analysis process to make the modeling tool compatible with the MWCOG Version 2.3 Travel Model and Round 8.0 land-use forecasts. The revised model is currently being used to evaluate alternative transit networks for the year 2040 to support WMATA's Regional Transit System Plan (RTSP).

WMATA's post-processor to the MWCOG v2.3 model uses the person trip tables, highway skims, and "inputs" directory from a Version 2.3 run. It then tests and evaluates multiple transit alternatives given a fixed set of person trip tables and highway travel times. The process includes a few changes to the way transit networks and access links are processed, but the transit fares, path building and assignment methods are virtually identical to the MWCOG model. The most significant change is in the mode choice model. The software was upgraded and models were recalibrated for three trip purposes in peak and offpeak time periods.

2.2.1 Transit Access

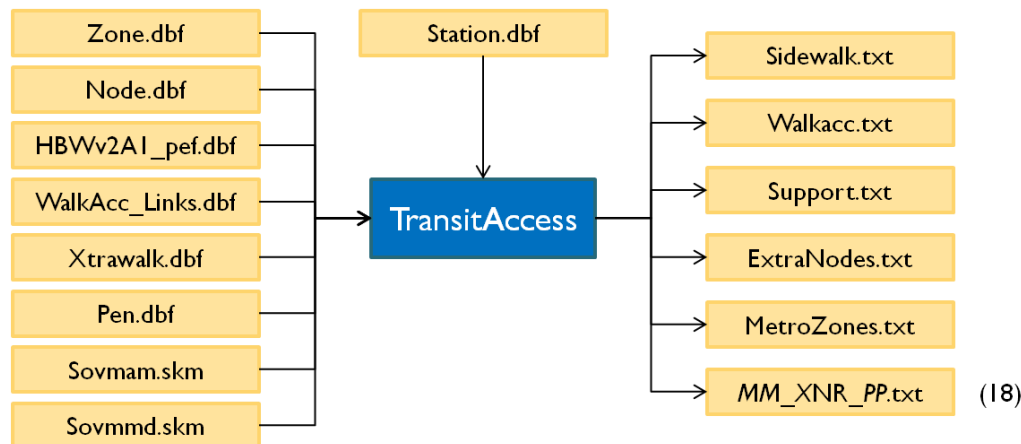
The WMATA transit access generation process builds walk and drive access files in TRNBUILD format similar to the MWCOG process. One difference involves the introduction of a Pedestrian Environment Factor (PEF) that is used to weight the straight-line distance between zone centroids and network nodes. PEF is calculated based on the number of blocks per square mile within each traffic analysis zone. The general assumption is that zones with greater block density will provide greater opportunities for more direct pedestrian movements in all directions. As a result, the walk distance adjustment factor increases as the PEF value for the zone decreases.

A similar adjustment is made to the distance between a park-&-ride lot and the nearby station. In this case the size of the parking lot affects the estimated time to find a parking space and walk from the parking space to the station. These time estimates also include the parking cost at the station and a shadow price concept used to calibrate parking capacity constraints.

In order to improve processing efficiency and reduce problems caused by coding errors, the programs and scripts included in the WMATA and MWCOG access generation processes were encapsulated into a new C++ program called TransitAccess. This program replaces the STAPROTP_V23 program and parker.s, walkacc.s, and autoacc4.s Cube scripts. It is fully parameterized with respect to file format

and processing flexibility and reduces the computer processing time required for generating the transit access links from about 40 minutes to 8 seconds. The overall input and output files processed by the TransitAccess program are shown in Figure 2-5.

Figure 2-5: TransitAccess Inputs and Outputs



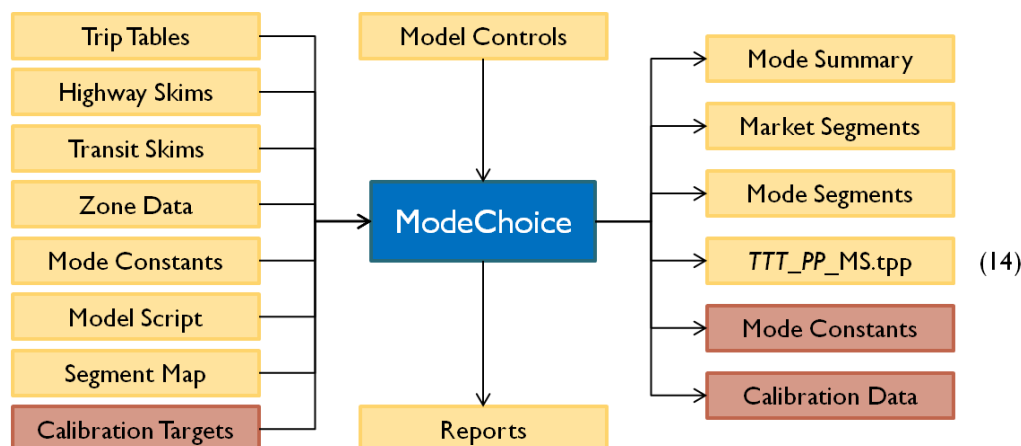
2.2.2 Mode Choice

The WMATA mode choice includes three purposes (HBW, HBO and NHB) and two time periods (peak and offpeak). The MWCOG v2.3 model includes five mode choice models (HBW, HBS, HBO, NHW, and NHO). In the MWCOG model, the HBW trips use peak period travel conditions and the non-work purposes use offpeak travel conditions. For the WMATA model, the person trips from the MWCOG model are collapsed into three trip purposes and each purpose is split using time of day factors. Modeling trips in this way enables WMATA to estimate current and future capacity problems on Metrorail segments and at park-&-ride lots.

As part of the conversion effort, AECOM re-wrote the AEMS mode choice software currently included in the MWCOG Version 2.3 model and previously used by the WMATA post-processor. AEMS was an older FORTRAN program that could no longer be compiled with readily available FORTRAN compilers. It also included a cumbersome user interface that was susceptible to undetected coding errors. It also requires some special configuration considerations when running on modern 64 bit computers.

The new ModeChoice program mimics the functionality of the AEMS software to apply a nested-logit or multinomial logit mode choice model. It reads trip tables and skim files in TPPlus or TransCAD format and supports sub-models by geographic market segments, traveler attribute, and short- and long-walk access markets. User generated processing scripts are available for manipulating the skim and zonal attribute data. The software also includes an iterative model calibration process to match calibration targets by mode, market segment, and traveler attribute. This process includes methods for controlling the acceptable range of mode-specific constants and convergence criteria based on root-mean-squared-error (RMSE) statistics. The overall inputs and outputs for the ModeChoice program are shown in Figure 2-6.

Figure 2-6: ModeChoice Inputs and Outputs



The change in person trip rates between the MWCOCG Version 2.2 model and the Version 2.3 model required a complete recalibration of the WMATA mode choice models. The six mode choice models included in the new WMATA process use a different set of geographic market segments than the MWCOCG Version 2.3 model and use the Pedestrian Environment Factor (PEF) extensively. The new calibration was based on the 2008 Metrorail and regional bus surveys. The calibration effort considered a model structure with 12 geographic market segments and one with no geographic market segments. It also included greater dependence on a modified PEF concept and shadow prices to control park-&-ride demand at Metrorail stations.

In order to minimize the impact of geographic market biases on future forecasts, the models without geographic market segments were selected. There was less confidence in the observed data for the 12 market segments and the Metrorail station boardings were not significantly better than the model without geographic market segments. The new PEF concept was found to be useful. This concept increases the PEF values assigned to a given zone based on the developable area of the zone, the PEF values of nearby zones, Metrorail and other transit services in the area, and the population and employment density of the zone. These relationships were calibrated using existing PEF data and used to adjust the PEF value assigned to future years and transit network scenarios.

The conversion of the WMATA model to the MWCOCG Version 2.3 platform was a much larger effort than originally anticipated. The conversion and calibration is by no means perfect and additional process improvements are being considered. Developing a logical set of calibration targets for the 12 market segment models from the 2008 survey data proved difficult. Comparing the results to other survey data would be desirable. In general, the walk-to-Metrorail and commuter rail targets were habitually underestimated by the calibration process. This may point to survey expansion problems or a need to adjust the path building parameters or walk access generation process. The WMATA process also adds Metrorail trips to selected stations to account for airports, tourists, and external transit trips. Subtracting these trips from the mode choice calibration process helped to improve the results. Improvements to the way park-&-ride capacity constraints are developed and applied would also help.

2.2.3 Points of Interest

The WMATA model conversion process identified a number of places MWCOG may wish to consider improvements to future versions of the regional modeling process. In the spirit of full disclosure, the source of many of the data formatting and processing concerns we experienced were a direct result of legacy software tools and procedures borrowed from older versions of transit models developed by AECOM for WMATA and other clients. In most cases, these tools were developed for a specific project and applied by AECOM staff. Unfortunately, they were not designed for or intended as “commercial” software products of the quality and sophistication needed for an official MPO process.

What this has led to is a model that includes the same basic data in multiple files and file formats. Walk access percentages, PEF values, station data and basic location coordinates are prime examples. Inconsistencies between these files will distort the mode choice calculations and model results. The modeling process is also more complex than it needs to be, due to the additional format conversion steps and file management overhead. AECOM would recommend constructing fewer input files that directly serve the needs of multiple programs or scripts. We also recommend reconfiguring the programs and scripts to write fewer output files. Consolidating all of the various record types needed for coding TRNBUILD access links into a single file is one possibility.

Error checking is another area that could use some improvement. For example, the current MWCOG process assumes the node number assigned to a Metrorail stations is between 8001 and 8150. If the number is greater than 8150, the station-to-station distance calculations are all zero and the Metrorail fare to any station is zero. This makes the Metrorail ridership too high. In addition, each Metrorail station is assigned a 5,000 series “zone” number as part of the drive access time calculation. A highway path builder is executed to build a path from each origin zone to these station zones. If the station zone number is not included in the highway skim file used by the drive access routine, the drive time to the station is zero. This leads to excessive park-&-ride and kiss-&-ride demand at the station. If any of these things occur, the user has coded the network improperly. The modeler, however, will have no idea that the results are wrong unless they investigate individual cells within the skim matrix files.

AECOM was able to design the WMATA modeling process to use a much larger number of CPUs than the four specified for the MWCOG Version 2.3 model. This reduced the computer processing time by an order of magnitude (3.5 hours to 30 minutes). We are also able to run multiple transit alternatives at the same time using our large modeling server. This raised a number of issues with the MWCOG and WMATA batch processing methods that may be worth improving. First of all, the batch processing approach frequently uses environment variables to pass a parameter into a Cube script file. The variables are typically used to identify the run year, alternative name, or processing options. Unfortunately, running multiple batch files in parallel overwrites the environment variables.

A slightly different, but similar issue relates to Cube Cluster processor IDs. Cube Cluster enables the software to execute some procedures in parallel. It uses a list of processor IDs to identify the processing instances that relate to a given application. For the most part, the WMATA process executes multiple instances of a given program rather than multiple threads within a program. There is, however, at least one matrix processing step that takes advantage of Cube Cluster parallel processing. If two alternatives

are running at the same time using the same processor IDs, Cube Cluster is unable to identify which processors belong to each program and as a result will close the second process prematurely when it exits the first process. This does not generate any error messages. It just does not complete the second process. In the WMATA application, this resulted in fewer trips included in the trip table used in the second transit assignment.

AECOM made a number of improvements to the modeling process and significant upgrades to the software tools used in the WMATA process. There are a number of additional places AECOM would recommend making improvements to the WMATA process and the model calibration. Notwithstanding, we believe there are some useful components of the WMATA process that MWCOCG may wish to consider. The new software tools are one logical place to start. The new programs are distributed through the TRANSIMS Open Source site and as such are well maintained and frequently upgraded.

3 Improving Mode Choice Modeling (Task Order 2)

As part of this task, AECOM reviewed the structure of the TPB Version 2.3 mode choice model; evaluated potential changes to the transit path building software; and identified the impacts of software changes on transit assignment and ridership reporting tools. The transit modeling components were also compared to the WMATA post processing procedures to identify places where additional coordination may be desirable. WMATA is currently upgrading their procedures to utilize the results of the TPB Version 2.3 Travel Model and is open to tighter integration that would simplify their analysis.

3.1 Structure of the Mode Choice Model

The mode choice model currently included in the TPB Version 2.3 Travel Model is based on work AECOM did for WMATA in 2005 for the D.C. Alternatives Analysis. Since then, AECOM updated the WMATA model with CTPP data for the White House Area Transportation Study in 2007 and the VRE Haymarket Extension in 2008. AECOM has continued to refine the WMATA mode choice process as part of the Regional Transit System Plan (RTSP).

The challenge for this task is to determine how best to coordinate these efforts to enhance consistency without sacrificing any of the primary modeling objectives of each agency. COG/TPB is focused on multi-modal regional planning and air quality conformity requirements while WMATA is focused on detailed transit ridership analysis. At the same time, both agencies generate forecasts of transit usage so it is generally desirable that both forecasts be approximately the same.

Fortunately, the latest WMATA mode choice model incorporates many of the changes COG/TPB identified as desirable. More specifically, the model structure was changed to eliminate the 20 geographic market segments³ by adding a pedestrian environment factor (PEF). This helps reduce the number of calibration constants by introducing a new explanatory variable. As a result, the model is easier to calibrate and responds more appropriately to future land use changes.

Both models are nested-logit models with multiple transit options. The modes include:

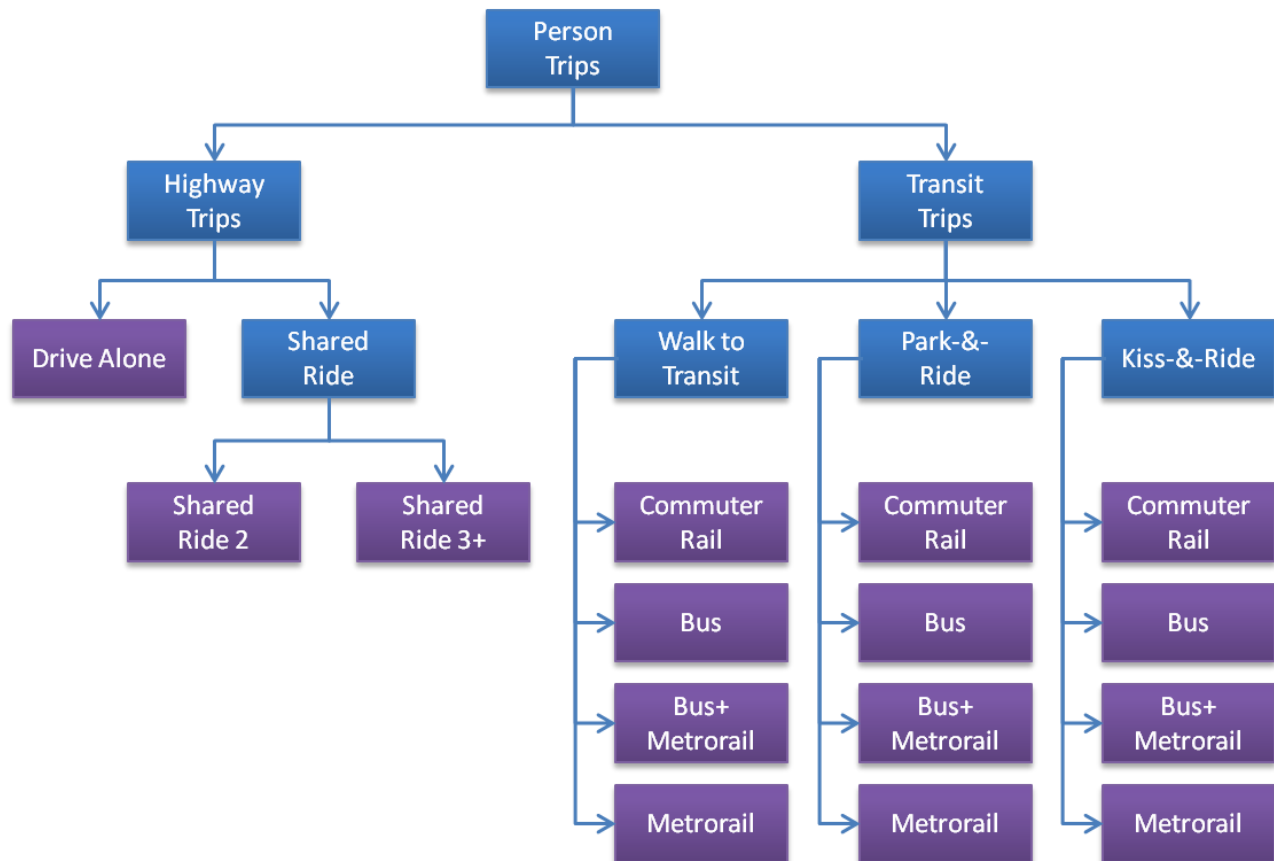
1. Drive Alone
2. Shared Ride 2
3. Shared Ride 3+
4. Walk to Commuter Rail
5. Walk to Bus
6. Walk to Bus+Metrorail
7. Walk to Metrorail
8. Park-&-Ride to Commuter Rail
9. Park-&-Ride to Bus
10. Park-&-Ride to Bus+Metrorail

³ Twelve geographic market segments are still used in the WMATA model: Not for the purpose of nesting constants, but for summary and analysis purposes. These 12 geographic market segments are described later in this chapter.

11. Park-&-Ride to Metrorail
12. Kiss-&-Ride to Commuter Rail⁴
13. Kiss-&-Ride to Bus
14. Kiss-&-Ride to Bus+Metrorail
15. Kiss-&-Ride to Metrorail

The nesting structure is shown in Figure 3-1.

Figure 3-1: Mode Choice Nesting Structure



The COG/TPB process includes daily mode choice models for five trip purposes:

1. Home-Based Work
2. Home-Based Shopping
3. Home-Based Other
4. Non-Home-Based-Work
5. Non-Home-Based Other

The WMATA process includes peak and off-peak mode choice models for three trip purposes:

1. Peak Home-Based Work

⁴ Park-&-Ride to Commuter Rail and Kiss-&-Ride to Commuter Rail use the same path data.

2. Off-peak Home-Based Work
3. Peak Home-Based Other
4. Off-peak Home-Based Other
5. Peak Non-Home-Based
6. Off-peak Non-Home-Based

The WMATA process adds the COG/TPB Home-Based Shopping to Home-Based Other and combines the two Non-Home-Based purposes. It then applies time-of-day factors to the daily person trip tables to create the peak and off-peak trip tables. The AM peak period network and congestion levels are used for the peak period models and the off-peak network and congestion levels are used for the off-peak period models. COG/TPB also uses AM peak and off-peak networks and travel conditions, but models all Home-Based Work trips using the AM peak conditions and all other purposes using off-peak conditions.

Both models include market segments by income quartile for home-based trips purposes. They also include geographic market segmentation, but the impact of geographic areas is significantly different. Both models using seven area types based on geographic area:

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

Market segments for mode choice coefficients and constants are then defined based on the origin and destination area type of the trip. The WMATA model defines 12 geographic market segments and the COG/TPB model defines 20 markets. The 12 market segments defined by WMATA are:

1. DC to DC Core
2. DC to Virginia Core
3. DC to Urban (DC, VA, MD)
4. DC to Other (VA, MD)
5. Maryland to DC Core
6. Maryland to Virginia Core
7. Maryland to Urban (DC, VA, MD)
8. Maryland to Other (VA, MD)
9. Virginia to DC Core
10. Virginia to Virginia Core
11. Virginia to Urban (DC, VA, MD)
12. Virginia to Other (VA, MD)

The actual models, however, adjust the nesting constants for only bus-related modes (walk to bus, walk to bus+Metrorail, park-&-ride to bus, and kiss-&-ride to bus) in three sub-markets. These markets include:

1. DC/Maryland to Urban (DC, VA, MD) and Maryland to Other (VA, MD)
2. Maryland/Virginia to DC Core
3. Virginia to Urban (DC, VA, MD)/Other (VA, MD)

The COG/TPB model currently includes 20 geographic markets with a unique set of calibrated nesting constants for each market. These markets include:

1. DC to DC Core
2. DC to Virginia Core
3. DC to Urban (DC, VA, MD)
4. DC to Other (VA, MD)
5. Maryland Urban to DC Core
6. Maryland Urban to Virginia Core
7. Maryland Urban to Urban (DC, VA, MD)
8. Maryland Urban to Other (VA, MD)
9. Virginia Core and Virginia Urban to DC Core
10. Virginia Core and Virginia Urban to Virginia Core
11. Virginia Core and Virginia Urban to Urban (DC, VA, MD)
12. Virginia Core and Virginia Urban to Other (VA, MD)
13. Maryland Other to DC Core
14. Maryland Other to Virginia Core
15. Maryland Other to Urban (DC, VA, MD)
16. Maryland Other to Other (VA, MD)
17. Virginia Other to DC Core
18. Virginia Other to Virginia Core
19. Virginia Other to Urban (DC, VA, MD)
20. Virginia Other to Other (VA, MD)

The variables defined in the COG/TPB zone file are used in the same way as the variables defined in the WMATA zone file. One difference is that the cost fields in the WMATA model are provided in 1980 cents while the cost fields in the COG/TPB file are defined in 2007 cents. The WMATA file includes additional data fields, but the only additional field used by the mode choice model is Pedestrian Environment Factor (PEF).

3.1.1 Pedestrian Environment Factor

Previous WMATA calibration efforts utilized the automatic calibration tools available for the advanced mode choice model in the same way COG/TPB did. These tools force fit the transit parameters and constants to match each dimension of the target transit shares. Automatic calibrations are relatively quick and easy to implement, but generate large bias constants that FTA does not accept. As a result,

WMATA decided to abandon the automatic calibration tool and manually calibrate the mode choice model using incremental adjustments from a base model with no bias constants.

Six rounds of incremental adjustments were implemented. Within each increment one or two target variables / parameters were selected for calibration. Multiple mode choice runs were used to manually adjust the target parameter until the estimated transit share reasonably matched to the survey target. It was often the case that an adjustment to one parameter would move another variable out of calibration. This meant that each subsequent calibration also required minor adjustment to the previous calibrated parameters.

The six calibration steps focused first on the total transit trips. This was followed by adjusting the total trips by market segment. Since these results had very high HOV mode shares, the next calibration brought the HOV share in line. This has a relatively large impact on the transit mode shares, so the transit factors needed adjusting. The overall transit mode shares by market segment were reasonable, but there was noticeable bias among the transit sub-modes. The next round of calibration adjusted the transit mode of access shares – walk, park-&-ride and kiss-&-ride. These results showed an imbalance between bus-to-Metrorail trips and the single mode paths. A constant was added to the bus-to-Metrorail model to make this mode less attractive. The final round fine-tuned all of these values to achieve the best fit with a minimum of market segment differences.

The resulting model included a 25-minute constant for walk-to-bus-to-Metrorail trips and 50 to 125 minute constants for park-&-ride and kiss-&-ride modes. The constants by market segment were high for Maryland to urban and other destinations and most of the Virginia interchanges. There were also a few interchanges with illogical negative bias constants. In general the calibrated constants were more acceptable than the original values, but still not within FTA standards.

Based on the manual calibration results, several new model formulations were designed in an attempt to identify a specific combination of parameters that could explain transit choices in the Washington D.C. region without depending on bias constants that cannot easily be explained or justified. These tests evaluated:

- Similar nest constants across all modes,
- Additional bias for trip ends in Maryland and Virginia,
- Zero auto household coefficients,
- Driving cost penalties by household type,
- Parking cost factors,
- Low income household factors, and
- Park-&-ride and kiss-&-ride constants.

The best results, however, were found using the Pedestrian Environment Factor (PEF). This factor is calculated as the sum of the number of 2000 Census blocks in a TAZ divided by the area of the TAZ in

square miles.⁵ Zones with a dense street grid will have a much higher pedestrian environment score (70+) than zones in suburban areas with relatively few intersecting streets (less than 10). Zones with more blocks will generally provide easier walk options for accessing transit services. In a few special cases, such as the Pentagon and National Airport, manual adjustments to the PEF were made to account for a concentrated access point to a relatively large single use zone. Examples of the PEF values for the region are shown in Figure 3-2 and Figure 3-3.

The mode choice model defines PEF constants for walk-to-transit and drive-to-transit and production and attraction zone at four break points (i.e., $2 * 2 * 4 = 16$ calibrated constants per model). The production constant is added to the attraction constant for any given zone-to-zone interchange. The actual constant applied to each trip end and mode is based on a linear interpolation between break points. The PEF value at the four break points is 0, 75, 150 and 265. So for example, if the calibrated attraction constants by break point are: -6.0, -2.2, 1.2 and 2.8, a PEF of 100 in the attraction zone generates a mode choice constant of $-2.2 + (1.2 + 2.2) * (100 - 75) / (150 - 75) = 1.067$. A similar calculation would be made for the production zone.

Adding the pedestrian environment factor to the mode choice model moved much of the explanatory power of the model out of the geographic market segment constants and into a defensible land-use attribute. This made the walk-to-transit options more attractive for D.C. urban residential areas. It also helped Maryland urban areas generate more walk-to-transit trips than Virginia urban areas as observed in the transit survey. This helped reduce the overall walk-to-transit trips in Virginia without significantly dampening the walk-to-transit from the denser areas of Arlington and Alexandria.

⁵ Please note that the PEF described in this chapter and shown in the two figures is an older version of the PEF that has recently been replaced by a newer version of the PEF. The revised PEF, which was developed after this chapter was first drafted, will be described in future documentation. The revised PEF is a function of zone proximity, transit access, and population/employment density.

Figure 3-2: Pedestrian Environment Factors near Metrorail Lines

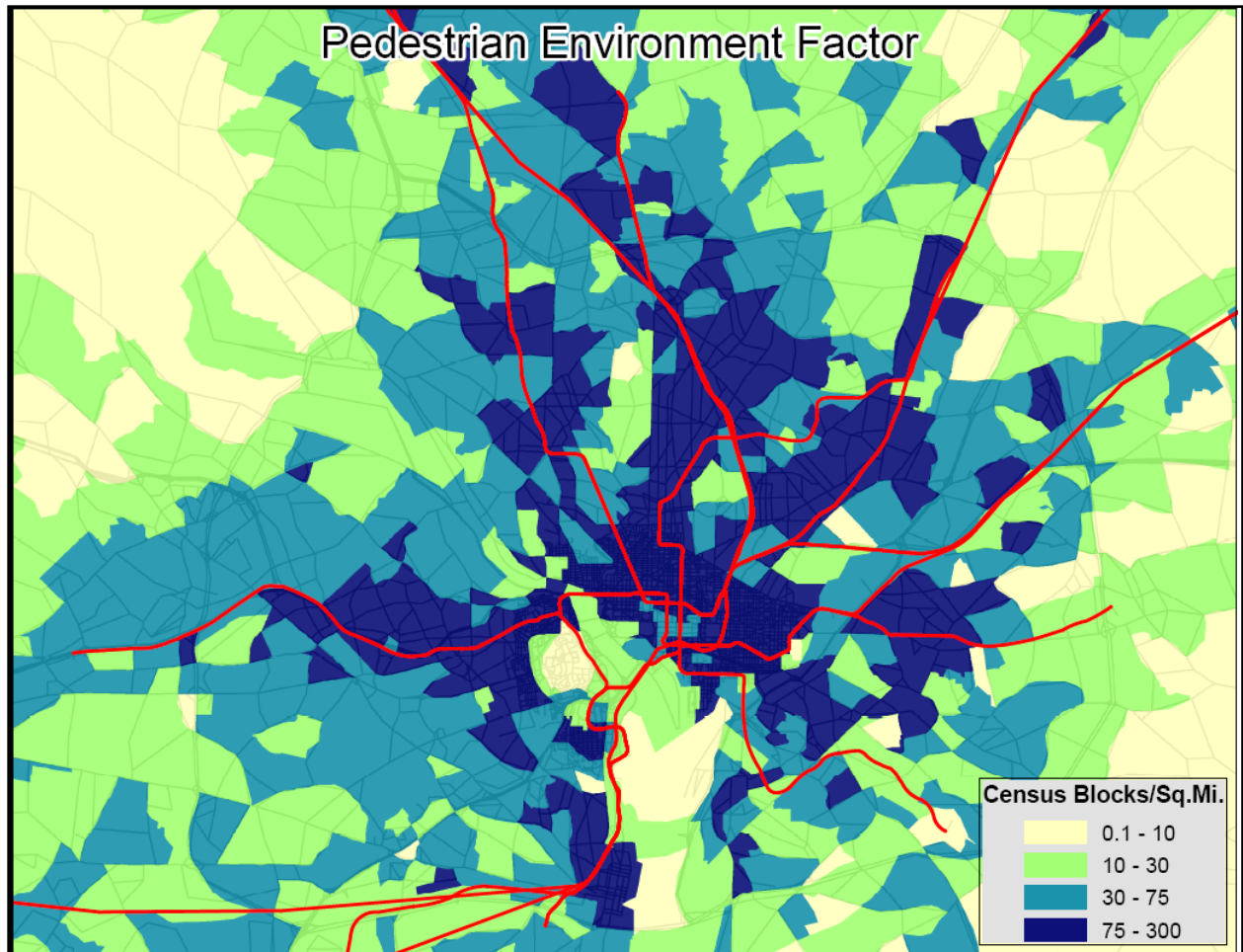
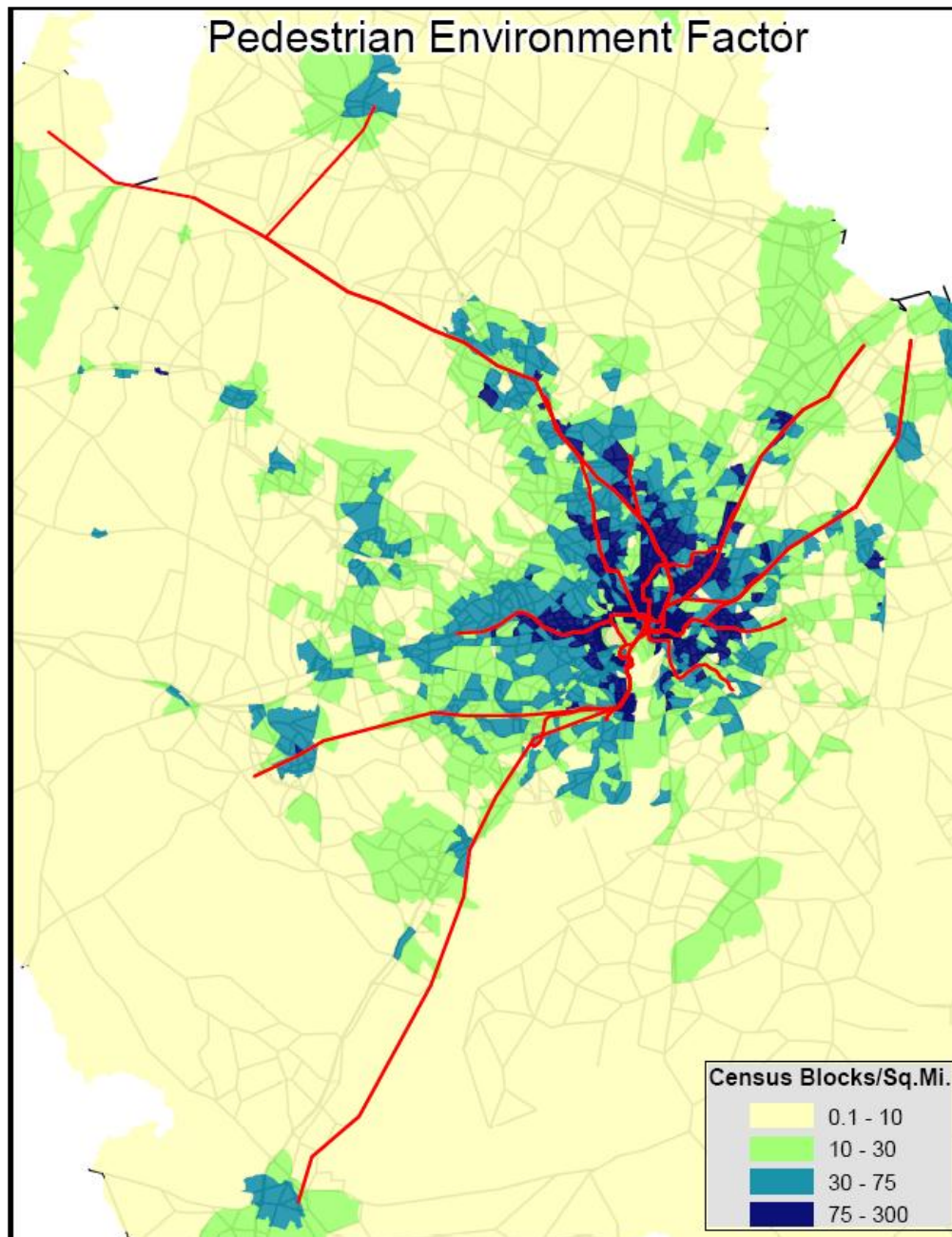


Figure 3-3: Pedestrian Environment Factors at a Regional Level



In addition, the mode choice results were analyzed by mode and market segment to determine if income explained the residual differences. Income data were not available on the Metrorail survey, but were included in the bus survey. The calibration targets by mode and market segment were subdivided into income quartiles using the bus survey data. The mode choice model already includes a separate model for each income quartile. Bias constants for low and high income households were used in the original WMATA model, but were removed by subsequent efforts because there was no income data used in the original calibration. The current model includes an income constant calibrated based on the subset of travelers in the regional bus survey.

In general, we would recommend migrating the COG/TBP model away from market segment constants and toward a zone-based variable like the Pedestrian Environment Factor. Pedestrian Environment Factor is a useful and relatively straightforward parameter to calculate for existing conditions. Future applications raise some concerns, but for most zones with significant transit service the value is reasonably stable. Greenfield development sites do require some additional work. On the other hand, the factor provides an opportunity to include a distinction between site plans that focus on transit-oriented development patterns versus typical suburban sprawl.

We also recommend that COG/TBP continue to use cost data in 2007 cents. We will encourage WMATA to update their cost data from 1980 cents to 2007 cents for improved compatibility and ease of use. The recommendation about trip purposes and time periods is less clear. WMATA will continue using mode choice models for peak and off-peak periods because it is important to them to model transit congestion by time of day and peak congestion is significantly affected by non-work trips in the peak period. It is unlikely WMATA will be interested in additional trip purposes, but they may not oppose including additional mode choice purposes if COG/TBP decides to migrate to peak and off-peak models for five trip purposes. If the modeling process takes full advantage of parallel processing, the overall processing time is not likely to be impacted significantly by modeling mode choice by time period and trip purpose.

3.2 Transit Path-Building Software or Procedures

The primary focus of this task was to assess the desirability of migrating the COG/TPB transit networks from TRNBUILD to PT. TRNBUILD is the older TP+ transit path building program that continues to be included within the Cube Voyager software suite, but is no longer maintained or improved. PT (Public Transport) is the new program Citilabs plans to support and improve moving forward.

AECOM and others have in the past expressed concerns about using PT for FTA New Starts analysis and other transit-related studies. AECOM has worked with Citilabs to address these concerns and believes PT is now mature enough to consider using. There are a number of attractive features about PT that make conversion desirable. COG/TPB staff is particularly attracted to the on-screen path tracing capabilities.

On the other hand, switching from TRNBUILD to PT requires a number of network changes that have time and effort implications. Interfacing PT with other components of the modeling process also needs to be considered. Moving to PT has minor impacts on the mode choice model and the transit assignment analysis tools.

Converting TRNBUILD (TB) networks to PT involves two basic activities:

- Converting existing TB format transit line files to PT format, and
- Generating non-transit legs compatible with the PT path building approach.

The TB and PT formats are quite similar with minor differences in field names. For example, TB line files use the field name 'freq' while PT files use the field name 'headway' to describe the frequency of transit service. The conversion of the TB line file format to a PT line file format can be easily accomplished using a custom script or text editor to 'find and replace' keywords such as 'freq'. We recommend developing

a VBScript to create an automated conversion tool with a simple user interface. VBScript is native to the Windows environment and thus does not need any external software.

Non-transit legs (walk access, drive access, and transfer) could either be input or generated. The MWCOCG Version 2.3 model generates these non-transit links using SUPPLINK in the script *walkacc.s*. Unless it is essential to provide the non-transit links as an input, it is recommended that the non-transit links be generated using the GENERATE statement in the PT module.

The PT path-building process requires a *Factor File* to specify the factors applied for weighting the different components of time and cost to find the best path. A *System File* defining the transit modes and wait curves and a *Fare File* defining the fare system are also required during PT path-building procedures. Each transit mode is defined as a *USERCLASS* in PT. Weighting factors and other parameters are to be defined by user class.

The PT path-building approach differs from TRNBUILD in the requirements for components of a transit path. A TRNBUILD path connects a sequence of links between origin and destination zones and could contain multiple non-transit links such as zone connectors (access or egress) and connector links between park-and-ride lots and transit stations. PT-based paths connect a series of legs where a leg can only be a connection from transit stop to transit stop or a transit stop to zone centroid.

AECOM researched the differences between TRNBUILD- and PT-based transit path building while developing the North Jersey Transit Demand Forecasting Model (NJTDFM). For this model, transit paths and the travel times for each leg of the path built using TRNBUILD and PT were compared. Table 3-1 and Table 3-2 present details of a typical drive-access transit path between an origin-destination pair in NJTDFM for TRNBUILD and PT respectively.

Table 3-1: Structure of a Drive-Access Transit Path using TRNBUILD

Leg	Time (Minutes)					Distance (mi)		Transit Lines
	Wait	Travel	Total	Boarding/ Transfer	Weighted Cumulative	Leg	Total	
Drive Access		12.57	12.57		16.34	3.36	3.36	
Walk Access		19.8	32.37		46.04	0	3.36	
Transit	7	25.38	64.75		92.3	8.38	11.74	321A1
Walk Egress		0.2	64.95		92.6	0.01	11.75	
Walk Zone Connector		6.4	71.35		102.2	0.32	12.07	

Table 3-2: Structure of a Drive-Access Transit Path using PT

Leg	Time (Minutes)					Distance (mi)		Transit Lines
	Wait	Travel	Total	Boarding/ Transfer	Weighted Cumulative	Leg	Total	
Access (Drive + Walk)		32.61	32.61		42.39	3.37	3.37	
Transit	7	25.38	64.99	5.3	88.65	8.38	11.75	321A1
Walk Egress		6.6	71.59		98.55	0.33	12.08	

While the TRNBUILD paths include multiple access or egress links, PT paths contain only one access and one egress link although it is possible to force PT to include extra links and replicate TRNBUILD. The paths presented in Table 3-1 and Table 3-2 use different versions of the same network. These networks have been coded to be identical; however, the networks have minor differences. These differences are necessary since TRNBUILD and PT use slightly different concepts for access and egress links to and from transit stations. For example, the total access time in the TRNBUILD model is 32.37 minutes while it is 32.61 minutes in the PT model. TRNBUILD and PT in-vehicle travel times are identical as expected. In addition, the difference in the total time for the trip is very small and most of these differences are due to differences in the network coding details.

The following tasks have been identified to convert the existing COG/TPB TRNBUILD-based model to a PT-based model.

- **Network conversion:** This involves adding Metrorail and commuter rail lines and stations to create a multimodal link database. PT requires that the rail lines be part of the multimodal link database. After the rail links are added to the link database, the existing transit line files in TRNBUILD format need to be converted to PT format. While Cube does not provide a tool to convert the transit line files from one format to another, a simple “replace” script can accomplish the task. In addition, TRNBUILD allows buses to traverse a one-way roadway in both directions, but PT does not. Any buses coded this way need to be re-routed. AECOM is not aware of any buses that need to be re-routed at this point, but we recommend that the bus routes be checked to ensure that the line files are accurate.
- **Scripting:** The scripts that generate access, egress, and transfer links need to be modified to use PT modules instead of TRNBUILD modules. Model parameters such as transfer prohibitions, fares, etc. are part of the script and use TRNBUILD modules. These parameters need to be converted to files and the scripts need to be updated to read these files using PT modules. It is important to ensure that the access, egress, and transfer links generated using PT modules — when used as inputs to generate paths using PT modules — are able to replicate the paths, and the travel time for each leg of the path built using TRNBUILD.

Table 3-3 presents the level of effort estimates for converting TRNBUILD path building to PT path building. This level of effort is greater than the estimate included in our Task 3 proposal. Significant contributions from COG/TPB staff would be necessary to implement the PT conversion.

Table 3-3: Level of Effort Estimates

Task	Sub task	Hours		
		Manager	Senior	Consultant
Network conversion				
	Develop a comprehensive highway and transit network	2	6	16
	Add Metrorail and commuter rail nodes in the highway network		4	16
	Convert TRNBUILD transit lines to PT format		4	4
	If needed, re-route bus lines on one-way links		6	16
Scripting				
	Generate access and egress links	4	8	32
	Generate transfer links		4	16
	Develop Factor, System, and Fare files	2	4	16
	Convert TRNBUILD script to PT script		4	8
Validation				
	Validate PT results against TRNBUILD results		4	16
	Validate PT results against transit survey observations		4	16
Total		8	48	156

3.3 Transit Assignment

If the decision is made to migrate from TRNBUILD to PT, the transit assignment analysis tools will need to be updated or replaced to work with new file structures. While the data fields output by TRNBUILD are also output by PT, the field names have changed. Table 3-4 presents the differences in the assignment output file field names. Since AECOM developed the LINESUM and LINEVOL programs that are currently included in the COG/TPB model, modifying these programs to read PT output will not be difficult. Alternatively, these programs could be replaced with newer analysis capabilities provided by Cube Voyager.

Table 3-5 presents a sample PT Stop-to-Stop output for consideration.

COG/TPB is also interested in including transit capacity constraints in the modeling process. Shadow pricing at park-and-ride lots is available to help constrain drive-to-transit trips and Metrorail station demand. The Cube Voyager software also includes a new transit crowding tool that may be worth considering.

The transit assignment scripts will need to be heavily edited in order to become usable for PT, but will be easier to follow by the user. The fare system, transit and waiting time factors and wait curves could be separate input files instead of being part of the script itself avoiding replication and the need to edit multiple scripts.

Table 3-4: TRNBUILD – PT Output File Field Name Differences

TRNBUILD (TB)	PUBLIC TRANSPORT (PT)	DESCRIPTION
A	A	Node A
B	B	Node B
MODE	MODE	Transit Mode
OWNER	OPERATOR	Transit Service Owner/Operator
NAME	NAME	Route Name
DIST	DIST	Route leg distance (between nodes)
TIME	TIME	Route leg travel time
SEQ	LINKSEQ	Route leg link sequence
FREQ	HEADWAY	Frequency / Headway
STOP_A	STOPA	Is Node A a stop? (Boolean)
STOP_B	STOPB	Is Node B a stop? (Boolean)
Forward (AB) direction fields		
AB_VOL	VOL	Volume on route leg
AB_BRDA	ONA	Boardings at Node A
AB_XITA	OFFA	Alights at Node A
AB_BRDB	ONB	Boardings at Node B
AB_XITB	OFFB	Alights at Node B
Reverse (BA) direction fields		
BA_VOL	REV_VOL	Volume on route leg
BA_BRDA	REV_ONA	Boardings at Node A
BA_XITA	REV_OFFA	Alights at Node A
BA_BRDB	REV_ONB	Boardings at Node B
BA_XITB	REV_OFFB	Alights at Node B

Table 3-5: PT Stop-to-Stop Output Sample

I	J	FromNode	ToNode	Mode	VOL
1	2	17305	17285	1	25
1	3	17305	17285	1	92
1	4	17305	16655	1	56
1	5	17305	17285	1	84
1	6	17305	22808	1	176
1	7	17305	22809	1	89
1	8	17305	22808	1	57
1	9	17305	22855	1	24
1	10	17305	22855	1	12

4 Transit Line Summaries and Access Considerations (Task Order 3)

This task was designed to upgrade the LineSum program used to summarize various aspects of the transit assignment results and propose transit access coding techniques that could accurately distinguish the modes used to arrive at and depart from transit stations. The deliverables include proposed changes to the station coding techniques, an upgraded LineSum program, and a software user's guide.

4.1 LineSum

The LineSum program was originally developed for New Jersey Transit in the year 2000. It was written to support a number of unique analysis needs of New Jersey Transit such as a summary of all of the bus routes that enter the Port Authority Bus Terminal in New York City. Its key feature is its ability to combine transit routes that share the same links in both the peak and offpeak periods in order to report total peak, offpeak and daily transit ridership statistics.

The program was written in C and has been updated sporadically over the years as bugs or issues were identified through applications in a number of places. It is still part of the standard modeling process at New Jersey Transit and several other transit agencies. It was used in the WMATA Version 2.2 post-processor and LineSum 1.8 was integrated by MWCOG into the MWCOG Version 2.3 travel model.

As part of this task, AECOM worked with MWCOG to address several issues that were identified through a close scrutiny of the output reports. The first issue was related to route names. The MWCOG transit network includes a number of routes whose route name includes special characters that the LineSum program interpreted as C format commands or comment statements. For example, lines that included a "%" or "#" would be mangled or truncated by the program. Extra logic was included to detect and correct these problems.

The more serious and complicated issue, however, was found in the way line data was merged or combined for multiple routes sharing some of the same links. The original code constructed a list of links from the anode-bnode sequence identified in the TPPlus transit assignment data. If the summary included multiple routes, the data for the other routes or time periods were compared to the anode-bnode combinations in the first route to determine where the new route data should be summed and in what direction that sum should be performed (i.e., AB direction or BA direction).

There were a few routes in the MWCOG transit network that traveled in both directions on the same link. This would typically be a link off of a major roadway used to enter and exit a transit station. If this route was traveling in the opposite direction of the first route used to establish the orientation of the combined service, the directional matching logic for the station access link would flag the route with the same orientation as the first route and fail to aggregate the ridership in the AB and BA directions correctly. To address this problem, AECOM added logic to compare all of the links in each route to the initial route and create a count of how many links matched the AB orientation and how many links matched the BA orientation. The direction with the greatest count was used to establish the orientation of the second route relative to the first route for combining the ridership data.

MWCOG also did some tests of peak and offpeak routes independently and in combination and found the total ridership reported in each direction to be slightly different depending on how the routes were combined. The problem boiled down to the way stop and non-stop nodes are identified and reported in the line reports. The program provided the user with the option to include or exclude non-stop nodes from the summary report. By default, non-stop nodes are excluded. The problem arose when a node was identified as a non-stop node in the peak period route, but a stop node in the offpeak route. In most cases this was detected properly, but if the new stop node was at the end of the offpeak route it failed to change the node from a non-stop to stop node and the boardings at this node in the offpeak period were not included in the ridership report.

AECOM also added some clarity in the way branch lines were reported. When MWCOG includes all of the Metrorail lines into a single summary report there were questions about how the boardings and alightings are reported at the stations where lines merge or diverge. For example, the boarding and alighting summary at the King Street station depends on the orientation of the line and how the branch lines (Yellow and Blue lines) are reported. Boardings at King Street going northbound reflect the boardings on the combined Yellow and Blue services in the northbound direction. The Yellow and Blue lines also include boardings at King Street in the southbound direction. These boardings are reported separately at the King Street station attached to each branch link. This makes an assessment of the total boardings or alightings in both directions at the King Street station somewhat confusing.

To help address this issue, MWCOG asked AECOM to modify the station access report to include an additional level of summary detail. The original implementation includes the option to report detailed information for a specified station node or the total boardings and alightings. The detailed information listed each link for each mode that entered or exited the station. Rather than listing the ridership on each link, MWCOG wanted the data summarized by access mode. This would report the number of people who access the station by walking, park-&-ride, and kiss-&-ride modes.

Five rounds of software updates to the LineSum program were implemented and provided to MWCOG for additional testing and analysis. LineSum 2.3 ultimately addressed the issues of primary concern. Once MWCOG was comfortable with the reports and methods, AECOM proceeded to upgrade the LineSum code to current C++ software standards and improve the user interface to provide additional help information and input flexibility. AECOM built this version of the software on top of the TRANSIMS Version 5.0 Open Source software library. This approach enabled the LineSum program to be built for 32 bit and 64 bit Windows and Linux operating systems and take full advantage of the user interface tools and standards included in the TRANSIMS package. This includes the routines for managing control keys, help messages, summary reports and file input and output formats.

The new LineSum program also adds the capability to merge multiple TPPlus transit ridership files (i.e., replaces the old LineVol program), includes additional selection options and selection range controls, and outputs combined link ridership data in Arcview shapefile and other formats. The link ridership summary was requested as part of Task Order 5, and integrated into the LineSum program for convenience. This includes the combination of link ridership by mode and line ranges with ridership and headway data for peak, offpeak, and daily time periods in the AB and BA orientation of the link. It

also includes a variety of performance statistics that utilize an estimate of the peak and offpeak hours of operation, vehicle capacity, and peaking factor to generate passenger miles and hours of demand and vehicle miles and hours of service for each time period. It also estimates the peak-hour riders and link load factor to flag and address transit congestion levels and demand/supply imbalances.

AECOM prepared a Quick Reference and User's Guide for the new LineSum program and provided the software and documentation to MWCOCG for review and comment. Documentation for the underlying TRANSIMS Execution Services was also provided. This includes generic information about how the TRANSIMS control keys, help facilities, and file processing works. Capabilities and features continued to be added to the LineSum program through work with WMATA for the Regional Transit System Plan. Additional report types, direct calculations of transfers at stations, and improved line segment ridership summaries were implemented. The latest version of LineSum is 5.0.11.

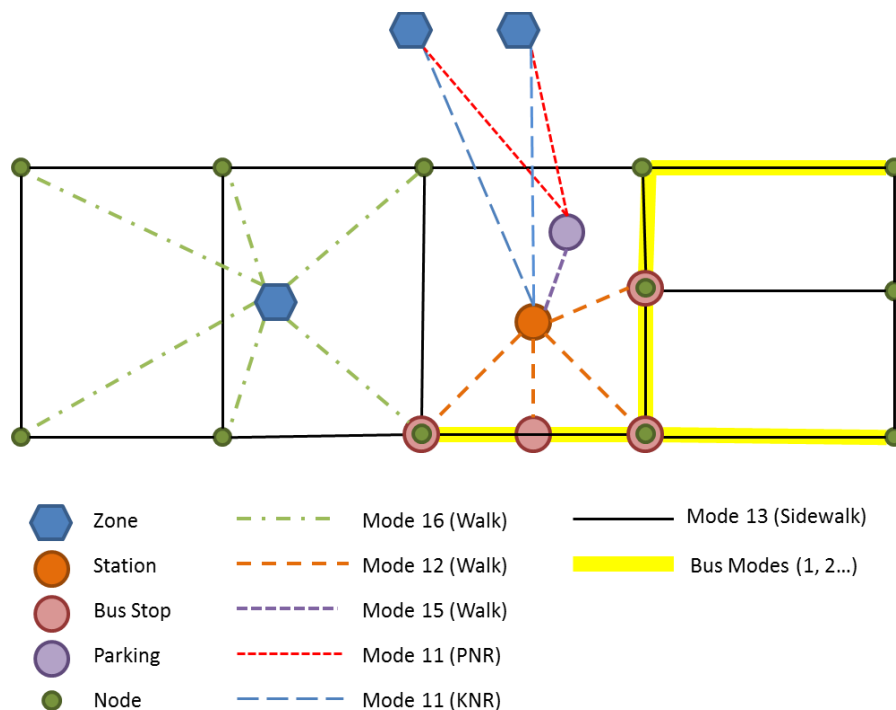
4.2 Station Access Coding

One problem with the way MWCOCG currently codes access to Metrorail and commuter rail stations is that there is no way of distinguishing boardings that transfer from a bus to the station from boardings that walk to the station directly from a zone. There is also no way of knowing if a traveler who alights at the station will subsequently board a bus or walk directly to their destination. As part of this task order, MWCOCG asked AECOM to propose potential changes to the coding methods that might address this issue.

Figure 4-1 shows how the transit access links are coded or constructed by the MWCOCG Version 2.3 Travel Model. The walk access routines connect zone centroids to all network nodes near the zone that satisfy various criteria. The area of the zone is used to define the maximum distance from the centroid to a candidate network node. These nodes must be located on a network link with a facility type that permits walking (e.g., not freeways or ramps). Each network link is also assigned to a zone number. The referenced zone must be within a short or long walk of a transit route. If all of these criteria are met, a mode 16 connector link is added to the network. Network links that satisfy these criteria are added as mode 13 links to the sidewalk layer.

The drive access links are constructed from information stored in the station database file. This file includes additional transit-only node numbers with X/Y coordinates that define transit stations and park-&-ride lots. Each station is also associated with a drive access zone number and up to four bus stop numbers. The drive access zone number is used to extract travel time and distance information from a special path-building procedure that builds paths on the highway network between each traffic analysis zone and each drive access zone. The four bus stop nodes are used to construct walk links to the station.

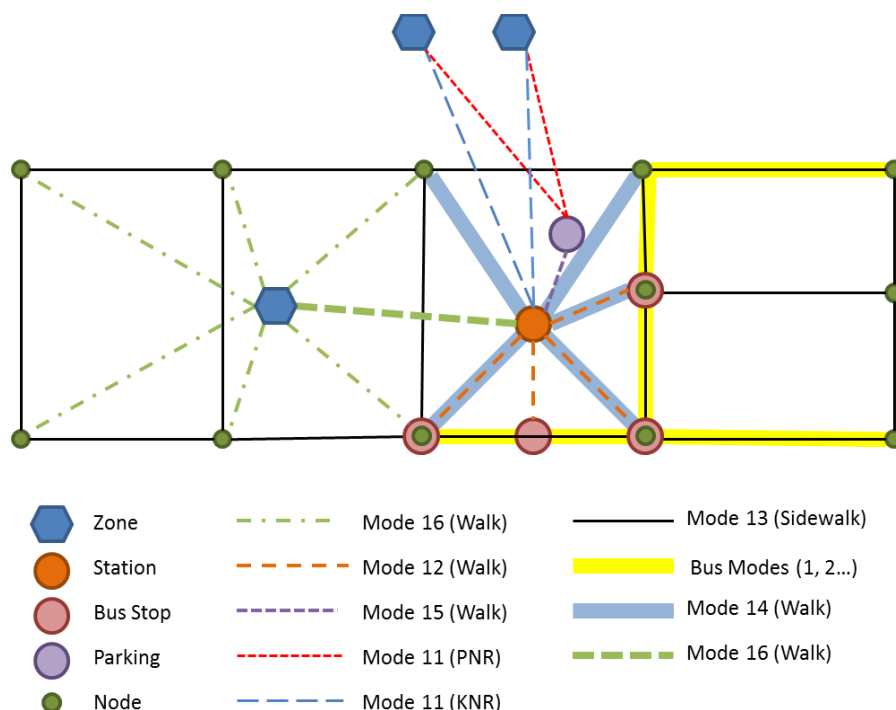
Figure 4-1: Current Transit Access Links



In the example shown in Figure 4-1, the station is connected with four mode-12 walk links to the four specified bus stops. Note that one of the stop nodes is not on the sidewalk network. This would be a station node with a direct link for transferring between the two routes (e.g., Metrorail to commuter rail). The figure also shows two zones connected to the station with park-&-ride and kiss-&-ride links. The kiss-&-ride links connect directly to the station while the park-&-ride links connect to the parking lot. The mode 15 walk link connects the station to the parking lot. Note that the parking lot does not need to be included in the highway network. Only the drive access zone number needs to be connected to the network to generate the travel time from the zone to the parking lot.

One concern about the current process is that traffic analysis zones with walk access to transit are only connected to nodes on the sidewalk network. Stations are also connected to the sidewalk and bus network through the four bus stop nodes specified in the station database. Since the regional roadway network is relatively limited and only a subset of facility types can be used to generate the sidewalk network, the walk paths between zone centroids and Metrorail or commuter rail stations can at times be unreasonably long. Expanding the zone access logic to permit direct walk connections between traffic analysis zones and Metrorail and commuter rail stations would seem advisable. This concept is shown in Figure 4-2 (the thick, dashed green line between the zone centroid and the station node). These connections should have distance limitations similar to other zone connector links.

Figure 4-2: Proposed Change in Transit Access Coding



This change will improve walk access at Metrorail stations, but still does not address the bus transfer requirement. The proposed solution to this problem involves separating walk access to the station from nearby zones from walk access from nearby bus routes. This concept is shown in Figure 4-2. Since the nearby bus stops are already defined in the station database, no change is required for generating the bus access links (mode 12). The drive access concepts are also unaffected. What is needed is a process that is very similar to the existing zone connector logic that identifies network nodes on the sidewalk network in the general vicinity of the station and creates new connector links to these locations. These links require a unique mode number to distinguish them from bus access links. In this case, mode 14 was selected.

Given the coding concepts outlined above, trips that arrive or depart the station by mode 12 would be considered transfers to bus routes. Trips that arrive or depart the station by modes 14 or 16 would be considered walk access. Mode 11 trips access the station by kiss-&-ride and mode 15 trips use the park-&-ride lot. (The WMATA coding also uses mode 13 links to represent pedestrian tunnels between stations and escalators between station platforms).

There are, however, two important keys to the success of this coding concept. The first is that all reasonable bus connections to the station must be identified in the station database.⁶ The second is that all bus transfers must use mode 12 links to access the station. In other words, the path builder needs to limit the mode-to-mode transfer options to ensure that all bus-to-station transfers use a mode 12 link and all walk-to-station transfers use a mode 14 link. This can be achieved using the NOX

⁶ Note that the current station database is limited to four bus nodes per station node.

commands in the TRNBUILD software. A transfer prohibition will need to be added between modes 11, 13, 14, 15 and 16 and mode 12, and between mode 14 and the transit modes (1-10). The coding must also minimize the possibility that the path builder will find a minimum impedance path that exits a bus onto the sidewalk network and then uses a mode 14 link to access the station. This can be achieved by ensuring that the mode 12 link between the station and the bus stop is always shorter than the alternative combination of walk links.

The proposed coding would be a relatively easy short-term solution to the access mode accounting concerns. If or when MWCOG decides to transition from TRNBUILD to PT, the coding requirements should be much more straightforward. The PT GENERATE statement should be able to build connection paths between zones and stations and between transit stops, while tracking the mode-to-mode transfer combinations directly. If this proves to be the case, distinguishing between walk access to the station from a bus or the sidewalk network will not be necessary.

5 Model Performance Enhancements (Task Order 4)

As part of this task, AECOM reviewed the structure of the TPB Version 2.3 travel demand forecasting model (build #28) and evaluated strategies designed to reduce the computing time for a model run by taking full advantage of multi-core and multi-machine distributed computing environments. AECOM used a combination of Cube Cluster and native Windows techniques, implemented in batch files, to accomplish the parallelization described in this chapter. These strategies were designed to fully utilize the capabilities of the Cube Cluster software. Cube Cluster utilizes multiple cores on a single machine or multiple machines to distribute the computing tasks and thus reduce run times. Multi-threading is limited to a few Voyager procedures (HIGHWAY and MATRIX), but parallel executions of independent tasks can be performed for most programs. In such cases, AECOM designed a strategy to effectively utilize available computing resources without using Cube Cluster. AECOM examined the output of the model in order to ensure that these strategies or model performance enhancements produce accurate and consistent output.⁷

The following sections present the model processes identified for performance enhancements, the implementation of the enhancements, a comparison of model output before and after these enhancements, and a quantification of time savings.

5.1 Model Processes Identified for Distributed Computing

AECOM reviewed Cube Cluster's capabilities to identify the most effective distributed computing methods. Cube Cluster provides two types of distributed computing features. They are:

1. ***Intra-step distributed processing*** (IDP): According to the Cube Voyager Reference Guide (ver. 5.1.3), IDP breaks up zone-based processing in a single step into zone groups that can be processed concurrently on multiple computing nodes (p. 976). As of Cube Voyager 5.1, only steps running under the HIGHWAY and MATRIX modules of Voyager can make use of IDP. Furthermore, IDP requires that there be zone independence in terms of calculations. For example, since the process of building highway paths and skimming highway path attributes between a given origin and destination is done using HIGHWAY and the calculations are zone independent, the skimming process can make use of IDP.
2. ***Multi-step distributed processing*** (MDP): According to the Cube Voyager Reference Guide (ver. 5.1.3), MDP breaks up blocks of one or more modeling *steps* and distributes them to multiple computing nodes for processing (p. 977). Unlike IDP, which can be used only with HIGHWAY and MATRIX steps, MDP can be used for any modeling step. Whereas IDP requires *zone* independence, MDP requires *step* independence (p. 977). For example, the peak and off-peak highway assignment modules are independent of each other and thus can be executed at the same time using MDP. Additionally, in this case, because the computing uses the HIGHWAY

⁷ However, as discussed later in this memo, using Cube Cluster can have a small effect on modeled outputs, such as estimated VMT, and the rounding appears to be a function of the number of cores used.

module, one can invoke IDP along with MDP, enabling the computer to utilize the available computing resources even more effectively.⁸

While it would be ideal to use as many IDP process threads as possible, MWCOG observed that the model results are different when the model is re-run with a different number of IDP process threads.^{9 10} To maintain model run consistency, MWCOG recommends that four IDP process threads be used for all model runs unless the computing resources cannot accommodate four threads. Hence, AECOM limited its tests to four IDP process threads.¹¹

AECOM received build 28 of the TPB Version 2.3 Travel Model (2.3.28) on August 23, 2011 from TPB staff. Prior to giving the travel model to AECOM, TPB staff had already implemented some Cube Cluster (only IDP) in the regional travel model, as described below:

- HIGHWAY steps
 - Traffic assignment (Highway_Assignment.s)
- MATRIX steps
 - Fare development (MFARE2.s)
 - Time of day processing
 - Time-of-Day.s
 - Misc_Time-of-Day.s
 - Preparation for traffic assignment (Prepare_Trip_Tables_for_Assignment.s)

AECOM reviewed the Voyager scripts and Windows batch files that are part of this model, with the goal of identifying other model processes that could be further parallelized without any changes to model output or fidelity. The following processes were identified for further parallelization efforts:

1. Highway and Transit Skimming procedures
2. Trip Distribution
3. Mode Choice
4. Highway and Transit Assignment procedures.

Figure 5-1 presents the model processes identified for further distributed computing and the following section discusses each step identified in conjunction with the strategy used to reduce computing time

⁸ It should be noted, however, that, while IDP supports loop statements, MDP does not. So, when implementing MDP in a script with a LOOP block, the loop must be decomposed, such that the script will now have duplicative blocks of code, which can be more challenging to maintain.

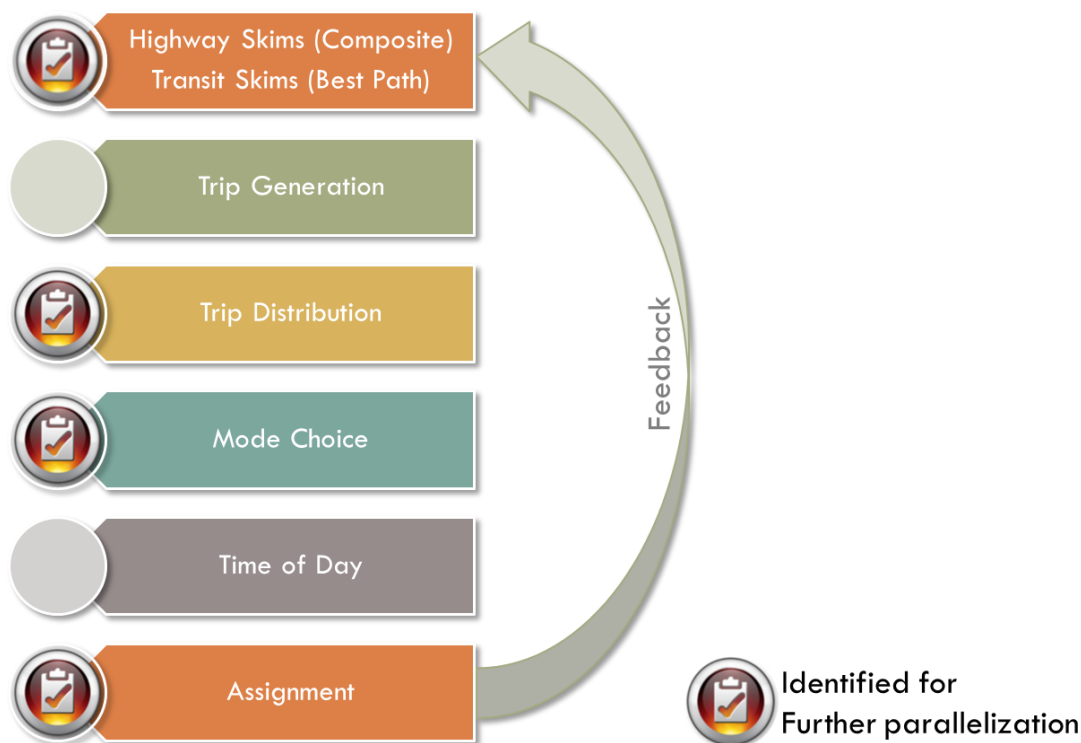
⁹ Due to rounding that occurs when using Cube Cluster, some model output metrics, such as estimated VMT, will vary slightly when Cube Cluster is used. Although the difference is small at the regional level (ca. 1-3 hundredths of a percent), it can be as much as 20% on some links. See, for example, slides 25-30 of Milone, R., & Moran, M. (2011, May 20). 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.

¹⁰ Furthermore, the rounding is a function of the number of cores used. See, for example, Martchouk, M. (2011, October 11). Cube Cluster Testing. Memorandum.

¹¹ Due to the use of IDP and MDP together, at times, there are eight concurrent cores/threads running.

for that step. AECOM used a combination of MDP, IDP, and native Windows techniques (implemented in batch files) to accomplish the parallelization.

Figure 5-1: Model Processes Identified for Further Distributed Computing

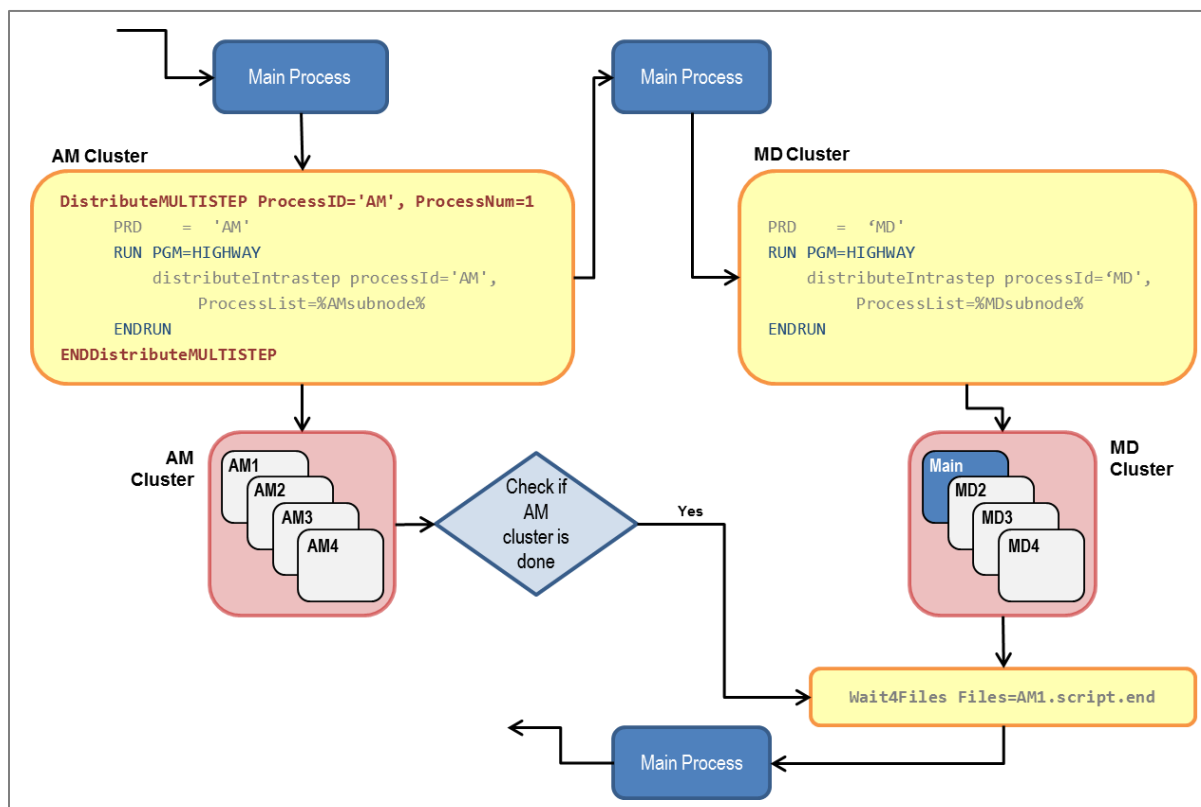


Highway and Transit Skims: The computing time for the highway skimming step was reduced by processing time period data in parallel instead of processing them sequentially. The computing time for the transit skimming step was reduced by processing the skims for the line-haul path options (i.e., commuter rail, bus, Metrorail with bus, and Metrorail only) in parallel. The highway and transit skimming procedures provide inputs to the mode choice model, namely the peak (AM) and the off-peak (mid-day) skims. Since the skims for the time periods and the skims for the transit line-haul paths are independent of one another, the peak and off-peak skims can be processed together using multi-step distributed processing or native Windows techniques. Along with MDP, IDP processing was also implemented for the highway skimming procedure in order to expedite the processing within each period. The recommended model setup uses four (4) intra-step processes within each of the two (2) multi-step processes. The resulting time savings are highlighted below:

Highway Skims Time Savings: about 40% = 3 minutes / feedback iteration
Transit Skims Time Savings: about 50% = 30 minutes / feedback iteration

Figure 5-2 depicts the distributed processing concepts used in the Highway Skims procedure and how these concepts interact with each other.

Figure 5-2: Highway Skims Procedure with Distributed Computing (both MDP and IDP)



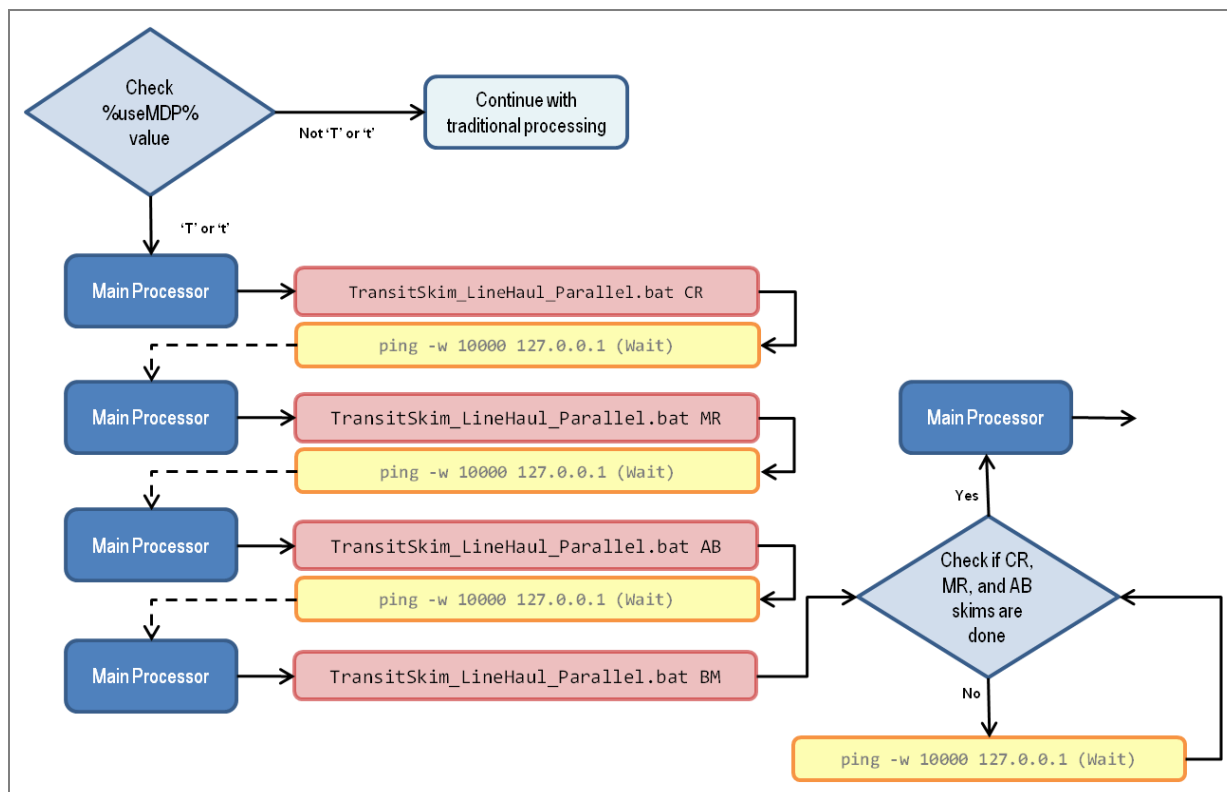
As shown in Figure 5-2, the highway skimming procedure’s main process or the master thread launches four (4) sub processes or slave threads (Cluster nodes), named AM1, AM2, AM3, and AM4, to process the AM highway skims. Once the information required to process the AM highway skims is passed to these slave threads, the master thread takes back control and launches three (3) slave threads (Cluster nodes), named MD2, MD3, and MD4, to process the mid-day highway skims. Only three slave threads are launched since the master uses itself as one of the threads to process the mid-day highway skims. Once the mid-day highway skims are processed, the master thread checks for the completion of the AM skimming procedure. This is accomplished by checking for a “process completion” file that is created by the AM slave threads upon completion of their processing. Thus the peak (AM) and the off-peak (mid-day) highway skims are processed using the MDP parallel processing procedures.

Note that the main process checks several Windows environment variables to enable or disable parallel processing. If the ‘useMDP’ environment variable is not defined or set to false (‘F’), the highway skimming process executes the peak and off-peak highway skims sequentially. In addition, if the ‘useIDP’ environment variable is not true (‘T’), the highway skimming process does not use slave threads and uses only the master thread to process the skims. The number of threads used by IDP for the peak and off-peak highway skimming procedures is also controlled by environment variables. These variables are: AMsubnodes, and MDsubnodes respectively. The default value for AMsubnodes is ‘2-4’ and the default value for MDsubnodes is ‘2-4’. Since the master thread is used to process the mid-day highway skims, the MDsubnodes should always start with a ‘2’. If sixteen (16) IDP threads are available, the

AMsubnodes environment variable should be set to '2-8' and the MDsubnodes variable should be set to '2-8'.

Figure 5-3 depicts the distributed processing concepts used in the Transit Skims procedure and how these concepts interact with each other.

Figure 5-3: Transit Skims Procedure with Distributed Computing



While the highway skimming process uses Cube Cluster’s MDP capabilities, the transit skimming process does not. As shown in Figure 5-3, it uses simple Windows batch scripting to launch all four (4) transit skimming procedures in parallel. The transit skimming batch file checks the ‘useMDP’ environment variable and launches the skims in parallel if the variable is set to true (‘T’). If the variable is not set to true, the transit skims are processed in series. If the ‘useMDP’ environment variable is set to true, the main processor (the Windows batch file) launches the transit skimming procedures for all four line-haul paths concurrently and waits for these processes to terminate successfully. When each process terminates successfully, a “process completion” file is written indicating successful completion to the master thread. The master thread then continues with the execution of the next set of procedures in the model.

In our initial tests, the transit skimming procedure terminated abnormally due to a sharing violation. This sharing violation occurred because the original skimming procedure outputs a temporary file using the same file name for each line-haul mode. When run in parallel, one of the processes locks this file and the other processes cannot create it; causing a sharing violation. AECOM modified the script to output a

unique intermediate file for each line-haul mode, thus avoiding the issue.¹² Also note that the batch process includes a “ping” command to force the process to wait ten second between executions. This helps avoid conflicts or collisions in reading license files and opening common data files.

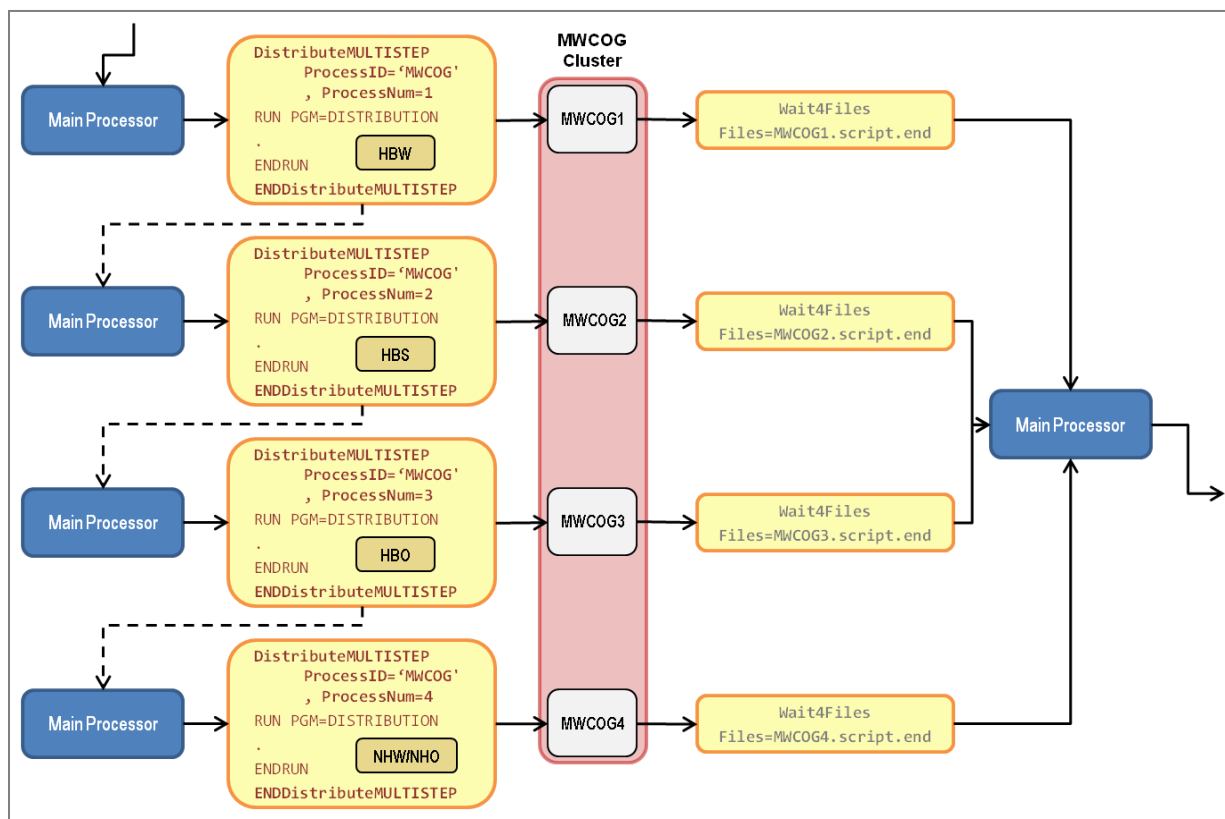
Trip Distribution: The computing time for this step was reduced by processing trip purposes in parallel instead of processing them sequentially. Since each trip purpose is independent of the others, the trip purposes can be processed in parallel. In addition, the trip distribution procedures use a significant amount of matrix computations; each of these computations is independent and thus can be processed in parallel using IDP. The trip distribution procedures for Home-Based Work (HBW), Home-Based Shop (HBS), Home-Based Other (HBO), and Non-Home-Based (NHB) are processed using MDP. The commercial vehicles and truck trip distribution procedures are not processed in parallel in order to limit the number of MDP processes to four (4). These could be processed in parallel if additional cores are available, but since the commercial vehicles and truck trip distribution processes converge relatively quickly, there is no significant benefit of processing them in parallel. The resulting time savings from parallel trip distribution is highlighted below:

Trip Distribution Time Savings: about 50% = 7 minutes / feedback iteration

Figure 5-4 presents the various processes in the Trip Distribution procedure with distributed computing and how they interact with each other. Trip distribution procedure’s main process or the master thread launches four (4) sub processes or slave threads named ‘MWC0G 1-4’ to distribute the trips for each trip purpose. This is accomplished using Cube’s MDP capabilities. Once the slave threads finish processing, the master thread regains control and continues with the execution of the next set of procedures in the model. Similar to the highway and transit skimming procedures, the trip distribution procedure checks the ‘useMDP’ environment variable to implement serial or parallel processing. If this variable is not set to true (‘T’), trip purposes are processed serially.

¹² So, for example, in the transit skimming script, we changed “mato = transit.skm” to “mato = transit_mr.skm.”

Figure 5-4: Trip Distribution Procedure with Multi-Step Distributed Processing

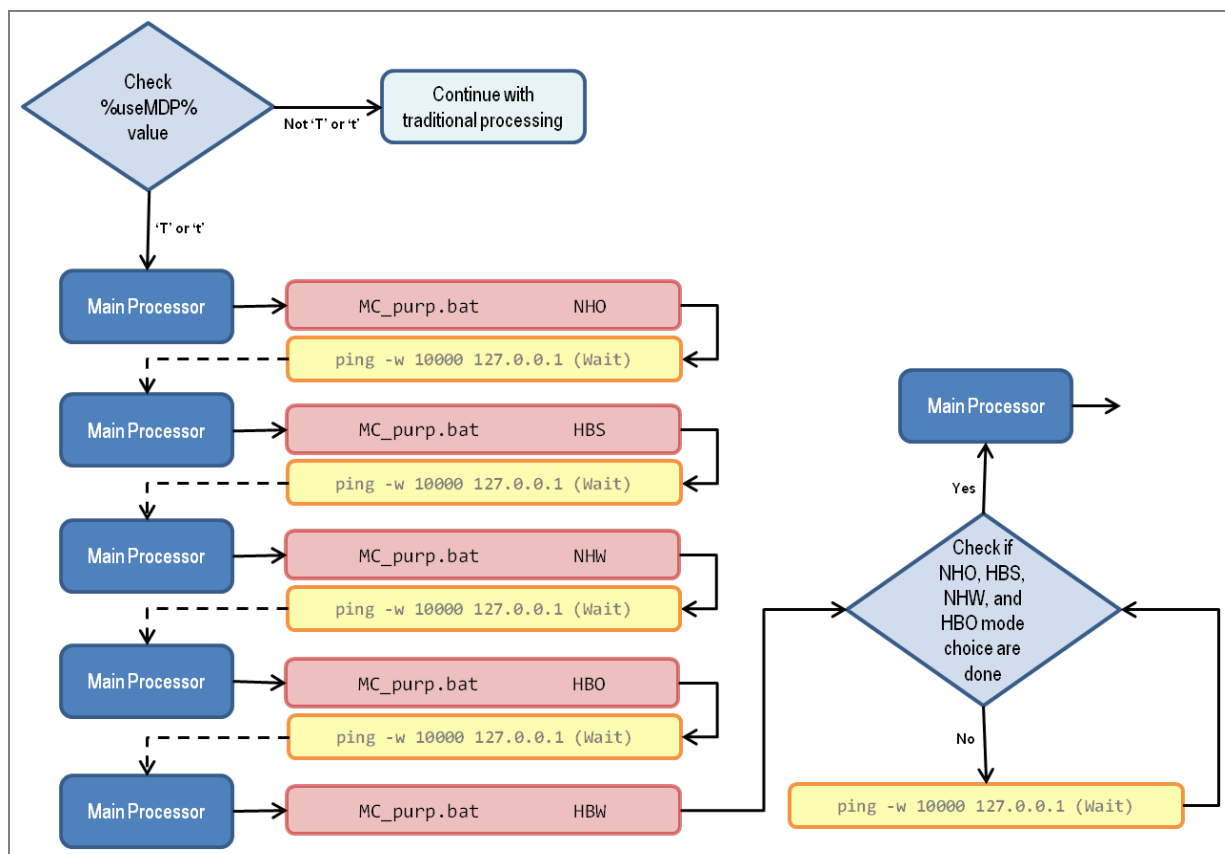


Mode Choice: The computing time for this step was reduced by processing trip purposes in parallel instead of processing them sequentially. While each trip purpose’s mode choice application (AEMS) is independent of the rest, the applications use the same input files. Since the input files are not modified, AEMS does not lock these files; however, FORTRAN-based applications such as AEMS are known to lock the files during the file open process. This process lasts less than a second but causes other instances of AEMS reading the same input file to experience an abnormal termination. In order to avoid these abnormal terminations, a ten (10) second delay was introduced between each launch of the mode choice application. The introduction of this delay overcomes the issue and the mode choice applications execute without any errors. The resulting time savings from parallel mode choice is highlighted below:

Mode Choice Time Savings: about 70% = 35 minutes / feedback iteration

Figure 5-5 presents the various processes in the Mode Choice procedure with distributed computing and how they interact with each other.

Figure 5-5: Mode Choice Procedure with Distributed Computing



Since the mode choice model is not a Cube-based application, it would need to be called through a special Cube script to take advantage of MDP procedures. Unfortunately, the mode choice software crashes if multiple instances attempt to open the same file at the same time. To avoid this problem, a simple Windows batch file process was used to execute the AEMS program for each trip purpose concurrently. This batch file launches all five (5) mode choice applications with a slight delay between each execution, waits for each process to create a “process completion” file and continues with the execution of the next steps in the model. The mode choice batch file checks the ‘useMDP’ environment variable and launches the mode choice applications in parallel if the variable is set to true (‘T’). If the variable is not set to true, each trip purpose is executed sequentially.

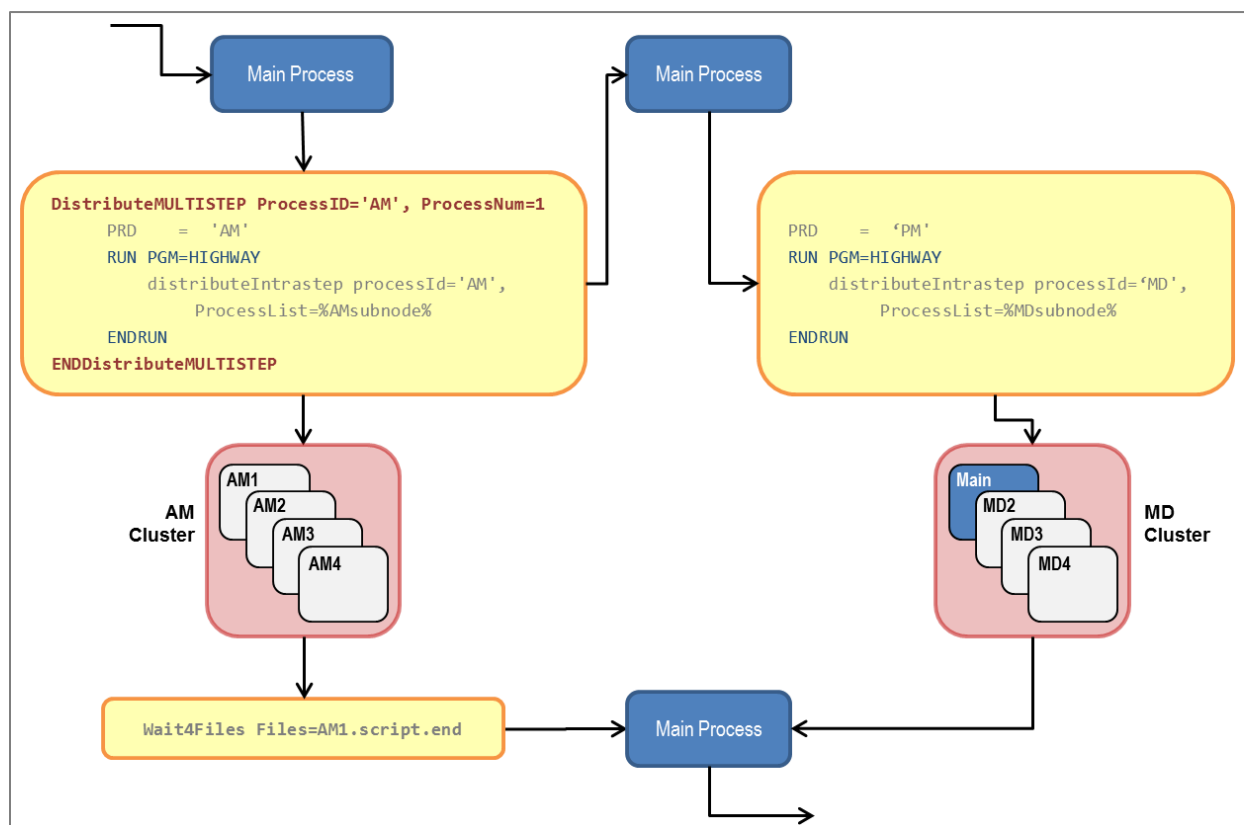
Highway and Transit Assignment: The computing time for this step was reduced by processing highway assignment time periods in parallel and by processing transit line-haul path assignments in parallel rather than sequentially. The four highway assignment time periods, namely AM, mid-day, PM, and night, could be processed together. Processing these time periods, however, is not computationally efficient since the highway assignment for mid-day and night time periods converge in fewer iterations than the AM or PM time periods. If all four are launched at the same time, the cores dedicated to the mid-day and night assignments will finish early and wait for the AM and PM assignments to finish.

In order to utilize computing resources more effectively, AECOM modified the existing model setup to process the AM and PM time periods together using MDP and then process the mid-day and the night time periods together after the peak assignments are completed. In addition, each of these MDP processes uses four (4) IDPs. This configuration limits the number of IDP process threads used by the assignment to four (4) to be consistent with MWCOG’s processing recommendations while taking advantage of computing environments that include eight (8) cores or hyper threads. This model setup provided the greatest computational benefit while at the same time reproducing the results of the sequential process. The highway and transit assignment time savings are highlighted below:

Highway Assignment Time Savings: about 40% = 45 minutes / feedback iteration
Transit Assignment Time Savings: about 50% = 15 minutes / model execution

Figure 5-6 presents the various processes in the Highway Assignment procedure with distributed computing and how they interact with each other.

Figure 5-6: Highway Assignment Procedure with Distributed Computing

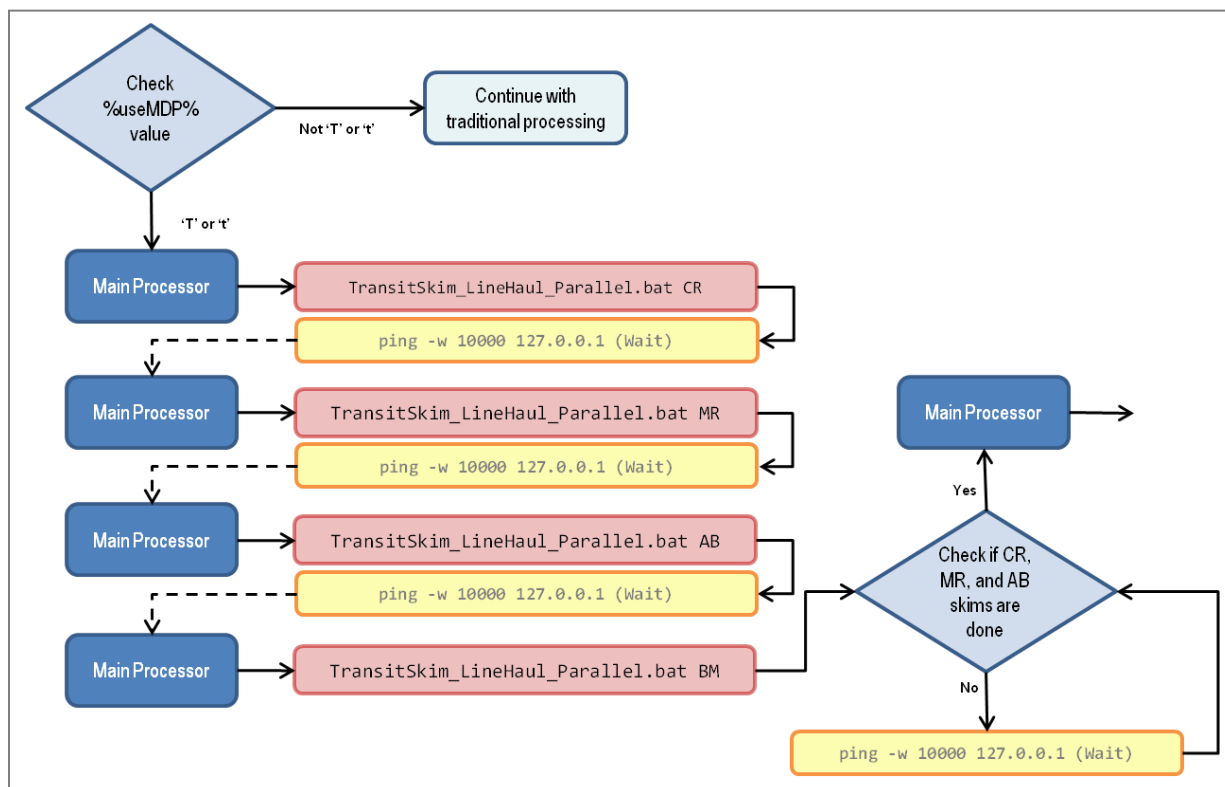


As shown in Figure 5-6, the highway assignment procedure’s main process or the master thread launches four (4) sub processes or slave threads (Cluster nodes) named AM to perform the AM highway assignment. Once the information required to assign the AM highway traffic is passed to these slave threads, the master thread takes back control and launches three (3) slave threads (Cluster nodes) named MD to process the PM highway assignment. Only three slave threads are launched since the

master uses itself as one of the threads to process the PM highway assignment. Once the PM highway assignment converges, the master thread checks for the completion of the AM assignment procedure. This is accomplished by checking for a “process completion” file that is created by the AM slave threads upon completion of their processing. Thus the AM and the PM highway assignment is performed in parallel using the MDP procedures with four (4) IDP threads per process for a total of eight (8) processing threads or cores. If the computer does not include eight cores, the Windows operating system shares the available cores among the two assignments. In most cases, this will still be faster than running the two assignments sequentially. When the peak period assignments are complete, the same process is used to execute the mid-day and night highway assignments.

As discussed previously, the highway assignment procedure uses the ‘useMDP’ and ‘useIDP’ environment variables to control parallel processing. The AMsubnodes and MDsubnodes variables are also used to control the number of IDP threads. The default value for AMsubnodes is ‘1-4’ and the default value for MDsubnodes is ‘2-4’.

Figure 5-7: Transit Assignment Procedure with Distributed Computing



While the highway assignment process uses Cube Cluster’s MDP capabilities, the transit assignment process uses the Windows batch script depicted in Figure 5-7 to launch all four (4) line-haul path-based transit assignment procedures at the same time. The transit assignment batch file checks the ‘useMDP’ environment variable and launches the assignments in parallel if the variable is set to true (‘T’). The main processor (the Windows batch file) launches the transit assignment procedures in parallel and waits for these processes to finish successfully. When the process terminates successfully a “process

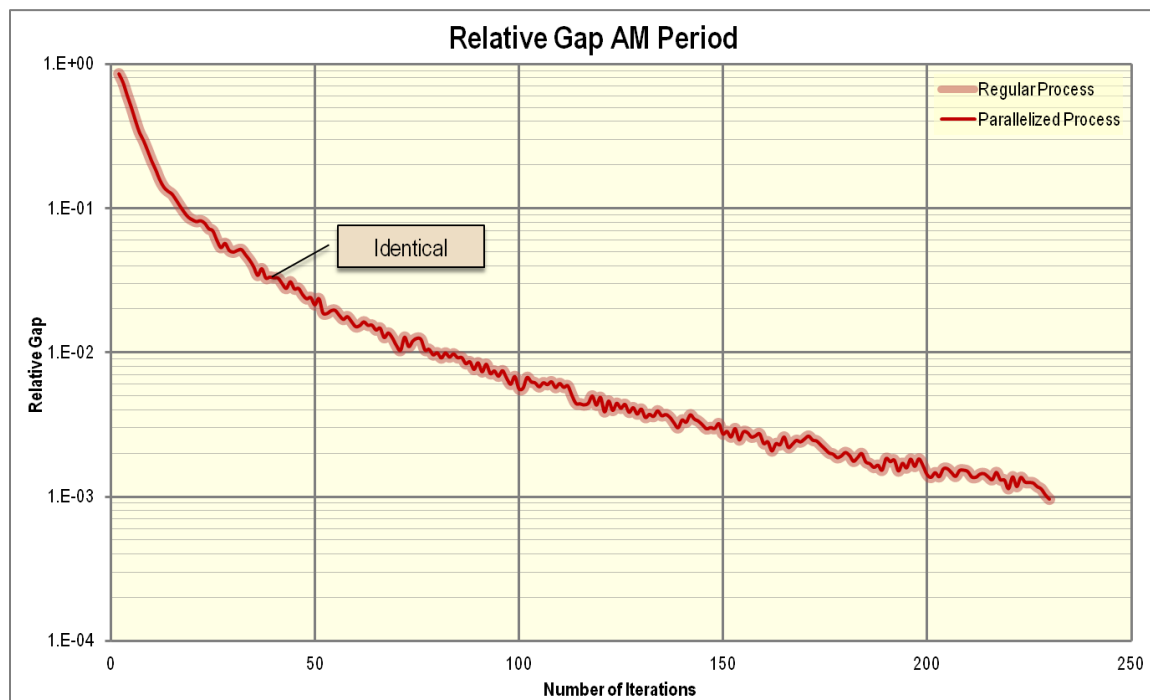
completion” file is written indicating successful completion to the master thread. The master thread then continues with the execution of the next set of procedures in the model.

5.2 Model Results Validation

In order to ensure that the model produces the same results with the performance enhancements, AECOM performed model executions with and without the parallelization procedures. These tests revealed that the model performance enhancements produce identical results. AECOM compared the highway and transit skim matrices, trip distribution matrices, and mode choice model output trip matrices and the compared matrices were identical. In addition, AECOM compared the Vehicle Miles of Travel (VMT) from the two model executions to ensure that they produce the same VMT.

During the validation process, AECOM also compared traffic assignment convergence using the relative gap measure. This test revealed that while AM, PM, and the night traffic assignments’ convergence plots were identical as presented in Figure 5-8, Figure 5-9, and Figure 5-11, but the traffic assignment convergence was not identical for the mid-day period as shown in Figure 5-10. Further investigation revealed that the mid-day traffic assignments for the two model executions converge at the same iteration but are not identical because Cube generates zero gaps at different iterations. The night traffic assignment convergence was identical; however, Cube generates zero gaps for the same iteration of the traffic assignment. These aberrations in gap calculations should be addressed by Citilabs.¹³ MWCOG reported the gap calculation anomalies to Citilabs for resolution or clarification.

Figure 5-8: Comparison of Relative Gap for Parallel and Serial Model Executions (AM Peak)



¹³ TPB staff informed Citilabs about these cases where the gap reached a value of zero. Citilabs concluded that the gap values of zero were due to “numerical rounding,” indicating that the gap value must be smaller than the 15 digits of the double precision numbers (personal communication from Citilabs, January 30, 2012).

Figure 5-9: Comparison of Relative Gap for Parallel and Serial Model Executions (PM Peak)

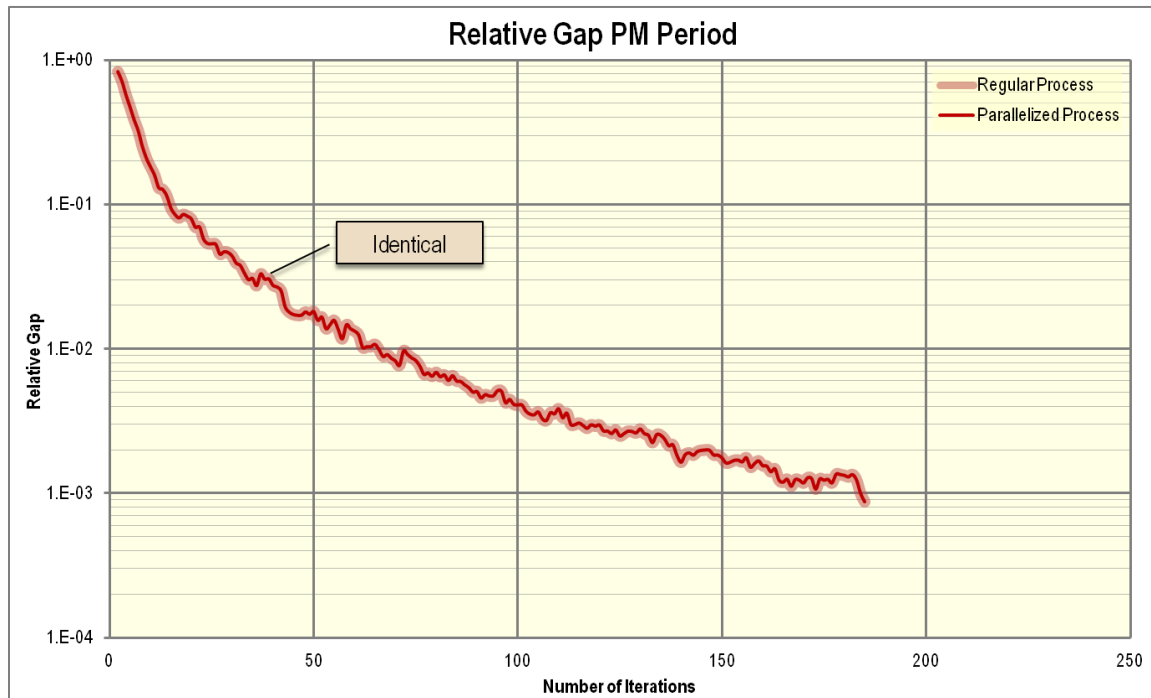


Figure 5-10: Comparison of Relative Gap for Parallel and Serial Model Executions (Mid-day)

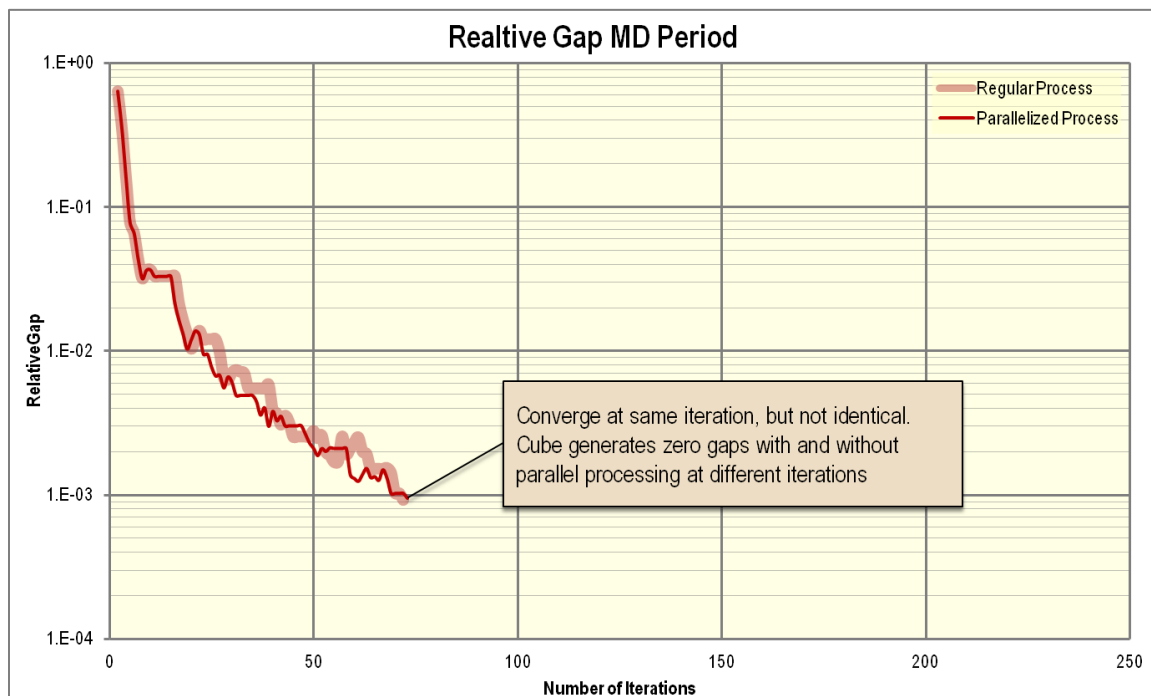
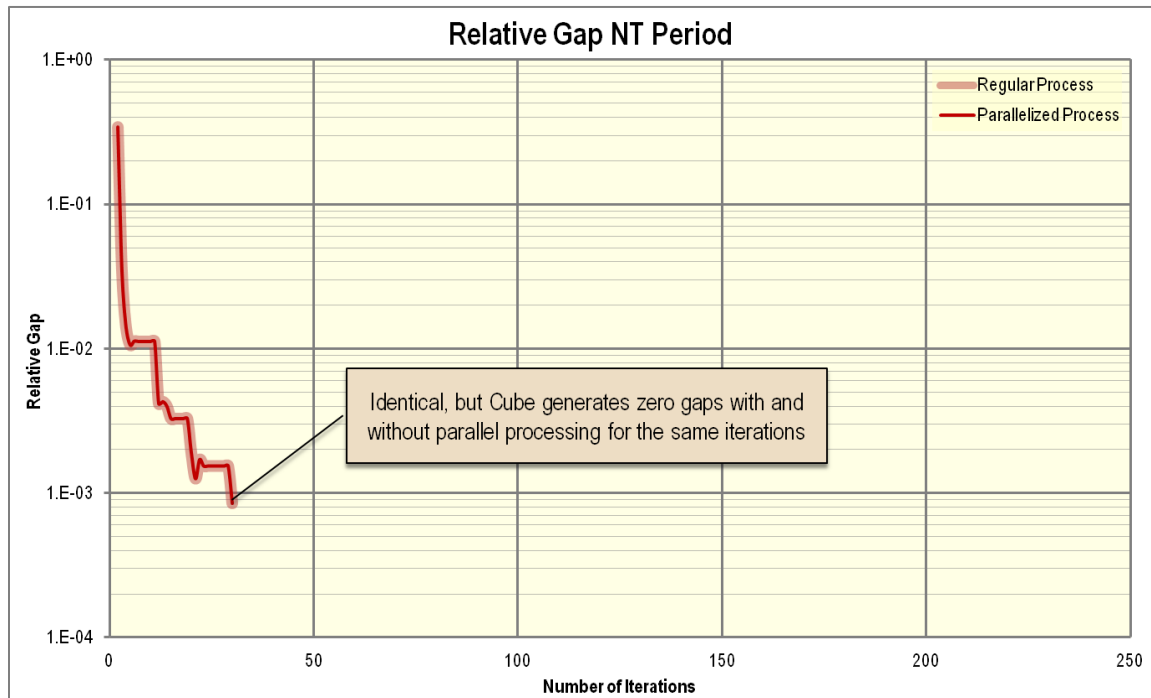


Figure 5-11: Comparison of Relative Gap for Parallel and Serial Model Executions (Night)



5.3 Benefits Quantification

AECOM executed the model with and without the performance enhancements in order to estimate the performance benefits of the enhancements. These tests revealed that parallel processing reduces processing time by about 40 percent. Table 5-1 presents the time savings for each feedback iteration of model while Table 5-2 presents the time savings for each step in the first feedback iteration of the model.

Table 5-1: Time Savings in Hours for each Feedback Iteration of the Model

Iteration	Regular Process	Parallelized Process	Savings	Percent Savings
Pump Prime	5:41	3:48	1:53	33%
Iteration 1	5:06	3:06	2:00	39%
Iteration 2	5:57	3:39	2:18	39%
Iteration 3	5:31	3:27	2:04	37%
Iteration 4	5:35	3:24	2:11	39%
Transit Assignment	0:30	0:15	0:15	50%
Full Run	28:20	17:39	10:41	38%

Table 5-2: Time Savings in Hours for each Step of the First Model Feedback Iteration

Iteration 1 Step	Regular Process	Parallelized Process	Savings	Percent Savings
Transit Skims	0:59	0:29	0:30	51%
Transit Fare	0:19	0:19		
Trip Generation	0:01	0:01		
Trip Distribution	0:14	0:07	0:07	50%
Mode Choice	0:51	0:16	0:35	69%
Auto Driver	0:08	0:08		
Time of day	0:26	0:26		
Hwy Assignment	2:00	1:15	0:45	38%
Hwy Skims	0:08	0:05	0:03	38%
Total	5:06	3:06	2:00	39%

5.4 Conclusions and Recommendations

The implementation of the performance enhancements required modifications to both the Windows batch processes and the Cube scripts. In order to implement MDP within Cube, some of the code in the scripts had to be replicated. This increases the level of effort required to manage the existing scripts and also when new scripts are to be created.

MWCOG model execution process contains procedures to capture all on-screen output and model error information to text files. The current MWCOG process uses ‘Tee’, an Open Source software, to direct the output to both the screen and to a log file. Since some of the serial processes are being parallelized as part of the model performance enhancements, multiple Windows batch files (or shells) need to be launched and ‘Tee’ captures only output from the main batch file. The on-screen output from the parallelized processes is lost and not written to the log file. AECOM modified the existing model setup to capture the output of each parallel process, perform checks to ensure that each of the parallel processes have successfully terminated, trap and records errors, and concatenate the on-screen output of each parallel process and append it to the log file.

AECOM recommends that the highway assignment procedures to assign non-HOV and HOV3+ trip tables be combined using a multi-class traffic assignment process. The current model performs a non-HOV assignment first and then uses the congested speeds from this assignment to perform the HOV3+ assignment. While the non-HOV traffic affects the HOV3+ travel, HOV3+ travel does not affect non-HOV travel (i.e. the additional congestion caused by HOV3+ vehicles on general purpose lanes does not impact non-HOV vehicles). Unless MWCOG believes that this two-step assignment process should not be changed, AECOM advises replacing the two-step assignment process with a single step multi-class assignment process. In addition to more accurately calculating congested speeds for all vehicle types, this conversion will reduce run times significantly. This modification would, however, require additional parameter estimation and sensitivity analysis.

6 Begin Converting TRNBUILD to PT (Task Order 5)

Task Order 5 was designed to gather aggregate transit ridership data for display purposes and explore the network generation and path building features of Public Transport (PT) to better estimate the overall effort required in converting the transit models from TRNBUILD to PT. The deliverables include the software updates needed to improve ridership summaries; the initial PT network data and path building tests; and an outline of the potential model impacts and changes to the support tools required by a PT conversion in the TPB Travel Model. A work plan for implementing the networks, tools, and models was also prepared.

6.1 Displaying Aggregate Ridership Data

Cube has a number of facilities to display transit ridership results for individual transit lines, but does not include a convenient way of aggregating ridership from all of the routes that share the same segment or link (e.g., “Orange Line A,” “Orange Line B,” and “Orange Line C”) and formatting these data so they can easily be displayed by ArcGIS or Cube software. The original plan was to upgrade the ArcLineSum program AECOM developed several years ago to generate Arcview shapefiles from the TPPlus transit ridership file. This approach was eventually abandoned for the following reasons:

- The ArcLineSum code was not consistent with current programming standards;
- The LineSum code was upgraded to current programming standards as part of Task Order 3;
- The LineSum program reads the same files and does the same aggregation as ArcLineSum;
- The new LineSum program is built on the TRANSIMS subroutine library which includes support for Arcview and other file formats; and
- Combining the functionality of LineSum and ArcLineSum into a single program minimizes duplication of effort and consolidates the software maintenance and documentation into one place.

The LineSum program functionality was expanded to include a Link Rider File group. This file group enables the LineSum program to generate one or more files of various formats based on a variety of selection criteria. The format options of primary interest include Arcview, dBase, and tab delimited. The selection options include mode and route name ranges. Since the input data can include peak and offpeak transit ridership files, a set of control keys was included to generate a variety of link performance statistics. The number of hours of peak period service and the peak vehicle capacity can be provided to convert the peak frequency of each route to the total number of runs, vehicle miles, vehicle hours, and total capacity of each link in the peak period. Similarly, the offpeak hours of service is used to estimate the quantity of service in the offpeak period. These values are coupled with a peak hour factor to generate an estimate of the peak hour volume and the link load factor.

Each output rider file includes the following data items:

- A node number
- B node number
- Peak period riders in the AB direction

- Peak period frequency in the AB direction
- Offpeak riders in the AB direction
- Offpeak frequency in the AB direction
- Daily riders in the AB direction
- Peak period riders in the BA direction
- Peak period frequency in the BA direction
- Offpeak riders in the BA direction
- Offpeak frequency in the BA direction
- Daily riders in the BA direction
- Peak period riders in both directions
- Peak period frequency in both directions
- Peak period link length
- Peak period link travel time
- Peak period passenger miles of travel
- Peak period passenger hours of travel
- Peak period vehicle miles of travel
- Peak period vehicle hours of travel
- Offpeak period riders in both directions
- Offpeak period frequency in both directions
- Offpeak period link length
- Offpeak period link travel time
- Offpeak period passenger miles of travel
- Offpeak period passenger hours of travel
- Offpeak period vehicle miles of travel
- Offpeak period vehicle hours of travel
- Daily riders in both directions
- Daily passenger miles in both directions
- Daily passenger hours in both directions
- Daily vehicle miles in both directions
- Daily vehicle hours in both directions
- Maximum directional riders in one hour
- Maximum directional riders per vehicle
- Station name at the A node
- Station name at the B node

If the link rider file is saved in Arcview format, XY coordinate data must be provided. At a minimum, a simple node file with the X and Y coordinate of each node is needed. In this case, a straight-line link between the A and B nodes is generated. Alternatively, the user can provide a link shape file to draw a shaped vector between the two nodes. The link shape file must be in Arcview shapefile format and include A-node and B-node fields that can be matched in either direction to link values in the transit

ridership file. The user can also provide coordinate projection information to specify the projection system (e.g., stateplane, UTM, LatLong) used to output the Arcview shapefile.

6.2 PT Conversion Analysis

The TRNBUILD network, analysis, and display tools described throughout this document are short-term solutions to address immediate needs while MWCOG pursues its longer term plan to convert the transit network coding, transit path-building and transit assignment software from TRNBUILD to Public Transport (PT). The full conversion from TRNBUILD to PT is planned for next fiscal year, but a number of useful and informative activities were completed in this fiscal year to initiate the process. The purpose of this task was, therefore, to move the process forward sufficiently enough to gain a clearer understanding of the level of effort involved, the implications for transit path building and mode choice calibration, and how MWCOG and AECOM might best contribute to this process.

The first step in this process was to convert the primary transit line files from TRNBUILD to PT for initial path building purposes. We then tested several configuration options within PT to identify the access generation and path building parameters that most closely replicate the paths built using the current MWCOG access generation tools and TRNBUILD procedures. If the majority of paths are identical, MWCOG could begin taking advantage of some of the benefits of switching to PT, such as on-screen transit path-building.

The initial exercise identified the level of effort needed to convert the transit access components of the model and how different the resulting paths and path attributes are likely to be. Given this information, AECOM estimated the potential impact to the mode choice model calibration and designed coding techniques or software tools or scripts to compensate for these differences. The proposed design distinguishes between walk, bus, and drive access to Metrorail stations and other transit services. It also effectively builds paths featuring the four transit modes currently included in the mode choice process (commuter rail, bus, Metrorail+bus, and Metrorail only).

These findings were packaged into a work plan for next fiscal year that estimates the level of effort and costs for both AECOM and MWCOG to complete the conversion process to PT and update the models and modeling scripts as appropriate.

6.3 Convert TRNBUILD Network to PT

The PT conversion has two primary components. The first is transit network and access link generation. The second is the path building and trip loading function. The objective of this first work effort was to convert the TRNBUILD routes and access links from the existing process to PT for the purpose of performing side-by-side tests of the path building algorithms. In other words, it was an attempt to eliminate any analysis complications caused by the PT network generation process on an assessment of the path building functions.

The MWCOG process uses GIS tools to generate the “transit-only” links and nodes required by TRNBUILD. This includes rail links and stations for Metrorail, commuter rail and light rail systems. It may also add nodes or links for bus and streetcar systems as well. Bus Rapid Transit (BRT) lines and bus routes that use local streets to serve neighborhoods often require additional links and nodes beyond

those provided by the standard roadway network. In addition, it is often easier to code a future year transit alternative using a whole new set of links and nodes rather than looking up all of the nodes the route would use in the future year roadway network or splitting links to add nodes for new stops or stations along existing links¹⁴.

TRNBUILD enables the user to add the “transit-only” network links and nodes through input text files during the path building or assignment process. The LINK, SUPPLINK and XY data types perform this function. In PT all links and nodes need to be included in the network file. In other words, the network file includes both highway and transit links and nodes. This requires greater consistency between the networks and methods of coding link attributes that restrict use to a specified set of modes (e.g., auto traffic cannot use rail links). The Network program is used to build the combined network required as input to the Public Transport program. The network needs to include all of the zones, links, and nodes (i.e., stop or station locations) required by the transit line files and access generation process.

6.3.1 PT Network

For this first exercise, the transit-only network data used by TRNBUILD is merged with the highway network to create the transit network file required by PT. The MWCOG process generates the following transit-only input files:

- Met_Link.tb, Met_Node.tb, Met_PNRN.tb //---- Metrorail ----
- Com_Link.tb, Com_Node.tb, Com_PNRN.tb //---- commuter rail ----
- LRT_Link.tb, LRT_Node.tb, LRT_PNRN.tb //---- light rail ----
- New_Link.tb, New_Node.tb, New_PNRN.tb //---- new transit modes ----
- Bus_PNRN.tb //---- bus modes

These files are merged with the highway network using a Network program script. Fortunately, the MWCOG process that creates these files writes the “free format” data in fixed column ranges. This makes it relatively straight forward to extract data from the input files and merge it into the network file or re-write a text file in the PT non-transit data format.

Figure 6-1 shows the script used to merge the transit-only link and node data into the highway network. It reads four link files with A, B, MODES, DISTANCE, and SPEED fields. It then reads four node files with station numbers and coordinates and five node files with park-n-ride lot numbers and coordinates. In the link merge phase, it converts the distance data to miles and calculates the transit travel time. Note that the network for any given analysis year may or may not include data for each of these transit modes. If a particular mode is not used, the file will be blank and Cube will generate a series of warning messages. If the batch processing script is checking the return code from the Network program, it should recognize a return code less than 2 as acceptable.

¹⁴ WMATA has traditionally implemented transit alternatives using a separate set of transit-only links and station nodes for each new line segment they wish to test in various combinations. This approach avoids the need to modify the MWCOG highway or transit network to insert new infill stations or route extensions and simplifies the coding processing. It has the disadvantage of separating the new transit facilities from the underlying MWCOG network.

Figure 6-1: Cube Script to Convert Transit-Only Network

```

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

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

LINKI[2] = "Inputs\Met Link.TB",
          VAR = A,12-17, B,20-26, MODES,34-35, DISTANCE,42-47, SPEED,66-71, REV=1
LINKI[3] = "Inputs\Com Link.TB",
          VAR = A,12-17, B,20-26, MODES,34-35, DISTANCE,42-47, SPEED,66-71, REV=1
LINKI[4] = "Inputs\LRT Link.TB",
          VAR = A,12-17, B,20-26, MODES,34-35, DISTANCE,42-47, SPEED,66-71, REV=1
LINKI[5] = "Inputs\New Link.TB",
          VAR = A,12-17, B,20-26, MODES,34-35, DISTANCE,42-47, SPEED,66-71, REV=1

;---- transit only nodes ----

NODEI[2] = "Inputs\Met_Node.TB",
          VAR = N,9-14, X,18-27, Y,31-40
NODEI[3] = "Inputs\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

;---- park-n-ride nodes ----

NODEI[6] = "Inputs\Met_PNRN.TB",
          VAR = N,9-14, X,18-27, Y,31-40
NODEI[7] = "Inputs\Com_PNRN.TB",
          VAR = N,9-14, X,18-27, Y,31-40
NODEI[8] = "Inputs\LRT_PNRN.TB",
          VAR = N,9-14, X,18-27, Y,31-40
NODEI[9] = "Inputs\New_PNRN.TB",
          VAR = N,9-14, X,18-27, Y,31-40
NODEI[10] = "Inputs\Bus_PNRN.TB",
          VAR = N,9-14, X,18-27, Y,31-40

;---- link processing ----

MERGE RECORD=TRUE

PHASE = LINKMERGE
  IF (SPEED >0)
    DISTANCE = DISTANCE / 100.0
    TRANTIME = DISTANCE * 60.0 / SPEED
  ELSE
    TRANTIME = AMHTIME
  ENDIF
ENDPHASE
ENDRUN

```

6.3.2 PT Transit Lines

Once the network includes the links and nodes required to support the transit service, the transit routes and access links can be converted. MWCOC defines transit routes for two time periods (AM peak period and offpeak) using 10 mode-specific files (i.e., MODE1AM.tb, MODE2AM.tb, ... MODE10AM.tb, MODE1OP.tb, ... MODE10OP.tb). The TRNBUILD line file format is defined using a free-format text file that utilizes multiple lines of text to define a single route. Fortunately, the PT line file format is very similar to the TRNBUILD format. All that needs to change is a few of the keywords buried in the text to convert the file. This can be implemented using Cube's search and replace capabilities. Two examples of

these conversion routines are shown in Figure 6-2. These scripts use the Matrix program to read each record from the input TRNBUILD line file and replace “Owner” with “LongName”, “Freq” with “Headway”, and the “Oneway” flag from Y/N to T/F. This last change is not absolutely necessary, but improves consistency with the Cube software documentation. The output file has the extension “*.Lin” to indicate it is in PT line file format.

Figure 6-2: Cube Script to Convert TRNBUILD Line Files to PT Format

```

RUN PGM=MATRIX
FILEI RECI = "inputs\Model1AM.TB"
FILEO PRINTO[1] = "Model1AM.Lin"
REP1 = REPLACESTRIC (RECI, 'OWNER', 'LONGNAME', 1)
REP2 = REPLACESTRIC (REP1, 'FREQ', 'HEADWAY_R', 2)
REP3 = REPLACESTRIC (REP2, 'FREQ', 'HEADWAY', 2)
REP4 = REPLACESTRIC (REP3, 'ONEWAY= Y', 'ONEWAY=T', 1)
REP5 = REPLACESTRIC (REP4, ' ONEWAY= N', 'ONEWAY=F', 1)
PRINT LIST=REP5, PRINTO=1
ENDRUN

RUN PGM=MATRIX
FILEI RECI = "inputs\Model1OP.TB"
FILEO PRINTO[1] = "Model1OP.Lin"
REP1 = REPLACESTRIC (RECI, 'OWNER', 'LONGNAME', 1)
REP2 = REPLACESTRIC (REP1, 'FREQ', 'HEADWAY R', 2)
REP3 = REPLACESTRIC (REP2, 'FREQ', 'HEADWAY', 2)
REP4 = REPLACESTRIC (REP3, 'ONEWAY= Y', 'ONEWAY=T', 1)
REP5 = REPLACESTRIC (REP4, 'ONEWAY= N', 'ONEWAY=F', 1)
PRINT LIST=REP5, PRINTO=1
ENDRUN

```

6.3.3 PT Access Links

The most significant difference between TRNBUILD and PT is in the way transit access links are defined and created. For this phase of the analysis, the access links created for the TRNBUILD implementation were converted directly to PT format. In other words, the PT generation logic was not used to construct the transit access links.

The procedure MWCOG uses to create the TRNBUILD access data generates separate files for each data type and each file has data items in fixed column ranges. This makes it possible to use the Network program to read variables in specific columns, calculate new variables, and print a new record to an output file in PT non-transit format. Figure 6-3 shows the script used to convert the 24 transit access files created by the MWCOG process. This includes the sidewalk and walk access links used to walk to and from zones and between transit lines. The next group of files is the walk links between transit stations and nearby bus stops. This is followed by the drive access (park-n-ride and kiss-n-ride) links by time period (AM peak period and offpeak). The bus and drive access links are processed as a group in order to avoid fatal errors that would otherwise be generated by the Network program if blank input files are processed independently.

Figure 6-3: Cube Script to Convert TRNBUILD Access Links to PT Format

```

;---- sidewalk links ----
RUN PGM = NETWORK
LINKI[1] = "sidewalk.asc",
VAR = A,11-16, B,18-23, MODE,30-31, SPEED,39,1, DIST,56-62

```

```

FILEO PRINTO[1] = "sidewalk.nt"

PHASE = INPUT FILEI = LI.1
  IF (A > 0)
    DIST = DIST / 100.0
    COST = DIST * 60.0 / SPEED
    PRINT LIST="NT LEG=",A(5.0),"-",B(5.0)," MODE=",MODE(2.0),
      " SPEED=",SPEED(4.1)," ONEWAY=T DIST=",DIST(6.2),
      " COST=",COST(7.2) PRINTO=1
  ENDIF
ENDPHASE
ENDRUN
;---- walkacc links ----
RUN PGM = NETWORK
  LINKI[1] = "walkacc.asc",
  VAR = A,11-16, B,18-23, MODE,40-41, SPEED,49-50, DIST,57-63
  FILEO PRINTO[1] = "walkacc.nt"

  PHASE = INPUT FILEI = LI.1
    IF (A > 0)
      DIST = DIST / 100.0
      COST = DIST * 60.0 / SPEED
      PRINT LIST="NT LEG=",A(5.0),"-",B(5.0)," MODE=",MODE(2.0),
        " SPEED=",SPEED(4.1)," ONEWAY=F DIST=",DIST(6.2),
        " COST=",COST(7.2) PRINTO=1
    ENDIF
  ENDPHASE
ENDRUN

;---- bus access to stations ----
RUN PGM = NETWORK
  LINKI[1] = "Met_Bus.tb",
  VAR = A,11-17, B,20-26, MODE,45-46, DIST,54,1, SPEED,63-64
  LINKI[2] = "Com_Bus.tb",
  VAR = A,11-17, B,20-26, MODE,45-46, DIST,54,1, SPEED,63-64
  LINKI[3] = "LRT_Bus.tb",
  VAR = A,11-17, B,20-26, MODE,45-46, DIST,54,1, SPEED,63-64
  LINKI[4] = "New_Bus.tb",
  VAR = A,11-17, B,20-26, MODE,45-46, DIST,54,1, SPEED,63-64

  FILEO PRINTO[1] = "Met_Bus.nt"
  FILEO PRINTO[2] = "Com_Bus.nt"
  FILEO PRINTO[3] = "LRT_Bus.nt"
  FILEO PRINTO[4] = "New_Bus.nt"

  PHASE = INPUT FILEI = LI.1
    IF (A > 0)
      DIST = DIST / 100.0
      COST = DIST * 60.0 / SPEED
      PRINT LIST="NT LEG=",A(5.0),"-",B(5.0)," MODE=",MODE(2.0),
        " SPEED=",SPEED(4.1)," ONEWAY=F DIST=",DIST(6.2),
        " COST=",COST(7.2) PRINTO=1
    ENDIF
  ENDPHASE

  PHASE = INPUT FILEI = LI.2
    IF (A > 0)
      DIST = DIST / 100.0
      COST = DIST * 60.0 / SPEED
      PRINT LIST="NT LEG=",A(5.0),"-",B(5.0)," MODE=",MODE(2.0),
        " SPEED=",SPEED(4.1)," ONEWAY=F DIST=",DIST(6.2),
        " COST=",COST(7.2) PRINTO=2
    ENDIF
  ENDPHASE

  PHASE = INPUT FILEI = LI.3
    IF (A > 0)
      DIST = DIST / 100.0
      COST = DIST * 60.0 / SPEED
      PRINT LIST="NT LEG=",A(5.0),"-",B(5.0)," MODE=",MODE(2.0),
        " SPEED=",SPEED(4.1)," ONEWAY=F DIST=",DIST(6.2),

```

```

                                " COST=", COST(7.2) PRINTO=3
        ENDIF
    ENDPHASE

    PHASE = INPUT   FILEI = LI.4
        IF (A > 0)
            DIST = DIST / 100.0
            COST = DIST * 60.0 / SPEED
            PRINT LIST="NT LEG=", A(5.0), "-", B(5.0), " MODE=", MODE(2.0),
                " SPEED=", SPEED(4.1), " ONEWAY=F DIST=", DIST(6.2),
                " COST=", COST(7.2) PRINTO=4
        ENDIF
    ENDPHASE
ENDRUN

;---- AM PNR access links ----
RUN PGM = NETWORK
    LINKI[1] = "mrpram.asc",
        VAR = A, 12-16, B, 18-23, DIST, 29-34, MODE, 50-51, SPEED, 59-62
    LINKI[2] = "cram.asc",
        VAR = A, 12-16, B, 18-23, DIST, 29-34, MODE, 50-51, SPEED, 59-62
    LINKI[3] = "lrtam.asc",
        VAR = A, 12-16, B, 18-23, DIST, 29-34, MODE, 50-51, SPEED, 59-62
    LINKI[4] = "newam.asc",
        VAR = A, 12-16, B, 18-23, DIST, 29-34, MODE, 50-51, SPEED, 59-62
    LINKI[5] = "buspram.asc",
        VAR = A, 12-16, B, 18-23, DIST, 29-34, MODE, 50-51, SPEED, 59-62

    FILEO PRINTO[1] = "mrpram.nt"
    FILEO PRINTO[2] = "cram.nt"
    FILEO PRINTO[3] = "lrtam.nt"
    FILEO PRINTO[4] = "newam.nt"
    FILEO PRINTO[5] = "buspram.nt"

    PHASE = INPUT   FILEI = LI.1
        IF (A > 0)
            DIST = DIST / 100.0
            COST = DIST * 60.0 / SPEED
            PRINT LIST="NT LEG=", A(5.0), "-", B(5.0), " MODE=", MODE(2.0),
                " SPEED=", SPEED(4.1), " ONEWAY=T DIST=", DIST(6.2),
                " COST=", COST(7.2) PRINTO=1
        ENDIF
    ENDPHASE

    PHASE = INPUT   FILEI = LI.2
        IF (A > 0)
            DIST = DIST / 100.0
            COST = DIST * 60.0 / SPEED
            PRINT LIST="NT LEG=", A(5.0), "-", B(5.0), " MODE=", MODE(2.0),
                " SPEED=", SPEED(4.1), " ONEWAY=T DIST=", DIST(6.2),
                " COST=", COST(7.2) PRINTO=2
        ENDIF
    ENDPHASE

    PHASE = INPUT   FILEI = LI.3
        IF (A > 0)
            DIST = DIST / 100.0
            COST = DIST * 60.0 / SPEED
            PRINT LIST="NT LEG=", A(5.0), "-", B(5.0), " MODE=", MODE(2.0),
                " SPEED=", SPEED(4.1), " ONEWAY=T DIST=", DIST(6.2),
                " COST=", COST(7.2) PRINTO=3
        ENDIF
    ENDPHASE

    PHASE = INPUT   FILEI = LI.4
        IF (A > 0)
            DIST = DIST / 100.0
            COST = DIST * 60.0 / SPEED
            PRINT LIST="NT LEG=", A(5.0), "-", B(5.0), " MODE=", MODE(2.0),
                " SPEED=", SPEED(4.1), " ONEWAY=T DIST=", DIST(6.2),
                " COST=", COST(7.2) PRINTO=4

```

```

        ENDIF
    ENDPHASE

    PHASE = INPUT   FILEI = LI.5
        IF (A > 0)
            DIST = DIST / 100.0
            COST = DIST * 60.0 / SPEED
            PRINT LIST="NT LEG=",A(5.0), "-",B(5.0), " MODE=",MODE(2.0),
                " SPEED=",SPEED(4.1), " ONEWAY=T DIST=",DIST(6.2),
                " COST=",COST(7.2) PRINTO=5
        ENDIF
    ENDPHASE
ENDRUN

;---- OP PNR access links ----
RUN PGM = NETWORK
LINKI[1] = "mrprop.asc",
    VAR = A,12-16, B,18-23, DIST,29-34, MODE,50-51, SPEED,59-62
LINKI[2] = "crop.asc",
    VAR = A,12-16, B,18-23, DIST,29-34, MODE,50-51, SPEED,59-62
LINKI[3] = "lrtop.asc",
    VAR = A,12-16, B,18-23, DIST,29-34, MODE,50-51, SPEED,59-62
LINKI[4] = "newop.asc",
    VAR = A,12-16, B,18-23, DIST,29-34, MODE,50-51, SPEED,59-62
LINKI[5] = "busprop.asc",
    VAR = A,12-16, B,18-23, DIST,29-34, MODE,50-51, SPEED,59-62

FILEO PRINTO[1] = "mrprop.nt"
FILEO PRINTO[2] = "crop.nt"
FILEO PRINTO[3] = "lrtop.nt"
FILEO PRINTO[4] = "newop.nt"
FILEO PRINTO[5] = "busprop.nt"

    PHASE = INPUT   FILEI = LI.1
        IF (A > 0)
            DIST = DIST / 100.0
            COST = DIST * 60.0 / SPEED
            PRINT LIST="NT LEG=",A(5.0), "-",B(5.0), " MODE=",MODE(2.0),
                " SPEED=",SPEED(4.1), " ONEWAY=T DIST=",DIST(6.2),
                " COST=",COST(7.2) PRINTO=1
        ENDIF
    ENDPHASE

    PHASE = INPUT   FILEI = LI.2
        IF (A > 0)
            DIST = DIST / 100.0
            COST = DIST * 60.0 / SPEED
            PRINT LIST="NT LEG=",A(5.0), "-",B(5.0), " MODE=",MODE(2.0),
                " SPEED=",SPEED(4.1), " ONEWAY=T DIST=",DIST(6.2),
                " COST=",COST(7.2) PRINTO=2
        ENDIF
    ENDPHASE

    PHASE = INPUT   FILEI = LI.3
        IF (A > 0)
            DIST = DIST / 100.0
            COST = DIST * 60.0 / SPEED
            PRINT LIST="NT LEG=",A(5.0), "-",B(5.0), " MODE=",MODE(2.0),
                " SPEED=",SPEED(4.1), " ONEWAY=T DIST=",DIST(6.2),
                " COST=",COST(7.2) PRINTO=3
        ENDIF
    ENDPHASE

    PHASE = INPUT   FILEI = LI.4
        IF (A > 0)
            DIST = DIST / 100.0
            COST = DIST * 60.0 / SPEED
            PRINT LIST="NT LEG=",A(5.0), "-",B(5.0), " MODE=",MODE(2.0),
                " SPEED=",SPEED(4.1), " ONEWAY=T DIST=",DIST(6.2),
                " COST=",COST(7.2) PRINTO=4
        ENDIF

```

```

ENDPHASE

PHASE = INPUT  FILEI = LI.5
  IF (A > 0)
    DIST = DIST / 100.0
    COST = DIST * 60.0 / SPEED
    PRINT LIST="NT LEG=",A(5.0),"-",B(5.0)," MODE=",MODE(2.0),
      " SPEED=",SPEED(4.1)," ONEWAY=T DIST=",DIST(6.2),
      " COST=",COST(7.2) PRINTO=5
  ENDIF
ENDPHASE
ENDRUN

;---- AM KNR access links ----
RUN PGM = NETWORK
LINKI[1] = "mrkram.asc",
  VAR = A,12-16, B,18-23, DIST,29-34, MODE,50-51, SPEED,59-62
LINKI[2] = "lrtkram.asc",
  VAR = A,12-16, B,18-23, DIST,29-34, MODE,50-51, SPEED,59-62
LINKI[3] = "newkram.asc",
  VAR = A,12-16, B,18-23, DIST,29-34, MODE,50-51, SPEED,59-62
LINKI[4] = "buskram.asc",
  VAR = A,12-16, B,18-23, DIST,29-34, MODE,50-51, SPEED,59-62

FILEO PRINTO[1] = "mrkram.nt"
FILEO PRINTO[2] = "lrtkram.nt"
FILEO PRINTO[3] = "newkram.nt"
FILEO PRINTO[4] = "buskram.nt"

PHASE = INPUT  FILEI = LI.1
  IF (A > 0)
    DIST = DIST / 100.0
    COST = DIST * 60.0 / SPEED
    PRINT LIST="NT LEG=",A(5.0),"-",B(5.0)," MODE=",MODE(2.0),
      " SPEED=",SPEED(4.1)," ONEWAY=T DIST=",DIST(6.2),
      " COST=",COST(7.2) PRINTO=1
  ENDIF
ENDPHASE

PHASE = INPUT  FILEI = LI.2
  IF (A > 0)
    DIST = DIST / 100.0
    COST = DIST * 60.0 / SPEED
    PRINT LIST="NT LEG=",A(5.0),"-",B(5.0)," MODE=",MODE(2.0),
      " SPEED=",SPEED(4.1)," ONEWAY=T DIST=",DIST(6.2),
      " COST=",COST(7.2) PRINTO=2
  ENDIF
ENDPHASE

PHASE = INPUT  FILEI = LI.3
  IF (A > 0)
    DIST = DIST / 100.0
    COST = DIST * 60.0 / SPEED
    PRINT LIST="NT LEG=",A(5.0),"-",B(5.0)," MODE=",MODE(2.0),
      " SPEED=",SPEED(4.1)," ONEWAY=T DIST=",DIST(6.2),
      " COST=",COST(7.2) PRINTO=3
  ENDIF
ENDPHASE

PHASE = INPUT  FILEI = LI.4
  IF (A > 0)
    DIST = DIST / 100.0
    COST = DIST * 60.0 / SPEED
    PRINT LIST="NT LEG=",A(5.0),"-",B(5.0)," MODE=",MODE(2.0),
      " SPEED=",SPEED(4.1)," ONEWAY=T DIST=",DIST(6.2),
      " COST=",COST(7.2) PRINTO=4
  ENDIF
ENDPHASE
ENDRUN

;---- OP KNR access links ----

```



```

RUN PGM = NETWORK
LINKI[1] = "mrkrop.asc",
  VAR = A,12-16, B,18-23, DIST,29-34, MODE,50-51, SPEED,59-62
LINKI[2] = "lrtkrop.asc",
  VAR = A,12-16, B,18-23, DIST,29-34, MODE,50-51, SPEED,59-62
LINKI[3] = "newkrop.asc",
  VAR = A,12-16, B,18-23, DIST,29-34, MODE,50-51, SPEED,59-62
LINKI[4] = "buskrop.asc",
  VAR = A,12-16, B,18-23, DIST,29-34, MODE,50-51, SPEED,59-62

FILEO PRINTO[1] = "mrkrop.nt"
FILEO PRINTO[2] = "lrtkrop.nt"
FILEO PRINTO[3] = "newkrop.nt"
FILEO PRINTO[4] = "buskrop.nt"

PHASE = INPUT FILEI = LI.1
  IF (A > 0)
    DIST = DIST / 100.0
    COST = DIST * 60.0 / SPEED
    PRINT LIST="NT LEG=",A(5.0),"-",B(5.0)," MODE=",MODE(2.0),
      " SPEED=",SPEED(4.1)," ONEWAY=T DIST=",DIST(6.2),
      " COST=",COST(7.2) PRINTO=1
  ENDIF
ENDPHASE

PHASE = INPUT FILEI = LI.2
  IF (A > 0)
    DIST = DIST / 100.0
    COST = DIST * 60.0 / SPEED
    PRINT LIST="NT LEG=",A(5.0),"-",B(5.0)," MODE=",MODE(2.0),
      " SPEED=",SPEED(4.1)," ONEWAY=T DIST=",DIST(6.2),
      " COST=",COST(7.2) PRINTO=2
  ENDIF
ENDPHASE

PHASE = INPUT FILEI = LI.3
  IF (A > 0)
    DIST = DIST / 100.0
    COST = DIST * 60.0 / SPEED
    PRINT LIST="NT LEG=",A(5.0),"-",B(5.0)," MODE=",MODE(2.0),
      " SPEED=",SPEED(4.1)," ONEWAY=T DIST=",DIST(6.2),
      " COST=",COST(7.2) PRINTO=3
  ENDIF
ENDPHASE

PHASE = INPUT FILEI = LI.4
  IF (A > 0)
    DIST = DIST / 100.0
    COST = DIST * 60.0 / SPEED
    PRINT LIST="NT LEG=",A(5.0),"-",B(5.0)," MODE=",MODE(2.0),
      " SPEED=",SPEED(4.1)," ONEWAY=T DIST=",DIST(6.2),
      " COST=",COST(7.2) PRINTO=4
  ENDIF
ENDPHASE
ENDRUN

```

6.3.4 PT Path Building

Converting a TRNBUILD path building script to Public Transport (PT) is not as straightforward as one might like. The process requires a different set of input files and uses a different set of variable names to select skim attributes. It also will fail when processing blank line or access link files. This means the MWCOG process of generating separate files for LRT and New transit modes will need to change. It will be necessary to combine these files with other mode data in the same path building group. For example, the LRT data can be combined with the Metrorail data and the New mode data can be combined with the Bus data. In this way, analysis years that do not include LRT or New modes will not

cause the path building to fail due to a blank file. Note this was also a problem for the Network program, but the problem could be overcome by processing blank and non-blank file in the same Network application. Unfortunately this trick did not work in the Public Transport program.

The Public Transport program requires several new file types. These include a system file that provides general system-wide data and descriptive text for mode, vehicle, and operator codes. An example of this file is shown in Figure 6-4. It also requires a factor file. This file includes specific attributes that control the path building process. This file includes several global parameters, fare methods, run time factors, and boarding and transfer penalties. An example of this file is shown in Figure 6-5.

Figure 6-4: Sample PT System File

```

;<<PT>><<SYSTEM>>;
; Mode numbers
MODE NUMBER=1 LONGNAME="Local Metrobus" NAME="BUS"
MODE NUMBER=2 LONGNAME="Express Metrobus" NAME="XBUS"
MODE NUMBER=3 LONGNAME="Metrorail" NAME="METRO"
MODE NUMBER=4 LONGNAME="Commuter Rail" NAME="RAIL"
MODE NUMBER=5 LONGNAME="Light Rail Transit" NAME="LRT"
MODE NUMBER=6 LONGNAME="Other Local Bus in WMATA Service Area" NAME="BUS1"
MODE NUMBER=7 LONGNAME="Other Express Bus in WMATA Service Area" NAME="XBUS1"
MODE NUMBER=8 LONGNAME="Other Local Bus beyond WMATA Service Area" NAME="BUS2"
MODE NUMBER=9 LONGNAME="Other Express Bus beyond WMATA Service Area" NAME="XBUS2"
MODE NUMBER=10 LONGNAME="Bus Rapid Transit and Streetcar" NAME="BRT"

MODE NUMBER=11 LONGNAME="Drive Access Connectors" NAME="DRCON"
MODE NUMBER=12 LONGNAME="Walk Transfer Links" NAME="XFERCON"
MODE NUMBER=13 LONGNAME="Sidewalk Links" NAME="SIDEWALK"
MODE NUMBER=14 LONGNAME="Unused" NAME="UNUSED"
MODE NUMBER=15 LONGNAME="Walk Transfer Link from PNR Lot to Transit Station" NAME="PNRWALK"
MODE NUMBER=16 LONGNAME="Walk Access Connectors" NAME="WKCON"

; Operator numbers
OPERATOR NUMBER=1 LONGNAME="WMATA" NAME="WMATA"

; Vehicle type (not used in the model)
VEHICLETYPE NUMBER=1 LONGNAME="BUS" NAME="BUS"
VEHICLETYPE NUMBER=2 LONGNAME="METRORAIL" NAME="METRO"
VEHICLETYPE NUMBER=3 LONGNAME="LIGHT RAIL" NAME="LRT"
VEHICLETYPE NUMBER=4 LONGNAME="COMMUTER RAIL" NAME="COMRAIL"
VEHICLETYPE NUMBER=5 LONGNAME="COMMUTER BUS" NAME="COMBUS"
VEHICLETYPE NUMBER=6 LONGNAME="PREMIUM BUS" NAME="BRT"
VEHICLETYPE NUMBER=7 LONGNAME="STREETCAR" NAME="STCAR"

; Wait curve definitions
WAITCRVDEF NUMBER=1 LONGNAME="Initial and Transfer Wait" NAME="InitXferWait" ,
    CURVE=0-0,1-0.5,4-2,60-30,120-60

```

Figure 6-5: Sample PT Factor File

```

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

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

;Global Settings

```

```

; DELMODE =1,2,9,10
; DELACCESSMODE = 14,16
; MUSTUSEMODE=1
; DELEGRESSMODE = 2

;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

;Runfactors
RUNFACTOR[1] = 1.00 ;--- in-vehicle time ----
RUNFACTOR[2] = 1.00
RUNFACTOR[3] = 1.00
RUNFACTOR[4] = 1.00
RUNFACTOR[5] = 1.00
RUNFACTOR[6] = 1.00
RUNFACTOR[7] = 1.00
RUNFACTOR[8] = 1.00
RUNFACTOR[9] = 1.00
RUNFACTOR[10] = 1.00
RUNFACTOR[11] = 1.50 ;--- drive access time ----
RUNFACTOR[12] = 2.00 ;--- transit transfer time ----
RUNFACTOR[13] = 1.00 ;--- walk network time ----
RUNFACTOR[14] = 2.00 ;--- dummy links to stations ----
RUNFACTOR[15] = 2.50 ;--- park-&-ride transfer time ----
RUNFACTOR[16] = 1.00 ;--- walk access time ----

;Boarding and Transfer Penalties
; IWAITCURVE=1, NODES=3012-99999
; XWAITCURVE=1, NODES=3012-99999
; WAITFACTOR=2.0, n=20000-20599,20998-20999
; WAITFACTOR=1.5, n=1-19999,20600-20997,21000-99999

BRDPEN=10*5.3
XFERPEN=1.6, from=1-10, to=1-10,
XFERPEN=9999, from=1-7,10, to=8,9,11,15
XFERPEN=9999, from=8-9,13-14, to=11,15
XFERPEN=9999, from=11, to=11,12,14
XFERPEN=9999, from=12, to=11,12,15
XFERPEN=9999, from=15, to=11-16
XFERPEN=9999, from=16, to=11,15,16

```

PT also makes a distinction between actual and perceived times in the skim variables generated for a path. The actual time on a given mode is available using the TIMEA variable. The perceived time is available using the TIMEP variable. Each of these variables is defined for a route-set, but the software currently doesn't support multiple route-sets so this value is always zero. The actual Metrorail (mode 3) travel time is therefore retrieved using TIMEA(0,3). Boardings, boarding penalties, distance, fares, and transfer penalties can be retrieved by mode, but transfer times and initial wait times are not available by mode. TRNBUILD gave the user access to transfer and initial wait times by mode. Fortunately, the current MWCOG process does not summarize waiting time for specific modes. Unfortunately, PT does not include the NODE0 and 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. PT does provide a stop-to-stop option in the transit loading function that might be configured to generate similar information. Alternatively, it might be possible to configure the PT fare logic to generate the Metrorail fare directly. In either case, the MWCOG Metrorail fare calculation procedure will need to be re-worked.

A sample script for a Public Transport application to build walk-access Metrorail skims similar to the current MWCOG process is shown in Figure 6-6. The script reads the transit network files created using the script in Figure 6-1.

. It also uses the converted transit line files and access links, the additional files, and the new variable names. This script runs and generates the output files. Unfortunately it does not build any paths. Unlike TRNBUILD, Public Transport cannot build paths that require more than one non-transit link between zones and transit stops or between two transit stops. Since the MWCOG access network is designed to have zones connected to nearby nodes or park-&-ride lots, sidewalk links between nodes, and walk access links between nodes or park-&-ride lots and stations, at least two non-transit links are required for all Metrorail or commuter rail paths built using TRNBUILD. In other words, the current access data generated by the MWCOG process cannot be used as input to Public Transport. The process must be redesigned or the PT Generate concepts must be used.

Figure 6-6: 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=1-20,
  NAME = IVLB, IVXB, IVMT, IVCR, IVNM, INIT, XFERT, WACCT, WLKT, BRDS,
  DACCT, DACCD, PRKT, PRKC, ISTOS, JSTOS, DMR, DLRT

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

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 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

ENDPROCESS
ENDRUN

```

6.4 Generating PT Network

The second phase of the analysis examined the tools available within PT for generating transit networks. As described above, all links and nodes required by transit routes need to be included in a network file. This can be done using the process described in section 5.3.1, but it is more likely that the MWCOG process that generates these data will be modified to create a link and node file in dBase format to add the transit-only data to the roadway network.

Generating walk access links between zones and transit stops is the next step. The example shown in Figure 6-7 uses the Generate command to build walk paths between each zone and the stops included in two line files (modes 2 and 3). In this case the maximum path length is set to two miles and the maximum number of access points from any zone is limited to five. The path cannot use links with facility types 1 or 2 and will be designated as mode 16¹⁵. The second Generate command is similar, but

¹⁵ It will be desirable to define a new set of access mode numbers for the PT network. At a minimum a new mode code will be required to distinguish park-n-ride access links from kiss-n-ride access links. It will also be helpful to add several more non-transit modes to capture mode-to-mode transfers more accurately (e.g., bus to Metrorail).

creates transfer links between stop nodes (10001-60000) and Metrorail station nodes (8000-8200). These links are limited to 10 minutes or a one-half mile walk and are designated as mode 12.

Figure 6-7: Script for Generating PT Walk Access Links

```

; The link and node files contain both highway and transit links/nodes

RUN PGM=NETWORK
  NETI = "ZONEHWY.NET"
  FILEI LINKI[2]="LINKS.DBF"
  FILEI NODEI[2]="NODES.DBF"
  FILEO NETO="PT.NET"
  MERGE RECORD=TRUE
ENDRUN

; WALK LINKS - 2 miles
RUN PGM=PUBLIC TRANSPORT

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

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

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

  PROCESS PHASE = DATAPREP
    GENERATE,
      COST=li.DISTANCE*20,
      MAXCOST=10*30.0,
      MAXNTLEGS = 5,
      NTLEGMODE = 16,
      INCLUDELINK = (li.FTYPE <> 1-2)
  ENDPROCESS
ENDRUN

;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_WK.FAC"

  FILEO NTLEGO = "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=10001-60000,
      TONODE=8001-8200,
      INCLUDELINK=(li.FTYPE <> 1-2),
      MAXNTLEGS=10
  ENDPROCESS
ENDRUN

```

Starting with Cube 6.0, the network file can also include station data. These data include the station name, number, zone, node, type, and attributes for park-n-ride and kiss-n-ride access (e.g., maximum distance, parking spaces, AM/MD parking costs, and terminal times). When combined with the new AUTOCON process and the PNR keyword, drive access generation can consider most of the attributes currently included in the MWCOCG station database.

If the process must be compatible with Cube 5.x, the park-n-ride nodes must be coded into generate statements. In the example shown in Figure 6-8 the kiss-n-ride links to Metrorail stations are generated in a straightforward way. The AM peak highway travel times are used to build paths between each zone and all Metrorail stations. The path must be less than 15 minutes long and is designated as mode 11. The fact that kiss-n-ride links are built to all stations is not consistent with the current MWCOCG process, but this could be corrected using logic similar to the park-n-ride links.

Since park-n-ride access is limited to specified stations, the PT Generate statement needs to be limited as well. In Cube 6.0 this is done using the station fields in the input network. In Cube 5.x a similar process can be implemented by adding a PNR flag to the node database. The script in Figure 6-8 detects the presents of this flag and if the value is 1, writes out a new set of Generate statements that includes a “ToNode” list for the Metrorail stations that permit parking. This generates a new script file similar to Figure 6-9. This script file can be executed to create the mode 11 park-&-ride access links from zones within 15 minutes of each station.

Figure 6-8: Script for Generating PT Drive Access Links

```
;KNR LINKS - ~5 miles = 15 minutes
RUN PGM=PUBLIC TRANSPORT

  FILEI LINEI[1] = "Mode3AM.lin"
  FILEI NETI = "PT.NET"
  FILEI FACTORI[1] = "MR_AM_DR.FAC"

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

  PROCESS PHASE = LINKREAD
    GENTRSTIME = li.AMHTIME
  ENDPROCESS

  PROCESS PHASE = DATAPREP
    GENERATE,
    COST=li.AMHTIME,
    MAXCOST=10*15.0,
    MAXNTLEGS = 10,
    NTLEGMODE = 11
  ENDPROCESS

ENDRUN

;PNR LINKS - ~5 miles = 15 minutes
RUN PGM=PUBLIC TRANSPORT

  FILEI NETI = PT.NET
  FILEI LINEI[1] = Mode3AM.lin
  FILEI FACTORI[1] = "MR AM DR.FAC"

  FILEO PRINTO[1] = "PNR.s"

  PHASE = LINKREAD
```

```

numPNR = 0
ENDPHASE

PHASE=NODEREAD
IF (NI.PNR == 1)
  IF (numPNR == 0)
    PRINT LIST = "RUN PGM=PUBLIC TRANSPORT", printo=1
    PRINT LIST = "  FILEI NETI = PTIN.NET", printo=1
    PRINT LIST = "  FILEI LINEI[1] = Mode3AM.lin", printo=1
    PRINT LIST = "  FILEI FACTORI[1] = MR AM DR.FAC", printo=1

    PRINT LIST = "  FILEO NTLEGO = METRO_PNR.LEG, XN=N", printo=1

    PRINT LIST = "  PARAMETERS TRANTIME = GENTRSTIME", printo=1

    PRINT LIST = "", printo=1
    PRINT LIST = "  PROCESS PHASE = LINKREAD", printo=1
    PRINT LIST = "    GENTRSTIME = li.AMHTIME", printo=1
    PRINT LIST = "  ENDPROCESS", printo=1
    PRINT LIST = "", printo=1

    PRINT LIST = "  PROCESS PHASE = DATAPREP", printo=1
    PRINT LIST = "    GENERATE,", printo=1
    PRINT LIST = "    TONODE = ", N(6), ",", printo=1
  ELSE
    PRINT LIST = "    ", N(6), ",", printo=1
  ENDIF
  numPNR = numPNR+1
ENDIF
ENDPHASE

PROCESS PHASE = DATAPREP
PARAMETERS TRANTIME = GENTRSTIME
GENTRSTIME = li.AMHTIME

PRINT LIST = "  COST=li.AMHTIME,", printo=1
PRINT LIST = "  MAXCOST=10*15.0,", printo=1
PRINT LIST = "  MAXNTLEGS = 10,", printo=1
PRINT LIST = "  NTLEGMODE = 11", printo=1
PRINT LIST = "  ENDPROCESS", printo=1
PRINT LIST = "", printo=1
PRINT LIST = "ENDRUN", printo=1
ENDPROCESS

ENDRUN

```

Figure 6-9: Script for Manually Selecting Access Points

```

RUN PGM=PUBLIC TRANSPORT
FILEI NETI = PTIN.NET
FILEI LINEI[1] = Mode3AM.lin
FILEI FACTORI[1] = MR AM DR.FAC
FILEO NTLEGO = METRO_PNR.LEG, XN=N
PARAMETERS TRANTIME = GENTRSTIME

PROCESS PHASE = LINKREAD
  GENTRSTIME = li.AMHTIME
ENDPROCESS

PROCESS PHASE = DATAPREP
  GENERATE,
  TONODE = 5001,
    5011,
    5021,
    5031,
    5051,
    5061,
    5071,

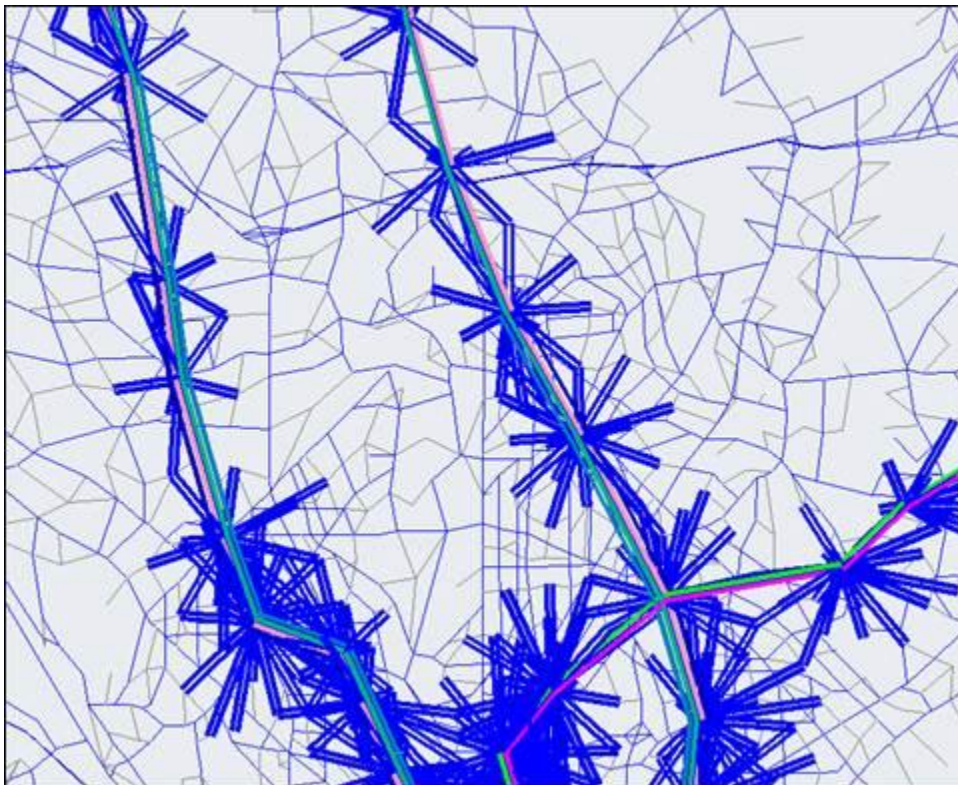
```



```
5081,  
5091,  
5101,  
8001,  
8011,  
8021,  
8024,  
8031,  
8041,  
8051,  
8061,  
8071,  
8081,  
COST=1i.AMHTIME,  
MAXCOST=10*15.0,  
MAXNTLEGS = 10,  
NTLEGMODE = 11  
ENDPROCESS  
ENDRUN
```

The PT Generate examples presented above are relatively easy to understand and run very quickly. The output network shown in Figure 6-10 is reasonable, but considerably different from the network generated by the current MWCOC process. Using PT to implement a process that more closely matches the current logic will involve significant creativity. The Cube 6.0 extensions do offer some additional functionality that should simplify this process.

Figure 6-10: Transit Links Generated by PT



6.5 Estimated Conversion Effort

As expected, converting TRNBUILD routes to PT line files is straightforward. Adding the transit-only links to the highway network is also not difficult. We would, however, recommend modifying the GIS process that generates the transit-only links to write the data in dBase format with the data fields expected by the Network program. Alternatively, the GIS process could create a composite highway and transit network directly. This would require link and node flags to clearly distinguish highway facilities from transit-only facilities.

Converting the TRNBUILD access links to PT non-transit links is significantly more complex. It is clear from the analysis above that the transit access logic needs to be totally reworked to be useable in PT. PT requires a single non-transit link record between zones and stops and between two stops. These records can include a list of nodes that comprise the path between the two points, but the PT path builder cannot build this path from a collection of non-transit leg types.

If MWCOCG wishes to modify their current procedures to work with PT, the walk access logic would need to abandon the concept of a sidewalk network and create walk access links between each zone and the bus stops and stations within a specified search radius. This radius could be defined based on the size of the zone and weighted by the PEF value if so desired. These links would not be exactly the same as the TRNBUILD process because the paths wouldn't be restricted by the sidewalk network. If using a sidewalk layer is desirable, the PT Generate statement can be used to build paths from the zones to the stops and stations. If this is done, logic would need to be developed to control the path length or weight the path cost by the zone size and PEF values.

Upgrading the drive access routines to support PT may not be especially difficult. The current logic uses the station database to identify the station numbers, coordinates, and access restrictions. The kiss-n-ride logic uses barrier information, station flags, and highway travel times to construct zone connectors to the station nodes. The park-n-ride logic inserts the parking lot and creates a walk access link between the parking lot and the station that includes parking cost, parking access time, and shadow prices. This link could be made consistent with PT by combining both the drive leg and the parking lot leg into a single PT non-transit leg record.

A Cube 6.0 implementation is likely to make the PT AUTOCON logic for generating the drive access links more consistent with the current MWCOCG process. If the logic needs to be compatible with all Cube versions, the process will be more cumbersome. The advantage of using the PT Generate statement is that a separate park-n-ride zone would not need to be assigned to each station to skim the drive access time to the station. This saves processing time and reduces the potential for coding errors or processing mistakes.

As mentioned earlier, the PT skim logic does not enable the user to save the first and last station used on the Metrorail path. The current process subtracts 8000 from the station number to identify the station-to-station "zone codes" used by the special path building routine that calculates Metrorail trip length. This length is needed for calculating the Metrorail fare. It is unclear at this point what would be required to replicate this process within PT. It is likely, however, that the PT fare processing logic could

be configured to generate a reasonable estimate of the WMATA fare policy. If this proves not to be the case, the PT loading tools do appear to have a station-to-station analysis option that might be able to generate the distance information needed by the existing fare calculation procedure.

Beyond the basic skim generation and path loading process, PT includes a number of features and capabilities that may ultimately prove useful. Multi-path, multiple user classes, select link, crowding, wait time curves, fare-based paths and other path building controls might be used to improve the overall transit modeling process¹⁶. Even without these capabilities there is likely to be some impact on the mode choice calibration and transit summary routines. For example, the mode of access distribution at rail stations or bus stops could be more easily summarized by properly designing the mode numbers assigned to PT non-transit links. Access to Metrorail could be subdivided into walk, park-n-ride, kiss-n-ride, local bus, express bus, commuter rail, commuter bus, LRT, BRT, streetcar and any other combination that might be of interest.

6.5.1 Proposed Work Plan

Moving from TRNBUILD to PT offers a number of advantages and challenges. It will, however, take time and require a considerable amount of testing and validation work. AECOM proposes to proceed with the conversion in several phases to keep the tasks manageable and demonstrate incremental progress. The effort should be shared between MWCOG staff and the consultant team to maximize understanding, proceed at a reasonable pace, and keep the costs down. AECOM would also recommend coordinating these efforts with WMATA to design a process that supports a full range of analysis needs.

The proposed work plan is divided into five primary phases. A brief description and estimated level of effort is provided for each phase.

Phase 1: Network Preparation

This phase would focus on developing the PT network, routes, and non-transit legs. It would decide how the transit-only links and nodes are integrated into the roadway network and consolidate the transit routes into logical path building groups that avoid the possibility of a blank line file. The bulk of the work, however, involves generating the non-transit legs. Though upgrading the existing procedures is a possibility, the general recommendation at this point is to utilize the PT Generate statements to the fullest extent possible – including the new Cube 6.0 station data. 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, should be designed to summarize all mode-of-access combinations at Metrorail stations.

¹⁶ FTA does not accept multi-path and other advanced path building techniques for New Starts analysis. FTA requires a single transit path for each mode defined in the mode choice model for User Benefit calculations. MWCOG or WMATA could still use these techniques as part of their modeling process and disable these features for FTA New Starts applications.

Phase 2: Path Building and Loading

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 phase is not so much to reproduce the paths generated by TRNBUILD, but to generate a set of reasonable paths under a variety of conditions. This would include an assessment of the pros and cons of the multi-path capabilities as well as the multiple user classes available within PT. Crowding and fare-based path building are also of potential interest.¹⁷ This effort will also assess the processing methods used to generate the 22 path combinations included in the current MWCOG process. It will consolidate path skimming and loading scripts to minimize code duplication and maximize the flexibility with regard to the number of CPU's available for parallel processing. It will also refine the batch processing scripts to avoid conflicts caused by running multiple alternatives simultaneously.

Phase 3: Transit Fares

The MWCOG process calculates fares after the 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 phase will analyze 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. If the PT fare tools cannot sufficiently replicate the Metrorail fare, methods of implementing the station-to-station fare calculation using a simpler and more intuitive procedure will be investigated.

Phase 4: Mode Choice Calibration

At a minimum the mode choice constants will need to be adjusted to account for the changes in path options and attributes generated by the PT conversion. This phase, however, is intended to reexamine the overall mode choice structure and application software to consider a reduction in the number of geographic market segments included in the model, constraints on the constants applied to specific modes within a given market segment, and integration of a pedestrian environment factor concept into the mode choice models. If the path building and fare calculation procedures find multi-class methods

¹⁷ The TPB travel model currently uses single, best-path routing for transit paths generated by TRNBUILD. PT offers both single, best-path routing and multirouting. According to the reference guide for Cube Voyager 5.1.3, "The best-path-only method cannot be used with either the crowding model or the service frequency and cost model" (p. 673). So, in order to use PT's crowding model, TPB staff may need to switch the travel model from single-path to multirouting.

to be effective, the mode choice models could be segregated into traveler-based market segments rather than or in addition to geographic market segments. The effort will also debate the pros and cons of disaggregating trip purposes by time period.

Phase 5: Advanced Applications

The Public Transport and related Cube software offer a number of opportunities to perform sophisticated analysis of the transit system. Select link and crowding analysis along with the ability to build and trace paths interactively on a network map are among the more attractive capabilities offered by the PT approach, although it may require switching to multi-routing in transit path building. It may also be the case that MWCOG chooses to postpone some of the suggestions proposed for earlier phases until after a more traditional model is implemented. This phase would investigate some of these more advanced applications and develop a set of tools for generating performance measures and analysis reports and graphics from the transit assignment results.

Table 6-1 summarizes the overall effort and work schedule envisioned for the five proposed tasks.

Table 6-1: Level of Effort Estimate

Phase	Consultant Hours	MWCOG Hours	Schedule Weeks
Network Preparation	100 to 130	100 to 140	8 to 10
Path Building and Loading	120 to 180	100 to 160	8 to 10
Transit Fares	120 to 200	120 to 140	10 to 12
Mode Choice Calibration	160 to 240	160 to 200	12 to 16
Advanced Applications	100 to 200	80 to 100	8 to 12

7 Non-Resident Metrorail Trips (Task Order 6)

The current MWCOG mode choice model includes only internal-to-internal trips and is composed of mainly resident travel. However, transit passenger surveys in the Washington region have routinely identified that a non-trivial number of non-residents use transit, especially the Metrorail system. This includes visitors, external commuters¹⁸ who use Metrorail as part of their trip to/from work, and non-residents in the area on business, who use Metrorail for intra-area travel during the day. It also includes non-residents using Metrorail to travel to/from Reagan National Airport, which is significant enough to warrant its own analysis (see Chapter 8).

Forecasting non-resident travel is difficult for several reasons, including: 1) the lack of good data describing their internal travel, 2) the wide variety of trip types, and 3) insufficient data on their travel sensitivities and relationships. It would probably be unwise to assume that non-residents use transit in exactly the same way (and for the same reasons) as do residents. On the other hand, the number of non-resident transit trips is not so great as to justify a very detailed forecasting procedure that would add much to the complexity and running time of the current model.

Thus, a relatively simple technique has been developed that makes the best use of already-existing data files and one particularly useful source of information on non-resident transit travel: the 2007 Metrorail on-board passenger survey. (A similar 2008 survey was also consulted, but it did not include information about external travel.) This analysis focuses on Metrorail travel, as it is assumed that the non-resident share of Metrobus (and other suburban bus) travel is extremely low.

So this analysis begins with a description of the base 2007 trip tables and the travel markets that comprise non-resident Metrorail travel.

7.1 Base Tables

As noted above, non-resident transit travel includes several different types of trip and traveler:

- External/Internal commuters (E/I Home-Based Work)
 - o auto to Metrorail (driver and passenger, park-and-ride [PnR] and kiss-and-ride [KnR])
 - o walk to or transfer from commuter rail or commuter bus
- External/Internal Non-Work
 - o auto to Metrorail (driver and passenger, PnR and KnR)
 - o walk to or transfer from commuter rail or commuter bus
- Non-Home-Based (NHB) trips
 - o tourists, visitors
 - o business-related (including students)
- Air passenger
 - o non-residents

¹⁸ The terms “internal” and “external” are in reference to the modeled area of the TPB Version 2.3 Travel Model, as shown on page 1-6 of “Calibration Report for the TPB Travel Forecasting Model, Version 2.3, on the 3,722-Zone Area System” (Final Report). National Capital Region Transportation Planning Board, January 20, 2012.

As noted above, air passenger Metrorail users (both resident and non-resident) are covered in much greater detail in a separate model, described in Chapter 7. Non-resident airport trips are included in Chapter 6 only as a “placeholder”.

Building trip tables from the survey required some basic pre-processing, including the following:

- Survey records that showed the boarding and alighting station being the same were removed.
- Records were converted to production/attraction (P/A) format based on whether the trip originated at home or a non-home location.
- Air passengers are those who boarded or alighted at the National Airport station.
- Records with a missing expansion factor or other important missing data (e.g., station number) were excluded.

The resulting analysis produces the trip totals shown in Table 7-1. These are shown stratified by “purpose” (market) and station jurisdiction in Table 7-2 – Table 7-9. They are shown stratified by market and Metrorail line segment in Table 7-10 – Table 7-17. The trip totals in Table 7-2 – Table 7-9 are slightly different from those in Table 7-1, due to rounding. The line segment definitions are shown in Figure 7-1.

Table 7-1: 2007 Non-Resident Metrorail Trips

Table	Definition	Trips
1	E/I Commuters, auto access	3,092
2	E/I Non-Work, auto access	1,649
3	E/I Commuters, walk access/train transfer	8,980
4	E/I Non-Work, walk access/train transfer	2,940
5	NHB Visitors (walk access)	10,645
6	NHB Business (walk access)	3,980
7	Air Passengers (walk access)	1,678
	Total	32,964

Source: 2007 Metrorail On-Board Survey

These trips are stored as tables 1-7 in the Cube binary matrix file SURV2007.TRP, which has been transmitted separately to TPB staff. This file also has a table 8, resident air passenger trips, totaling 3,207. That table is not used in the non-resident analysis, but was created to assist the subsequent analysis of air passenger travel.

Table 7-2: 2007 Non-Resident Metrorail Trips by Station Jurisdiction (E/I Work, Auto Access)

		E/I Commuters, Auto Access Station-to-Station, P/A Format							
		Destination District							
		1	2	3	4	5	6	7	Total
O	1 DC	458	0	27	88	7	0	0	580
r	2 Montgmry	583	22	0	82	0	0	0	687
i	3 PrinGeor	854	0	0	170	19	32	0	1075
g	4 Arlingtn	99	0	13	107	7	0	0	226
i	5 Alexandr	13	8	0	0	0	0	0	21
n	6 Fairfax	284	20	0	179	10	14	0	507
	7 Loudoun	0	0	0	0	0	0	0	0
Total		2291		40		43		0	3096
			50		626		46		

Table 7-3: 2007 Non-Resident Metrorail Trips by Station Jurisdiction (E/I Non-Work Auto Access)

		E/I Non-Work, Auto Access Station-to-Station, P/A Format							
		Destination District							
		1	2	3	4	5	6	7	Total
O	1 DC	168	13	0	9	11	0	0	201
r	2 Montgmry	287	25	0	9	0	15	0	336
i	3 PrinGeor	345	0	0	24	0	24	0	393
g	4 Arlingtn	128	0	0	0	0	0	0	128
i	5 Alexandr	39	0	0	43	0	0	0	82
n	6 Fairfax	429	0	0	68	11	0	0	508
	7 Loudoun	0	0	0	0	0	0	0	0
Total		1396		0		22		0	1648
			38		153		39		

Table 7-4: 2007 Non-Resident Metrorail Trips by Station Jurisdiction (E/I Work, Walk Access)

		E/I Commuters, Walk Acc/Train Transfer Station-to-Station, P/A Format							
		Destination District							
		1	2	3	4	5	6	7	Total
O	1 DC	5209	588	182	1140	227	67	0	7413
r	2 Montgmry	334	269	42	30	10	0	0	685
i	3 PrinGeor	146	10	33	29	0	0	0	218
g	4 Arlingtn	239	0	0	192	7	21	0	459
i	5 Alexandr	21	0	0	25	0	0	0	46
n	6 Fairfax	105	12	0	31	0	10	0	158
	7 Loudoun	0	0	0	0	0	0	0	0
Total		6054		257		244		0	8979
			879		1447		98		

Table 7-5: 2007 Non-Resident Metrorail Trips by Station Jurisdiction (E/I Non-Work, Walk Access)

		E/I Non-Work, Walk Acc/Train Transfer Station-to-Station, P/A Format							
		Destination District							
		1	2	3	4	5	6	7	Total
Origin Station	1 DC	1241	225	44	224	56	69	0	1859
	2 Montgmrly	247	18	0	0	11	7	0	283
	3 PrinGeor	29	0	0	22	0	0	0	51
	4 Arlingtn	224	0	0	178	0	20	0	422
	5 Alexandr	138	0	0	0	14	15	0	167
	6 Fairfax	129	0	15	14	0	0	0	158
	7 Loudoun	0	0	0	0	0	0	0	0
Total		2008		59		81		0	2940
			243		438		111		

Table 7-6: 2007 Non-Resident Metrorail Trips by Station Jurisdiction (Visitor)

		Visitor, Walk Access Station-to-Station, P/A Format							
		Destination District							
		1	2	3	4	5	6	7	Total
Origin Station	1 DC	4488	415	225	929	295	357	0	6709
	2 Montgmrly	404	18	0	36	9	0	0	467
	3 PrinGeor	412	11	19	43	0	8	0	493
	4 Arlingtn	1197	37	64	560	31	34	0	1923
	5 Alexandr	436	0	16	79	8	9	0	548
	6 Fairfax	411	0	0	75	16	0	0	502
	7 Loudoun	0	0	0	0	0	0	0	0
Total		7348		324		359		0	10642
			481		1722		408		

Table 7-7: 2007 Non-Resident Metrorail Trips by Station Jurisdiction (NHB Business)

		NHB Business-Related, Walk Access Station-to-Station, P/A Format							
		Destination District							
		1	2	3	4	5	6	7	Total
Origin Station	1 DC	1515	285	131	478	44	76	0	2529
	2 Montgmrly	190	48	16	40	0	0	0	294
	3 PrinGeor	87	0	18	10	10	0	0	125
	4 Arlingtn	428	40	30	233	15	37	0	783
	5 Alexandr	51	0	0	47	0	0	0	98
	6 Fairfax	96	0	0	44	15	0	0	155
	7 Loudoun	0	0	0	0	0	0	0	0
Total		2367		195		84		0	3984
			373		852		113		

Table 7-8: 2007 Non-Resident Metrorail Trips by Station Jurisdiction (Air Passenger)

		Air Passenger, Walk Access Station-to-Station, P/A Format							
		Destination District							
		1	2	3	4	5	6	7	Total
Origin Station	1 DC	0	0	0	178	0	0	0	178
	2 Montgmrly	0	0	0	8	0	0	0	8
	3 PrinGeor	0	0	0	9	0	0	0	9
	4 Arlingtn	1062	78	17	161	58	81	0	1457
	5 Alexandr	0	0	0	14	0	0	0	14
	6 Fairfax	0	0	0	12	0	0	0	12
	7 Loudoun	0	0	0	0	0	0	0	0
Total		1062	78	17	382	58	81	0	1678

Table 7-9: 2007 Non-Resident Metrorail Trips by Station Jurisdiction (Total)

		Total Non-Resident Passengers Station-to-Station, P/A Format							
		Destination District							
		1	2	3	4	5	6	7	Total
Origin Station	1 DC	13076	1526	614	3044	641	570	0	19471
	2 Montgmrly	2041	399	59	206	30	22	0	2757
	3 PrinGeor	1871	22	69	309	28	64	0	2363
	4 Arlingtn	3377	155	122	1428	121	196	0	5399
	5 Alexandr	702	8	16	206	20	25	0	977
	6 Fairfax	1454	31	15	420	53	25	0	1998
	7 Loudoun	0	0	0	0	0	0	0	0
Total		22521	2141	895	5613	893	902	0	32965

Table 7-10: 2007 Non-Resident Metrorail Trips by Line Segment (E/I Work, Auto Access)

		E/I Commuters, Auto Access Station-to-Station, P/A Format										
		Destination District										
		1	2	3	4	5	6	7	8	9	10	Total
Origin Station	1 NW Red	157	0	14	25	36	9	33	36	143	0	453
	2 N Red	75	8	0	10	0	24	2	11	118	0	248
	3 NE Green	28	0	0	17	0	31	56	55	366	0	553
	4 DC G/Y/R	0	0	0	0	0	0	0	0	59	0	59
	5 E Yel/Bl	51	0	0	0	29	9	95	31	309	0	524
	6 SE Grn	14	0	0	17	15	36	0	0	25	0	107
	7 S Yel/Bl	29	0	0	0	0	13	192	15	128	0	377
	8 W Orange	43	20	0	0	14	0	18	92	190	0	377
	9 DC Core	51	0	0	0	18	7	56	23	243	0	398
	10 Silver	0	0	0	0	0	0	0	0	0	0	0
Total		448	28	14	69	112	129	452	263	1581	0	3096

Table 7-11: 2007 Non-Resident Metrorail Trips by Line Segment (E/I Non-Work, Auto Access)

		E/I Non-Work, Auto Access Station-to-Station, P/A Format										
		Destination District										
		1	2	3	4	5	6	7	8	9	10	Total
O	1 NW Red	82	0	0	0	14	0	9	0	189	0	294
r	2 N Red	0	0	0	0	0	0	0	15	47	0	62
i	3 NE Green	27	0	0	9	5	10	9	32	68	0	160
g	4 DC G/Y/R	0	0	0	0	0	0	0	0	0	0	0
i	5 E Yel/Bl	25	0	0	0	10	0	0	7	77	0	119
n	6 SE Grn	0	0	0	0	21	0	0	0	93	0	114
	7 S Yel/Bl	12	0	0	0	38	8	21	78	187	0	344
D	8 W Orange	25	0	0	15	32	0	13	10	279	0	374
i	9 DC Core	27	8	0	11	0	0	11	9	115	0	181
s	10 Silver	0	0	0	0	0	0	0	0	0	0	0
Total		198	8	0	35	120	18	63	151	1055	0	1648

Table 7-12: 2007 Non-Resident Metrorail Trips by Line Segment (E/I Work, Walk Access)

		E/I Commuters, Walk Acc/Train Transfer Station-to-Station, P/A Format										
		Destination District										
		1	2	3	4	5	6	7	8	9	10	Total
O	1 NW Red	384	20	0	9	15	20	29	19	151	0	647
r	2 N Red	0	14	42	28	0	0	0	0	51	0	135
i	3 NE Green	0	10	13	0	0	20	0	0	8	0	51
g	4 DC G/Y/R	5	24	0	25	0	13	0	0	23	0	90
i	5 E Yel/Bl	12	8	0	0	65	0	18	19	71	0	193
n	6 SE Grn	0	0	0	0	0	11	0	15	26	0	52
	7 S Yel/Bl	30	0	0	0	13	13	145	88	165	0	454
D	8 W Orange	0	12	0	0	16	10	10	43	118	0	209
i	9 DC Core	1962	299	19	266	114	303	947	456	2782	0	7148
s	10 Silver	0	0	0	0	0	0	0	0	0	0	0
Total		2393	387	74	328	223	390	1149	640	3395	0	8979

Table 7-13: 2007 Non-Resident Metrorail Trips by Line Segment (E/I Non-Work, Walk Access)

		E/I Non-Work, Walk Acc/Train Transfer Station-to-Station, P/A Format										
		Destination District										
		1	2	3	4	5	6	7	8	9	10	Total
O	1 NW Red	199	0	0	24	0	0	29	7	436	0	695
r	2 N Red	24	4	0	10	0	18	11	0	29	0	96
i	3 NE Green	0	0	0	0	0	0	0	8	11	0	19
g	4 DC G/Y/R	39	56	0	0	6	23	16	0	64	0	204
i	5 E Yel/Bl	27	0	0	0	12	0	11	27	38	0	115
n	6 SE Grn	0	0	0	0	13	0	22	0	18	0	53
	7 S Yel/Bl	15	0	0	0	0	11	96	29	171	0	322
D	8 W Orange	43	0	0	0	20	0	45	71	246	0	425
i	9 DC Core	319	43	0	54	61	17	167	91	259	0	1011
s	10 Silver	0	0	0	0	0	0	0	0	0	0	0
Total		666	103	0	88	112	69	397	233	1272	0	2940

Table 7-14: 2007 Non-Resident Metrorail Trips by Line Segment (Visitor)

		Visitor, Walk Access Station-to-Station, P/A Format										
		Destination District										
		1	2	3	4	5	6	7	8	9	10	Total
O r i g i n D i s	1 NW Red	323	25	0	16	93	35	96	82	969	0	1639
	2 N Red	30	8	0	11	0	0	8	8	58	0	123
	3 NE Green	40	11	19	26	8	0	36	7	99	0	246
	4 DC G/Y/R	86	0	0	35	0	9	30	0	92	0	252
	5 E Yel/Bl	152	0	0	0	42	18	77	138	338	0	765
	6 SE Grn	0	0	0	0	5	0	0	16	76	0	97
	7 S Yel/Bl	52	10	45	46	114	49	588	123	1244	0	2271
	8 W Orange	72	4	0	28	117	0	58	43	380	0	702
	9 DC Core	1108	113	126	101	390	84	727	452	1446	0	4547
	10 Silver	0	0	0	0	0	0	0	0	0	0	0
Total		1863		190		769		1620		4702		10642
			171		263		195		869		0	

Table 7-15: 2007 Non-Resident Metrorail Trips by Line Segment (NHB Business)

		NHB Business-Related, Walk Access Station-to-Station, P/A Format										
		Destination District										
		1	2	3	4	5	6	7	8	9	10	Total
O r i g i n D i s	1 NW Red	139	11	18	11	10	23	64	53	299	0	628
	2 N Red	8	23	0	37	19	0	0	8	42	0	137
	3 NE Green	0	0	18	9	0	8	10	0	13	0	58
	4 DC G/Y/R	11	0	8	0	0	7	39	0	50	0	115
	5 E Yel/Bl	32	0	15	14	10	0	11	0	66	0	148
	6 SE Grn	0	0	0	0	0	19	25	7	42	0	93
	7 S Yel/Bl	49	23	0	0	29	49	171	61	227	0	609
	8 W Orange	25	0	0	7	38	12	76	83	186	0	427
	9 DC Core	447	57	23	46	140	18	279	162	597	0	1769
	10 Silver	0	0	0	0	0	0	0	0	0	0	0
Total		711		82		246		675		1522		3984
			114		124		136		374		0	

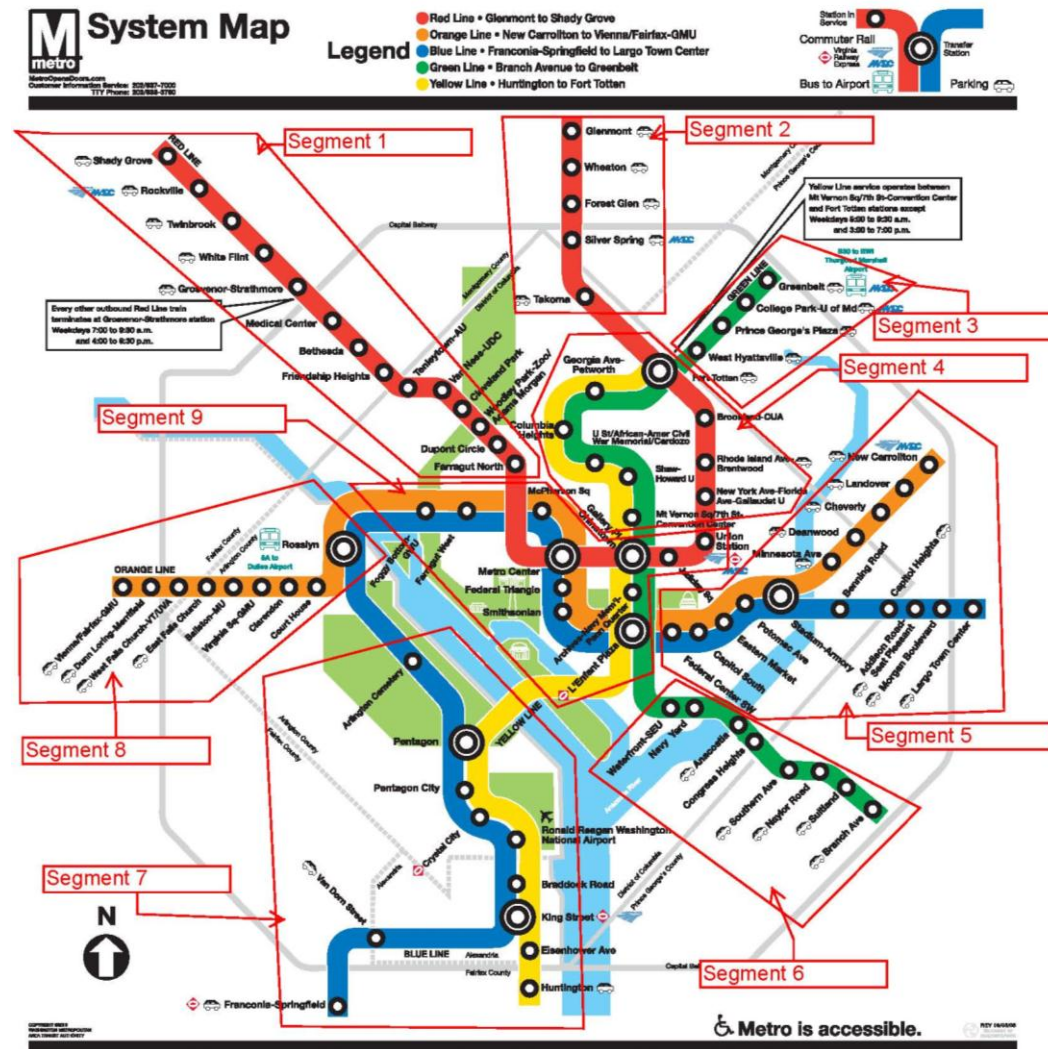
Table 7-16: 2007 Non-Resident Metrorail Trips by Line Segment (Air Passenger)

		Air Passenger, Walk Access Station-to-Station, P/A Format										
		Destination District										
		1	2	3	4	5	6	7	8	9	10	Total
O r i g i n D i s	1 NW Red	0	0	0	0	0	0	12	0	0	0	12
	2 N Red	0	0	0	0	0	0	0	0	0	0	0
	3 NE Green	0	0	0	0	0	0	0	0	0	0	0
	4 DC G/Y/R	0	0	0	0	0	0	24	0	0	0	24
	5 E Yel/Bl	0	0	0	0	0	0	17	0	0	0	17
	6 SE Grn	0	0	0	0	0	0	9	0	0	0	9
	7 S Yel/Bl	318	10	0	60	143	44	139	150	582	0	1446
	8 W Orange	0	0	0	0	0	0	37	0	0	0	37
	9 DC Core	0	0	0	0	0	0	133	0	0	0	133
	10 Silver	0	0	0	0	0	0	0	0	0	0	0
Total		318		0		143		371		582		1678
			10		60		44		150		0	

Table 7-17: 2007 Non-Resident Metrorail Trips by Line Segment (Total)

		Total Non-Resident Passengers Station-to-Station, P/A Format										
		Destination District										
		1	2	3	4	5	6	7	8	9	10	Total
Origins	1 NW Red	1292	54	33	85	166	87	270	198	2183	0	4368
	2 N Red	136	57	43	96	18	40	21	42	346	0	799
	3 NE Green	94	22	50	61	14	68	112	103	562	0	1086
	4 DC G/Y/R	141	83	8	59	6	52	111	0	289	0	749
	5 E Yel/Bl	294	8	15	14	166	28	230	223	902	0	1880
	6 SE Grn	14	0	0	17	54	67	56	38	279	0	525
	7 S Yel/Bl	505	41	45	106	340	185	1347	545	2709	0	5823
	8 W Orange	209	36	0	50	238	22	260	342	1394	0	2551
	9 DC Core	3915	519	169	476	728	428	2317	1193	5439	0	15184
	10 Silver	0	0	0	0	0	0	0	0	0	0	0
Total		6600	820	363	964	1730	977	4724	2684	14103	0	32965

Figure 7-1: Metrorail Line Segments



Note: the future Silver Line is designated as segment 10.

Table 7-18 lists the non-resident sum of boardings and alightings (B+A) by station. Union Station has by far the largest non-resident activity, over 11,000 daily B+A, obviously due to the commuter rail systems. Next is National Airport, with about 5,400, followed by Metro Center and Smithsonian (tourism) with 3,500-4,000 each. Farragut North and Dupont Circle (2,400 each) are major employment sites and L'Enfant Plaza (2,800) is both an employment and tourist destination.

Table 7-18: 2007 Non-Resident Ridership

Station No.	Station	Total B+A	NonRes B+A	NonRes Pct
1	METRO CENTER	61,493	3,962	6.4%
6	FARRAGUT NORTH	59,926	2,452	4.1%
10	DUPONT CIRCLE	46,808	2,458	5.3%
13	WOODLEY PARK ZOO	14,458	1,468	10.2%
14	CLEVELAND PARK	9,856	258	2.6%
15	VAN NESS UDC	14,849	241	1.6%
16	TENLEYTOWN	16,517	530	3.2%
17	FRIENDSHIP HEIGHTS	19,196	486	2.5%
20	BETHESDA	21,858	786	3.6%
21	MEDICAL CENTER	11,217	433	3.9%
22	GROSVENOR	12,586	335	2.7%
23	WHITE FLINT	7,870	283	3.6%
24	TWINBROOK	9,218	297	3.2%
25	ROCKVILLE	9,346	329	3.5%
26	SHADY GROVE	29,567	1,245	4.2%
27	GALLERY PLACE	41,930	1,913	4.6%
31	JUDICIARY SQUARE	21,603	767	3.5%
34	UNION STATION	67,304	11,388	16.9%
37	RHODE ISLAND AVENUE	10,215	129	1.3%
38	BROOKLAND CUA	14,336	398	2.8%
39	FORT TOTTEN	12,548	146	1.2%
40	TAKOMA PARK	13,379	315	2.4%
41	SILVER SPRING	30,700	867	2.8%
44	FOREST GLEN	5,723	131	2.3%
45	WHEATON	10,086	155	1.5%
46	GLENMONT	13,763	387	2.8%
47	MCPHERSON SQUARE	34,997	1,436	4.1%
50	FARRAGUT WEST	47,736	802	1.7%
53	FOGGY BOTTOM GWU	40,754	1,656	4.1%
54	ROSSLYN	32,598	1,659	5.1%
55	ARLINGTON CEMETERY	1,854	1,305	70.4%
56	PENTAGON	36,414	1,209	3.3%

Station No.	Station	Total B+A	NonRes B+A	NonRes Pct
57	PENTAGON CITY	25,667	1,306	5.1%
58	CRYSTAL CITY	27,595	2,000	7.2%
59	NATIONAL AIRPORT	9,019	5,453	60.5%
62	BRADDOCK ROAD	9,412	244	2.6%
63	KING STREET	17,356	1,171	6.7%
66	EISENHOWER AVENUE	3,553	208	5.9%
67	HUNTINGTON	18,267	694	3.8%
70	FEDERAL TRIANGLE	20,781	953	4.6%
71	SMITHSONIAN	21,831	3,598	16.5%
74	L'ENFANT PLAZA	47,121	2,846	6.0%
78	FEDERAL CENTER SW	11,641	722	6.2%
79	CAPITOL SOUTH	15,392	919	6.0%
80	EASTERN MARKET	13,462	472	3.5%
81	POTOMAC AVENUE	6,785	121	1.8%
82	STADIUM ARMORY	5,445	187	3.4%
85	MINNESOTA AVENUE	5,499	0	0.0%
86	DEANWOOD	3,277	77	2.4%
87	CHEVERLY	3,347	72	2.1%
88	LANDOVER	5,818	58	1.0%
89	NEW CARROLLTON	19,668	919	4.7%
90	MT VERNON SQUARE-UDC	5,443	227	4.2%
91	SHAW HOWARD U	6,309	209	3.3%
94	U STREET CARDOZO	10,822	307	2.8%
97	COLUMBIA HEIGHTS	14,724	179	1.2%
98	GEORGIA AVENUE	8,434	140	1.7%
99	WEST HYATTSVILLE	6,412	48	0.8%
100	PRINCE GEORGE'S PLAZA	8,903	94	1.1%
101	COLLEGE PARK - U OF MD	10,186	569	5.6%
102	GREENBELT	14,117	844	6.0%
103	ARCHIVES-NAVY MEMORIAL	16,141	629	3.9%
104	WATERFRONT	6,364	221	3.5%
105	NAVY YARD	7,199	652	9.1%
108	ANACOSTIA	12,155	158	1.3%
111	CONGRESS HEIGHTS	4,081	0	0.0%
112	NAYLOR ROAD	5,621	104	1.8%
113	SUITLAND	13,496	229	1.7%
114	BRANCH AVENUE	14,256	290	2.0%
115	BENNING ROAD	5,485	0	0.0%
116	CAPITOL HEIGHTS	4,451	59	1.3%

Station No.	Station	Total B+A	NonRes B+A	NonRes Pct
117	ADDISON ROAD	8,579	70	0.8%
118	VAN DORN STREET	8,550	495	5.8%
119	FRANCONIA-SPRINGFIELD	19,405	935	4.8%
120	COURT HOUSE	15,191	601	4.0%
121	CLARENDON	8,565	170	2.0%
122	VIRGINIA SQUARE GMU	7,903	231	2.9%
123	BALLSTON	24,454	736	3.0%
124	EAST FALLS CHURCH	8,749	277	3.2%
125	WEST FALLS CHURCH	21,899	646	3.0%
126	DUNN LORING	10,602	459	4.3%
127	VIENNA	26,699	1,005	3.8%
128	SOUTHERN AVENUE	10,182	0	0.0%
129	NEW YORK AVE	5,090	190	3.7%
132	MORGAN BLVD	2,351	0	0.0%
133	LARGO TOWN CENTER	11,051	167	1.5%
Total		1,435,508	73,186	5.1%

Source: 2007 Metrorail On-Board Survey

Note: Station numbers are those used in the 2007 on-board survey.

7.2 Forecasting Theory

As noted above, modeling non-resident travel is complex, mainly because of the lack of observed data describing their trips. Some data is available on external-internal travel (trips into and outside the modeled region by people living outside the region), in terms of traffic counts at the modeled area cordon. But non-residents also make internal-internal trips during the day and there is no household travel survey data on these trips, though they might appear in the on-board Metrorail surveys to some extent. They are only known in terms of the traffic counts on area roadways and passenger counts on the transit systems. In the current model, these trips may be accounted for indirectly by adjustment factors that increase the number of resident non-home-based trips.

Thus, it is not generally feasible to estimate non-resident person trips and let the existing mode choice model split them by travel mode. Fortunately, surveys of Metrorail riders exist that somewhat clearly identify non-resident riders. The term *somewhat* is used because it is unclear how temporary residents of the area are classified (college students, transient military personnel, Federal workers on temporary duty in Washington). Nonetheless, these surveys (particularly the 2007 survey) do provide some useful data on the number and O-D patterns of non-resident Metrorail riders.

After reviewing the data and the existing MWCOG travel model, it is clear that the best approach to forecasting these trips is to create a base year (2007) table as described above and to project future trips by applying growth factors to the station production zone and the station attraction zone. As the above analysis indicates, this should be done separately by travel market. This approach should produce

a reasonable forecast of this small group of trips without imposing a significant burden on the existing travel model in terms of new data requirements or computer run time.

There are three reasons for identifying the non-resident trips by separate market: 1) it facilitates their identification and description, 2) their time periods and access modes are generally different, and 3) their methods of being forecasted are slightly different. The forecasting process uses a two-part approach. Part 1 is to Fratar the base trip table using growth factors derived from external volumes and employment near Metrorail stations. Part 2 is to separately forecast trips for new stations and merge the forecasts for the new and existing stations. This accomplishes the objectives of a well-considered process that is not overly complex or time-consuming to apply.

7.2.1 Existing Metrorail Stations

For existing stations, the process is straightforward. Two sets of growth factors are developed: external and internal. For each Metrorail station, the consultant has defined a range of external station numbers, representing the external stations that are most likely to feed external trips to the Metrorail station. These are shown in Table 7-19. For example, the Shady Grove station was judged to be most directly served by external stations 3696 through 3710, inclusive. So for the E/I purposes, the boarding (production) growth factor is the ratio of the sum of the future cordon volume at all of those stations (from the Ext_PsAs.dbf file) to the sum of the 2007 cordon volume at all of those stations.

The internal factor is computed similarly, as the future-to-2007 ratio of the total employment for all zones whose centroid lies within 0.5 mi of each Metrorail station (straight line). The external and internal factors are applied as shown in Table 7-20

Table 7-19: Metrorail Station Codes

2007 Survey Station No.	Station Name	Station Zone	Station Node	Segment	Starting Cordon Station	Ending Cordon Station
1	METRO CENTER	23	8015	9	3676	3722
6	FARRAGUT NORTH	39	8014	1	3696	3710
10	DUPONT CIRCLE	45	8013	1	3696	3710
13	WOODLEY PARK ZOO	150	8012	1	3696	3710
14	CLEVELAND PARK	149	8011	1	3696	3710
15	VAN NESS UDC	146	8010	1	3696	3710
16	TENLEYTOWN	101	8009	1	3696	3710
17	FRIENDSHIP HEIGHTS	97	8008	1	3696	3710
20	BETHESDA	662	8007	1	3696	3710
21	MEDICAL CENTER	664	8006	1	3696	3710
22	GROSVENOR	681	8005	1	3696	3710
23	WHITE FLINT	687	8004	1	3696	3710
24	TWINBROOK	692	8003	1	3696	3710
25	ROCKVILLE	717	8002	1	3696	3710
26	SHADY GROVE	520	8001	1	3696	3710

27	GALLERY PLACE	20	8016	9	3676	3722
31	JUDICIARY SQUARE	18	8017	9	3676	3722
34	UNION STATION	283	8018	9	3676	3722
37	RHODE ISLAND AVENUE	218	8019	4	3713	3721
38	BROOKLAND CUA	231	8020	4	3713	3721
39	FORT TOTTEN	243	8021	4	3713	3721
40	TAKOMA PARK	620	8022	2	3704	3712
41	SILVER SPRING	624	8023	2	3704	3712
44	FOREST GLEN	629	8024	2	3704	3712
45	WHEATON	560	8025	2	3704	3712
46	GLENMONT	552	8026	2	3704	3712
47	MCPHERSON SQUARE	28	8068	9	3676	3722
50	FARRAGUT WEST	37	8067	9	3676	3722
53	FOGGY BOTTOM GWU	57	8066	9	3676	3722
54	ROSSLYN	1472	8065	8	3685	3695
55	ARLINGTON CEMETERY	1487	8056	7	3676	3684
56	PENTAGON	1496	8055	7	3676	3684
57	PENTAGON CITY	1493	8054	7	3676	3684
58	CRYSTAL CITY	1501	8053	7	3676	3684
59	NATIONAL AIRPORT	1510	8052	7	3676	3684
62	BRADDOCK ROAD	1596	8051	7	3676	3684
63	KING STREET	1593	8050	7	3676	3684
66	EISENHOWER AVENUE	1591	8049	7	3676	3684
67	HUNTINGTON	2059	8048	7	3676	3684
70	FEDERAL TRIANGLE	22	8069	9	3676	3722
71	SMITHSONIAN	13	8070	9	3676	3722
74	L'ENFANT PLAZA	380	8037	9	3676	3722
78	FEDERAL CENTER SW	379	8071	5	3720	3722
79	CAPITOL SOUTH	287	8072	5	3720	3722
80	EASTERN MARKET	288	8073	5	3720	3722
81	POTOMAC AVENUE	297	8074	5	3720	3722
82	STADIUM ARMORY	305	8075	5	3720	3722
85	MINNESOTA AVENUE	257	8076	5	3720	3722
86	DEANWOOD	258	8077	5	3720	3722
87	CHEVERLY	1021	8078	5	3720	3722
88	LANDOVER	1014	8079	5	3720	3722
89	NEW CARROLLTON	1126	8080	5	3720	3722
90	MT VERNON SQUARE-UDC	195	8035	4	3713	3721
91	SHAW HOWARD U	191	8034	4	3713	3721
94	U STREET CARDOZO	189	8033	4	3713	3721

97	COLUMBIA HEIGHTS	173	8032	4	3713	3721
98	GEORGIA AVENUE	162	8031	4	3713	3721
99	WEST HYATTSVILLE	968	8030	3	3713	3721
100	PRINCE GEORGE'S PLAZA	972	8029	3	3713	3721
101	COLLEGE PARK - U OF MD	981	8028	3	3713	3721
102	GREENBELT	895	8027	3	3713	3721
103	ARCHIVES-NAVY MEMORIAL	16	8036	9	3676	3722
104	WATERFRONT	386	8038	6	3722	3722
105	NAVY YARD	372	8039	6	3722	3722
108	ANACOSTIA	363	8040	6	3722	3722
111	CONGRESS HEIGHTS	357	8041	6	3722	3722
112	NAYLOR ROAD	853	8043	6	3722	3722
113	SUITLAND	844	8044	6	3722	3722
114	BRANCH AVENUE	830	8045	6	3722	3722
115	BENNING ROAD	256	8081	5	3720	3722
116	CAPITOL HEIGHTS	1039	8082	5	3720	3722
117	ADDISON ROAD	1064	8083	5	3720	3722
118	VAN DORN STREET	1546	8046	7	3676	3684
119	FRANCONIA-SPRINGFIELD	2031	8047	7	3676	3684
120	COURT HOUSE	1477	8064	8	3685	3695
121	CLARENDON	1480	8063	8	3685	3695
122	VIRGINIA SQUARE GMU	1458	8062	8	3685	3695
123	BALLSTON	1415	8061	8	3685	3695
124	EAST FALLS CHURCH	1435	8060	8	3685	3695
125	WEST FALLS CHURCH	1925	8059	8	3685	3695
126	DUNN LORING	1968	8058	8	3685	3695
127	VIENNA	1806	8057	8	3685	3695
128	SOUTHERN AVENUE	858	8042	6	3722	3722
129	NEW YORK AVE	206	8098	4	3713	3721
132	MORGAN BLVD	1090	8085	5	3720	3722
133	LARGO TOWN CENTER	1093	8086	5	3720	3722
134	<i>Tysons East</i>	<i>1869</i>	<i>8087</i>	<i>10</i>	<i>3689</i>	<i>3695</i>
135	<i>Tysons Central RT123</i>	<i>1844</i>	<i>8088</i>	<i>10</i>	<i>3689</i>	<i>3695</i>
136	<i>Tysons West</i>	<i>1833</i>	<i>8089</i>	<i>10</i>	<i>3689</i>	<i>3695</i>
137	<i>Tyson Central Rt 7</i>	<i>1845</i>	<i>8090</i>	<i>10</i>	<i>3689</i>	<i>3695</i>
138	<i>Wiehle Ave</i>	<i>1727</i>	<i>8091</i>	<i>10</i>	<i>3689</i>	<i>3695</i>
139	<i>Reston Parkway</i>	<i>1741</i>	<i>8092</i>	<i>10</i>	<i>3689</i>	<i>3695</i>
140	<i>Herndon/Monroe</i>	<i>1744</i>	<i>8093</i>	<i>10</i>	<i>3689</i>	<i>3695</i>
141	<i>Route 28</i>	<i>1704</i>	<i>8094</i>	<i>10</i>	<i>3689</i>	<i>3695</i>
142	<i>Dulles Airport</i>	<i>1703</i>	<i>8095</i>	<i>10</i>	<i>3689</i>	<i>3695</i>

143	<i>Western Regional</i>	2321	8096	10	3689	3695
144	<i>Route 772/DGWay</i>	2323	8097	10	3689	3695
145	<i>Potomac Yards</i>	1577	8084	7	3676	3684

Note: Stations 134-145, shown in italics, are future stations that were not in the 2007 survey; these station numbers were assigned by the consultant and are shown only for reference. The “Segments” are shown in Figure 7-1.

Table 7-20: Application Growth Factors

Table	Trips	Production Factor	Attraction Factor
1	E/I work auto acc	external	internal
2	E/I non-work auto acc	external	internal
3	E/I work walk/train acc	external	internal
4	E/I non-work walk/train acc	external	internal
5	NHB visitor	internal	internal
6	NHB business-related	internal	internal
7	air passenger	internal	internal

7.2.2 New Stations

According to the MWCOG model input file used to represent rail stations (STATION.DBF), 12 new Metrorail stations are currently being considered, as shown in Table 7-21. These are all on the new Silver Line except for Potomac Yards, which is on the Blue/Yellow Line.

Because of the importance of new stations in the planning process and because the Fratar step described above cannot account for new stations, it became necessary to develop a separate procedure for new stations that could be integrated with the process for existing stations. This turned out to be more complex than was initially thought.

Table 7-21: New Metrorail Stations

MWCOG Station Node	Station	Nearest Zone
8087	Tysons East	1869
8088	Tysons Central Rt123	1844
8089	Tysons West	1833
8090	Tysons Central Rt 7	1845
8091	Wiehle Ave	1727
8092	Reston Parkway	1741
8093	Herndon/Monroe	1744
8094	Route 28	1704
8095	Dulles Airport	1703
8096	Western Regional	2321
8097	Route 772/DGWay	2323

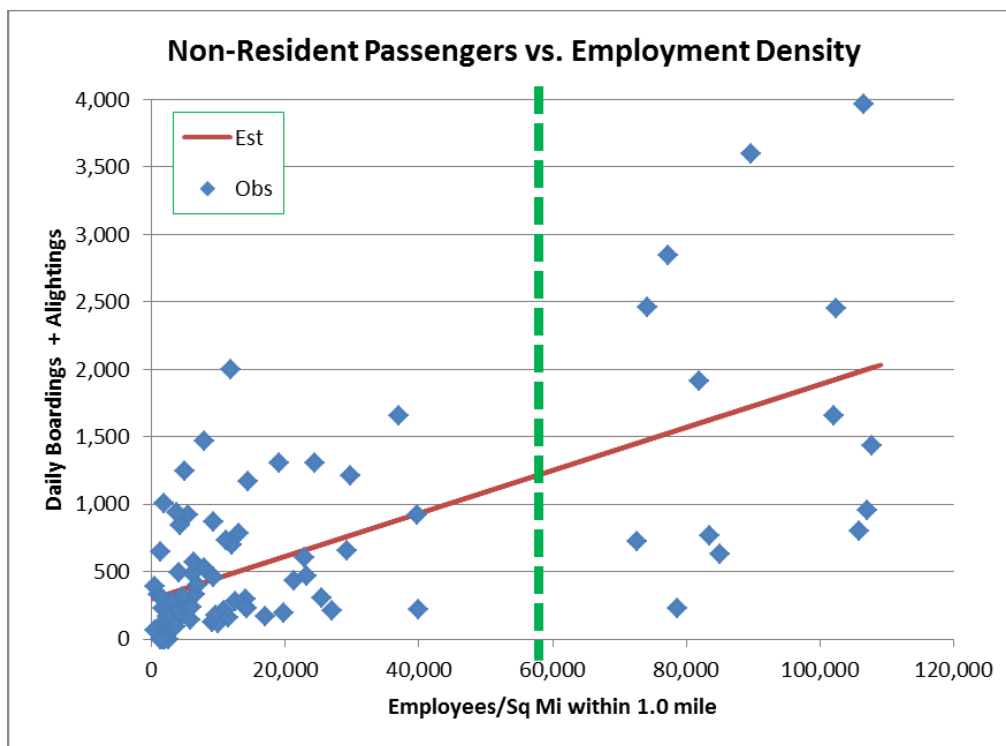
8084	Potomac Yards	1577
------	---------------	------

The first step was to develop a submodel to estimate trip ends for the new stations. Since non-resident trips are currently only 5.1% of the Metrorail total, an extremely rigorous procedure is not warranted. It turns out that the total non-resident activity at a station (B+A) is related to the level of development near the station. The independent variables should be limited to the available data so as not to require the creation of any new data. It seemed that population and employment near the station, and the associated densities, would be a reasonable choice. The consultant created a calibration file consisting of 86 observations (one per station). The dependent variable is the observed number of non-resident B+A, as shown in Table 7-18. The consultant created several candidate independent variables, based on “floating zones” surrounding each rail station. For each station, the consultant considered zones whose centroids are within 0.25 mi, 0.50 mi, and 1.0 mi of the station (straight line) and calculated the total population, total employment, and total land area within those radii. The area type of the “station zone” (STAZ in STATION.DBF) was also included.

The initial analysis of this data indicated that the Union Station and National Airport stations were clear outliers with respect to non-resident travel. Union Station has 11,388 non-resident trips and National Airport’s non-resident share is 60.5%. So those stations were dropped from the model estimation. For the remaining 84 stations, the consultant calculated the correlation between non-resident B+A and the independent variables. The employment variables gave the highest correlation, with the best of those being the employment density within 1.0 mile ($r = 0.648$). Figure 7-2 is the graph of that variable.

This figure suggests an approximately linear relationship between non-resident B+A and employment density. Also, the data is clearly split between the smaller area stations (less than 40,000 empl/sq mi) and larger area stations (more than 70,000 empl/sq mi). Most of the new stations will be in suburban areas that will have the smaller employment density (although a few, such as some of the Tyson’s Corner stations and Potomac Yards, may have fairly high employment densities within 1.0 mile).

Figure 7-2: Non-Resident B+A vs. Employment Density



The straight line shown in Figure 7-2 is the equation:

$$\text{non-resident B+A} = 300 + 0.016 * \text{empl/sq mi within 1.0 mi} \quad (1)$$

This equation was based on 84 data points, and it seems to reasonably represent both the low density and high density stations (although the high density stations have a *lot* of scatter). Another equation was estimated for only the low density stations, but it was nearly identical and so is not shown. The r^2 is 0.42 and the percent RMSE is 66% (42% when considering only the low density stations).

Next, a process is needed to split the total B+A into boardings and alightings. Since the data are maintained in P/A format, the variation by station and by trip type can be significant (although the grand total for all stations is a 50/50 split, as it must be). The data are too thin to permit separate estimation by the seven trip purposes, but when the four E/I purposes are grouped, it is clear that their split is much different from that of the three NHB purposes, which makes sense. Logically, the E/I production split should be higher for suburban stations and lower for urban stations. For the NHB purposes, the production split should generally be near 50/50 and for most stations, it is.

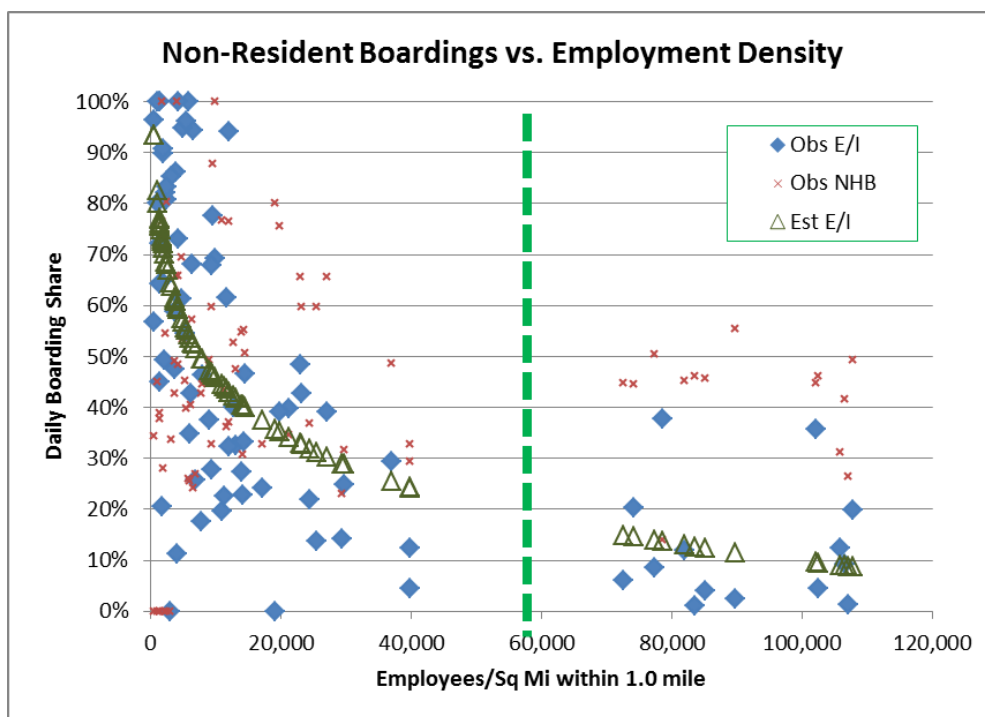
As Figure 7-3 shows, the 2007 survey data shows that for E/I trips the boarding (production) share of trips by station is in fact high for low density stations and steadily decreases as the employment density near the station increases. The observed variation in the ratio of boardings (productions) to total B+A is substantial, ranging from over 96% at Glenmont to 1% at Judiciary Square. This graph also shows that for NHB trips, the boarding share does not vary with employment density. A non-linear least squares line was estimated with the equation:

$$E/I \text{ boarding proportion} = 1.895 - 0.156 * \ln(\text{empl/sq mi within 1.0 mi}) \quad (2)$$

It is also necessary to set bounds on this function so that the estimated value is always between 0.0 and 1.0. The r^2 is 0.50 and the %RMSE is 47%. For the NHB purposes, a 50/50 split will be used.

The above process estimates the trip ends by purpose and direction at the new Metrorail stations. In order to produce trip patterns, a key assumption is made: that the O/D patterns for a new station will be similar to those of the nearest existing station. Given this simplified method, that seems reasonable. The exception is Dulles Airport, whose pattern will probably be closer to that of National Airport than of Vienna (the closest existing station to Dulles). The final step is to again Fratar the initial table to account for the estimated trip ends and synthesized trip patterns of the new stations.

Figure 7-3: Boarding Share



7.3 Forecasting Script

The Voyager script to implement this model is shown in the Appendix. The entire process uses ten steps, described as follows:

1. *2007 trip end summary*

The process must always start with the file SURV2007.TRP, which contains the seven tables by purpose built from the on-board survey. This step summarizes the rows and columns for later use.

2. *External growth factor*

This step summarizes the regional external trip ends by external stations for 2007 and the forecast year. These are grouped for each Metrorail station, as shown in the fixed input file

STANAME07.DBF (this file contains the data shown in Table 7-19). The external factor is calculated for each Metrorail station as the ratio of the future external group total to the 2007 external group total. The factor is output along with the station's zone number (STAZ).

3. *Internal growth factor*

This step is similar to step 2 except that it uses the future and 2007 employment for all zones whose centroid is within 0.5 mi (straight line) of the Metrorail station, based on coordinates. If no centroid is within 0.5 mi, use the data for the zone nearest the Metrorail station (STAZ). The factor is output along with the Metrorail station's zone number.

4. *Reconcile growth factors and trip ends*

Steps 2 and 3 calculate two growth factors for each station. These will be applied to the 2007 trip tables by purpose. In Voyager, the FRATAR program issues a warning if a growth factor is applied to a row or column whose total is zero. This step checks the growth factors to the trip ends and if any zone's trip ends are zero, then that zone's growth factor is set to zero. This step does not affect the calculations; its only purpose is to keep FRATAR from issuing a Return Code of 1 ("warning") for this run.

5. *Initial Fratar*

This step applies the reconciled growth factors to the 2007 trip tables by purpose, using the protocol shown in Table 7-20. For the E/I purposes, the desired production (boarding) total is considered the governing total. For the NHB purposes, the governing total is the average of the factored production and attraction totals. This creates interim tables for the existing stations.

6. *For new stations, estimate B+A and boarding split*

Apply equation (1) to estimate the total B+A for the new stations. "New" stations are those that had no non-resident B+A in the 2007 survey, except for Congress Heights and Morgan Blvd, which also had no B+A in that survey. (Those stations are assumed to never have non-resident trips.) Then, calculate the employment per square mile for those zones whose centroid is within 1.0 mi of the station and apply the above model to calculate the B+A total. This produces the desired trip ends at the new stations, but not the station-station O/D pattern. (Throughout this chapter, "station-station O/D" refers to a zone-zone matrix where the zones represent the station zones [STAZ]. This process assumes there will never be more than one Metrorail station in a zone.) Then, apply equation (2) to estimate the E/I boarding share for the new stations.

7. *Find nearest existing station to each new Metrorail station*

For each new station, this process assumes that its station-station O/D pattern will be similar to that of the existing station that is closest to it. In this step, find the closest existing station to each new station (straight line, based on coordinates). For the Silver Line stations, that is West Falls Church, Dunn Loring, and Vienna.

8. *Adjust interim tables for new Metrorail stations*

In this step, modify the interim table from step 5. For each new station X's zone, find the associated existing station Y's zone and replace X's row with Y's data. Also, for all rows, replace

X's column with Y's data. Do this separately by trip purpose and output a new trip end summary by purpose.

9. *Merge trip ends for existing and new Metrorail stations*

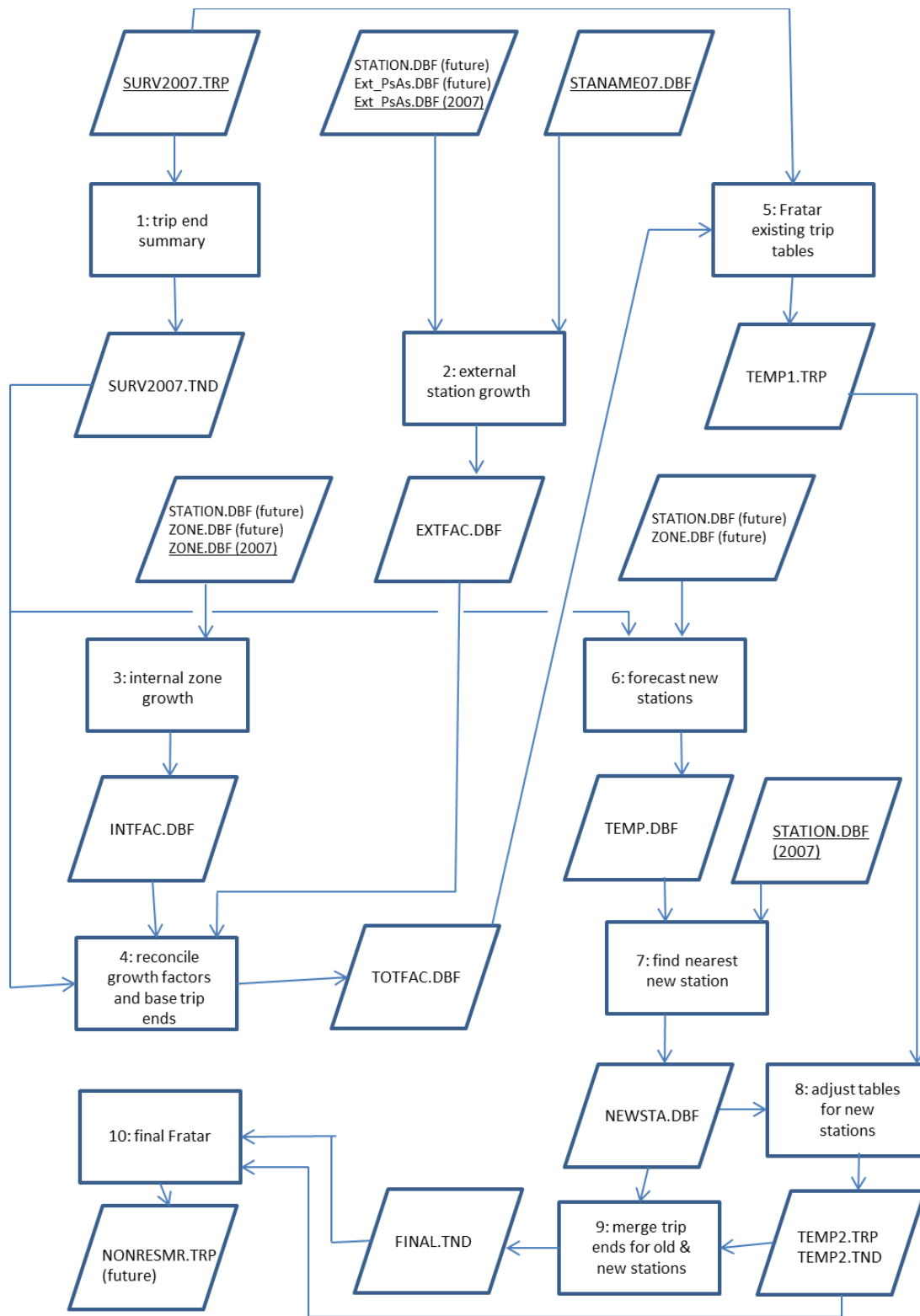
Merge the interim future trip ends from step 5 with the new Metrorail station trip ends from step 6. Split the new station data by the seven non-resident trip purposes, based on data from the 2007 Orange Line stations from Vienna to Ballston. For the E/I purposes, use the boarding shares estimated in step 6. For Dulles Airport, use the purpose and boarding splits from National Airport. Reconcile the desired trip ends with the row and column totals from the interim table, as in step 4 (this requires a surprisingly large amount of code).

10. *Final Fratar*

Fratar the interim non-resident trip table (as modified in step 8) to match the merged trip ends from step 9. As in step 3, set the maximum number of iterations to 19. This is somewhat arbitrary, but in testing it seemed to produce reasonable results.

This process is depicted in Figure 7-4. The final output file is shown in the lower left corner.

Figure 7-4: Model Application Flowchart



Note: Underlined files are basic inputs. Numbers in process boxes are the sequential step numbers.

7.4 Test Results

This process was applied to estimate the 2040 non-resident Metrorail trips. The results for the new stations are shown in Table 7-22.

Table 7-22: Estimated New Station Non-Resident Metrorail Boardings+Alightings, 2040

MWCOG Station Node	Station	Non-Resident Trips (B+A)	Nearest Zone
8087	Tysons East	565	1869
8088	Tysons Central Rt123	884	1844
8089	Tysons West	705	1833
8090	Tysons Central Rt 7	967	1845
8091	Wiehle Ave	539	1727
8092	Reston Parkway	671	1741
8093	Herndon/Monroe	616	1744
8094	Route 28	556	1704
8095	Dulles Airport*	1,382	1703
8096	Western Regional	379	2321
8097	Route 772/DGWay	436	2323
8084	Potomac Yards	457	1577

* This model should *not* be applied to the future Dulles Airport station. Since National Airport is an outlier in the 2007 data, one could reasonably expect Dulles Airport to be a similar anomaly in the future. The non-resident Metrorail trips for Dulles Airport should be obtained directly from the new air passenger model being developed for that purpose. The 2007 surveyed non-resident total for National Airport is 5,453.

Table 7-23 – Table 7-30 show the final 2040 trips squeezed to station jurisdiction and Table 7-31 – Table 7-38 show the final 2040 trips squeezed to line segment. Table 7-39 – Table 7-46 compare the 2040 and 2007 values. In these tables, some of the values are negative. This means that in 2040, the presence of the Silver Line actually pulls a few non-resident trips away from other parts of the Metrorail system. This may be surprising, but it does not seem impossible.

In summary, there were about 33,000 total non-resident Metrorail trips in 2007 and the initial Fratar estimated that the trips at those same stations would grow by about 45% to 47,700 in 2040. The new stations are estimated to add another 9,600 non-resident trips, for a total of about 57,300.

Table 7-23: Station Jurisdiction 2040 Forecast Summary (E/I Work, Auto Access)

E/I Commuters, Auto Access (incl. new stations) Station-to-Station, P/A Format									
		Destination District						Total	
		1	2	3	4	5	6	7	
O	1 DC	649	0	85	115	20	0	0	869
r	2 Montgmry	777	22	0	203	0	4	0	1006
i	3 PrinGeor	1180	0	0	233	36	180	0	1629
g	4 Arlingtn	85	0	16	222	9	0	0	332
i	5 Alexandr	19	10	0	0	0	0	0	29
n	6 Fairfax	509	59	0	530	12	181	0	1291
	7 Loudoun	30	0	0	51	0	20	0	101
Total		3249		101		77		0	5257
			91		1354		385		

Table 7-24: Station Jurisdiction 2040 Forecast Summary (E/I Non-Work, Auto Access)

E/I Non-Work, Auto Access (incl. new stations) Station-to-Station, P/A Format									
		Destination District						Total	
		1	2	3	4	5	6	7	
O	1 DC	258	13	0	16	7	0	0	294
r	2 Montgmry	452	59	0	8	0	80	11	610
i	3 PrinGeor	539	0	0	30	0	78	0	647
g	4 Arlingtn	187	0	0	0	0	0	0	187
i	5 Alexandr	131	0	0	51	0	0	0	182
n	6 Fairfax	1127	0	0	113	12	0	0	1252
	7 Loudoun	115	0	0	0	0	0	0	115
Total		2809		0		19		11	3287
			72		218		158		

Table 7-25: Station Jurisdiction 2040 Forecast Summary (E/I Work, Walk Access)

E/I Commuters, Walk Acc/Train Transfer (incl. new stations) Station-to-Station, P/A Format									
		Destination District						Total	
		1	2	3	4	5	6	7	
O	1 DC	6779	809	811	1890	355	572	0	11216
r	2 Montgmry	470	393	57	51	13	0	0	984
i	3 PrinGeor	215	13	51	46	0	0	0	325
g	4 Arlingtn	347	0	0	315	11	105	0	778
i	5 Alexandr	32	0	0	37	0	0	0	69
n	6 Fairfax	284	23	0	121	0	97	0	525
	7 Loudoun	42	0	0	0	0	0	0	42
Total		8169		919		379		0	13939
			1238		2460		774		

Table 7-26: Station Jurisdiction 2040 Forecast Summary (E/I Non-Work, Walk Access)

E/I Non-Work, Walk Acc/Train Transfer (incl. new stations) Station-to-Station, P/A Format									
		Destination District							Total
		1	2	3	4	5	6	7	
Origin Station	1 DC	1783	338	88	342	65	358	97	3071
	2 Montgmy	347	38	0	0	30	31	0	446
	3 PrinGeor	35	0	0	38	0	0	0	73
	4 Arlingtn	267	0	0	276	0	75	0	618
	5 Alexandr	253	0	0	0	21	68	0	342
	6 Fairfax	449	0	78	52	0	0	0	579
	7 Loudoun	77	0	0	0	0	0	0	77
Total		3211		166		116		97	5206
			376		708		532		

Table 7-27: Station Jurisdiction 2040 Forecast Summary (Visitor)

Visitor, Walk Access (incl. new stations) Station-to-Station, P/A Format									
		Destination District							Total
		1	2	3	4	5	6	7	
Origin Station	1 DC	4902	805	490	1362	508	1775	135	9977
	2 Montgmy	822	20	0	48	9	0	0	899
	3 PrinGeor	682	16	56	129	0	62	13	958
	4 Arlingtn	1485	53	101	813	27	104	0	2583
	5 Alexandr	616	0	48	124	5	77	16	886
	6 Fairfax	1265	4	0	397	19	0	0	1685
	7 Loudoun	53	0	0	30	0	0	0	83
Total		9825		695		568		164	17071
			898		2903		2018		

Table 7-28: Station Jurisdiction 2040 Forecast Summary (NHB Business)

NHB Business-Related, Walk Access (incl. new stations) Station-to-Station, P/A Format									
		Destination District							Total
		1	2	3	4	5	6	7	
Origin Station	1 DC	1893	389	225	780	77	503	46	3913
	2 Montgmy	251	72	40	81	0	15	0	459
	3 PrinGeor	268	0	35	59	18	0	0	380
	4 Arlingtn	658	89	54	383	46	271	0	1501
	5 Alexandr	69	0	0	101	0	0	0	170
	6 Fairfax	335	15	0	240	35	13	0	638
	7 Loudoun	31	0	0	24	0	0	0	55
Total		3505		354		176		46	7116
			565		1668		802		

Table 7-29: Station Jurisdiction 2040 Forecast Summary (Air Passenger)

Air Passenger, Walk Access (incl. new stations) Station-to-Station, P/A Format									
		Destination District							Total
		1	2	3	4	5	6	7	
Origin Station	1 DC	0	0	0	206	0	232	0	438
	2 Montgmy	0	0	0	9	0	10	0	19
	3 PrinGeor	0	0	0	14	0	16	0	30
	4 Arlingtn	2851	214	58	434	152	439	24	4172
	5 Alexandr	0	0	0	21	0	23	0	44
	6 Fairfax	453	34	9	84	24	54	0	658
	7 Loudoun	0	0	0	0	0	0	0	0
Total		3304		67		176		24	5361
			248		768		774		

Table 7-30: Station Jurisdiction 2040 Forecast Summary (Total)

Total Non-Resident Passengers Station-to-Station, P/A Format									
		Destination District							Total
		1	2	3	4	5	6	7	
Origin Station	1 DC	16261	2357	1703	4713	1032	3434	277	29777
	2 Montgmy	3122	605	97	403	52	139	12	4430
	3 PrinGeor	2917	30	141	552	53	337	13	4043
	4 Arlingtn	5883	355	230	2442	244	994	24	10172
	5 Alexandr	1122	10	47	334	25	168	17	1723
	6 Fairfax	4447	132	86	1521	101	345	0	6632
	7 Loudoun	351	0	0	103	0	19	0	473
Total		34103		2304		1507		343	57250
			3489		10068		5436		

Table 7-31: Line Segment 2040 Forecast Summary (E/I Work, Auto Access)

E/I Commuters, Auto Access (incl. new stations) Station-to-Station, P/A Format												
		Destination District										Total
		1	2	3	4	5	6	7	8	9	10	
Origin Station	1 NW Red	134	0	25	34	93	16	49	104	210	4	669
	2 N Red	60	8	0	35	0	39	10	40	170	0	362
	3 NE Green	15	0	0	19	0	38	100	83	477	110	842
	4 DC G/Y/R	0	0	0	0	0	0	0	0	88	0	88
	5 E Yel/Bl	91	0	0	0	48	12	133	44	466	0	794
	6 SE Grn	25	0	0	22	15	73	0	0	19	0	154
	7 S Yel/Bl	56	0	0	0	0	16	269	53	151	0	545
	8 W Orange	29	23	0	0	9	0	93	253	154	77	638
	9 DC Core	56	0	0	0	24	51	80	34	350	0	595
	10 Silver	71	36	0	0	10	0	124	156	173	0	570
Total		537		25		199		858		2258		5257
			67		110		245		767		191	

Table 7-32: Line Segment 2040 Forecast Summary (E/I Non-Work, Auto Access)

E/I Non-Work, Auto Access (incl. new stations) Station-to-Station, P/A Format												
		Destination District										Total
		1	2	3	4	5	6	7	8	9	10	Total
Origin Districts	1 NW Red	108	0	0	0	21	0	8	0	287	0	424
	2 N Red	0	0	0	0	0	0	0	15	124	76	215
	3 NE Green	21	0	0	9	0	12	13	33	164	54	306
	4 DC G/Y/R	0	0	0	0	0	0	0	0	0	0	0
	5 E Yel/Bl	16	0	0	0	9	0	0	8	143	0	176
	6 SE Grn	0	0	0	0	44	0	0	0	121	0	165
	7 S Yel/Bl	8	0	0	0	140	12	26	91	285	0	562
	8 W Orange	49	0	0	22	69	0	11	13	386	0	550
	9 DC Core	34	8	0	10	0	0	7	16	190	0	265
	10 Silver	64	0	0	0	89	0	16	19	436	0	624
Total		300	8	0	41	372	24	81	195	2136	130	3287

Table 7-33: Line Segment 2040 Forecast Summary (E/I Work, Walk Access)

E/I Commuters, Walk Acc/Train Transfer (incl. new stations) Station-to-Station, P/A Format												
		Destination District										Total
		1	2	3	4	5	6	7	8	9	10	Total
Origin Districts	1 NW Red	536	26	0	13	26	41	47	29	208	0	926
	2 N Red	0	17	57	52	0	0	0	0	75	0	201
	3 NE Green	0	13	23	0	0	28	0	0	12	0	76
	4 DC G/Y/R	7	35	0	35	0	19	0	0	33	0	129
	5 E Yel/Bl	11	8	0	0	119	0	30	28	93	0	289
	6 SE Grn	0	0	0	0	0	15	0	24	36	0	75
	7 S Yel/Bl	41	0	0	0	22	25	243	142	231	47	751
	8 W Orange	0	15	0	0	30	16	17	71	170	64	383
	9 DC Core	2181	407	41	556	259	1093	1580	728	3523	461	10829
	10 Silver	2	8	0	0	59	0	22	80	109	0	280
Total		2778	529	121	656	515	1237	1939	1102	4490	572	13939

Table 7-34: Line Segment 2040 Forecast Summary (E/I Non-Work, Walk Access)

E/I Non-Work, Walk Acc/Train Transfer (incl. new stations) Station-to-Station, P/A Format												
		Destination District										Total
		1	2	3	4	5	6	7	8	9	10	Total
Origin Districts	1 NW Red	250	0	0	59	0	0	29	10	659	21	1028
	2 N Red	25	4	0	15	0	25	30	0	55	0	154
	3 NE Green	0	0	0	0	0	0	0	18	10	0	28
	4 DC G/Y/R	57	79	0	0	10	38	22	0	92	0	298
	5 E Yel/Bl	42	0	0	0	16	0	17	42	51	0	168
	6 SE Grn	0	0	0	0	18	0	33	0	25	0	76
	7 S Yel/Bl	15	0	0	0	0	18	152	44	291	45	565
	8 W Orange	31	0	0	0	34	0	107	113	348	0	633
	9 DC Core	397	51	0	129	115	32	253	128	375	358	1838
	10 Silver	0	0	0	0	48	0	0	31	339	0	418
Total		817	134	0	203	241	113	643	386	2245	424	5206

Table 7-35: Line Segment 2040 Forecast Summary (Visitor)

		Visitor, Walk Access (incl. new stations) Station-to-Station, P/A Format										
		Destination District										
		1	2	3	4	5	6	7	8	9	10	Total
Origin District	1 NW Red	401	25	0	60	94	53	96	102	1321	0	2152
	2 N Red	32	8	0	29	0	0	10	12	55	0	146
	3 NE Green	48	16	56	63	10	0	119	10	153	0	475
	4 DC G/Y/R	198	0	0	54	0	44	95	0	170	0	561
	5 E Yel/Bl	246	0	0	0	41	98	133	229	415	326	1488
	6 SE Grn	0	0	0	0	10	0	0	35	120	0	165
	7 S Yel/Bl	60	19	59	120	156	109	882	296	1654	97	3452
	8 W Orange	89	9	0	30	178	0	84	91	529	0	1010
	9 DC Core	1292	131	295	132	446	131	1166	607	1546	1101	6847
	10 Silver	0	4	0	0	182	0	143	19	427	0	775
Total		2366		410		1117		2728		6390		17071
			212		488		435		1401		1524	

Table 7-36: Line Segment 2040 Forecast Summary (NHB Business)

		NHB Business-Related, Walk Access (incl. new stations) Station-to-Station, P/A Format										
		Destination District										
		1	2	3	4	5	6	7	8	9	10	Total
Origin District	1 NW Red	152	12	26	10	14	35	90	91	357	15	802
	2 N Red	9	30	0	45	43	0	0	12	30	0	169
	3 NE Green	0	0	35	20	0	62	18	0	19	0	154
	4 DC G/Y/R	22	0	9	0	0	23	122	0	81	47	304
	5 E Yel/Bl	24	0	53	13	21	0	12	0	62	0	185
	6 SE Grn	0	0	0	0	0	103	115	11	151	14	394
	7 S Yel/Bl	88	52	0	0	46	100	342	137	349	25	1139
	8 W Orange	16	0	0	22	69	25	115	114	246	203	810
	9 DC Core	567	73	30	99	187	46	538	183	710	311	2744
	10 Silver	17	9	0	0	13	19	177	0	180	0	415
Total		895		153		393		1529		2185		7116
			176		209		413		548		615	

Table 7-37: Line Segment 2040 Forecast Summary (Air Passenger)

		Air Passenger, Walk Access (incl. new stations) Station-to-Station, P/A Format										
		Destination District										
		1	2	3	4	5	6	7	8	9	10	Total
Origin District	1 NW Red	0	0	0	0	0	0	13	0	0	15	28
	2 N Red	0	0	0	0	0	0	0	0	0	0	0
	3 NE Green	0	0	0	0	0	0	0	0	0	0	0
	4 DC G/Y/R	0	0	0	0	0	0	30	0	0	34	64
	5 E Yel/Bl	0	0	0	0	0	0	20	0	0	22	42
	6 SE Grn	0	0	0	0	0	0	14	0	0	16	30
	7 S Yel/Bl	744	22	0	313	419	157	365	425	1468	228	4141
	8 W Orange	0	0	0	0	0	0	54	0	0	59	113
	9 DC Core	0	0	0	0	0	0	152	0	0	171	323
	10 Silver	118	3	0	50	67	24	58	66	234	0	620
Total		862		0		486		706		1702		5361
			25		363		181		491		545	

Table 7-38: Line Segment 2040 Forecast Summary (Total)

		Total Non-Resident Passengers Station-to-Station, P/A Format										
		Destination District										
		1	2	3	4	5	6	7	8	9	10	Total
Origin Station	1 NW Red	1586	63	51	177	249	145	336	336	3038	54	6035
	2 N Red	129	67	57	175	43	64	50	77	509	78	1249
	3 NE Green	85	30	114	111	10	137	247	148	835	164	1881
	4 DC G/Y/R	284	112	10	89	10	124	270	0	466	79	1444
	5 E Yel/Bl	432	9	53	13	252	111	345	352	1227	349	3143
	6 SE Grn	25	0	0	22	86	191	163	70	473	29	1059
	7 S Yel/Bl	1012	93	58	433	779	439	2272	1190	4436	444	11156
	8 W Orange	212	46	0	75	393	41	481	652	1835	402	4137
	9 DC Core	4532	670	366	921	1034	1353	3778	1696	6693	2396	23439
	10 Silver	272	59	0	50	466	42	527	369	1922	0	3707
Total		8569		709		3322		8469		21434		57250
			1149		2066		2647		4890		3995	

Table 7-39: 2040-2007 Difference by Station Jurisdiction (E/I Work, Auto Access)

		E/I Commuters, Auto Access (incl. new stations) Station-to-Station, P/A Format							
		Destination District							
		1	2	3	4	5	6	7	Total
Origin Station	1 DC	191	0	58	27	13	0	0	289
	2 Montgmy	194	0	0	121	0	4	0	319
	3 PrinGeor	326	0	0	63	17	148	0	554
	4 Arlingtn	-14	0	3	115	2	0	0	106
	5 Alexandr	6	2	0	0	0	0	0	8
	6 Fairfax	225	39	0	351	2	167	0	784
	7 Loudoun	30	0	0	51	0	20	0	101
Total		958		61		34		0	2161
			41		728		339		

Table 7-40: 2040-2007 Difference by Station Jurisdiction (E/I Non-Work, Auto Access)

		E/I Non-Work, Auto Access (incl. new stations) Station-to-Station, P/A Format							
		Destination District							
		1	2	3	4	5	6	7	Total
Origin Station	1 DC	90	0	0	7	-4	0	0	93
	2 Montgmy	165	34	0	-1	0	65	11	274
	3 PrinGeor	194	0	0	6	0	54	0	254
	4 Arlingtn	59	0	0	0	0	0	0	59
	5 Alexandr	92	0	0	8	0	0	0	100
	6 Fairfax	698	0	0	45	1	0	0	744
	7 Loudoun	115	0	0	0	0	0	0	115
Total		1413		0		-3		11	1639
			34		65		119		

Table 7-41: 2040-2007 Difference by Station Jurisdiction (E/I Work, Walk Access)

E/I Commuters, Walk Acc/Train Transfer (incl. new stations) Station-to-Station, P/A Format									
		Destination District							Total
		1	2	3	4	5	6	7	
Origin Station	1 DC	1570	221	629	750	128	505	0	3803
	2 Montgmy	136	124	15	21	3	0	0	299
	3 PrinGeor	69	3	18	17	0	0	0	107
	4 Arlingtn	108	0	0	123	4	84	0	319
	5 Alexandr	11	0	0	12	0	0	0	23
	6 Fairfax	179	11	0	90	0	87	0	367
	7 Loudoun	42	0	0	0	0	0	0	42
Total		2115		662		135		0	4960
			359		1013		676		

Table 7-42: 2040-2007 Difference by Station Jurisdiction (E/I Non-Work, Walk Access)

E/I Non-Work, Walk Acc/Train Transfer (incl. new stations) Station-to-Station, P/A Format									
		Destination District							Total
		1	2	3	4	5	6	7	
Origin Station	1 DC	542	113	44	118	9	289	97	1212
	2 Montgmy	100	20	0	0	19	24	0	163
	3 PrinGeor	6	0	0	16	0	0	0	22
	4 Arlingtn	43	0	0	98	0	55	0	196
	5 Alexandr	115	0	0	0	7	53	0	175
	6 Fairfax	320	0	63	38	0	0	0	421
	7 Loudoun	77	0	0	0	0	0	0	77
Total		1203		107		35		97	2266
			133		270		421		

Table 7-43: 2040-2007 Difference by Station Jurisdiction (Visitor)

Visitor, Walk Access (incl. new stations) Station-to-Station, P/A Format									
		Destination District							Total
		1	2	3	4	5	6	7	
Origin Station	1 DC	414	390	265	433	213	1418	135	3268
	2 Montgmy	418	2	0	12	0	0	0	432
	3 PrinGeor	270	5	37	86	0	54	13	465
	4 Arlingtn	288	16	37	253	-4	70	0	660
	5 Alexandr	180	0	32	45	-3	68	16	338
	6 Fairfax	854	4	0	322	3	0	0	1183
	7 Loudoun	53	0	0	30	0	0	0	83
Total		2477		371		209		164	6429
			417		1181		1610		

Table 7-44: 2040-2007 Difference by Station Jurisdiction (NHB Business)

		Station-to-Station, P/A Format NHB Business-Related, Walk Access (incl. new stations)							
		Destination District							
		1	2	3	4	5	6	7	Total
Origin Station	1 DC	378	104	94	302	33	427	46	1384
	2 Montgmy	61	24	24	41	0	15	0	165
	3 PrinGeor	181	0	17	49	8	0	0	255
	4 Arlingtn	230	49	24	150	31	234	0	718
	5 Alexandr	18	0	0	54	0	0	0	72
	6 Fairfax	239	15	0	196	20	13	0	483
	7 Loudoun	31	0	0	24	0	0	0	55
Total		1138		159		92		46	3132
			192		816		689		

Table 7-45: 2040-2007 Difference by Station Jurisdiction (Air Passenger)

		Air Passenger, Walk Access (incl. new stations) Station-to-Station, P/A Format							
		Destination District							
		1	2	3	4	5	6	7	Total
Origin Station	1 DC	0	0	0	28	0	232	0	260
	2 Montgmy	0	0	0	1	0	10	0	11
	3 PrinGeor	0	0	0	5	0	16	0	21
	4 Arlingtn	1789	136	41	273	94	358	24	2715
	5 Alexandr	0	0	0	7	0	23	0	30
	6 Fairfax	453	34	9	72	24	54	0	646
	7 Loudoun	0	0	0	0	0	0	0	0
Total		2242		50		118		24	3683
			170		386		693		

Table 7-46: 2040-2007 Difference by Station Jurisdiction (Total)

		Total Non-Resident Passengers Station-to-Station, P/A Format							
		Destination District							
		1	2	3	4	5	6	7	Total
Origin Station	1 DC	3185	831	1089	1669	391	2864	277	10306
	2 Montgmy	1081	206	38	197	22	117	12	1673
	3 PrinGeor	1046	8	72	243	25	273	13	1680
	4 Arlingtn	2506	200	108	1014	123	798	24	4773
	5 Alexandr	420	2	31	128	5	143	17	746
	6 Fairfax	2993	101	71	1101	48	320	0	4634
	7 Loudoun	351	0	0	103	0	19	0	473
Total		11582		1409		614		343	24285
			1348		4455		4534		

7.5 Model Integration

The Voyager script in the Appendix is almost ready for production use. It was tested using 2040 data and so the hard-coded references to 2040 must be replaced with substitutable parameters. The fixed files SURV2007.TRP and STANAME07.DBF (being transmitted separately) should probably be stored in the \Support directory. (On the other hand, since STANAME07.DBF contains the names of all Metrorail stations, it may be better to store this in the \inputs directory of the scenario being tested.) In addition, the following 2007 standard input files must always be available: ZONE.DBF, STATION.DBF, EXT_PSAS.DBF. The procedure described in this memo does not vary by speed feedback iteration and thus needs to be performed only once. The script runs in about 2 minutes on a reasonably fast computer.

Otherwise, the script is ready to be incorporated into the MWCOG model stream. This would probably be done by renaming the script to something like Non_Resident_MR.s and including it as the first step in Transit_Assignment.bat. The non-resident trip tables would then be included in the subsequent Combine_Tables_For_TrAssign.s script. The non-resident tables would be added to the tables from mode choice based on the time periods and access modes shown in Table 7-47.

Table 7-47: Trip Tables for Assignment

Table	Definition	Period	Access Mode
1	E/I Commuters, auto access	AM	PnR
2	E/I Non-Work, auto access	OP	PnR
3	E/I Commuters, walk access/train transfer	AM	PnR
4	E/I Non-Work, walk access/train transfer	OP	PnR
5	NHB Visitors (walk access)	OP	Walk
6	NHB Business (walk access)	OP	Walk
7	Air Passengers (walk access)*	OP	walk

* The air passenger Metrorail trip process described here is a temporary substitute for the more rigorous procedure described the next chapter.

8 Air Passenger Model (Task Order 6)

There are few examples of travel demand models that have the ability to estimate the number of air passengers to and from regional airports. This is mainly because households make relatively few trips to the airport, so most home interviews have trouble capturing this type of travel. But travel demand models which include these air passenger trips can be estimated if the air passengers are interviewed at the airport and if the total number of air passenger enplanements¹⁹ is known. Fortunately, MWCOG has an on-going program of air passenger surveys that cover the three commercial airports in the Metropolitan Washington region: Baltimore/Washington International Thurgood Marshall Airport (BWI), Washington Dulles International Airport (IAD), and Ronald Reagan Washington National Airport (DCA). The latest of these surveys that is documented occurred in 2009 (“2009 Washington-Baltimore Regional Air Passenger Survey” dated September 2010). It is possible to obtain the number of enplanements at the three airports from the Federal Aviation Administration (FAA) and/or from the two airport authorities in the region: the Metropolitan Washington Airports Authority (MWAA), which operates Washington Dulles and Reagan National airports, and the Maryland Aviation Authority (MAA), which operates Baltimore/Washington Marshall Airport.

As noted in Table 8-1, the 2009 survey (as well as earlier surveys) found that a majority of air passengers chose the closest airport to their home or business origin/destination. However, the survey also noted that there are other accessibility and quality of service issues that also have a significant impact on airport choice. These issues include: availability of public transportation, convenient flight times, least expensive airfare, availability of direct (non-stop) service, and airline loyalty. It is important to recognize that the air passenger model described in this chapter is not an *airport choice* model and does not attempt to address any issues other than travel to and from each of the airports. While it takes into consideration various aspects of the ground transportation services available at each airport, it does not consider government policy, air carrier service, air fare issues, or other such factors that influence passengers’ choice of airport.

Table 8-1: Primary Reason for Selecting Airport Used

Reason	BWI Marshall (BWI)	Washington Dulles (IAD)	Reagan National (DCA)	All Airports
Closest airport	56%	53%	72%	61%
Better public transportation	1%	1%	6%	3%
Better access roads and parking	3%	3%	1%	2%
Subtotal - Accessibility	60%	57%	79%	66%
More convenient flight times	5%	8%	8%	7%
Direct / Non-Stop Service	4%	11%	2%	5%

¹⁹ Enplanements refer to the number of passengers boarding a flight and include passengers originating at the local airport, passengers on a plane that has a stopover en route between another origin and destination, and passengers connecting from another flight.

Less expensive airfare	27%	16%	7%	17%
Frequent flyer	1%	4%	2%	2%
Only airport serving market	1%	3%	1%	2%
Subtotal – Quality of Service	39%	42%	20%	33%
Other	1%	1%	1%	1%
Total	100%	100%	100%	100%

Source: “2009 Washington-Baltimore Regional Air Passenger Survey”, September 2010, page 13.

In the early 1990’s a team of consultants developed a new type of airport access model that could be incorporated into a regional model structure. This model was developed jointly for San Francisco and Washington, using data from both urban areas. It used sample enumeration to pivot off of an air passenger survey and included a nested logit mode choice model (Harvey, 1988). In 2002, this model structure was modified to be a completely synthetic model – instead of relying on a survey, it included trip generation, distribution, and time of day components and was applied in New Orleans (Regional Planning Commission, 2002). In 2009, that same model structure was converted for application in Cube Voyager by the Baltimore Metropolitan Council, though the work is not documented. As part of the current work effort, this Cube version of the model was further adapted for estimating travel to/from the three major airports in the MWCOG model area. In this phase of the work (FY 12) the focus is on estimating Metrorail trips to Reagan National Airport. The air passenger model consists of a three step process. These steps are:

1. Estimate total air passengers by obtaining the total average daily enplanements. This data is from external sources, mainly FAA information and forecasts.
2. Allocate the daily originating passengers to their “ground side” trip ends. Passengers on a stopover or transferring from another flight do not have an origin or destination in the local area. This allocation model primarily uses information on households and employment stratified by the four income groups.
3. Estimate the mode used to travel between the airport and the “ground side” trip end. For this study a nested logit model has been used to develop these estimates.

The first step uses information from external sources and, in the application program, is specified by the user. The second step uses information from the normal travel demand models and, in the application program, is specified by including specific information from these models to the program. The third step required a review of air passenger models and the development of a nested logit mode choice model for the MWCOG region. In addition to developing the procedures and models to estimate air passenger travel, a computer program was written using CUBE Voyager to implement all three steps.

8.1 Step One: Estimating Total Air Passengers

The initial step in estimating the air passenger travel for the region is to estimate the average daily air passengers to and from each of the three airports. This estimate consists of estimating (or obtaining) enplanements and estimating the purpose of these enplanements. The best source of annual

enplanements is the FAA. Annual enplanement forecasts through 2030 were taken from the MWCOG report “Washington-Baltimore Regional Air Passenger Origin/Destination Forecast Update,” dated May 2011, which includes the FAA enplanement data. These forecasts are shown in Table 8-2.

Table 8-2: Annual Air Passenger Enplanements

Year	BWI Marshall (BWI)	Washington Dulles (IAD)	Reagan National (DCA)
2000	9,784,850	9,971,832	7,855,323
2001	9,801,066	10,656,740	8,066,071
2002	9,817,281	11,341,648	8,276,819
2003	9,833,497	12,026,557	8,487,567
2004	9,849,712	12,711,465	8,698,315
2005	9,865,928	13,396,373	8,909,063
2006	10,092,085	13,079,679	8,934,359
2007	10,318,242	12,762,986	8,959,655
2008	10,544,399	12,446,292	8,984,952
2009	10,770,556	12,129,599	9,010,248
2010	10,996,713	11,812,905	9,035,544
2011	11,364,688	12,437,565	9,245,702
2012	11,732,663	13,062,224	9,455,860
2013	12,100,638	13,686,884	9,666,018
2014	12,468,613	14,311,543	9,876,176
2015	12,836,588	14,936,203	10,086,334
2016	13,200,242	15,628,793	10,189,235
2017	13,563,897	16,321,382	10,292,136
2018	13,927,551	17,013,972	10,395,037
2019	14,291,206	17,706,561	10,497,938
2020	14,654,860	18,399,151	10,600,839
2021	15,070,264	19,254,334	10,708,988
2022	15,485,668	20,109,518	10,817,137
2023	15,901,071	20,964,701	10,925,287
2024	16,316,475	21,819,885	11,033,436
2025	16,731,879	22,675,068	11,141,585
2026	17,206,428	23,731,536	11,255,252
2027	17,680,977	24,788,003	11,368,919
2028	18,155,526	25,844,471	11,482,586
2029	18,630,075	26,900,938	11,596,253
2030	19,104,624	27,957,406	11,709,920

Source: Metropolitan Washington Council of Governments, “Regional Air Passenger Origin/Destination Forecast Update”, Page 16-17. Forecasts based on FAA TAF forecasts. Includes both local originations and transfers.

Airport survey data was used to determine the percent of transferring passengers from other airports. The 2009 data was taken from the “2009 Washington-Baltimore Regional Air Passenger Survey”, dated September 2010. The transfer percentages for each airport for Years 2005, 2007 and 2009 are shown in Table 8-3. The number of passengers that travel between the airport and a local origin or destination can be computed by multiplying the total number of passengers by (1 - transfer percentage).

Table 8-3: Percent of Air Passengers Connecting from Another Flight

BWI Marshall (BWI)			Washington Dulles (IAD)			Reagan National (DCA)		
2005	2007	2009	2005	2007	2009	2005	2007	2009
13%	16%	15%	49%	47%	40%	10%	11%	9%

The annual enplanements then have to be converted to average weekday estimates. This is done by dividing by 365. It is assumed that for every enplaning passenger there is one deplaning passenger, a not unreasonable assumption, producing the average daily local air passengers to and from each airport shown in Table 8-4. For example, the 2005 BWI values is $9,865,928 * (1 - 0.13) / 365 = 23,516$.

Table 8-4: Average Daily Local Air Passengers

BWI Marshall (BWI)			Washington Dulles (IAD)			Reagan National (DCA)		
2005	2007	2009	2005	2007	2009	2005	2007	2009
23,516	23,746	25,082	18,718	18,533	19,939	21,968	21,847	22,464

The next phase of this step is to estimate the “purpose” of the air passenger. For this model the purpose is defined in two ways: the residence of the air passenger and the purpose of the trip. The air passenger is either a resident of the region or a non-resident of the region. The purpose is either business or non-business/pleasure. Therefore this model has four purposes which are:

- Residents on Business trips (BusRes)
- Non-residents on Business trips (BusNonRes)
- Residents on non-business or pleasure trips (NonBusRes)
- Non-residents on non-business or pleasure trips (NonBusNonRes)

A review of the raw survey data was done in order to estimate the proportion of air passengers for each of the three airports. The results of this estimation process are shown in Table 8-5.

Table 8-5: 2009 Air Passenger Trip Purpose by Airport (Percentage)

BWI Marshall (BWI)				Washington Dulles (IAD)				Reagan National (DCA)			
Business		Non-Business		Business		Non-Business		Business		Non-Business	
Res	Non Res	Res	Non Res	Res	Non Res	Res	Non Res	Res	Non Res	Res	Non Res
11.2%	18.0%	29.1%	41.7%	10.1%	21.6%	17.5%	50.8%	14.1%	29.6%	18.3%	38.1%

Notes:

1. Percentages estimated from raw survey data (airsvy09.xls) from the 2009 Washington-Baltimore Regional Air Passenger Survey, documented in a September 2010 report.
2. Percentages may not total to 100.0% due to rounding.

These parameters need to be forecasted for future years. It is suggested that the FAA and/or the airport authorities would be good sources for the total enplanements. Unless other surveys are taken, the percent transfers and the percent by purpose developed in this study are adequate values for forecasting. The application computer program for this model has defaults for the percent transfers and percent by purpose, the defaults being the values presented in this section, but the user has the option to change these values. The total annual enplanements must be determined from an external source and are read from the FAA forecast data.

8.2 Step Two: Allocating Ground Side Trip Ends

In Step One, the total daily air passengers are estimated. In this step, these total air passengers are allocated to the ground side locations – either the homes of the residents or the offices, hotels, etc. for the non-residents. The following distribution was used in the Baltimore model and is summarized below:

Airport Trip Purpose	Ground-Side Purpose
Resident Business	HBO Productions
Resident Non-Business	HBO Productions
Non-Resident Business	HBW Attractions
Non-Resident Non-Business	HBO Attractions

A review of the survey data and other studies indicated that the most appropriate allocation procedure would be to use the home-based other trip ends by income level. That is, if a zone had two percent of all home-based other trip ends, it would also have two percent of the total air passenger trip ends. Home-based productions consider the size of the household and the income of the traveler, with the number of trips increasing as income and family size increases. For individual households, there can be a wide spread of trips. For example, a high-income household with five persons generates six and a half times as many home-based other trips as does a low-income household with one person. The home-based-other attraction model gives a substantial weight to the retail employment in the zone, but also uses the number of households as a variable. Thus, the home-based other attractions include both employment and households to estimate trip ends. The home-based other productions are used as the zone-level “weights” for the residential air passenger trips, while the home-based other attractions are used as the zone-level “weights” for the non-residential air passenger trips. This procedure, though, is

more detailed when income is considered and when the air passenger purposes are considered. The travel demand models produce trip ends by income level, and wealthier people tend to travel more. Therefore, the home-based other trip ends were “weighted” by the income level in order to match the income distribution from the airport survey. There are eight income ranges in the airport survey, but only four income ranges in the MWCOG Version 2.3 model. In addition, two other adjustments were necessary. First, the airport income surveys were converted from 2009 dollars to 2007 dollars by applying the appropriate CPI. Second, the airport incomes were aggregated to the four MWCOG ranges assuming a uniform income distribution.

The four household annual income groups were (in 2007 dollars):

- Income Group 1: \$0 - \$50,000
- Income Group 2: \$50,000 - \$100,000
- Income Group 3: \$100,000 - \$150,000
- Income Group 4: \$150,000+

In order to match the income distribution for each airport, an income weight was applied to each of the four income categories. In general, the income weight for the lowest income category was less than 0.50; the weight for the second category was approximately 1.00; the third was weighted by an average of 2.00 for all airports; and the highest income group was weighted by 1.15 on average. The income weights and resulting income distribution are summarized for each airport and compared to the survey data in Table 8-6.

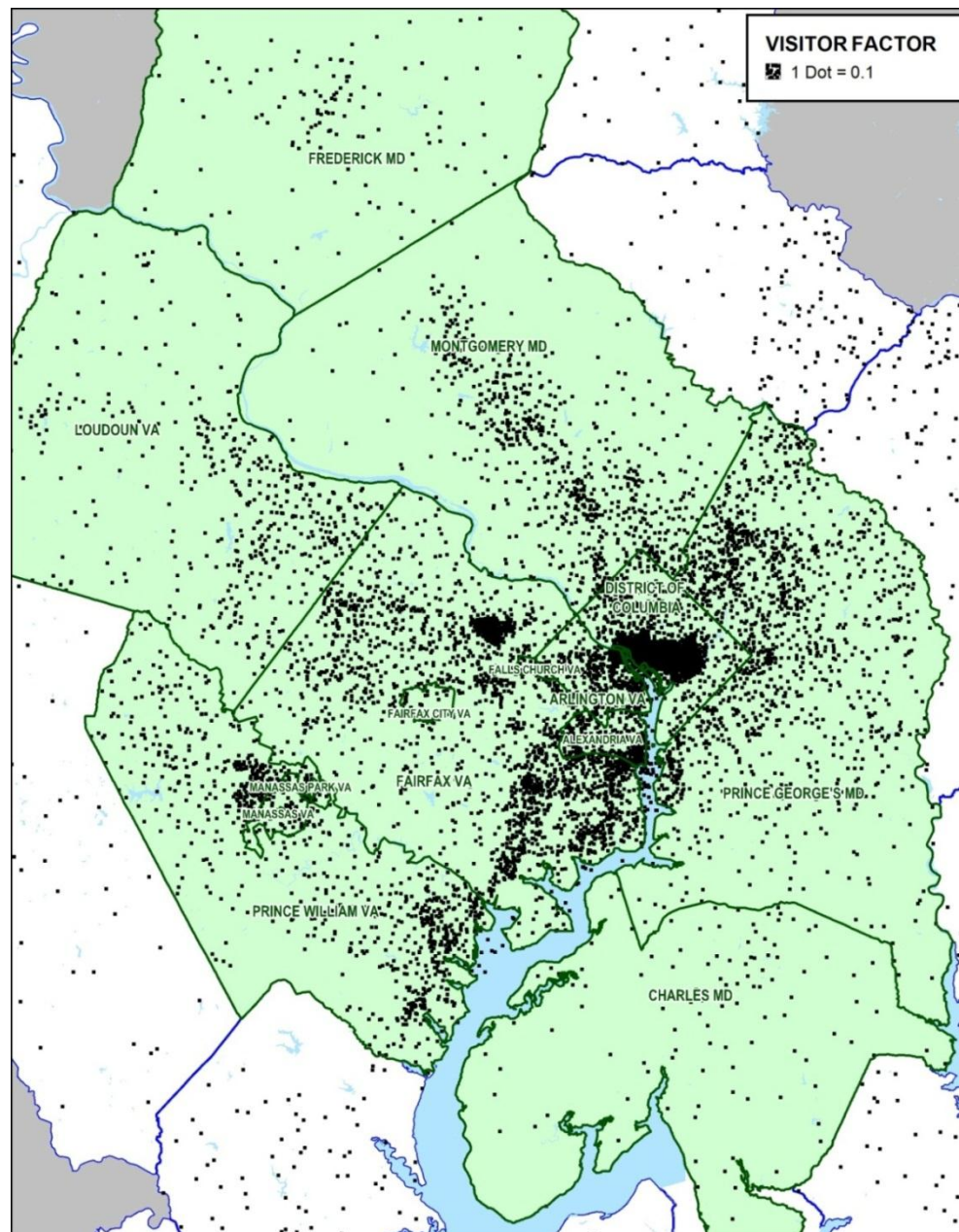
Table 8-6: Income Weights and Income Distribution by Airport

Income Group	BWI Marshall (BWI)			Washington Dulles (IAD)			Reagan National (DCA)		
	Input Weight Factor	Distribution		Input Weight Factor	Distribution		Input Weight Factor	Distribution	
		Model Output	Survey Target		Model Output	Survey Target		Model Output	Survey Target
1	0.55	7%	7%	0.40	5%	5%	0.30	4%	4%
2	1.05	26%	26%	1.10	26%	26%	0.95	21%	21%
3	1.95	41%	41%	1.95	39%	39%	2.10	40%	39%
4	1.00	26%	26%	1.15	31%	30%	1.30	36%	36%
Total		100%	100%		100%	100%		100%	100%

There are areas within the MWCOG region that generate a larger percentage of visitor trips compared with the number of home-based other and/or home-based work trips. To better account for these areas, the “Visitor Factor” was introduced. The “Visitor Factor” is applied to each traffic analysis zone (TAZ) to better match the airport trip distribution. In Phase 1, this airport trip distribution was compared at a county level. In Phase 2, it may be desirable to compare the airport trip distribution at

the Airport Analysis Zone (AAZ) level. The “Visitor Factor” is shown graphically in Figure 8-1²⁰ The “Visitor Factor” approach is adequate for both Baltimore and New Orleans, since they each have only a single major airport. However, this approach is insufficient to differentiate between the different markets served by the three airports in the MWCOG region, since it produces the same “attractiveness” distribution for all three airports.

Figure 8-1: Visitor Factor by TAZ



²⁰ The area covered by the 2009 air passenger survey is larger than the modeled area. Specifically, it includes three jurisdictions that are not in the TPB modeled area: Baltimore City, Baltimore County, and Harford County, Maryland. Consequently, in Figure 8-1 and other subsequent figures, these three jurisdictions have been included with Anne Arundel County. The green area on the map represents the TPB member jurisdictions.

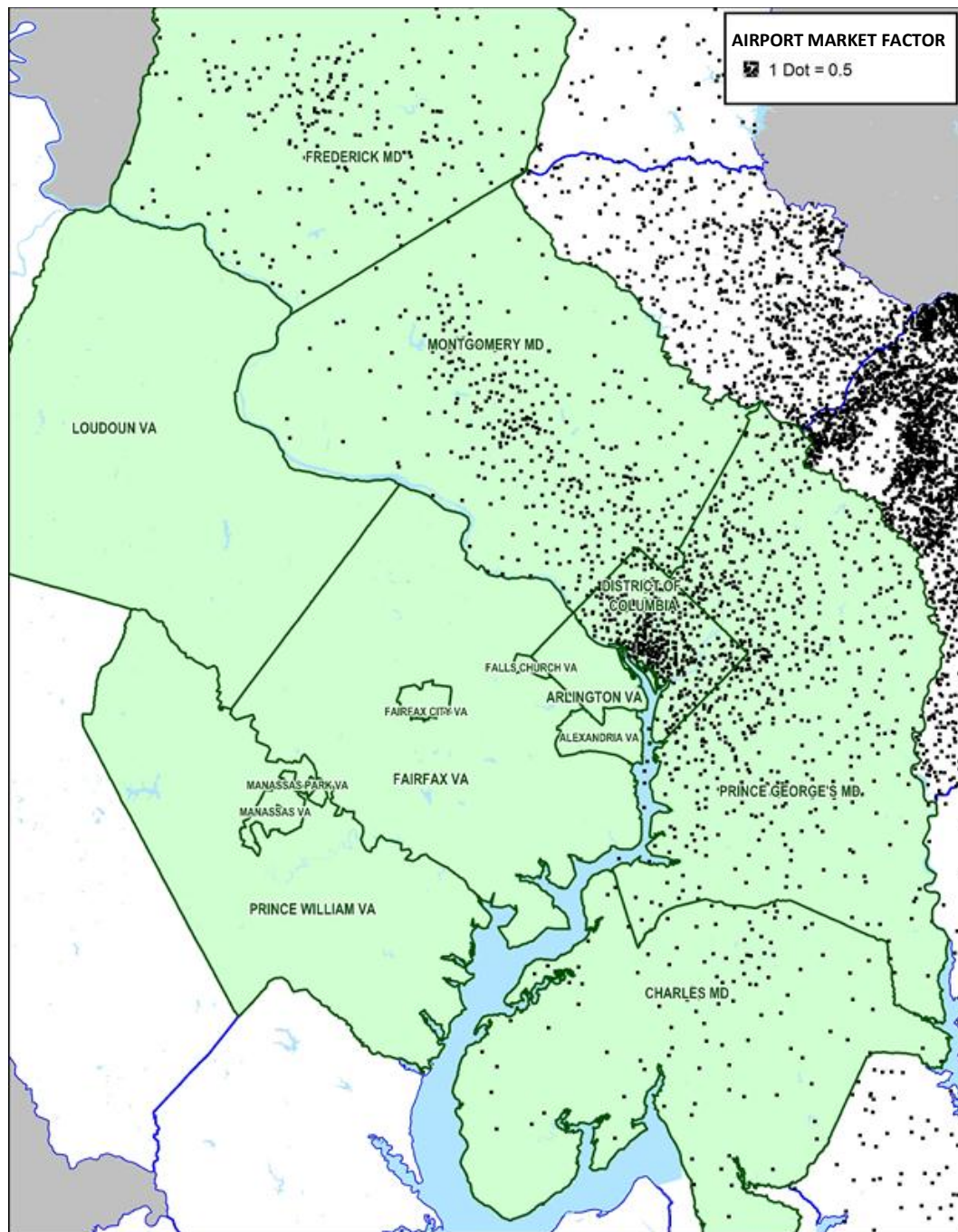
To correct for this limitation, a second “Airport Market Factor” is introduced which is developed uniquely for each of the three airports. Consider three zones of similar demographics: the first zone is in Loudoun County, the second zone is in Arlington County and the third zone is in Anne Arundel County. The “Visitor Factor” would be the same for all three zones however the “Airport Market Factor” would differ greatly among the three airports. For Dulles Airport, the Loudoun County factor would be much greater than 1.0 while the Arlington County and Anne Arundel County factors would be much less than 1.0. For National Airport, the Arlington County factor would be much greater than 1.0 while the Loudoun County and Anne Arundel County factors would be much less than 1.0. For BWI Airport, the Anne Arundel County factor would be much greater than 1.0 while the Loudoun County and Arlington County factors would be much less than 1.0. The Airport Market Factors for each airport are listed by County in Table 8-7 and are shown graphically in Figure 8-2, Figure 8-3, and Figure 8-4. In Phase 1, the “Airport Market Factor” is applied on a county basis. In Phase 2, it may be desirable to apply it on an AAZ basis. The resulting airport trip distributions for each of three airports are shown in Table 8-8. As seen in the table, for each airport the county distributions closely match the survey data distributions. Survey totals represent the survey sample.

Table 8-7: Airport Market Factor by Airport

County / State	BWI Marshall (BWI)	Washington Dulles (IAD)	Reagan National (DCA)
Washington, DC	0.600	1.900	4.800
Montgomery, MD	0.800	2.100	1.600
Prince George’s, MD	0.800	0.005	0.400
Arlington, VA	0.200	2.200	7.000
Alexandria, VA	0.100	1.000	4.000
Fairfax County, VA	0.001	3.800	0.800
Loudoun, VA	0.001	15.000	0.050
Prince William, VA	0.002	2.750	0.200
Frederick, MD	1.000	0.010	0.010
Howard, MD	6.000	0.010	0.010
Anne Arundel, MD (1)	33.000	0.010	0.010
Charles, MD	0.500	0.005	0.200
Carroll, MD	1.000	0.001	0.200
Calvert, MD	1.000	0.001	0.200
St. Mary’s, MD	1.000	0.010	0.200
King George, VA	0.010	2.000	0.200
Fredericksburg, VA	0.010	1.000	0.200
Stafford, VA	0.001	1.000	0.200
Spotsylvania, VA	0.001	1.000	0.200
Fauquier, VA	0.050	3.000	0.200
Clarke, VA	0.050	3.000	0.200
Jefferson, VA	0.1000	3.000	0.200

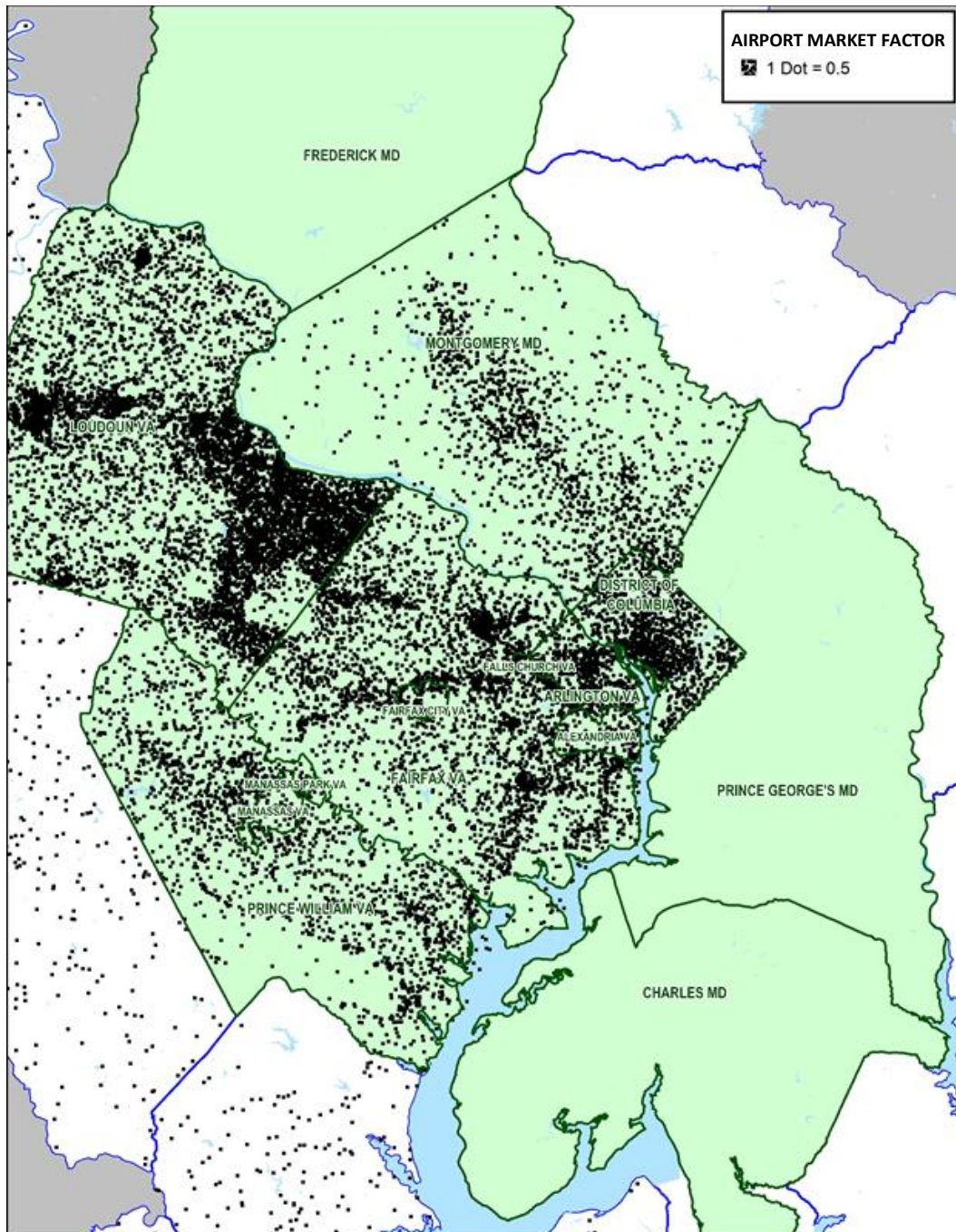
Note: (1) Anne Arundel includes Baltimore City, Baltimore County and Harford County

Figure 8-2: BWI Marshall (BWI) Airport Market Factor



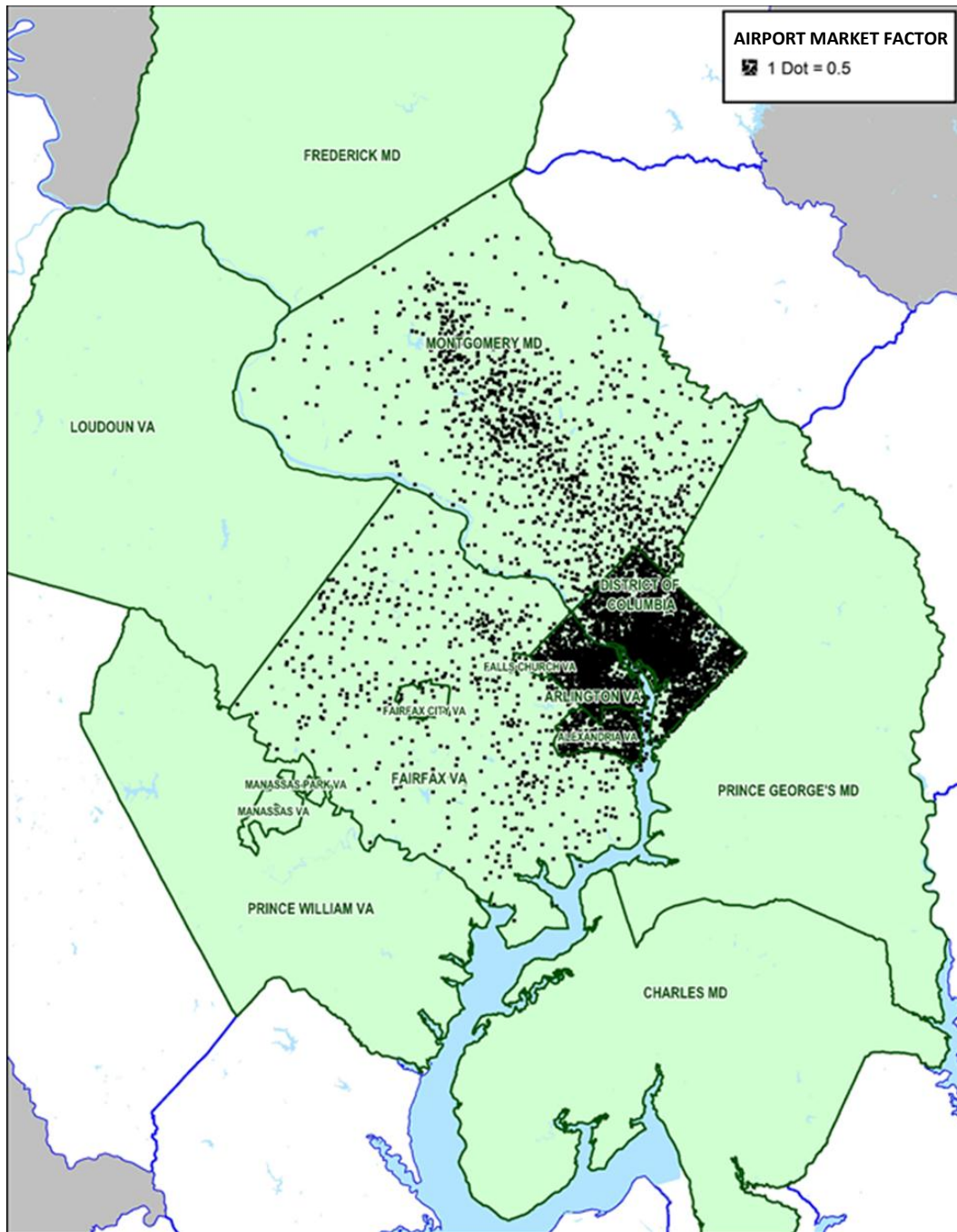
Baltimore City, Baltimore County, and Harford County, MD are shown as generated from Anne Arundel County, MD

Figure 8-3: Washington Dulles (IAD) Airport Market Factor



Baltimore City, Baltimore County, and Harford County, MD are shown as generated from Anne Arundel County, MD

Figure 8-4: Reagan National (DCA) Airport Market Factor



Baltimore City, Baltimore County, and Harford County, MD are shown as generated from Anne Arundel County, MD

Table 8-8: 2009 Estimated and Observed Airport Trip Distribution by Airport

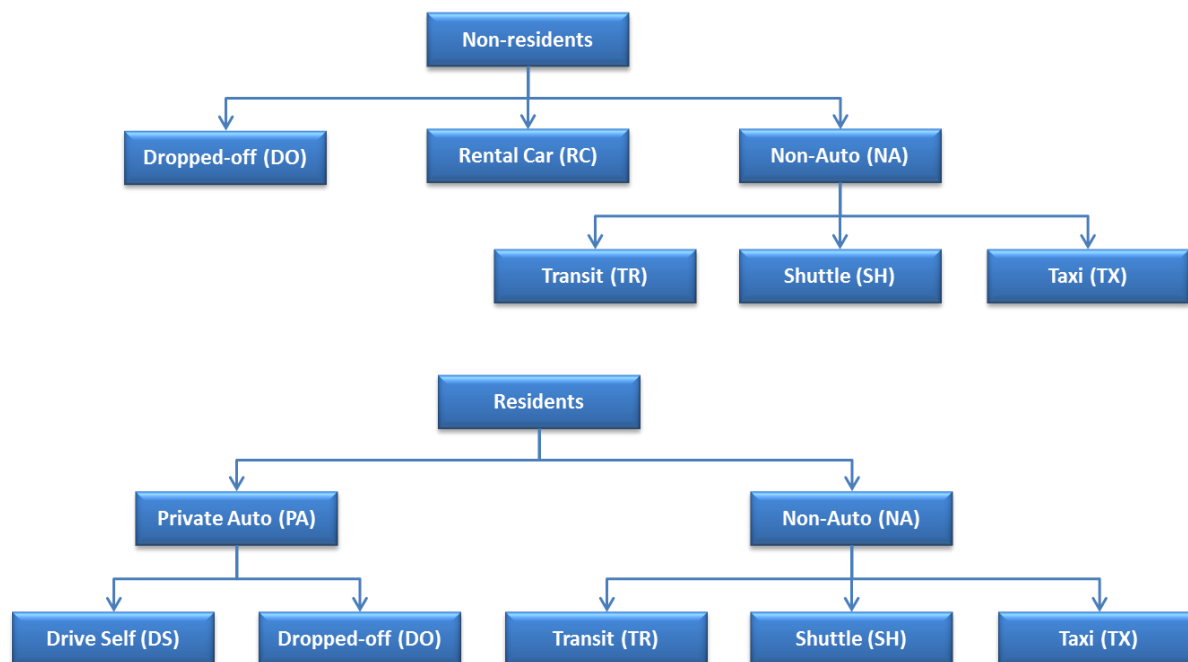
County/State	BWI Marshall (BWI)				Washington Dulles (IAD)				Reagan National (DCA)			
	Model Output		Survey Target		Model Output		Survey Target		Model Output		Survey Target	
Washington, DC	5,625	11.2%	3,438	11.3%	7,523	18.9%	4,548	19.6%	18,280	40.7%	12,850	42.8%
Montgomery, MD	5,170	10.3%	3,256	10.7%	4,544	11.4%	2,762	11.9%	4,587	10.2%	3,191	10.6%
Prince George's, MD	3,553	7.1%	2,280	7.5%	1,327	3.3%	347	1.5%	2,040	4.5%	1,140	3.8%
Arlington, VA	952	1.9%	724	2.4%	2,173	5.5%	1,413	6.1%	6,337	14.1%	4,763	15.9%
Alexandria, VA	467	0.9%	292	1.0%	746	1.9%	439	1.9%	2,492	5.5%	1,843	6.1%
Fairfax, VA	3,817	7.6%	1,331	4.4%	12,547	31.5%	7,630	32.8%	5,090	11.3%	3,867	12.9%
Loudoun, VA	868	1.7%	339	1.1%	4,583	11.5%	2,752	11.8%	658	1.5%	184	0.6%
Prince William, VA	1,345	2.7%	269	0.9%	2,764	6.9%	1,617	7.0%	1,113	2.5%	681	2.3%
Frederick, MD	1,052	2.1%	694	2.3%	405	1.0%	255	1.1%	528	1.2%	123	0.4%
Howard, MD	3,531	7.0%	2,121	7.0%	505	1.3%	151	0.6%	677	1.5%	38	0.1%
Anne Arundel, MD (1)	20,783	41.4%	5,390	17.8%	925	2.3%	303	1.3%	1,211	2.7%	166	0.6%
Charles, MD	492	1.0%	180	0.6%	236	0.6%	97	0.4%	322	0.7%	187	0.6%
Carroll, MD	618	1.2%	523	1.7%	265	0.7%	40	0.2%	359	0.8%	2	0.0%
Calvert, MD	395	0.8%	279	0.9%	171	0.4%	-	0.0%	235	0.5%	144	0.5%
St. Mary's, MD	388	0.8%	218	0.7%	162	0.4%	41	0.2%	218	0.5%	244	0.8%
King George, VA	64	0.1%	37	0.1%	51	0.1%	82	0.4%	45	0.1%	7	0.0%
Fredericksburg, VA	50	0.1%	31	0.1%	161	0.4%	18	0.1%	58	0.1%	87	0.3%
Stafford, VA	354	0.7%	88	0.3%	246	0.6%	165	0.7%	263	0.6%	166	0.6%
Spotsylvania, VA	306	0.6%	23	0.1%	205	0.5%	32	0.1%	220	0.5%	116	0.4%
Fauquier, VA	163	0.3%	48	0.2%	163	0.4%	177	0.8%	118	0.3%	45	0.2%
Clarke, VA	40	0.1%	28	0.1%	41	0.1%	20	0.1%	28	0.1%	5	0.0%
Jefferson, WV	82	0.2%	39	0.1%	82	0.2%	82	0.4%	55	0.1%	-	0.0%
Baltimore City, MD	N/A	N/A	4,244	14.0%	N/A	N/A	166	0.7%	N/A	N/A	122	0.4%
Baltimore County, MD	N/A	N/A	3,424	11.3%	N/A	N/A	55	0.2%	N/A	N/A	21	0.1%
Harford, MD	N/A	N/A	1,004	3.3%	N/A	N/A	43	0.2%	N/A	N/A	-	0.0%
Total	50,111	99.9%	30,300	100.0%	39,826	99.9%	23,235	100.0%	44,934	99.9%	29,992	100.0%

Notes: 1. Model Output for Anne Arundel includes Anne Arundel, Baltimore City, Baltimore County and Harford County
 2. Totals may not add to 100.0% due to rounding

8.3 Step Three: Mode Choice Model

A relatively sophisticated approach to mode choice modeling was taken in order to give the air passengers a wide array of modal options to access to the airport. A nested logit model is used, with different structures and modal options for Resident and Non-Resident air passengers, as shown in Figure 8-5:.

Figure 8-5: Air Passenger Model Structure



Non-residents are assumed to have three primary modal choices: being dropped off (or picked up) by someone in a private car, using a rental car, or using one of the non-private-auto modes. Within the latter, there are three choices: public transit (regularly scheduled service), shuttle services, including private and hotel limos, and taxi. Residents of the MWCOG region have a different set of choices, involving one fewer mode at the top level (they are presumed not to be car renters). Within the Private Auto mode, they are either Dropped Off, or they Drive Themselves. Within the “Non-Auto” mode, they have the same Transit, Taxi and Shuttle options as Non-Residents. These choice sets are based on other air passenger models and data, including the Washington, D.C. model.

It has been widely observed that decision makers exhibit preferences for alternatives which cannot be explained by the observed attributes of those alternatives. These preferences are described as alternative-specific preference or bias; they measure the average preference of individuals with different characteristics for an alternative relative to a “reference” alternative. (Koppelman and Bhat, “A Self Instructing Course in Mode Choice Modeling: Multinomial and Nested Logit Models, January 2006). In the “TPB nested logit mode choice model: 2002, 2005 and 2030 results” presentation by Mark Moran, dated March 21, 2008, these preferences, or bias, are referred to as nesting constants. As noted in the TPB mode choice model presentation, calibration of these bias constants can frequently be problematic. The revised calibration approach used by TPB was to develop a unique set of bias

constants for each travel mode for each of seven superdistricts (DC core, VA core, DC urban, MD urban, VA urban, MD suburban, and VA suburban). A similar decision was made for the Airport Model: To develop a unique set of bias constants for each travel mode for each of the three airports (BWI, IAD and DCA). For the TPB mode choice model, an automated calibration routine was used. For the airport choice model, there was a desire to match as closely as possible the bias coefficients (nesting constants) previously estimated as part of the New Orleans and Baltimore models. A manual process was therefore employed that began with the New Orleans / Baltimore model bias coefficients (nesting constants). After each model run for each of the three airports, the mode share outputs were compared with the results of the 2009 Washington-Baltimore Regional Air Passenger Survey. Each of the bias coefficients (nesting constants) were then incrementally adjusted up or down and the Airport Model run again. This process was repeated until the model closely estimated the results of the 2009 survey. The TPB presentation noted that large bias coefficients (nesting constants) “could be symptomatic of upstream problems” in the model chain and “could lead to unreasonable forecasts”. These two points, particularly with regard to the Transit mode, are beyond the scope of the Phase 1 effort but will be further investigated in Phase 2.

The model is a nested logit, which is similar to the more commonly known multinomial logit, whose formula is shown below:

$$p(m) = e^{U(m)} / \sum e^{U(m)}$$

where:

$p(m)$ = probability of choosing mode m

$U(m)$ = disutility of mode m

The disutility equations for each mode and each model are shown in Table 8-9. The lower nest values are calculated first. For example, in the non-resident model, the disutilities are calculated for Transit, Shuttle, and Taxi. The exponentials of these three disutilities are taken and then summed. The natural log of that sum is the “log sum” term that is used in the “top level” nest to compare the Non-Auto mode with Rental Car and Dropped Off. With this structure, trips that “leave” a mode are more likely to go to other modes in the same nest, rather than modes on a different “level”.

Table 8-9: Air Passenger Mode Choice Disutility Equations

Business, Residents

$$U(\text{Drive Self}) = (-0.071 * \text{HWYTIME} - 0.00277 * (\text{HWYCOST} + \text{PCOST}) + \text{biasDS})/0.3$$

$$U(\text{Dropped Off}) = (-0.071 * \text{HWYTIME} - 0.00277 * \text{HWYCOST})/0.3$$

$$U(\text{Transit}) = (-0.093 * \text{WALK} - 0.107 * \text{WAIT} - 0.00277 * \text{TRFARE} - 0.053 * \text{TRRUN} + \text{biasTR})/0.3$$

$$U(\text{Shuttle}) = (-0.071 * \text{SHRUN} - 0.00277 * \text{SHFARE} + \text{biasSH})/0.3$$

$$U(\text{Taxi}) = (-0.071 * \text{HWYTIME} - 0.00277 * \text{TXFARE} + \text{biasTX})/0.3$$

$$\text{NonAuto logsum} = \ln(e^{U(\text{Transit})} + e^{U(\text{Taxi})} + e^{U(\text{Shuttle})})$$

$$\text{Auto logsum} = \ln(e^{U(\text{Dropped Off})} + e^{U(\text{Drive Self})})$$

$$U(\text{Non-Auto}) = 0.3 * \text{NonAuto logsum} + \text{biasNA}$$

$$U(\text{Private Auto}) = 0.3 * \text{Auto logsum}$$

Business, Non-residents

$$U(\text{Dropped Off}) = -0.068 * \text{HWYTIME} - 0.00256 * \text{HWYCOST}$$

$$U(\text{Rental Car}) = \text{biasRC}$$

$$U(\text{Transit}) = (-0.089 * \text{WALK} - 0.096 * \text{WAIT} - 0.00256 * \text{TRFARE} - 0.050 * \text{TRRUN} + \text{biasTR})/0.3$$

$$U(\text{Shuttle}) = (-0.089 * \text{SHRUN} - 0.00256 * \text{SHFARE} + \text{biasSH})/0.3$$

$$U(\text{Taxi}) = (-0.068 * \text{HWYTIME} - 0.00256 * \text{TXFARE} + \text{biasTX})/0.3$$

$$\text{NonAuto logsum} = \ln(e^{U(\text{Transit})} + e^{U(\text{Taxi})} + e^{U(\text{Shuttle})})$$

$$U(\text{Non-Auto}) = 0.3 * \text{NonAuto logsum} + \text{biasNA}$$

Non- Business Residents

$$U(\text{Drive Self}) = (-0.044 * \text{HWYTIME} - 1.04 * (\text{HWYCOST} + \text{PCOST})/\text{FINC} + \text{biasDS})/0.3$$

$$U(\text{Dropped Off}) = (-0.044 * \text{HWYTIME} - 1.04 * \text{HWYCOST}/\text{FINC})/0.3$$

$$U(\text{Transit}) = (-0.051 * \text{WALK} - 0.077 * \text{WAIT} - 1.04 * \text{TRFARE}/\text{FINC} - 0.031 * \text{TRRUN} + \text{biasTR})/0.3$$

$$U(\text{Shuttle}) = (-0.044 * \text{SHRUN} - 1.04 * \text{SHFARE} + \text{biasSH})/0.3$$

$$U(\text{Taxi}) = (-0.044 * \text{HWYTIME} - 1.04 * \text{TXFARE}/\text{FINC} + \text{biasTX})/0.3$$

$$\text{NonAuto logsum} = \ln(e^{U(\text{Transit})} + e^{U(\text{Taxi})} + e^{U(\text{Shuttle})})$$

$$\text{Auto logsum} = \ln(e^{U(\text{Dropped Off})} + e^{U(\text{Drive Self})})$$

$$U(\text{Non-Auto}) = 0.3 * \text{NonAuto logsum} + \text{biasNA}$$

$$U(\text{Private Auto}) = 0.3 * \text{Auto logsum}$$

Non-business, Non-residents

$$U(\text{Dropped Off}) = -0.039 * \text{HWYTIME} - 0.973 * \text{HWYCOST}/\text{FINC}$$

$$U(\text{Rental Car}) = \text{biasRC}$$

$$U(\text{Transit}) = (-0.045 * \text{WALK} - 0.071 * \text{WAIT} - 0.973 * \text{TRFARE}/\text{FINC} - 0.029 * \text{TRRUN} + \text{BiasTR})/0.3$$

$$U(\text{Shuttle}) = (-0.045 * \text{SHRUN} - 0.973 * \text{SHFARE}/\text{FINC} + \text{biasSH})/0.3$$

$$U(\text{Taxi}) = (-0.039 * \text{HWYTIME} - 0.973 * \text{TXFARE}/\text{FINC} + \text{biasTX})/0.3$$

$$\text{NonAuto logsum} = \ln(e^{U(\text{Transit})} + e^{U(\text{Taxi})} + e^{U(\text{Shuttle})})$$

$$U(\text{Non-Auto}) = 0.3 * \text{NonAuto logsum} + \text{biasNA}$$

Where:

HWYTIME = off-peak travel time from the highway network (minutes)
 HWYCOST = off-peak distance from the highway network * 8.74 cents/mile
 PCOST = half the daily long-term parking cost at BWI (cents), multiplied by the average duration of the trip in days.
 Average trip length based on survey data BWI - \$8.00 per day; Business – 3.9 days, Non-Business = 5.2 days
 IAD – \$10.00 per day; Business – 4.0 days, Non-Business = 7.2 days
 DCA - \$12.00 per day; Business – 3.9 days, Non-Business = 5.6 days
 WALK = access + egress time from the AM peak transit network (minutes)
 WAIT = initial wait + transfer wait time from the AM peak transit network (minutes)
 TRRUN = total in-vehicle time from the AM peak transit network (minutes)
 TRFARE = transit fare (cents)
 TXFARE = taxi fare (cents); based on the standard distance fare for each airport
 BWI - \$2.90 for first 1/10 mile plus \$2.50 per mile
 IAD – \$3.50 for first 1/4 mile + \$2.00 per mile
 DCA - \$2.75 for first 1/6 mile + \$2.10 per mile

SHRUN = shuttle in-vehicle time (minutes); computed as highway time * 1.50 circuitry factor
 SHFARE = fixed shuttle fare based on www.supershuttle.com. Note: For Phase 1, use one representative zip code for each county to determine round-trip fare to/from each airport. For Phase 2, it may be desirable to use one representative zip code for each analysis zone.
 FINC = power function of average income for each income group;
 $FINC = (\text{income}/1000)^{1.5}$ and the average income by group is: group 1 = \$25,000, group 2 = \$75,000, group 3 = \$125,000, group 4 = \$175,000 (2007 dollars)
 (Note: auto, taxi, and shuttle costs are not divided by average vehicle occupancy.)
 biasMM = bias coefficients by mode and purpose, as follows:

The system coefficients (on time and cost) were obtained from other air passenger models, mainly Washington and San Francisco’s models. As discussed earlier, a unique set of modal bias coefficients (nesting constants) was estimated for each airport using the Baltimore / New Orleans data as an initial estimate and using the 2009 air passenger survey data set as target values. The bias coefficients (nesting constants) for each of the three airports are shown in Table 8-10.

Table 8-10: Bias Coefficients and Nesting Constants by Airport, Mode, and Trip Market

BWI Marshall (BWI)		Trip Market		
Mode (MM)	Bus.,Res.	Bus.,Non-Res.	Non-Bus., Res.	Non-Bus. Non-Res.
Drive Self (DS)	0.27	N/A	0.22	N/A
Rental Car (RC)	N/A	-4.90	N/A	-5.8
Transit (TR)	7.55	3.21	8.00	5.02
Shuttle (SH)	6.42	4.51	5.19	2.58
Taxi (TX)	-0.88	2.61	-2.11	0.68
Non-Auto Nest (NA)	-1.31	2.25	-1.81	0.33
Washington Dulles (IAD)		Trip Market		
Mode (MM)	Bus.,Res.	Bus.,Non-Res.	Non-Bus., Res.	Non-Bus. Non-Res.
Drive Self (DS)	0.77	N/A	0.72	N/A
Rental Car (RC)	N/A	-9.25	N/A	-10.15
Transit (TR)	5.15	4.61	5.60	6.42
Shuttle (SH)	3.52	8.11	2.29	6.18
Taxi (TX)	3.52	-0.39	2.29	-2.32
Non-Auto Nest (NA)	0.21	3.47	-0.29	1.55
Reagan National (DCA)		Trip Market		
Mode (MM)	Bus.,Res.	Bus.,Non-Res.	Non-Bus., Res.	Non-Bus. Non-Res.
Drive Self (DS)	2.17	N/A	2.12	N/A
Rental Car (RC)	N/A	-5.90	N/A	-6.80
Transit (TR)	5.15	4.61	5.80	6.42
Shuttle (SH)	2.02	6.11	0.79	4.18
Taxi (TX)	5.22	5.01	3.99	3.08
Non-Auto Nest (NA)	1.36	4.20	0.86	2.28

Table 8-11 shows these target mode shares and the mode shares estimated by the model, using the Baltimore / New Orleans system coefficients as shown in Table 8-9. The table indicates that the model replicates the target mode shares with reasonable accuracy.

The mode choice model calculates trips in the Drive Self and Dropped Off private auto modes. The model does not do a separate calculation of auto occupancy, but uses user-entered average occupancies as follows: Drive Self = 1.0, Dropped Off = 1.1, Rental Car = 1.1, Taxi = 1.1, Shuttle = 5.0. In the case of Dropped Off trips, it is assumed that each air passenger being dropped off generates two vehicle trips – one going and one coming.

Table 8-11: Observed and Estimated Air Passenger Trips by Mode

BWI Marshall (BWI)		Residents				Non-Residents				Total			
Mode (MM)	Model Output		Survey Target		Model Output		Survey Target		Model Output		Survey Target		
Drive Self/Drop Off (DS/DO)	17835	88.3%	2547	86.7%	12498	41.8%	2001	41.6%	30332	60.5%	4548	58.7%	
Rental Car (RC)	0	0.0%	43	1.5%	8155	27.2%	1321	27.5%	8155	16.3%	1364	17.6%	
Transit (TR)	608	3.0%	92	3.1%	1295	4.3%	214	4.4%	1903	3.8%	306	3.9%	
Shuttle (SH)	1167	5.8%	190	6.5%	6048	20.2%	971	20.2%	7216	14.4%	1161	15.0%	
Taxi (TX)	591	2.9%	65	2.2%	1937	6.5%	305	6.3%	2528	5.0%	370	4.8%	
Total	20201	100.0%	2937	100.0%	29932	100.0%	4812	100.0%	50133	100.0%	7749	100.0%	
Washington Dulles (IAD)		Residents				Non-Residents				Total			
Mode (MM)	Model Output		Survey Target		Model Output		Survey Target		Model Output		Survey Target		
Drive Self/Drop Off (DS/DO)	15437	76.4%	1774	75.7%	9409	39.5%	1170	39.8%	24846	56.4%	2944	55.7%	
Rental Car (RC)	0	0.0%	15	0.6%	6479	27.2%	796	27.1%	6479	14.7%	811	15.4%	
Transit (TR)	0	0.0%	17	0.7%	0	0.0%	43	1.5%	0	0.0%	60	1.1%	
Shuttle (SH)	1540	7.6%	156	6.7%	4719	19.8%	574	19.5%	6259	14.2%	730	13.8%	
Taxi (TX)	3222	16.0%	380	16.2%	3212	13.5%	358	12.2%	6435	14.6%	738	14.0%	
Total	20199	100.0%	2342	100.0%	23819	100.0%	2941	100.0%	44019	100.0%	5283	100.0%	
Reagan National (DCA)		Residents				Non-Residents				Total			
Mode (MM)	Model Output		Survey Target		Model Output		Survey Target		Survey Target		Model Output		
Drive Self/Drop Off (DS/DO)	10670	52.8%	1353	53.1%	4373	18.4%	861	19.3%	15043	34.2%	2214	31.6%	
Rental Car (RC)	0	0.0%	15	0.6%	3306	13.9%	581	13.1%	3306	7.5%	596	8.5%	
Transit (TR)	3461	17.1%	445	17.5%	3842	16.1%	725	16.3%	7303	16.6%	1170	16.7%	
Shuttle (SH)	919	4.6%	80	3.1%	3834	16.1%	729	16.4%	4753	10.8%	809	11.6%	
Taxi (TX)	5145	25.5%	653	25.6%	8465	35.5%	1555	34.9%	13610	30.9%	2208	31.6%	
Total	20195	100.0%	2546	100.0%	23820	100.0%	4451	100.0%	44015	100.0%	6997	100.0%	

Note: Transit includes Metrorail (DCA), Light Rail (BWI) and Commuter Rail.
 Shuttle includes airport limo / van / bus services; hotel / motel courtesy vehicles; and "other" modes as reported in the survey.

8.4 Time of Day

The airport model contains a procedure to “split” vehicle and transit trips into four time periods for purposes of assignment: AM Peak (5 – 10 AM), Midday (10 AM – 2 PM), PM Peak (2 – 6:30 PM) and Night (6:30 PM – 12 AM). The temporal distribution for airport trips is much different than work or other purposes from the regional model. Rather than develop temporal factors based on HBW or HBO trips from the model, the flight distribution from each airport was obtained from the survey and used instead. These distributions are applied to all four trip markets: business-resident, business-nonresident, nonbusiness-resident, and nonbusiness-nonresident and are shown in Table 8-12 for each of the three airports. After the time-of-day calculations, the model application program outputs two trip table matrix files in binary format. The first file has four tables containing vehicle trips by time period, in O/D format. The second file has four tables containing transit trips by time period, in O/D format. The relationship between the survey time period and the MWCOG regional model time period is also shown in Table 8-12.

Table 8-12: Time of Day Percentages for the Air Passenger Model

Survey Time Period	BWI	DCA	IAD	All Airports	MWCOG Time Period
AM Peak (5 – 10 AM)	26.9%	32.2%	23.4%	27.2%	AM Peak (6 -9 AM)
Midday (10 AM – 2 PM)	23.3%	22.1%	21.7%	22.4%	Midday (9 AM – 3 PM)
PM Peak (2 – 6:30 PM)	30.4%	27.4%	32.0%	30.0%	PM Peak (3 – 7 PM)
Night (6:30 PM – 12 AM)	19.4%	18.3%	23.0%	20.3%	Night (7 PM – 6 AM)
Total	100.0%	100.0%	100.0%	100.0%	

Source: 2009 Washington-Baltimore Regional Air Passenger Survey, September 2010, page 4.

8.5 Implementing the Model

A Cube Voyager script written originally by BMC staff was extensively modified for MWCOG to accommodate the increased complexity of the airport mode choice process in the region. The modifications enable the user to run the script for any of the three Washington area airports (BWI, IAD, DCA) individually. To run the model for a specific airport, the user must edit three items within the script: Airport Code, the Airport TAZ, and the Column number. An excerpt from the script is shown below for running BWI Marshall with the user input text bolded. The script would then be run again to generate outputs for the other airports.

```
AIRPORT= 'BWI'
AIRPORTBWI = 3036 ; BWI- From MWCOG Model
AIRPORTIAD = 2388 ; IAD- From MWCOG Model
AIRPORTDCA = 1498 ; DCA- From MWCOG Model
AIRPORTTAZ=3036
COLUMN = 2 ; BWI=2, IAD=3, DCA=4
```

The user must also select several additional files from the MWCOG regional model run. The HBW and HBO person trip tables are used to distribute total airport demand among the different TAZs. The highway and transit skim files are used to obtain travel distance, travel time and fare data for travel

between the airport and each of the TAZs. The lines within the script that need to be modified are shown below.

```
;HBO I/I person trip table (income 1, 2, 3, 4)
HBOIITT = 'c:\data\jobs\MWCOG\Model\HBO I4 .ptt'
;HBW I/I person trip table (income 1, 2, 3, 4)
HBWIITT = 'c:\data\jobs\MWCOG\Model\HBW I4 .ptt'

; AM Peak Period Highway Skim
HWYSKM='c:\data\jobs\MWCOG\Model\SOVMI4MD.skm'; AM Peak Period Transit Walk to Bus Skim
BUSSSKM='c:\data\jobs\MWCOG\Model\TRNAM_AB.SKM'; AM Peak Period Transit Walk to Metrorail Skim
MRSKM='c:\data\jobs\MWCOG\Model\TRNAM_MR.SKM'; AM Peak Period Transit Walk to Commuter Skim
CRSKM='c:\data\jobs\MWCOG\Model\TRNAM_CR.SKM'
```

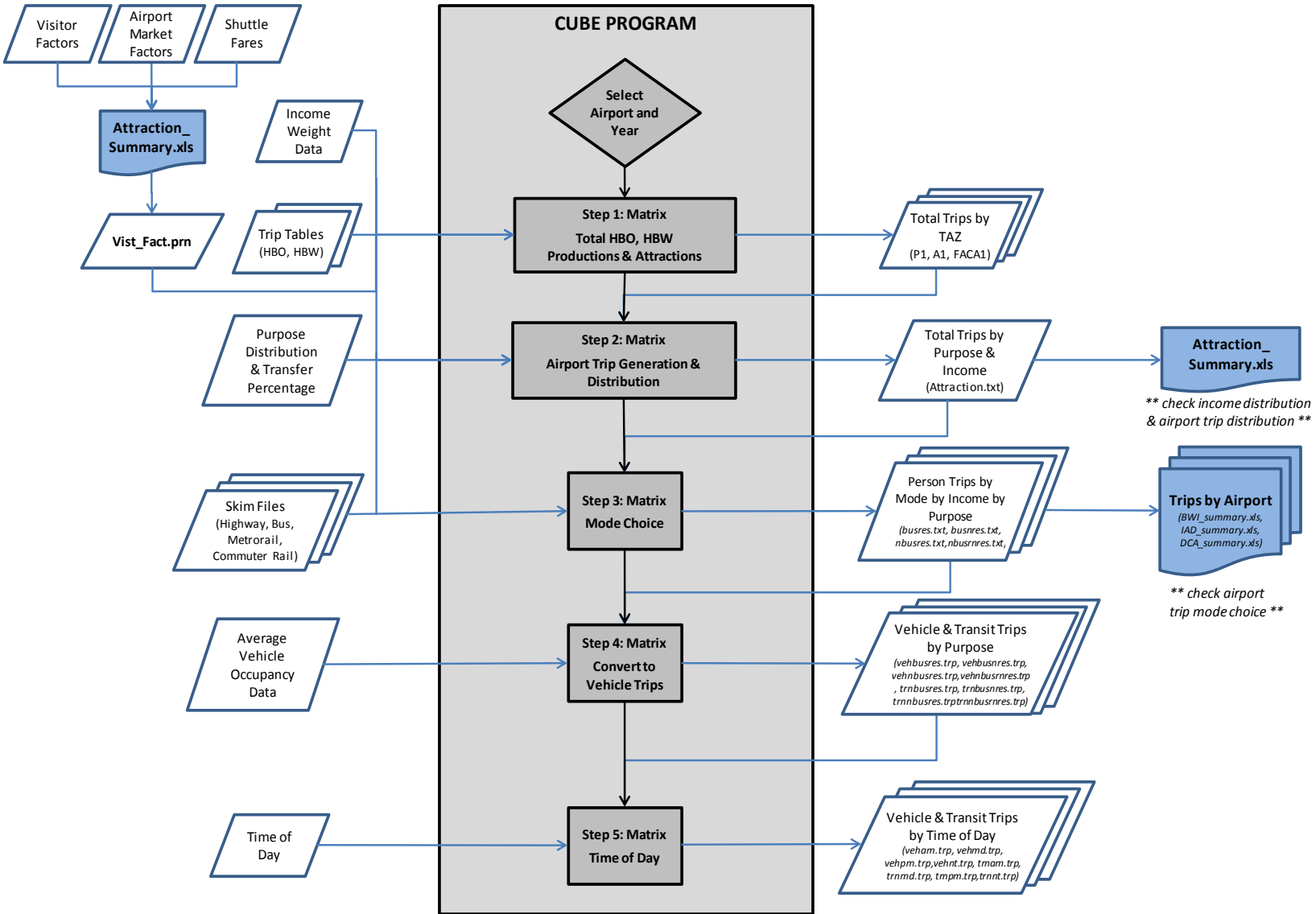
The final output trip tables have the following form where @AIRPORT@ refers to the three letter airport code input by the user.

```
veham@AIRPORT@.trp, mo=1, name=AMVehAirPass ;am veh trip table
vehmd@AIRPORT@.trp, mo=2, name=MDVehAirPass ;mid day veh trip table
vehpm@AIRPORT@.trp, mo=3, name=PMVehAirPass ;pm veh trip table
vehnt@AIRPORT@.trp, mo=4, name=NTVehAirPass ;over night veh trip table

trnam@AIRPORT@.trp, mo=5, name=AMTrnAirPass ;am veh trip table
trnmd@AIRPORT@.trp, mo=6, name=MDTrnAirPass ;mid day veh trip table
trnpm@AIRPORT@.trp, mo=7, name=PMTrnAirPass ;pm veh trip table
trnnt@AIRPORT@.trp, mo=8, name=NTTrnAirPass ;over night veh trip table
```

A flowchart illustrating the modeling process is shown in Figure 8-6:. The flow chart shows the main Cube programming process and also includes all of the input and output data files.

Figure 8-6: Air Passenger Model Flowchart



8.6 Sources and References

Atlanta Regional Commission, “Plan 2040 RTP – Appendix E Model Documentation” July 2011.

http://documents.atlantaregional.com/plan2040/docs/tp_PLAN2040RTP_appendE_0702711.pdf

Harvey, Greig (1988) *Access: Models of Airport Access and Airport Choice for the San Francisco Bay Region*, Version 1.2, Revision 2. Developed for the Metropolitan Transportation Commission, Oakland, California. December 1988.

Koppelman, Frank S. and Chandra Bhat, “A Self Instructing Course in Mode Choice Modeling: Multinomial and Nested Logit Models”, prepared for Federal Transit Administration. January 31, 2006 (modified June 30, 2006).

http://www.ce.utexas.edu/prof/bhat/COURSES/LM_Draft_060131Final-060630.pdf

Metropolitan Transportation Commission, “ACCESS: San Francisco Bay Area Airport Choice Model and Ground Access Mode Choice Model version 1.2”

Metropolitan Washington Council of Governments, “2009 Washington-Baltimore Regional Air Passenger Survey”, September 2010.

Metropolitan Washington Council of Governments, “TPB Nested Logit Mode Choice Model”, presentation to the Travel Forecasting Subcommittee. March 21, 2008.

<https://www.mwcog.org/uploads/committee-documents/kV5fWF9Z20080321114401.pdf>

Metropolitan Washington Council of Governments, “Washington-Baltimore Regional Air Passenger Origin/Destination Forecast Update”, May 2011.

Metropolitan Washington Council of Governments, “Washington-Baltimore Regional Airports Air Passenger Ground Access Forecast Update”, May 2012.

Parsons, Brinckerhoff, Quade and Douglas, Inc. and KPMG, “Dulles Corridor Study Model”, developed for Metropolitan Washington Council of Governments, 1995.

Regional Planning Commission, “Development and Calibration of Travel Demand Models for the New Orleans Region”, developed by Parsons, Brinckerhoff, Quade and Douglas, Inc. November 4, 2002.

Southern California Association of Governments, “SCAG Regional Airport Demand Model”, prepared by Cambridge Systematics, Inc., Dr. Geoffrey D. Gosling, and SH&E, Inc. June 2003.

9 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 recommends performing a TransCAD assignment using all of the network attributes and trip tables from a congested MWCOG Version 2.3 model application. This will provide a direct software comparison that could help Citilabs focus on any assignment convergence issues that may be identified.
2. AECOM recommends constructing fewer input files that directly serve the needs of multiple programs or scripts. We also recommend reconfiguring the programs and scripts to write fewer output files.
3. AECOM recommends including additional error checking in the process to minimize the possibility of application errors.
4. AECOM recommends adding logic to the modeling process that minimizes the possibility of conflicts between multiple applications running on the same computer at the same time.
5. AECOM recommends that MWCOG consider some of the processing changes and software tools developed for the WMATA post-processor. We believe it is in the best interest of both agencies and the region to minimize the differences between the two modeling processes.
6. AECOM believes modeling all trip purposes by time of day is important for WMATA's transit modeling. It is less clear how important this is for MWCOG's modeling needs. At a minimum, however, AECOM would recommend adding the Pedestrian Environment Factor concept to the mode choice models and recalibrating these models with fewer geographic market segments and constrained constants.
7. AECOM recommends converting the transit summary processing from LineSum 2.3 to LineSum 5.0.11.
8. As a short-term solution, AECOM recommends adding logic to the TRNBUILD walk access scripts (or the TransitAccess program) to connect stations to nearby zones and sidewalk nodes and adding transfer prohibitions in the path-builder to force transfers to or from bus routes to use these new links to access the station. This would enable the model to distinguish bus transfers from walk access at Metrorail stations.
9. AECOM recommends implementing Intra-step Distributed Processing (IDP) and Multi-step Distributed Processing (MDP) to the fullest extent possible. We also recommend making it easier to use more than four CPUs when the computer resources are available and the results are not affected by the change.
10. AECOM still believes that it "should" be possible to assign non-HOV and HOV3+ trip tables in the same multi-class assignment step. We recognize that implementing this process using the current trip tables underestimates HOV volumes. We recommend that MWCOG consider potential adjustments to trip distribution and path-building parameters that are likely to improve the HOV volumes generated by a multi-class traffic assignment process.

11. AECOM recommends formally starting the conversion process from TRNBUILD to PT. We also recommend making this a coordinated and cooperative effort between MWCOG, WMATA, and the consultant team.
12. Section 6.5.1 outlines a five phase work program for implementing the PT conversion and upgrading the transit modeling process to take advantage of PT capabilities. The five phases include: transit network preparation, transit path building and loading, transit fares, mode choice calibration, and advanced applications. AECOM recommends pursuing at least the first two phases during fiscal year 2013. Phases 3 and 4 can be implemented if funds and time permit, but Phase 5 is perhaps more appropriate for fiscal year 2014.
13. Stump/Hausman recommends that the air passenger model be recalibrated (for all modes) using the most recently available air passenger survey.
14. Stump/Hausman recommends a partial Airport Choice model to address the portion of travelers that choose an airport based on distance and/or travel time.
15. Stump/Hausman suggests that MWCOG investigate revising the external trip model so that it estimates person trips instead of only vehicle trips. This includes changes to transit network coding and the mode choice model to permit that model to specifically include external trips.
16. Stump/Hausman recommends that a separate four-step model of Visitor travel be developed for the MWCOG region.

10 Appendix

10.1 Script to Forecast Non-Resident Metrorail Trips

```
run pgm=matrix msg="Step 1: Summarize 2007 observed non-resident MR trips"
```

```
; nonRes.s
```

```
; Forecast non-resident Metrorail ridership
```

```
id = "Summarize 2007 observed non-resident MR trips"
```

```
; MATI must always be 2007 data.
```

```
mati = surv2007.trp
```

```
; Define table numbers:
```

```
; 1 E/I auto work (peak)
```

```
; 2 E/I auto non-work (off-pk)
```

```
; 3 E/I walk/train work (peak)
```

```
; 4 E/I walk/train non-work (off-pk)
```

```
; 5 NHB visitor/tourist (walk, off-pk)
```

```
; 6 NHB business (walk, off-pk)
```

```
; 7 non-resident air passenger (walk, off-pk)
```

```
fillmw mw[1] = mi.1.1,2,3,4,5,6,7
```

```
mw[8] = mw[1] + mw[2] + mw[3] + mw[4] + mw[5] + mw[6] + mw[7]
```

```
; Write out total trip ends by table and directionality and also the
```

```
; grand total P+A.
```

```
report marginrec=y, form=7, list=j,r1,c1,r2,c2,r3,c3,r4,c4,r5,c5,  
r6,c6,r7,c7,r8_c8, file=surv2007.tnd
```

```
endrun
```

```
-----  
run pgm=matrix msg="Step 2: Ext station growth by Metrorail station shed"
```

```
id = "Ext station growth by Metrorail station shed"
```

```
reci = Ver2.3.38_Conf_Xmittal\2040_base\inputs\Station.dbf
```

```
; ZDATI[1] must always be 2007 data.
```

```
zdati[1] = Ver2.3.38_Conf_Xmittal\2007_pseu\inputs\Ext_PsAs.dbf
```

```
zdati[2] = Ver2.3.38_Conf_Xmittal\2040_base\inputs\Ext_PsAs.dbf
```

```
zdati[3] = surv2007.tnd, z=#1, tot=#2
```

```
reco = extfac.dbf, fields=staz(5),sname(27),efac(6.2),shed(3)
```

```
; For each Metrorail station, look up the starting and ending
```

```
; numbers of the external stations that make up the shed for that MR sta.
```

```
; Lookup table is staName07.dbf, which is tab staName07 of MetrorailStations.xlsx.
```

```
; This file never changes.
```

```
lookupi[1] = staName07.dbf
```

```
; Define station equivalency lookup.
```

```
lookup lookupi=1, name=staName,
```

```
lookup[1]=sta_node, result=extSta1,
```

```
lookup[2]=sta_node, result=extSta2,
```

```
lookup[3]=sta_node, result=Segment,
```

```
fail=0,0,0, interpolate=n
```

```
; Do this only for Metrorail stations.
```

```
if (ri.mm = 'M')
```

```

; Sum the person X/I trip ends for each shed, for base (2007) and
; future years. The ratio is the E/I non-resident growth factor at
; the external end.
  _ext1 = staName(1,ri.stat)
  _ext2 = staName(2,ri.stat)
  loop _q = _ext1,_ext2
    _base = _base + zi.1.hbw_xi[_q] + zi.1.hbs_xi[_q] + zi.1.hbo_xi[_q] +
    zi.1.nhb_xi[_q]
    _futr = _futr + zi.2.hbw_xi[_q] + zi.2.hbs_xi[_q] + zi.2.hbo_xi[_q] +
    zi.2.nhb_xi[_q]
  endloop

  ro.efac = 1.0
  if (_base > 0) ro.efac = _futr/_base
  ro.shed = staName(3,ri.stat)

  write reco=1

endif

endrun

;-----
run pgm=matrix msg="Step 3: Empl growth at zones near Metrorail stations"

id = "Empl growth at zones near Metrorail stations"

reci = Ver2.3.38_Conf_Xmittal\2040_base\inputs\Station.dbf
; ZDATI[1] must always be 2007 data.
zdati[1] = Ver2.3.38_Conf_Xmittal\2007_pseu\inputs\zone.dbf
zdati[2] = Ver2.3.38_Conf_Xmittal\2040_base\inputs\zone.dbf
zdati[3] = surv2007.tnd, z=#1, pr=#2, at=#3, tot=#4

reco = intfac.dbf, fields=staz(5),sname(27),ifac(6.2)

; Do this only for Metrorail stations.
if (ri.mm = 'M')

  baseEmp = 0
  futrEmp = 0

; Loop on each zone.
  loop iz = 1,3675
    dist = sqrt((ri.stax - zi.1.tazxcrd[iz])^2 + (ri.stay - zi.1.tazycrd[iz])^2)

; Use 0.50 mi distance (2640 ft, because coordinates are in feet), as per R. Milone's
; suggestion. If the zone's centroid is this close to the station, include the
; zone's employment in the calculation.
    if (dist <= 2640)
      baseEmp = baseEmp + zi.1.totemp[iz]
      futrEmp = futrEmp + zi.2.totemp[iz]

      if (ri.staz=520)print form=8,list=iz,dist,zi.1.totemp[iz],zi.2.totemp[iz]
    endif
  endloop

; If no zone was within the specified distance, use the data for the zone
; nearest the station.
  if (baseEmp * futrEmp < 0.01)
    baseEmp = zi.1.totemp[ri.staz]
    futrEmp = zi.2.totemp[ri.staz]
  endif

; Sum the total employment for each Metrorail station's zone, for base (2007) and
; future years. The ratio is the Metrorail non-resident growth factor at the

```



```

; internal end.
  ro.ifac = 1.0
  if (baseEmp * futrEmp > 0)
    ro.ifac = futrEmp/baseEmp
  endif

  write reco=1

endif

endrun

;-----
run pgm=matrix msg="Step 4: Reconcile growth factors and base trip end totals"

id = "Reconcile growth factors and base trip end totals"

; The above steps calculate one set of factors by station. Fratar
; issues a warning if you try to factor a row or column that totals zero.
; Here, examine the starting trip ends and zero out the factor for any
; that are zero. This produces two sets of factors for each station and
; each table. This step does not change the answers, it only makes the
; final Return Code equal zero. This is important to some people.

zdati[1] = surv2007.tnd, z=#1, pr1=#2, at1=#3, pr2=#4, at2=#5, pr3=#6, at3=#7,
  pr4=#8, at4=#9, pr5=#10, at5=#11, pr6=#12, at6=#13, pr7=#14, at7=#15, tot=#16
zdati[2] = intFac.dbf, z=staz
zdati[3] = extFac.dbf, z=staz

reco = totFac.dbf, form=6.2, fields=staz(5),sname(27),ip1,ia1,ip2,ia2,
  ip3,ia3,ip4,ia4,ip5,ia5,ip6,ia6,ip7,ia7,ep1,ea1,ep2,ea2,ep3,ea3,ep4,
  ea4,ep5,ea5,ep6,ea6,ep7,ea7

zones = 3722

ro.ip1 = 0
ro.ip2 = 0
ro.ip3 = 0
ro.ip4 = 0
ro.ip5 = 0
ro.ip6 = 0
ro.ip7 = 0
ro.ia1 = 0
ro.ia2 = 0
ro.ia3 = 0
ro.ia4 = 0
ro.ia5 = 0
ro.ia6 = 0
ro.ia7 = 0
ro.ep1 = 0
ro.ep2 = 0
ro.ep3 = 0
ro.ep4 = 0
ro.ep5 = 0
ro.ep6 = 0
ro.ep7 = 0
ro.ea1 = 0
ro.ea2 = 0
ro.ea3 = 0
ro.ea4 = 0
ro.ea5 = 0
ro.ea6 = 0
ro.ea7 = 0
if (tot > 0)
  if (pr1 > 0)
    ro.ip1 = ifac
  
```

```

    ro.ep1 = efac
endif
if (pr2 > 0)
    ro.ip2 = ifac
    ro.ep2 = efac
endif
if (pr3 > 0)
    ro.ip3 = ifac
    ro.ep3 = efac
endif
if (pr4 > 0)
    ro.ip4 = ifac
    ro.ep4 = efac
endif
if (pr5 > 0)
    ro.ip5 = ifac
    ro.ep5 = efac
endif
if (pr6 > 0)
    ro.ip6 = ifac
    ro.ep6 = efac
endif
if (pr7 > 0)
    ro.ip7 = ifac
    ro.ep7 = efac
endif

if (at1 > 0)
    ro.ia1 = ifac
    ro.ea1 = efac
endif
if (at2 > 0)
    ro.ia2 = ifac
    ro.ea2 = efac
endif
if (at3 > 0)
    ro.ia3 = ifac
    ro.ea3 = efac
endif
if (at4 > 0)
    ro.ia4 = ifac
    ro.ea4 = efac
endif
if (at5 > 0)
    ro.ia5 = ifac
    ro.ea5 = efac
endif
if (at6 > 0)
    ro.ia6 = ifac
    ro.ea6 = efac
endif
if (at7 > 0)
    ro.ia7 = ifac
    ro.ea7 = efac
endif

ro.staz = zi.2.staz
ro.sname = zi.2.sname

write reco = 1
endif

endrun

;-----
run pgm=fratar msg="Step 5: Fratar Metrorail non-resident trip tables"

```

```

; This is part 1 of 2 parts to the forecasting. This step
; deals only with existing stations.

id = "Fratar Metrorail non-resident trip tables

; MATI must always be 2007 data.
mati = surv2007.trp
zdati[1] = totFac.dbf, z=staz

mato = temp1.trp, mo=1-7

; Fratar base 2007 Metrorail non-resident trips using the growth rates
; calculated above. Use external sta factors for the E/I prods and
; internal zone factors for the E/I attrs and for both ends of the Visitor
; and NHB business trips. Input trips are in P/A format.
setpa pgf[1] = ep1, agf[1] = ia1, mw[1] = mi.1.1,
control[1] = P
setpa pgf[2] = ep2, agf[2] = ia2, mw[2] = mi.1.2,
control[2] = P
setpa pgf[3] = ep3, agf[3] = ia3, mw[3] = mi.1.3,
control[3] = P
setpa pgf[4] = ep4, agf[4] = ia4, mw[4] = mi.1.4,
control[4] = P
setpa pgf[5] = ip5, agf[5] = ia5, mw[5] = mi.1.5,
control[5] = PA
setpa pgf[6] = ip6, agf[6] = ia6, mw[6] = mi.1.6,
control[6] = PA
setpa pgf[7] = ip7, agf[7] = ia7, mw[7] = mi.1.7,
control[7] = PA

; Set convergence and iteration control parameters.
maxrmse = 3, maxiters = 19
margins = 1,2,3,4,5,6,7

endrun

*del temp.dbf

;-----
run pgm=matrix msg="Step 6: Est board+alight for new Metrorail sta"

; Part 2: apply non-resident Metrorail model for new stations

id = "Est board+alight for new Metrorail sta

reci = Ver2.3.38_Conf_Xmittal\2040_base\inputs\Station.dbf
zdati[1] = surv2007.tnd, z=#1, tend=#16
zdati[2] = Ver2.3.38_Conf_Xmittal\2040_base\inputs\zone.dbf

reco = temp.dbf, form=6, fields=staz,sname(27),newtrp,boardPct(7.4),stax(10),stay

; Do this only for Metrorail stations.
if (ri.mm = 'M')

; RECI lists all Metrorail stations for this scenario. ZDATI lists trip ends
; for the stations in the 2007 on-board survey. Check one against the other.
; Any station in RECI that didn't have boardings + alightings in 2007 must
; be a new station. Except: stations 8041 Congress Heights and 8085 Morgan Blvd/
; Summerfield had no surveyed non-resident riders in 2007, so we must specifically
; exclude them from the list of new stations.
trips = zi.1.tend[ri.staz]
if (trips == 0 && ri.stat != 8041 && ri.stat != 8085)

emp100 = 0
area100 = 0

```

```

; Loop on internal zones. Calculate straight line distance to this station.
loop iz = 1,3675
  dist = sqrt((ri.stax - zi.2.tazxcrd[iz])^2 + (ri.stay - zi.2.tazycrd[iz])^2)

; Calculate floating zone variables within 1.0 mile. Coordinates are in feet.
if (dist <= 5280)
  emp100 = emp100 + zi.2.totemp[iz]
  area100 = area100 + zi.2.landarea[iz]
endif

endloop

; If no zone was within the specified distance, use the data for the zone
; nearest the station.
if (area100 < 0.01)
  emp100 = zi.2.totemp[ri.staz]
  area100 = zi.2.landarea[ri.staz]
endif

; Apply model to estimate total boardings and alightings.
; Model developed in rail07_final_3722.xlsx, tab "NRmodel".
; Include Dulles Airport. Later, replace those trips with data from the
; air passenger model.
den = 0
if (emp100 * area100 > 0) den = emp100/area100
newtrp = 300 + 0.016*den

; Apply separate model to estimate daily boarding share by station, for E/I
; trips. Model developed in rail07_final_3722.xlsx, tab "NRmodel".
boardPct = 0.5
if (den > 0) boardPct = max(min(1.895 - 0.156*ln(den),1),0)

write reco=1
endif

endif

endrun

;-----
run pgm=matrix msg="Step 7: Find nearest old sta to each new sta"

id = "Find nearest old sta to each new sta"

reci = temp.dbf
; ZDATI[1] must always be 2007 data.
zdati = Ver2.3.38_Conf_Xmittal\2007_pseu\inputs\Station.dbf, z=staz

reco = newSta.dbf, form=6, fields=staz,sname(27),newtrp,boardPct(7.4),stax(10),
stay,nr,closest

; Loop on new stations. Find closest 2007 existing Metrorail station.
mindist = 999999
loop iz = 1,3675
  if (zi.1.mm[iz] != 'M' || zi.1.seqno[iz] == 0) continue
  dist = sqrt((ri.stax - zi.1.stax[iz])^2 + (ri.stay - zi.1.stay[iz])^2)

; Calculate floating zone variables within 1.0 mile.
if (dist <= mindist)
  mindist = dist
  closest = iz
endif
endloop
nr = reci.recno

```

```

; Special case for future Dulles Airport station. Instead of copying its trip
; patterns from the closest Metrorail station (Vienna), copy them from National
; Airport station.
if (ri.staz == 1703) closest = 1510

write reco=1

; Write the number of new stations to the Cube log file.
if (i = 0)
  recs = reci.numrecords
  log var=recs
endif

endrun

;-----
run pgm=matrix msg="Step 8: Adjust non-res Metrorail tables for new sta"

id = "Adjust non-res Metrorail tables for new sta

; For new stations, replace the table row & column with the row & column
; from the nearest existing station.

mati = temp1.trp
lookupi[1] = newSta.dbf

mato = temp2.trp, mo=1-7

; Define station equivalency lookup.
lookup lookupi=1, name=nSta,
  lookup[1]=nr, result=staz,
  lookup[2]=nr, result=closest,
  fail=0,0,0, interpolate=n

fillmw mw[1] = mi.1.1,2,3,4,5,6,7

; For each I zone, loop on the list of new stations until we find one
; where the new zone's station matches I.
loop s = 1,@matrix.recs@
  new = nSta(1,s)
  old = nSta(2,s)

; If this row is a new station's row, copy the row from the closest
; existing station.
if (i == new)
  jloop
    mw[1] = matval(1,1,old,j,0)
    mw[2] = matval(1,2,old,j,0)
    mw[3] = matval(1,3,old,j,0)
    mw[4] = matval(1,4,old,j,0)
    mw[5] = matval(1,5,old,j,0)
    mw[6] = matval(1,6,old,j,0)
    mw[7] = matval(1,7,old,j,0)

  endjloop

else

; Copy the column for this station's zone.
  mw[1][new] = matval(1,1,i,old,0)
  mw[2][new] = matval(1,2,i,old,0)
  mw[3][new] = matval(1,3,i,old,0)
  mw[4][new] = matval(1,4,i,old,0)
  mw[5][new] = matval(1,5,i,old,0)
  mw[6][new] = matval(1,6,i,old,0)
  mw[7][new] = matval(1,7,i,old,0)

```

```

endif
endloop

report margins=1,2,3,4,5,6,7
report marginrec=y, form=6, list=j,r1,c1,r2,c2,r3,c3,r4,c4,r5,c5,
r6,c6,r7,c7, file=temp2.tnd

endrun

;-----
run pgm=matrix msg="Step 9: Merge est non-res B+A trip ends for exist & new sta"

id = "Merge est non-res B+A trip ends for exist & new sta

zdati[1] = temp2.tnd, z=#1, p1=#2, a1=#3, p2=#4, a2=#5, p3=#6, a3=#7,
p4=#8, a4=#9, p5=#10, a5=#11, p6=#12, a6=#13, p7=#14, a7=#15
zdati[2] = newSta.dbf, z=staz

zones = 3722

; For existing stations, use data from ZDAT11 as is.
if (zi.2.nr == 0)
pr1 = p1
pr2 = p2
pr3 = p3
pr4 = p4
pr5 = p5
pr6 = p6
pr7 = p7

at1 = a1
at2 = a2
at3 = a3
at4 = a4
at5 = a5
at6 = a6
at7 = a7

else
; For new stations, split the E/I total trip ends (board + alight) by
; direction using boardPct estimated above. For the NHB total trip ends,
; use 50/50 B/A split. Split the total trips by purpose using the
; percentages from the 2007 survey for the west Orange Line suburban
; stations (Vienna - Ballston).
pr1 = newtrp * 0.20 * boardPct
pr2 = newtrp * 0.23 * boardPct
pr3 = newtrp * 0.08 * boardPct
pr4 = newtrp * 0.15 * boardPct
pr5 = newtrp * 0.20 * 0.5
pr6 = newtrp * 0.13 * 0.5
pr7 = newtrp * 0.01 * 0.5

at1 = newtrp * 0.06 * (1-boardPct)
at2 = newtrp * 0.05 * (1-boardPct)
at3 = newtrp * 0.20 * (1-boardPct)
at4 = newtrp * 0.09 * (1-boardPct)
at5 = newtrp * 0.40 * 0.5
at6 = newtrp * 0.13 * 0.5
at7 = newtrp * 0.07 * 0.5

; Must treat Dulles Airport station differently. Use the purpose and boarding
; splits from National Airport.
if (i == 1703)
pr1 = newtrp * 0.03 * 0.67
pr2 = 0

```

```
pr3 = newtrp * 0.07 * 0.67
pr4 = 0
pr5 = newtrp * 0.02 * 0.5
pr6 = newtrp * 0.08 * 0.5
pr7 = newtrp * 0.80 * 0.5
```

```
at1 = newtrp * 0.04 * 0.33
at2 = 0
at3 = newtrp * 0.14 * 0.33
at4 = 0
at5 = 0
at6 = newtrp * 0.27 * 0.5
at7 = newtrp * 0.55 * 0.5
endif
```

; Check to see if original row total is zero; if so, set new row total
; to zero to keep Fratar from giving a warning. Add the lost trips back
; to the other tables. This sure is a lot of code just to keep Fratar happy.

```
if (p1 < 1)
pr1 = 0
temp2 = pr1 + at1 + pr2 + at2 + pr3 + at3 + pr4 + at4 + pr5 + at5 +
pr6 + at6 + pr7 + at7
fac = 1.0
if (temp2 > 0) fac = newtrp/temp2
at1 = at1 * fac
pr2 = pr2 * fac
at2 = at2 * fac
pr3 = pr3 * fac
at3 = at3 * fac
pr4 = pr4 * fac
at4 = at4 * fac
pr5 = pr5 * fac
at5 = at5 * fac
pr6 = pr6 * fac
at6 = at6 * fac
pr7 = pr7 * fac
at7 = at7 * fac
endif
```

```
if (p2 < 1)
pr2 = 0
temp2 = pr1 + at1 + pr2 + at2 + pr3 + at3 + pr4 + at4 + pr5 + at5 +
pr6 + at6 + pr7 + at7
fac = 1.0
if (temp2 > 0) fac = newtrp/temp2
pr1 = pr1 * fac
at1 = at1 * fac
at2 = at2 * fac
pr3 = pr3 * fac
at3 = at3 * fac
pr4 = pr4 * fac
at4 = at4 * fac
pr5 = pr5 * fac
at5 = at5 * fac
pr6 = pr6 * fac
at6 = at6 * fac
pr7 = pr7 * fac
at7 = at7 * fac
endif
```

```
if (p3 < 1)
pr3 = 0
temp2 = pr1 + at1 + pr2 + at2 + pr3 + at3 + pr4 + at4 + pr5 + at5 +
pr6 + at6 + pr7 + at7
fac = 1.0
if (temp2 > 0) fac = newtrp/temp2
```

```

pr1 = pr1 * fac
at1 = at1 * fac
pr2 = pr2 * fac
at2 = at2 * fac
at3 = at3 * fac
pr4 = pr4 * fac
at4 = at4 * fac
pr5 = pr5 * fac
at5 = at5 * fac
pr6 = pr6 * fac
at6 = at6 * fac
pr7 = pr7 * fac
at7 = at7 * fac
endif

```

```

if (p4 < 1)
pr4 = 0
temp2 = pr1 + at1 + pr2 + at2 + pr3 + at3 + pr4 + at4 + pr5 + at5 +
pr6 + at6 + pr7 + at7
fac = 1.0
if (temp2 > 0) fac = newtrp/temp2
pr1 = pr1 * fac
at1 = at1 * fac
pr2 = pr2 * fac
at2 = at2 * fac
pr3 = pr3 * fac
at3 = at3 * fac
at4 = at4 * fac
pr5 = pr5 * fac
at5 = at5 * fac
pr6 = pr6 * fac
at6 = at6 * fac
pr7 = pr7 * fac
at7 = at7 * fac
endif

```

```

if (p5 < 1)
pr5 = 0
temp2 = pr1 + at1 + pr2 + at2 + pr3 + at3 + pr4 + at4 + pr5 + at5 +
pr6 + at6 + pr7 + at7
fac = 1.0
if (temp2 > 0) fac = newtrp/temp2
pr1 = pr1 * fac
at1 = at1 * fac
pr2 = pr2 * fac
at2 = at2 * fac
pr3 = pr3 * fac
at3 = at3 * fac
pr4 = pr4 * fac
at4 = at4 * fac
at5 = at5 * fac
pr6 = pr6 * fac
at6 = at6 * fac
pr7 = pr7 * fac
at7 = at7 * fac
endif

```

```

if (p6 < 1)
pr6 = 0
temp2 = pr1 + at1 + pr2 + at2 + pr3 + at3 + pr4 + at4 + pr5 + at5 +
pr6 + at6 + pr7 + at7
fac = 1.0
if (temp2 > 0) fac = newtrp/temp2
pr1 = pr1 * fac
at1 = at1 * fac
pr2 = pr2 * fac

```



```

at2 = at2 * fac
pr3 = pr3 * fac
at3 = at3 * fac
pr4 = pr4 * fac
at4 = at4 * fac
pr5 = pr5 * fac
at5 = at5 * fac
at6 = at6 * fac
pr7 = pr7 * fac
at7 = at7 * fac
endif

```

```

if (p7 < 1)
pr7 = 0
temp2 = pr1 + at1 + pr2 + at2 + pr3 + at3 + pr4 + at4 + pr5 + at5 +
pr6 + at6 + pr7 + at7
fac = 1.0
if (temp2 > 0) fac = newtrp/temp2
pr1 = pr1 * fac
at1 = at1 * fac
pr2 = pr2 * fac
at2 = at2 * fac
pr3 = pr3 * fac
at3 = at3 * fac
pr4 = pr4 * fac
at4 = at4 * fac
pr5 = pr5 * fac
at5 = at5 * fac
pr6 = pr6 * fac
at6 = at6 * fac
at7 = at7 * fac
endif

```

; Check to see if original column total is zero; if so, set new column total
; to zero to keep Fratar from giving a warning. Add the lost trips back
; to the other tables.

```

if (a1 < 1)
at1 = 0
temp2 = pr1 + at1 + pr2 + at2 + pr3 + at3 + pr4 + at4 + pr5 + at5 +
pr6 + at6 + pr7 + at7
fac = 1.0
if (temp2 > 0) fac = newtrp/temp2
pr1 = pr1 * fac
pr2 = pr2 * fac
at2 = at2 * fac
pr3 = pr3 * fac
at3 = at3 * fac
pr4 = pr4 * fac
at4 = at4 * fac
pr5 = pr5 * fac
at5 = at5 * fac
pr6 = pr6 * fac
at6 = at6 * fac
pr7 = pr7 * fac
at7 = at7 * fac
endif

```

```

if (a2 < 1)
at2 = 0
temp2 = pr1 + at1 + pr2 + at2 + pr3 + at3 + pr4 + at4 + pr5 + at5 +
pr6 + at6 + pr7 + at7
fac = 1.0
if (temp2 > 0) fac = newtrp/temp2
pr1 = pr1 * fac
at1 = at1 * fac
pr2 = pr2 * fac

```

```

pr3 = pr3 * fac
at3 = at3 * fac
pr4 = pr4 * fac
at4 = at4 * fac
pr5 = pr5 * fac
at5 = at5 * fac
pr6 = pr6 * fac
at6 = at6 * fac
pr7 = pr7 * fac
at7 = at7 * fac
endif

```

```

if (a3 < 1)
at3 = 0
temp2 = pr1 + at1 + pr2 + at2 + pr3 + at3 + pr4 + at4 + pr5 + at5 +
        pr6 + at6 + pr7 + at7
fac = 1.0
if (temp2 > 0) fac = newtrp/temp2
pr1 = pr1 * fac
at1 = at1 * fac
pr2 = pr2 * fac
at2 = at2 * fac
pr3 = pr3 * fac
pr4 = pr4 * fac
at4 = at4 * fac
pr5 = pr5 * fac
at5 = at5 * fac
pr6 = pr6 * fac
at6 = at6 * fac
pr7 = pr7 * fac
at7 = at7 * fac
endif

```

```

if (a4 < 1)
at4 = 0
temp2 = pr1 + at1 + pr2 + at2 + pr3 + at3 + pr4 + at4 + pr5 + at5 +
        pr6 + at6 + pr7 + at7
fac = 1.0
if (temp2 > 0) fac = newtrp/temp2
pr1 = pr1 * fac
at1 = at1 * fac
pr2 = pr2 * fac
at2 = at2 * fac
pr3 = pr3 * fac
at3 = at3 * fac
pr4 = pr4 * fac
pr5 = pr5 * fac
at5 = at5 * fac
pr6 = pr6 * fac
at6 = at6 * fac
pr7 = pr7 * fac
at7 = at7 * fac
endif

```

```

if (a5 < 1)
at5 = 0
temp2 = pr1 + at1 + pr2 + at2 + pr3 + at3 + pr4 + at4 + pr5 + at5 +
        pr6 + at6 + pr7 + at7
fac = 1.0
if (temp2 > 0) fac = newtrp/temp2
pr1 = pr1 * fac
at1 = at1 * fac
pr2 = pr2 * fac
at2 = at2 * fac
pr3 = pr3 * fac
at3 = at3 * fac

```

```

pr4 = pr4 * fac
at4 = at4 * fac
pr5 = pr5 * fac
pr6 = pr6 * fac
at6 = at6 * fac
pr7 = pr7 * fac
at7 = at7 * fac
endif

if (a6 < 1)
at6 = 0
temp2 = pr1 + at1 + pr2 + at2 + pr3 + at3 + pr4 + at4 + pr5 + at5 +
pr6 + at6 + pr7 + at7
fac = 1.0
if (temp2 > 0) fac = newtrp/temp2
pr1 = pr1 * fac
at1 = at1 * fac
pr2 = pr2 * fac
at2 = at2 * fac
pr3 = pr3 * fac
at3 = at3 * fac
pr4 = pr4 * fac
at4 = at4 * fac
pr5 = pr5 * fac
at5 = at5 * fac
pr6 = pr6 * fac
pr7 = pr7 * fac
at7 = at7 * fac
endif

if (a7 < 1)
at7 = 0
temp2 = pr1 + at1 + pr2 + at2 + pr3 + at3 + pr4 + at4 + pr5 + at5 +
pr6 + at6 + pr7 + at7
fac = 1.0
if (temp2 > 0) fac = newtrp/temp2
pr1 = pr1 * fac
at1 = at1 * fac
pr2 = pr2 * fac
at2 = at2 * fac
pr3 = pr3 * fac
at3 = at3 * fac
pr4 = pr4 * fac
at4 = at4 * fac
pr5 = pr5 * fac
at5 = at5 * fac
pr6 = pr6 * fac
at6 = at6 * fac
pr7 = pr7 * fac
endif

endif

print form=7,list=i,pr1,at1,pr2,at2,pr3,at3,pr4,at4,pr5,at5,
pr6,at6,pr7,at7, file=final.tnd

endrun

;-----
run pgm=fratar msg="Step 10: Final Fratar non-res Metrorail trips"

id = "Final Fratar non-res Metrorail trips

; Initial estimate, with new stations added.
mati = temp2.trp

```

```
; External station growth factor file (computed above)
zdati = final.tnd,z=#1, pr1=#2, at1=#3, pr2=#4, at2=#5, pr3=#6, at3=#7,
pr4=#8, at4=#9, pr5=#10, at5=#11, pr6=#12, at6=#13, pr7=#14, at7=#15

mato = nonresMR2040.trp, mo=1-7, name=ElpkDrv,ElpkWlk,ElopDrv,ElopWlk,VisOpWlk,
NHBopWlk,AirPassOpWlk
```

```
; Fratar initial est trips by trip type. For E/I purposes (1-4), control
; on P total. For NHB purposes (5-7) control on the P+A average.
setpa p[1] = zi.1.pr1, a[1] = zi.1.at1, mw[1] = mi.1.1,
control[1] = P
setpa p[2] = zi.1.pr2, a[2] = zi.1.at2, mw[2] = mi.1.2,
control[2] = P
setpa p[3] = zi.1.pr3, a[3] = zi.1.at3, mw[3] = mi.1.3,
control[3] = P
setpa p[4] = zi.1.pr4, a[4] = zi.1.at4, mw[4] = mi.1.4,
control[4] = P
setpa p[5] = zi.1.pr5, a[5] = zi.1.at5, mw[5] = mi.1.5,
control[5] = PA
setpa p[6] = zi.1.pr6, a[6] = zi.1.at6, mw[6] = mi.1.6,
control[6] = PA
setpa p[7] = zi.1.pr7, a[7] = zi.1.at7, mw[7] = mi.1.7,
control[7] = PA
```

```
; Total
mw[8] = mw[1] + mw[2] + mw[3] + mw[4] + mw[5] + mw[6] + mw[7]
```

```
; Set convergence and iteration control parameters.
maxrmse = 3, maxiters = 19
margins = 1,2,3,4,5,6,7,8
```

```
endrun
```

```
*del surv2007.tnd
*del temp.dbf
*del temp1.trp
*del temp2.trp
*del intFac.dbf
*del extFac.dbf
*del totFac.dbf
```

10.2 Airport Passenger Script

```
*****
;
; Baltimore Metropolitan Council Air Passenger Model
; Model written for the New Orleans Model Update Project, March 2002
; by Bill Allen. Translated from FORTRAN by CMB.
;
; The model performs trip generation, distribution, mode choice, and time
; of day calculations for air passenger trips to/from Baltimore Washington
; International (BWI) airport.
*****
;Purposes
; 1 business, residents          3 pleasure, residents
; 2 business, non-residents     4 pleasure, non-residents

;Modes
;1 drive self          4 transit
;2 dropped off        5 limo/shuttle
;3 rental car         6 taxi
*****
;Step 1 of BWI Airport Model
;This step reads in the I-I, I-X, and X-I HBO and HBW matrix for incomes 1,2,3,4.
```

;The Matrix is transposed and row sum is used to total Productions and Attractions.
 ;Grand Total is Log for later use.
 ;*****

; PARAMETERS :
 ZONESIZE = 3722 ; Max. TAZ No. (Param)
 LSTITAZ = 3675 ; Last Internal Zone No. (Param)
 FstHwyNode = 20000 ; First Highway node (Param)

 ; I/O Files :
 NODEFILE = 'inputs\NODE.dbf' ; Node X/Y File (I/P file)
 LINKFILE = 'inputs\LINK.dbf' ; Link File (I/P file)
 ZONEFILE = 'inputs\ZONE.dbf' ; Zonal Land Use File (I/P file)
 ;;AT_OVR = 'AREAOVER.ASC' ; Area Type Override file (I/P file)
 ATYPFILE = 'AreaType_File.dbf' ; Zonal Area Type file (I/P file)

 AMSPD = '..\support\AMSPD.LKP' ; AM Speed lookup ATxFT (I/P file)
 MDSPD = '..\support\MDSPD.LKP' ; Midday Speed lookup ATxFT (I/P file)

 TOLL_Esc = 'inputs\TOLL_Esc.dbf' ; INPUT Toll Escalation Param file
 HWY_Defl = 'HWY_Deflator.txt' ; INPUT Default Highway Deflator (I/P file)
 LKTAZFILE = 'LinkTAZ.DBF' ; Nearest Taz to each link file(O/P file)
 OU_BSNET = 'ZONEHWY.NET' ; OUTPUT BUILT network FILE

 valueoftime = 3517.*0.75 / 0.6 ; Value of time - use 75% of average income
 ; Convert to cents per 0.01 minutes

AIRPORT='BWI'
 AIRPORTBWI = 3036 ; BWI- From MWCOG Model
 AIRPORTIAD = 2388 ; IAD- From MWCOG Model
 AIRPORTDCA = 1498 ; DCA- From MWCOG Model
 AIRPORTTAZ=3036
 COLUMN = 2 ; BWI=2, IAD=3, DCA=4

daysperyer=365

autocostmi = 8.74 ; Used to compute drive cost (per "Air Passenger Model", page V-7)

circuitry = 1.50 ; Used to compute shuttle in-vehicle time (per "Air Passenger Model", page V-7)

;; OUTPUT matrices for mode choice model consumption
 ; HBWforMC = '%_iter_%_hbw_NL.ptt' ; HBW Person Trips-4TABS (INC1..INC4)
 ; HBOforMC = '%_iter_%_hbo_NL.ptt' ; HBO Person Trips-4TABS (INC1..INC4)
 HBOIITT = 'c:\data\jobs\MWCOG\Model\HBO_I4_.ptt'
 HBWIIIT = 'c:\data\jobs\MWCOG\Model\HBW_I4_.ptt'
 HWYSKM = 'c:\data\jobs\MWCOG\Model\SOVMI4MD.skm'
 TRNSKMAB = 'c:\data\jobs\MWCOG\Model\TRNAM_AB.skm'
 TRNSKMMR = 'c:\data\jobs\MWCOG\Model\TRNAM_MR.skm'
 TRNSKMCR = 'c:\data\jobs\MWCOG\Model\TRNAM_CR.skm'

ZoneBlank = '3676-3722'

*if exist look.txt del look.txt
 *if exist attraction.txt del attraction.txt
 *if exist busres.txt del busres.txt
 *if exist busnres.txt del busnres.txt
 *if exist nbusres.txt del nbusres.txt
 *if exist nbusnres.txt del nbusnres.txt

 *if exist vehbusres@AIRPORT@.trp del vehbusres@AIRPORT@.trp
 *if exist vehbusnonres@AIRPORT@.trp del vehbusnonres@AIRPORT@.trp
 *if exist vehplsres@AIRPORT@.trp del vehplsres@AIRPORT@.trp
 *if exist vehplsnonres@AIRPORT@.trp del vehplsnonres@AIRPORT@.trp

 *if exist trnbusres@AIRPORT@.trp del trnbusres@AIRPORT@.trp

```

*if exist trnbusnonres@AIRPORT@.trp del trnbusnonres@AIRPORT@.trp
*if exist trnplsres@AIRPORT@.trp del trnplsres@AIRPORT@.trp
*if exist trnplsnonres@AIRPORT@.trp del trnplsrenonres@AIRPORT@.trp

*if exist veham@AIRPORT@.trp del veham@AIRPORT@.trp
*if exist vehmd@AIRPORT@.trp del vehmd@AIRPORT@.trp
*if exist vehpm@AIRPORT@.trp del vehpm@AIRPORT@.trp
*if exist vehnt@AIRPORT@.trp del vehnt@AIRPORT@.trp

*if exist trnam@AIRPORT@.trp del trnam@AIRPORT@.trp
*if exist trnmd@AIRPORT@.trp del trnmd@AIRPORT@.trp
*if exist trnpm@AIRPORT@.trp del trnpm@AIRPORT@.trp
*if exist trnnt@AIRPORT@.trp del trnnt@AIRPORT@.trp

run pgm=matrix

zones=@ZONESIZE@

mati[1]=@HBOIITT@ ;HBO I/I person trip table (income 1, 2, 3, 4)
;mati[2]=vehextop.trp ;Non-Work X/I (table 5) and I/X (table 1) vehicle trips
mati[3]=@HBWIITT@ ;HBW I/I person
;mati[4]=vehextpk.trp ;Work X/I (table 2) and I/X (table 1) vehicle trips

;zdati[1]=.\model\tazbwidata.dat, z=#1, rpd=#2, vistfac=#3, shtfare=#4
zdati[1]=VISIT_FACT.prn, z=#1, aaz=2, vistfac=3, juris=4, dca=5, iad=6, bwi=7, shdca=8, shiad=9, shbwi=10

;if(i=1900-2928,2932-2950) ;washington region
; residential business and pleasure
mw[1]=(mi.1.1) ;HBO income 1
mw[2]=(mi.1.2) ;HBO income 2
mw[3]=(mi.1.3) ;HBO income 3
mw[4]=(mi.1.4) ;HBO income 4

;HBO is used for non-residential pleasure
mw[5]=(mi.1.1.t) ;HBO income 1 transposed
mw[6]=(mi.1.2.t) ;HBO income 2 transposed
mw[7]=(mi.1.3.t) ;HBO income 3 transposed
mw[8]=(mi.1.4.t) ;HBO income 4 transposed

;HBW is used for non-residential business
mw[9]=(mi.3.1.t) ;HBW income 1 transposed
mw[10]=(mi.3.2.t) ;HBW income 1 transposed
mw[11]=(mi.3.3.t) ;HBW income 1 transposed
mw[12]=(mi.3.4.t) ;HBW income 1 transposed
;else
; mw[1]=mi.1.1+(mi.2.5*@extstationinc1@*@XIHBOBWIFAC1@) ;HBO income 1
; mw[2]=mi.1.2+(mi.2.5*@extstationinc2@*@XIHBOBWIFAC2@) ;HBO income 2
; mw[3]=mi.1.3+(mi.2.5*@extstationinc3@*@XIHBOBWIFAC3@) ;HBO income 3
; mw[4]=mi.1.4+(mi.2.5*@extstationinc4@*@XIHBOBWIFAC4@) ;HBO income 4

; ;HBO is used for non-residential pleasure
; mw[5]=mi.1.1.t+(mi.2.1.t*@extstationinc1@*@IXHBOBWIFAC1PLS@) ;HBO income 1 transposed
; mw[6]=mi.1.2.t+(mi.2.1.t*@extstationinc2@*@IXHBOBWIFAC2PLS@) ;HBO income 2 transposed
; mw[7]=mi.1.3.t+(mi.2.1.t*@extstationinc3@*@IXHBOBWIFAC3PLS@) ;HBO income 3 transposed
; mw[8]=mi.1.4.t+(mi.2.1.t*@extstationinc4@*@IXHBOBWIFAC4PLS@) ;HBO income 4 transposed

; ;HBW is used for non-residential business
; mw[9]=mi.3.1.t+(mi.4.1.t**@extstationinc1@*@IXHBOBWIFAC1BUS@) ;HBW income 1 transposed
; mw[10]=mi.3.2.t+(mi.4.1.t**@extstationinc1@*@IXHBOBWIFAC2BUS@) ;HBW income 1 transposed
; mw[11]=mi.3.3.t+(mi.4.1.t**@extstationinc1@*@IXHBOBWIFAC3BUS@) ;HBW income 1 transposed
; mw[12]=mi.3.4.t+(mi.4.1.t**@extstationinc1@*@IXHBOBWIFAC4BUS@) ;HBW income 1 transposed
;endif

,*****
,

```

```

; create a visitor factor for non-residential pleasure trips
; factor increases the HBO Attractions for the TAZ, so more non-residential pleasure trips
; are distributed to these TAZs

```

```

,*****
vistfactor=zi.1.VISTFAC*zi.1.@AIRPORT@

```

```

,*****

```

```

; adjust for the fact that air passengers are more likely to be higher
; income than travellers in general.

```

```

,*****

```

```

; Income Group 1 = <$25K
; Income Group 2 = <$80K
; Income Group 3 = <$160K
; Income Group 4 = >$160K
; Source: Survey Page 30
; BWI Airport

```

```

;Income Group 1=0.07
;Income Group 2=0.26
;Income Group 3=0.41
;Income Group 4=0.26
incweight1BWI=0.55
incweight2BWI=1.05
incweight3BWI=1.95
incweight4BWI=1.00

```

```

; Dulles Airport

```

```

;Income Group 1=0.05
;Income Group 2=0.26
;Income Group 3=0.39
;Income Group 4=0.30
incweight1IAD=0.40
incweight2IAD=1.10
incweight3IAD=1.95
incweight4IAD=1.15

```

```

; Reagan National Airport

```

```

; Target Distribution
;Income Group 1=0.04
;Income Group 2=0.21
;Income Group 3=0.39
;Income Group 4=0.36
incweight1DCA=0.30
incweight2DCA=0.95
incweight3DCA=2.10
incweight4DCA=1.30

```

```

incwts1=incweight1@AIRPORT@
incwts2=incweight2@AIRPORT@
incwts3=incweight3@AIRPORT@
incwts4=incweight4@AIRPORT@

```

```

;Exclude the Airport TAZ in both the Production and Attraction Total
if(i=@AIRPORTTAZ@) continue
;if(i=@zoneblank@) continue

```

```

hbototP1=(rowsum(1)-mw[1][@AIRPORTTAZ@])*incwts1
hbototP2=(rowsum(2)-mw[2][@AIRPORTTAZ@])*incwts2
hbototP3=(rowsum(3)-mw[3][@AIRPORTTAZ@])*incwts3
hbototP4=(rowsum(4)-mw[4][@AIRPORTTAZ@])*incwts4

```

```

hbwtotA1=(rowsum(9)-mw[9][@AIRPORTTAZ@])*incwts1*vistfactor
hbwtotA2=(rowsum(10)-mw[10][@AIRPORTTAZ@])*incwts2*vistfactor
hbwtotA3=(rowsum(11)-mw[11][@AIRPORTTAZ@])*incwts3*vistfactor
hbwtotA4=(rowsum(12)-mw[12][@AIRPORTTAZ@])*incwts4*vistfactor

```

```

hbototvFACA1=(rowsum(5)-mw[5][@AIRPORTTAZ@])*incwts1*vistfactor

```

```

hbototvFACA2=(rowsum(6)-mw[6][@AIRPORTTAZ@])*incwts2*vistfactor
hbototvFACA3=(rowsum(7)-mw[7][@AIRPORTTAZ@])*incwts3*vistfactor
hbototvFACA4=(rowsum(8)-mw[8][@AIRPORTTAZ@])*incwts4*vistfactor

```

```

hbogtotP1= hbogtotP1 + hbototP1 + hbototP2 + hbototP3 + hbototP4
hbwgtotA1= hbwgtotA1 + hbwtotA1 + hbwtotA2 + hbwtotA3 + hbwtotA4
hbogtotvFACA1= hbogtotvFACA1 + hbototvFACA1 + hbototvFACA2 + hbototvFACA3 + hbototvFACA4

```

```
log prefix=Airport var=hbogtotP1,hbwgtotA1,hbogtotvFACA1
```

```
endrun
```

```

,*****
;STEP 2 Trip Generation and Trip Distribution
,*****

```

```
run pgm=matrix
```

```
zones=@ZONESIZE@
```

```

mati[1]=@HBOIITT@ ;HBO I/I person trip table (income 1, 2, 3, 4)
;mati[2]=vehextop.trp ;Non-Work X/I (table 5) and I/X (table 1) vehicle trips
mati[3]=@HBWIIITT@ ;HBW I/I person
;mati[4]=vehextpk.trp ;Work X/I (table 2) and I/X (table 1) vehicle trips

```

```

;mati[1]=perhbo.trp ;HBO I/I person trip table (income 1, 2, 3, 4)
;mati[2]=vehextop.trp ;Non-Work X/I and I/X vehicle trips
;mati[3]=perhbw.trp ;HBW I/I person
;mati[4]=vehextpk.trp ;Work X/I (table 2) and I/X (table 1) vehicle trips

```

```

;zdati[1]=.\model\tazbwidata.dat, z=#1, rpd=#2, vistfac=#3, shtfare=#4
zdati[1]=VISIT_FACT.prn, z=#1, aaz=2, vistfac=3, juris=4, dca=5, iad=6, bwi=7, shdca=8, shiad=9, shbwi=10

```

```
if(i=1900-2928,2932-2950);washington region
```

```

mw[1]=(mi.1.1) ;HBO income 1
mw[2]=(mi.1.2) ;HBO income 2
mw[3]=(mi.1.3) ;HBO income 3
mw[4]=(mi.1.4) ;HBO income 4
; mw[1]=(mi.1.1*@MDSUBRESAPCHOICERES@)+(mi.2.5*@extstationinc1@*@IXHBOBWIFAC1WASH@) ;HBO income 1
; mw[2]=(mi.1.2*@MDSUBRESAPCHOICERES@)+(mi.2.5*@extstationinc2@*@IXHBOBWIFAC2WASH@) ;HBO income 2
; mw[3]=(mi.1.3*@MDSUBRESAPCHOICERES@)+(mi.2.5*@extstationinc3@*@IXHBOBWIFAC3WASH@) ;HBO income 3
; mw[4]=(mi.1.4*@MDSUBRESAPCHOICERES@)+(mi.2.5*@extstationinc4@*@IXHBOBWIFAC4WASH@) ;HBO income 4

```

```

;HBO is used for non-residential pleasure
mw[5]=(mi.1.1.t) ;HBO income 1 transposed
mw[6]=(mi.1.2.t) ;HBO income 2 transposed
mw[7]=(mi.1.3.t) ;HBO income 3 transposed
mw[8]=(mi.1.4.t) ;HBO income 4 transposed
; mw[5]=(mi.1.1.t*@MDSUBRESAPCHOICENONRES@)+(mi.2.1.t*@extstationinc1@*@IXHBOBWIFAC1PLS@) ;HBO income 1 transposed
; mw[6]=(mi.1.2.t*@MDSUBRESAPCHOICENONRES@)+(mi.2.1.t*@extstationinc2@*@IXHBOBWIFAC2PLS@) ;HBO income 2 transposed
; mw[7]=(mi.1.3.t*@MDSUBRESAPCHOICENONRES@)+(mi.2.1.t*@extstationinc3@*@IXHBOBWIFAC3PLS@) ;HBO income 3 transposed
; mw[8]=(mi.1.4.t*@MDSUBRESAPCHOICENONRES@)+(mi.2.1.t*@extstationinc4@*@IXHBOBWIFAC4PLS@) ;HBO income 4 transposed

```

```

;HBW is used for non-residential business
mw[9]=(mi.3.1.t) ;HBW income 1 transposed
mw[10]=(mi.3.2.t) ;HBW income 1 transposed
mw[11]=(mi.3.3.t) ;HBW income 1 transposed
mw[12]=(mi.3.4.t) ;HBW income 1 transposed
; mw[9]=(mi.3.1.t*@MDSUBRESAPCHOICENONRES@)+(mi.4.1.t**@extstationinc1@*@IXHBOBWIFAC1BUS@) ;HBW income 1 transposed
; mw[10]=(mi.3.2.t*@MDSUBRESAPCHOICENONRES@)+(mi.4.1.t**@extstationinc1@*@IXHBOBWIFAC2BUS@) ;HBW income 1 transposed
; mw[11]=(mi.3.3.t*@MDSUBRESAPCHOICENONRES@)+(mi.4.1.t**@extstationinc1@*@IXHBOBWIFAC3BUS@) ;HBW income 1 transposed
; mw[12]=(mi.3.4.t*@MDSUBRESAPCHOICENONRES@)+(mi.4.1.t**@extstationinc1@*@IXHBOBWIFAC4BUS@) ;HBW income 1 transposed
;else

```



```

; mw[1]=mi.1.1+(mi.2.5*@extstationinc1@*@XIHBOBWIFAC1@) ;HBO income 1
; mw[2]=mi.1.2+(mi.2.5*@extstationinc2@*@XIHBOBWIFAC2@) ;HBO income 2
; mw[3]=mi.1.3+(mi.2.5*@extstationinc3@*@XIHBOBWIFAC3@) ;HBO income 3
; mw[4]=mi.1.4+(mi.2.5*@extstationinc4@*@XIHBOBWIFAC4@) ;HBO income 4

;HBO is used for non-residential pleasure
; mw[5]=mi.1.1.t+(mi.2.1.t*@extstationinc1@*@IXHBOBWIFAC1PLS@) ;HBO income 1 transposed
; mw[6]=mi.1.2.t+(mi.2.1.t*@extstationinc2@*@IXHBOBWIFAC2PLS@) ;HBO income 2 transposed
; mw[7]=mi.1.3.t+(mi.2.1.t*@extstationinc3@*@IXHBOBWIFAC3PLS@) ;HBO income 3 transposed
; mw[8]=mi.1.4.t+(mi.2.1.t*@extstationinc4@*@IXHBOBWIFAC4PLS@) ;HBO income 4 transposed

;HBW is used for non-residential business
; mw[9]=mi.3.1.t+(mi.4.1.t**@extstationinc1@*@IXHBOBWIFAC1BUS@) ;HBW income 1 transposed
; mw[10]=mi.3.2.t+(mi.4.1.t**@extstationinc1@*@IXHBOBWIFAC2BUS@) ;HBW income 1 transposed
; mw[11]=mi.3.3.t+(mi.4.1.t**@extstationinc1@*@IXHBOBWIFAC3BUS@) ;HBW income 1 transposed
; mw[12]=mi.3.4.t+(mi.4.1.t**@extstationinc1@*@IXHBOBWIFAC4BUS@) ;HBW income 1 transposed
;endif

,*****
;
; create a visitor factor for non-residential pleasure trips
; factor increases the HBO Attractions for the TAZ, so more non-residential pleasure trips
; are distributed to these TAZs
,*****
vistfactor=zi.1.VISTFAC*zi.1.@AIRPORT@

,*****
;
; Trip generation. Calculate total daily trips as a function of
; enplanements and the market shares. Multiply by 2 to get both
; ends of the trip (coming and going).
,*****

;enplanement lookup table 'https://www.mwcog.org/uploads/committee-documents/ol1dX1Y20040805144744.pdf
; Airport enplanement data (Source: Washington-Baltimore Regaional Air Passenger O/D Forecast Update May 2011)
    lookup name=enplanement,
    lookup[1]=1, result=@COLUMN@,
interpolate=n,
fail=0,0,0,
;year enplanement
;      BWI      IAD      DCA
r=' 2000  9784850  9971832  7855323',
' 2001  9801066  10656740  8066071',
' 2002  9817281  11341648  8276819',
' 2003  9833497  12026557  8487567',
' 2004  9849712  12711465  8698315',
' 2005  9865928  13396373  8909063',
' 2006  10092085  13079679  8934359',
' 2007  10318242  12762986  8959655',
' 2008  10544399  12446292  8984952',
' 2009  10770556  12129599  9010248',
' 2010  10996713  11812905  9035544',
' 2011  11364688  12437565  9245702',
' 2012  11732663  13062224  9455860',
' 2013  12100638  13686884  9666018',
' 2014  12468613  14311543  9876176',
' 2015  12836588  14936203  10086334',
' 2016  13200242  15628793  10189235',
' 2017  13563897  16321382  10292136',
' 2018  13927551  17013972  10395037',
' 2019  14291206  17706561  10497938',
' 2020  14654860  18399151  10600839',
' 2021  15070264  19254334  10708988',
' 2022  15485668  20109518  10817137',
' 2023  15901071  20964701  10925287',
' 2024  16316475  21819885  11033436',
' 2025  16731879  22675068  11141585',

```

```
' 2026 17206428 23731536 11255252',
' 2027 17680977 24788003 11368919',
' 2028 18155526 25844471 11482586',
' 2029 18630075 26900938 11596253',
' 2030 19104624 27957406 11709920'
```

; Need to update table with either May 2011 or May 2012 Regional Air Passenger O/D Forecast Update

```
; data help call maa toni 410-859-7742
passenger=enplanement(1,{year})
```

```
; Transferring Passengers
; BWI= 15%
pctxferBWI=0.15
; Dulles = 40%
pctxferIAD=0.40
; National = 9%
pctxferDCA=0.09
; Source: Survey , page 9
```

```
; BWI Airport
; Business-Resident - 11.2%
; Business-NonResident - 18.0%
; NonBusiness-Resident - 29.1%
; NonBusiness-NonResident - 41.7%
pbusresBWI=0.112
pbusnresBWI=0.180
pplsresBWI=0.291
pplsnresBWI=0.417
```

```
; Dulles Airport
; Business-Resident - 10.1%
; Business-NonResident - 21.6%
; NonBusiness-Resident - 17.5%
; NonBusiness-NonResident - 50.8%
pbusresIAD=0.101
pbusnresIAD=0.216
pplsresIAD=0.175
pplsnresIAD=0.508
```

```
; Reagan National Airport
; Business-Resident - 14.1%
; Business-NonResident - 29.6%
; NonBusiness-Resident - 18.3%
; NonBusiness-NonResident - 38.1%
pbusresDCA=0.141
pbusnresDCA=0.296
pplsresDCA=0.183
pplsnresDCA=0.381
```

```
if(i=1) ;calculate total trips during first pass only TAZ=1
trips=2.0 * passenger*(1.0-pctxfer@AIRPORT@)/@daysperyer@
busrestot=trips*pbusres@AIRPORT@
busnrestot=trips*pbusnres@AIRPORT@
plsrestot=trips*pplsres@AIRPORT@
plsnrestot=trips*pplsnres@AIRPORT@
endif
```

```
,*****
; Trip distribution.
,*****
; adjust for the fact that air passengers are more likely to be higher
; income than travellers in general.
,*****
```

```

; Income Group 1 = <$25K
; Income Group 2 = <$80K
; Income Group 3 = <$160K
; Income Group 4 = >$160K
; Source: Survey Page 30
; BWI Airport
;Income Group 1=0.07
;Income Group 2=0.26
;Income Group 3=0.41
;Income Group 4=0.26
incweight1BWI=0.55
incweight2BWI=1.05
incweight3BWI=1.95
incweight4BWI=1.00

```

```

; Dulles Airport
;Income Group 1=0.05
;Income Group 2=0.26
;Income Group 3=0.39
;Income Group 4=0.30
incweight1IAD=0.40
incweight2IAD=1.10
incweight3IAD=1.95
incweight4IAD=1.15

```

```

; Reagan National Airport
; Target Distribution
;Income Group 1=0.04
;Income Group 2=0.21
;Income Group 3=0.39
;Income Group 4=0.36
incweight1DCA=0.30
incweight2DCA=0.95
incweight3DCA=2.10
incweight4DCA=1.30

```

```

incwts1=incweight1@AIRPORT@
incwts2=incweight2@AIRPORT@
incwts3=incweight3@AIRPORT@
incwts4=incweight4@AIRPORT@

```

```

;Exclude the Airport TAZ in both the Production and Attraction Total
if(i=@AIRPORTTAZ@) continue
;if(i=@zoneblank@) continue

```

```

hbototP1=(rowsum(1)-mw[1][@AIRPORTTAZ@])*incwts1
hbototP2=(rowsum(2)-mw[2][@AIRPORTTAZ@])*incwts2
hbototP3=(rowsum(3)-mw[3][@AIRPORTTAZ@])*incwts3
hbototP4=(rowsum(4)-mw[4][@AIRPORTTAZ@])*incwts4

```

```

hbwtotA1=(rowsum(9)-mw[9][@AIRPORTTAZ@])*incwts1*vistfactor
hbwtotA2=(rowsum(10)-mw[10][@AIRPORTTAZ@])*incwts2*vistfactor
hbwtotA3=(rowsum(11)-mw[11][@AIRPORTTAZ@])*incwts3*vistfactor
hbwtotA4=(rowsum(12)-mw[12][@AIRPORTTAZ@])*incwts4*vistfactor

```

```

hbototvFACA1=(rowsum(5)-mw[5][@AIRPORTTAZ@])*incwts1*vistfactor
hbototvFACA2=(rowsum(6)-mw[6][@AIRPORTTAZ@])*incwts2*vistfactor
hbototvFACA3=(rowsum(7)-mw[7][@AIRPORTTAZ@])*incwts3*vistfactor
hbototvFACA4=(rowsum(8)-mw[8][@AIRPORTTAZ@])*incwts4*vistfactor

```

```

,*****
;
; Loop through each TAZ and proportion out Airport passenger trips
; Based on proportion of HBO trips to log total from step 1
,*****

```

```

attrbusresinc1=busrestot*hbototP1/@Airport.hbototP1@

```

```

attrbusresinc2=busrestot*hbototp2/@Airport.hbogtotP1@
attrbusresinc3=busrestot*hbototp3/@Airport.hbogtotP1@
attrbusresinc4=busrestot*hbototp4/@Airport.hbogtotP1@

attrbusresinc1=busnrestot*hbwtota1/@Airport.hbwgtotA1@
attrbusresinc2=busnrestot*hbwtota2/@Airport.hbwgtotA1@
attrbusresinc3=busnrestot*hbwtota3/@Airport.hbwgtotA1@
attrbusresinc4=busnrestot*hbwtota4/@Airport.hbwgtotA1@

attrplsresinc1=plsrestot*hbototp1/@Airport.hbogtotP1@
attrplsresinc2=plsrestot*hbototp2/@Airport.hbogtotP1@
attrplsresinc3=plsrestot*hbototp3/@Airport.hbogtotP1@
attrplsresinc4=plsrestot*hbototp4/@Airport.hbogtotP1@

attrplsresinc1=plsrestot*hbototvFACA1/@Airport.hbogtotvFACA1@
attrplsresinc2=plsrestot*hbototvFACA2/@Airport.hbogtotvFACA1@
attrplsresinc3=plsrestot*hbototvFACA3/@Airport.hbogtotvFACA1@
attrplsresinc4=plsrestot*hbototvFACA4/@Airport.hbogtotvFACA1@

print file=attraction.txt, list=z(5), attrbusresinc1, attrbusresinc2, attrbusresinc3, attrbusresinc4,
      attrbusresinc1, attrbusresinc2, attrbusresinc3, attrbusresinc4,
      attrplsresinc1, attrplsresinc2, attrplsresinc3, attrplsresinc4,
      attrplsresinc1, attrplsresinc2, attrplsresinc3, attrplsresinc4

endrun
.*****
/
.*****
;step 3 Mode Choice
.*****
/
.*****

run pgm=matrix

ZONES=@ZONESIZE@

;mati[2]=hwyofpk.skm      ;SOVTime,SOVDist,SOVToll,HOVTime,
      ;HOVDist,HOVToll,TDTime

mati[2]=@HWYSKM@

mati[3]=@TRNSKMAB@           ; AM Peak Period Walk to Bus Skim

mati[4]=@TRNSKMMR@           ; AM Peak Period Walk to Metrorail Skim

mati[5]=@TRNSKMCR@           ; AM Peak Period Walk to Commuterrail Skim

;mati[3]=wbuspeak.skm      ;walk to bus peak
;mati[4]=wralpeak.skm      ;walk to light rail
;mati[5]=wcrlpeak.skm      ;walk to commuter rail
;mati[6]=vehextop.trp      ;HBO X/I and I/X vehicle trips

;create working matrix for highway and transit input skims
mw[1]=mi.2.1      ;sovdist
mw[21]=mi.2.2      ;sovtime

;walk bus peak
;mw[2]=mi.3.1      ;initwait
;mw[3]=mi.3.2      ;xferwait
;mw[4]=mi.3.3      ;walk
;mw[5]=mi.3.8      ;jvt bus
mw[2]=mi.3.3      ;initwait
mw[3]=mi.3.4      ;xferwait
mw[4]=mi.3.1+mi.3.2      ;walk
mw[5]=mi.3.7+mi.3.10+mi.3.11 ;jvt bus

```

```

;walk Metrorail peak
;mw[6]=mi.4.1      ;initwait
;mw[7]=mi.4.2      ;xferwait
;mw[8]=mi.4.3      ;walk
;mw[9]=mi.4.8      ;ivt bus
;mw[10]=mi.4.9     ;ivt rail
;mw[11]=mi.4.11    ;first mode
;mw[12]=mi.4.12    ;last transfer
mw[6]=mi.4.3      ;initwait
mw[7]=mi.4.4      ;xferwait
mw[8]=mi.4.1+mi.4.2 ;walk
mw[9]=mi.4.7+mi.4.10+mi.4.11 ;ivt bus
mw[10]=mi.4.8+mi.4.9 ;ivt rail
;mw[11]=mi.4.2     ;first mode
;mw[12]=mi.4.12    ;transfers

```

```

;walk MARC (commuter rail)peak
;mw[13]=mi.5.1     ;initwait
;mw[14]=mi.5.2     ;xferwait
;mw[15]=mi.5.3     ;walk
;mw[16]=mi.5.8     ;ivt bus
;mw[17]=mi.5.9     ;ivt rail
;mw[18]=mi.5.10    ;ivt marc
;mw[19]=mi.5.11    ;first mode
;mw[20]=mi.5.12    ;last transfer
mw[13]=mi.5.3     ;initwait
mw[14]=mi.5.4     ;xferwait
mw[15]=mi.5.1+mi.5.2 ;walk
mw[16]=mi.5.7+mi.5.10+mi.5.11 ;ivt bus
mw[17]=mi.5.8+mi.5.9 ;ivt rail
mw[18]=mi.5.6     ;ivt marc
;mw[19]=mi.5.2     ;first mode
;mw[20]=mi.5.12    ;transfers

```

```

;transit fare
;mw[22]=mi.3.6     ;walk bus
;mw[23]=mi.4.6     ;walk rail
;mw[24]=mi.5.6     ;walk marc
mw[22]=mi.3.13    ;walk bus
mw[23]=mi.4.13    ;walk rail
mw[24]=mi.5.13    ;walk marc

```

```

; Structure for Transit Skims (Peak or Off-Peak)
;MW[1] = MI.1.9 ;--- wlk walk acc time (0.01 min)
;MW[2] = MI.1.10 ;--- wlk other walk time (0.01 min)
;MW[3] = MI.1.7 ;--- wlk ini.wait time (0.01 min)
;MW[4] = MI.1.8 ;--- wlk xfr wait time (0.01 min)
;MW[5] = MI.1.3 ;--- wlk ivt-total (0.01 min)
;MW[6] = MI.1.4 ;--- wlk ivt-commuter rail(0.01 min)
;MW[7] = MI.1.2 ;--- wlk ivt-exp bus (0.01 min)
;MW[8] = MI.1.3 ;--- wlk ivt-metrorail (0.01 min)
;MW[9] = MI.1.5 ;--- wlk ivt-new rail mode(0.01 min)
;MW[10] = MI.1.6 ;--- wlk ivt-new bus mode (0.01 min)
;MW[11] = MI.1.1 ;--- wlk ivt-local bus (0.01 min)
;MW[12] = MI.1.12 ;--- wlk transfers (0+)
;MW[13] = MI.2.1 ;--- wlk fare (2007 cents)
;MW[14] = MI.1.11 ;--- wlk added board time (0.01 min)
;MW[15] = MI.3.9 ;--- drv walk acc time (0.01 min)
;MW[16] = MI.3.10 ;--- drv other walk time (0.01 min)
;MW[17] = MI.3.7 ;--- drv ini.wait time (0.01 min)
;MW[18] = MI.3.8 ;--- drv xfr wait time (0.01 min)
;MW[19] = MI.3.3 ;--- drv ivt-total (0.01 min)
;MW[20] = MI.3.4 ;--- drv ivt-commuter rail(0.01 min)
;MW[21] = MI.3.2 ;--- drv ivt-exp bus (0.01 min)
;MW[22] = MI.3.3 ;--- drv ivt-metrorail (0.01 min)

```

```

;MW[23] = MI.3.5 ;---- drv ivt-new rail mode(0.01 min)
;MW[24] = MI.3.6 ;---- drv ivt-new bus mode (0.01 min)
;MW[25] = MI.3.1 ;---- drv ivt-local bus (0.01 min)
;MW[26] = MI.3.12 ;---- drv transfers (0+)
;MW[27] = MI.4.1 ;---- drv fare (2007 cents)
;MW[28] = MI.3.11 ;---- drv added board time (0.01 min)
;MW[29] = MI.3.13 ;---- drv acc time (0.01 min)
;MW[30] = MI.3.14 ;---- drv acc distance (0.01 mile)
;MW[31] = MI.3.16 ;---- drv park cost (2007 cents)
;MW[32] = MI.3.15 ;---- drv park time (0.01 min)
;MW[33] = MI.5.9 ;---- knr walk acc time (0.01 min)
;MW[34] = MI.5.10 ;---- knr other walk time (0.01 min)
;MW[35] = MI.5.7 ;---- knr ini.wait time (0.01 min)
;MW[36] = MI.5.8 ;---- knr xfr wait time (0.01 min)
;MW[37] = MI.5.3 ;---- knr ivt-total (0.01 min)
;MW[38] = MI.5.4 ;---- knr ivt-commuter rail(0.01 min)
;MW[39] = MI.5.2 ;---- knr ivt-exp bus (0.01 min)
;MW[40] = MI.5.3 ;---- knr ivt-metrorail (0.01 min)
;MW[41] = MI.5.5 ;---- knr ivt-new rail mode(0.01 min)
;MW[42] = MI.5.6 ;---- knr ivt-new bus mode (0.01 min)
;MW[43] = MI.5.1 ;---- knr ivt-local bus (0.01 min)
;MW[44] = MI.5.12 ;---- knr transfers (0+)
;MW[45] = MI.6.1 ;---- knr fare (2007 cents)
;MW[46] = MI.5.11 ;---- knr added board time (0.01 min)
;MW[47] = MI.5.13 ;---- knr acc time (0.01 min)
;MW[48] = MI.5.14 ;---- knr acc distance (0.01 mile)

```

; Visitor factors used for non-residents and Shuttle fare in cents

```

;zdati[1]=.\model\tazbwidata.dat, z=#1, rpd=#2, vistfac=#3, shtfare=#4
;zdati[1]=VISIT_FACT.prn, z=#1, aaz=2, vistfac=3, juris=4, dca=5, iad=6, bwi=7,shdca=8, shiad=9, shbwi=10

```

; Average walk time (peak and off peak) for each mode

```

;zdati[2]=inputs\walktime.dat, z=#1, ppct=#2, BusTPK=#3, RailTPK=#4, MARCTPK=#5, OPCT=#6, BusTOP=#7, RailTOP=#8, MARCTOP=#9
; Use MWCOG Transit Skim matrix -- no need for this file

```

; attractions allocated above <step 2>

```

;zdati[3]=attraction.txt, z=#1, attrbusresinc1=#2, attrbusresinc2=#3, attrbusresinc3=#4, attrbusresinc4=#5,
attrbusresinc1=#6, attrbusresinc2=#7, attrbusresinc3=#8, attrbusresinc4=#9,
attrplsresinc1=#10, attrplsresinc2=#11, attrplsresinc3=#12, attrplsresinc4=#13,
attrplsresinc1=#14, attrplsresinc2=#15, attrplsresinc3=#16, attrplsresinc4=#17

```

```

;mato[1]=busres.trp, mo=41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,
; name=inc1DA, inc1DRoFF, inc1RC, inc1Tran, inc1SH, inc1TX,
; inc2DA, inc2DRoFF, inc2RC, inc2Tran, inc2SH, inc2TX,
; inc3DA, inc3DRoFF, inc3RC, inc3Tran, inc3SH, inc3TX,
; inc4DA, inc4DRoFF, inc4RC, inc4Tran, inc4SH, inc4TX

```

```

;mato[2]=busnonres.trp, mo=65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,
; name=inc1DA, inc1DRoFF, inc1RC, inc1Tran, inc1SH, inc1TX,
; inc2DA, inc2DRoFF, inc2RC, inc2Tran, inc2SH, inc2TX,
; inc3DA, inc3DRoFF, inc3RC, inc3Tran, inc3SH, inc3TX,
; inc4DA, inc4DRoFF, inc4RC, inc4Tran, inc4SH, inc4TX

```

```

;mato[3]=pleares.trp, mo=89,90,91,92,93,94,95,96,97,98,99,100,101,102,103,104,105,106,107,108,109,110,111,112,
; name=inc1DA, inc1DRoFF, inc1RC, inc1Tran, inc1SH, inc1TX,
; inc2DA, inc2DRoFF, inc2RC, inc2Tran, inc2SH, inc2TX,
; inc3DA, inc3DRoFF, inc3RC, inc3Tran, inc3SH, inc3TX,
; inc4DA, inc4DRoFF, inc4RC, inc4Tran, inc4SH, inc4TX

```

```

;mato[4]=pleanonres.trp, mo=113,114,115,116,117,118,119,120,121,122,123,124,125,126,127,128,129,130,131,132,133,134,135,136,
; name=inc1DA, inc1DRoFF, inc1RC, inc1Tran, inc1SH, inc1TX,
; inc2DA, inc2DRoFF, inc2RC, inc2Tran, inc2SH, inc2TX,
; inc3DA, inc3DRoFF, inc3RC, inc3Tran, inc3SH, inc3TX,

```

```

; inc4DA, inc4DRoFF, inc4RC, inc4Tran, inc4SH, inc4TX

mato[5]=minwalktime.mtx, mo=200
,*****
; Estimate the Taxi and Shuttle Service Fare to/from BWI (cents)
; Taxi assumptions from http://www.bwiairporttaxi.com/rates.htm
; first 1/8 mile is $2.90 thereafter 20 cents for each 1/8 mile ($1.60 per mile)
; Updated Information: first 1/10 mile is $2.90 thereafter 25 cents for each 1/10 mile ($2.50 per mile)
txfareBWI=(mw[1][@AIRPORTTAZ@]-(1/10))*250+290 ; BWI
; Shuttle assumptions from http://www.rmalimo.com/home.php

; Dulles Airport - Washington Flyer - $3.50 for initial 1/4 mile then $0.50 for each quarter mile ($2.00 per mile)
; Source: www.wmatc.gov/order/new/taxicabfares/WashFlyer.pdf
txfareAD=(mw[1][@AIRPORTTAZ@]-(1/4))*200+350 ; Dulles

; Reagan National Airport - Arlington Taxis - $2.75 for initial 1/6 mile then $0.35 for each 1/6 ($2.10 per mile)
; DC Taxis - $3.00 for initial 1/8 mile then $0.27 for each 1/8 ($2.02 per mile)
; Source: www.wmatc.gov/order/new/taxicabfares/Arl.pdf
txfareDCA=(mw[1][@AIRPORTTAZ@]-(1/6))*210+275 ; Reagan
,taxifare=(mw[1][@DCATAZ@]-(1/8))*202+300

taxifare=txfare@AIRPORT@
,*****

shuttlefare=zi.1.SH@AIRPORT@

,*****
; Calculate the "income function". This is the average HH income
; for each income group, in thousands, to the 1.5 power. From the
; Washington model.
,*****

Array finc=4, avgingrp=4

set val=0, vars=finc, avgingrp

;BMC income groups 10%, 15%, 20%, 55% 1999 DOLLARS Census
;incgrp1 11800
;incgrp2 26000
;incgrp3 44200
;incgrp4 >44200

;MWCOG Income Groups - 2007$
; HH Income Quartile Range
; Group 1 - $0 - $ 50K - Assumed Midpoint = $ 25K
; Group 2 - $ 50K - $100K - Assumed Midpoint = $ 75K
; Group 3 - $100K - $150K - Assumed Midpoint = $125K
; Group 4 - $150K - - Assumed Midpoint = $175K
; Source: User's Guide for the TPB Travel Forecasting Model, V2.3, Build 38, page 10-3

;MWCOG Income Groups - Adjusted to 2009$
; cpi 2007 = 126.16 , cpi 2009 = 131.38
; HH Income Quartile Range
; Group 1 - $0 - $ 52068 - Assumed Midpoint = $ 26034
; Group 2 - $52068 - $104135 - Assumed Midpoint = $ 83986
; Group 3 - $104135 - $156203 - Assumed Midpoint = $141938
; Group 4 - $156203 - - Assumed Midpoint = $199890
; Source: Consumer Price Index for All Urban Consumers, Washington-Baltimore Area USDOL

;1999 to 1990 CPI factor 0.82057
cpi=1.000

avgingrp[1]=26034*cpi
avgingrp[2]=83986*cpi
avgingrp[3]=141938*cpi
avgingrp[4]=199890*cpi

```

```

loop cnt=1,4
  finc[cnt]=pow((avgingrp[cnt]/1000),1.5)
endloop

```

```

;*****
; Mode choice. Borrow the general model structure and coefficients
; from the Washington, D.C. air passenger model, then simplify it
; for this model (the Washington model modelled airport selection,
; had many complex modes, and operated in pivot point fashion using
; survey records).
;
; Model structure:
;
;           Non-Residents (purp 2,4)
;           |
;           |-----|-----|
;           |         |         |
; Dropped Off  Rental Car  Non-Auto
;
;                               |
;                               |-----|-----|
;                               |         |         |
;                               Transit  Limo/Shuttle  Taxi
;
;
;           Residents (purp 1,3)
;           |
;           |-----|-----|
;           |         |         |
; Pvt Auto      Dropped Off  Non-Auto
;
;                               |
;                               |-----|-----|
;                               |         |         |
; Drive Self    Dropped Off  Transit  Limo/Shuttle  Taxi
;*****

```

;Baltimore travel mode has skims by time period (peak/off peak), mode (bus, rail, and MARC) and access (walk/drive). For BWI access, assume walk peak. For mode of access, sort through bus, rail, and MARC skim selecting the best option (shorts travel time).

;Create travel time matrix - Shorts runtime walk to transit (bus, rail, MARC) for each I to BWI
;The walk access and egress is by submode. Need to select correct travel time.
;Since we are only selecting the mininum path for transit representation, add fare * value of time
;to the calculation.

```

;set wlk transit modes to 99999 (if we have travel time by a mode, then it will be the shortest)
;and access egress to zero
wlkbus=99999
wlkbusaccegress=0
wlkrail=99999
wlkrailaccegress=0
wlkmarc=99999
wlkmaraaccegress=0

```

```

;first skim (wbuspeak) is bus only so access and egress is walk to bus
; Bus - available modes BUS only
; lwait  xwait  walk      bus      access  egress  fare(dollars)*VOT(cents per minute)
if(mw[5][@AIRPORTTAZ@]>0) ;bus travel time >0
;
wlkbus=mw[2][@AIRPORTTAZ@]+mw[3][@AIRPORTTAZ@]+mw[4][@AIRPORTTAZ@]+mw[5][@AIRPORTTAZ@]+busTPK+busTPK[@AIRPORTTAZ@]+(mw[22][@AIRPORTTAZ@]/@valueoftime@)
wlkbus=mw[2][@AIRPORTTAZ@]+mw[3][@AIRPORTTAZ@]+mw[4][@AIRPORTTAZ@]+mw[5][@AIRPORTTAZ@]+(mw[22][@AIRPORTTAZ@]/@valueoftime@)
wlkbusaccegress=11
endif

```

;need to do a jloop to test if rail or MARC have travel time

```

;Light Rail and Metro - available modes bus and rail
if(mw[10][@AIRPORTTAZ@]>0) ;rail travel time >0

```



```

; if(mw[11][@AIRPORTTAZ@]=4-5) ;first mode was rail access rail
; if(mw[12][@AIRPORTTAZ@]=0 || mw[12][@AIRPORTTAZ@]<5000) ;no transfer or last transfer to rail egress rail
IF (mw[9][@AIRPORTTAZ@]=0 & mw[10][@AIRPORTTAZ@]>0) ; Metro Rail Only
;
wlrail=mw[6][@AIRPORTTAZ@]+mw[7][@AIRPORTTAZ@]+mw[8][@AIRPORTTAZ@]+mw[9][@AIRPORTTAZ@]+mw[10][@AIRPORTTAZ@]+Rail
TPK+RailTPK[ @AIRPORTTAZ@ ]+(mw[23][@AIRPORTTAZ@]/@valueoftime@)

wlrail=mw[6][@AIRPORTTAZ@]+mw[7][@AIRPORTTAZ@]+mw[8][@AIRPORTTAZ@]+mw[9][@AIRPORTTAZ@]+mw[10][@AIRPORTTAZ@]+(m
w[23][@AIRPORTTAZ@]/@valueoftime@)
wlrailacccess=22 ; Metro Rail Only
else ;last transfer is > 5000 so egress is bus
;
wlrail=mw[6][@AIRPORTTAZ@]+mw[7][@AIRPORTTAZ@]+mw[8][@AIRPORTTAZ@]+mw[9][@AIRPORTTAZ@]+mw[10][@AIRPORTTAZ@]+Rail
TPK+BusTPK[ @AIRPORTTAZ@ ]+(mw[23][@AIRPORTTAZ@]/@valueoftime@)

wlrail=mw[6][@AIRPORTTAZ@]+mw[7][@AIRPORTTAZ@]+mw[8][@AIRPORTTAZ@]+mw[9][@AIRPORTTAZ@]+mw[10][@AIRPORTTAZ@]+(m
w[23][@AIRPORTTAZ@]/@valueoftime@)
wlrailacccess=21 ; Metro Rail and Bus
endif
else ;first mode was not rail access bus
; if(mw[12][@AIRPORTTAZ@]<5000) ;last transfer was rail egress rail
IF (mw[9][@AIRPORTTAZ@]>0 & mw[10][@AIRPORTTAZ@]>0) ; Bus and Rail
;
wlrail=mw[6][@AIRPORTTAZ@]+mw[7][@AIRPORTTAZ@]+mw[8][@AIRPORTTAZ@]+mw[9][@AIRPORTTAZ@]+mw[10][@AIRPORTTAZ@]+Bus
TPK+RailTPK[ @AIRPORTTAZ@ ]+(mw[23][@AIRPORTTAZ@]/@valueoftime@)

wlrail=mw[6][@AIRPORTTAZ@]+mw[7][@AIRPORTTAZ@]+mw[8][@AIRPORTTAZ@]+mw[9][@AIRPORTTAZ@]+mw[10][@AIRPORTTAZ@]+(m
w[23][@AIRPORTTAZ@]/@valueoftime@)
wlrailacccess=12
else ; Bus Only ;last transfer was not rail egress bus
;
wlrail=mw[6][@AIRPORTTAZ@]+mw[7][@AIRPORTTAZ@]+mw[8][@AIRPORTTAZ@]+mw[9][@AIRPORTTAZ@]+mw[10][@AIRPORTTAZ@]+Bus
TPK+BusTPK[ @AIRPORTTAZ@ ]+(mw[23][@AIRPORTTAZ@]/@valueoftime@)

wlrail=mw[6][@AIRPORTTAZ@]+mw[7][@AIRPORTTAZ@]+mw[8][@AIRPORTTAZ@]+mw[9][@AIRPORTTAZ@]+mw[10][@AIRPORTTAZ@]+(m
w[23][@AIRPORTTAZ@]/@valueoftime@)
wlrailacccess=11
endif
; endif
endif

;MARC - available modes bus, rail and marc
if(mw[18][@AIRPORTTAZ@]>0) ;MARC travel time >0
; if(mw[19][@AIRPORTTAZ@]=6) ;fist mode was MARC access MARC
; if(mw[20][@AIRPORTTAZ@]=0 || mw[20][@AIRPORTTAZ@]=4000-4070) ;no transfer or last transfer to MARC
IF (mw[16][@AIRPORTTAZ@]=0 & mw[17][@AIRPORTTAZ@]=0 & mw[18][@AIRPORTTAZ@]>0)
; Commuter Rail Only
;
wlmarc=mw[13][@AIRPORTTAZ@]+mw[14][@AIRPORTTAZ@]+mw[15][@AIRPORTTAZ@]+mw[16][@AIRPORTTAZ@]+mw[17][@AIRPORTTAZ
@]+mw[18][@AIRPORTTAZ@]+MARCTPK+MARCTPK[ @AIRPORTTAZ@ ]+(mw[24][@AIRPORTTAZ@]/@valueoftime@)

wlmarc=mw[13][@AIRPORTTAZ@]+mw[14][@AIRPORTTAZ@]+mw[15][@AIRPORTTAZ@]+mw[16][@AIRPORTTAZ@]+mw[17][@AIRPORTTAZ
@]+mw[18][@AIRPORTTAZ@]+(mw[24][@AIRPORTTAZ@]/@valueoftime@)
wlmarcacccess=33
; elseif(mw[20][@AIRPORTTAZ@]=4105-4999) ;last transfer to rail
ELSEIF ((mw[16][@AIRPORTTAZ@]=0 | mw[16][@AIRPORTTAZ@]>0) &
mw[17][@AIRPORTTAZ@]>0 & mw[18][@AIRPORTTAZ@]>0) ; Metrorail and Commuter Rail Only
;
wlmarc=mw[13][@AIRPORTTAZ@]+mw[14][@AIRPORTTAZ@]+mw[15][@AIRPORTTAZ@]+mw[16][@AIRPORTTAZ@]+mw[17][@AIRPORTTAZ
@]+mw[18][@AIRPORTTAZ@]+MARCTPK+RailTPK[ @AIRPORTTAZ@ ]+(mw[24][@AIRPORTTAZ@]/@valueoftime@)

wlmarc=mw[13][@AIRPORTTAZ@]+mw[14][@AIRPORTTAZ@]+mw[15][@AIRPORTTAZ@]+mw[16][@AIRPORTTAZ@]+mw[17][@AIRPORTTAZ
@]+mw[18][@AIRPORTTAZ@]+(mw[24][@AIRPORTTAZ@]/@valueoftime@)
wlmarcacccess=32
; else ;last transfer to bus

```

```

ELSEIF (mw[16][@AIRPORTTAZ@]>0 & mw[17][@AIRPORTTAZ@]=0 &
mw[18][@AIRPORTTAZ@]>0) ; Bus and Commuter Rail Only
;
wlmkarc=mw[13][@AIRPORTTAZ@]+mw[14][@AIRPORTTAZ@]+mw[15][@AIRPORTTAZ@]+mw[16][@AIRPORTTAZ@]+mw[17][@AIRPORTTAZ
@]+mw[18][@AIRPORTTAZ@]+MARCTPK+BusTPK[@AIRPORTTAZ@]+(mw[24][@AIRPORTTAZ@]/@valueoftime@)

wlmkarc=mw[13][@AIRPORTTAZ@]+mw[14][@AIRPORTTAZ@]+mw[15][@AIRPORTTAZ@]+mw[16][@AIRPORTTAZ@]+mw[17][@AIRPORTTAZ
@]+mw[18][@AIRPORTTAZ@]+(mw[24][@AIRPORTTAZ@]/@valueoftime@)
wlmkarcaccegress=31
endif
elseif(mw[19][@AIRPORTTAZ@]=4-5) ;first mode was rail access rail
if(mw[20][@AIRPORTTAZ@]=4000-4070) ;last transfer to MARC
;
wlmkarc=mw[13][@AIRPORTTAZ@]+mw[14][@AIRPORTTAZ@]+mw[15][@AIRPORTTAZ@]+mw[16][@AIRPORTTAZ@]+mw[17][@AIRPORTTAZ
@]+mw[18][@AIRPORTTAZ@]+RailTPK+MARCTPK[@AIRPORTTAZ@]+(mw[24][@AIRPORTTAZ@]/@valueoftime@)

wlmkarc=mw[13][@AIRPORTTAZ@]+mw[14][@AIRPORTTAZ@]+mw[15][@AIRPORTTAZ@]+mw[16][@AIRPORTTAZ@]+mw[17][@AIRPORTTAZ
@]+mw[18][@AIRPORTTAZ@]+(mw[24][@AIRPORTTAZ@]/@valueoftime@)
wlmkarcaccegress=23
elseif(mw[20][@AIRPORTTAZ@]=4105-4999) ;last transfer to rail
;
wlmkarc=mw[13][@AIRPORTTAZ@]+mw[14][@AIRPORTTAZ@]+mw[15][@AIRPORTTAZ@]+mw[16][@AIRPORTTAZ@]+mw[17][@AIRPORTTAZ
@]+mw[18][@AIRPORTTAZ@]+RailTPK+RailTPK[@AIRPORTTAZ@]+(mw[24][@AIRPORTTAZ@]/@valueoftime@)

wlmkarc=mw[13][@AIRPORTTAZ@]+mw[14][@AIRPORTTAZ@]+mw[15][@AIRPORTTAZ@]+mw[16][@AIRPORTTAZ@]+mw[17][@AIRPORTTAZ
@]+mw[18][@AIRPORTTAZ@]+(mw[24][@AIRPORTTAZ@]/@valueoftime@)
wlmkarcaccegress=22
else ;last transfer to bus
;
wlmkarc=mw[13][@AIRPORTTAZ@]+mw[14][@AIRPORTTAZ@]+mw[15][@AIRPORTTAZ@]+mw[16][@AIRPORTTAZ@]+mw[17][@AIRPORTTAZ
@]+mw[18][@AIRPORTTAZ@]+RailTPK+BusTPK[@AIRPORTTAZ@]+(mw[24][@AIRPORTTAZ@]/@valueoftime@)

wlmkarc=mw[13][@AIRPORTTAZ@]+mw[14][@AIRPORTTAZ@]+mw[15][@AIRPORTTAZ@]+mw[16][@AIRPORTTAZ@]+mw[17][@AIRPORTTAZ
@]+mw[18][@AIRPORTTAZ@]+(mw[24][@AIRPORTTAZ@]/@valueoftime@)
wlmkarcaccegress=21
endif
else ;first mode was bus access bus
if(mw[20][@AIRPORTTAZ@]=4000-4070) ;last transfer to MARC
;
wlmkarc=mw[13][@AIRPORTTAZ@]+mw[14][@AIRPORTTAZ@]+mw[15][@AIRPORTTAZ@]+mw[16][@AIRPORTTAZ@]+mw[17][@AIRPORTTAZ
@]+mw[18][@AIRPORTTAZ@]+BusTPK+MARCTPK[@AIRPORTTAZ@]+(mw[24][@AIRPORTTAZ@]/@valueoftime@)

wlmkarc=mw[13][@AIRPORTTAZ@]+mw[14][@AIRPORTTAZ@]+mw[15][@AIRPORTTAZ@]+mw[16][@AIRPORTTAZ@]+mw[17][@AIRPORTTAZ
@]+mw[18][@AIRPORTTAZ@]+(mw[24][@AIRPORTTAZ@]/@valueoftime@)
wlmkarcaccegress=13
elseif(mw[20][@AIRPORTTAZ@]=4105-4999) ;last transfer to rail
;
wlmkarc=mw[13][@AIRPORTTAZ@]+mw[14][@AIRPORTTAZ@]+mw[15][@AIRPORTTAZ@]+mw[16][@AIRPORTTAZ@]+mw[17][@AIRPORTTAZ
@]+mw[18][@AIRPORTTAZ@]+BusTPK+RailTPK[@AIRPORTTAZ@]+(mw[24][@AIRPORTTAZ@]/@valueoftime@)

wlmkarc=mw[13][@AIRPORTTAZ@]+mw[14][@AIRPORTTAZ@]+mw[15][@AIRPORTTAZ@]+mw[16][@AIRPORTTAZ@]+mw[17][@AIRPORTTAZ
@]+mw[18][@AIRPORTTAZ@]+(mw[24][@AIRPORTTAZ@]/@valueoftime@)
wlmkarcaccegress=12
else ;last transfer to bus
;
wlmkarc=mw[13][@AIRPORTTAZ@]+mw[14][@AIRPORTTAZ@]+mw[15][@AIRPORTTAZ@]+mw[16][@AIRPORTTAZ@]+mw[17][@AIRPORTTAZ
@]+mw[18][@AIRPORTTAZ@]+BusTPK+BusTPK[@AIRPORTTAZ@]+(mw[24][@AIRPORTTAZ@]/@valueoftime@)

wlmkarc=mw[13][@AIRPORTTAZ@]+mw[14][@AIRPORTTAZ@]+mw[15][@AIRPORTTAZ@]+mw[16][@AIRPORTTAZ@]+mw[17][@AIRPORTTAZ
@]+mw[18][@AIRPORTTAZ@]+(mw[24][@AIRPORTTAZ@]/@valueoftime@)
wlmkarcaccegress=11
endif
endif
endif

```

```

;get shortest total travel time for walk to transit
minwlktransit=min(wlkbus,wlkrail,wlkmarc)

;set travel time to zero for no transit service
if(minwlktransit=99999)
  minwlktransit=0
endif

;minwlktransitmode will contain which submode is the shorts 0-no service, 1-bus, 2-rail, 3-MARC
;minsubmode will contain which access and egress mode is used 12 - walk to bus/walk from rail
if(minwlktransit=0)
  minwlktransitmode=0
  minsubmode=0
elseif(minwlktransit=wlkbus)
  minwlktransitmode=1
  minsubmode=wlkbusacccess
elseif(minwlktransit=wlkrail)
  minwlktransitmode=2
  minsubmode=wlkrailacccess
elseif(minwlktransit=wlkmarc)
  minwlktransitmode=3
  minsubmode=wlkmarcacccess
endif

;walk time is access and egress based on sub mode 1-bus, 2-rail, 3-MARC (12 walk to bus/walk from rail)

if(minsubmode=0)
  trnwalk=0 ;no service
elseif(minsubmode=11) ; bus only
; trnwalk=BusTPK+BusTPK[@AIRPORTTAZ@]
  trnwalk=mw[4]
elseif(minsubmode=12) ; bus / MetroRail
; trnwalk=BusTPK+RailTPK[@AIRPORTTAZ@]
  trnwalk=mw[8]
elseif(minsubmode=13) ; bus / Commuter Rail
; trnwalk=BusTPK+MARCTPK[@AIRPORTTAZ@]
  trnwalk=mw[15]
elseif(minsubmode=21) ; MetroRail / bus
; trnwalk=RailTPK+BusTPK[@AIRPORTTAZ@]
  trnwalk=mw[8]
elseif(minsubmode=22) ; MetroRail only
; trnwalk=RailTPK+RailTPK[@AIRPORTTAZ@]
  trnwalk=mw[8]
elseif(minsubmode=23) ; MetroRail / Commuter Rail
; trnwalk=RailTPK+MARCTPK[@AIRPORTTAZ@]
  trnwalk=mw[15]
elseif(minsubmode=31)
; trnwalk=MARCTPK+BusTPK[@AIRPORTTAZ@]
  trnwalk=mw[15]
elseif(minsubmode=32)
; trnwalk=MARCTPK+RailTPK[@AIRPORTTAZ@]
  trnwalk=mw[15]
elseif(minsubmode=33)
; trnwalk=MARCTPK+MARCTPK[@AIRPORTTAZ@]
  trnwalk=mw[15]
endif

trnwait=0
trnfare=0
trnrun=0

;if(minwlktransitmode=1) ;Bus to Airport is shortest travel time
; trnwait=(mw[2][@AIRPORTTAZ@]+mw[3][@AIRPORTTAZ@])
; trnfare=mw[22][@AIRPORTTAZ@]
; trnrun=mw[5][@AIRPORTTAZ@]

```

```

;elseif(minwlktransitmode=2) ;Rail to Airport is shortest travel time
; trnwait=(mw[6][@AIRPORTTAZ@]+mw[7][@AIRPORTTAZ@])
; trnfare=mw[23][@AIRPORTTAZ@]
; trnrun=(mw[9][@AIRPORTTAZ@]+mw[10][@AIRPORTTAZ@])
;elseif(minwlktransitmode=3) ;MARC to Airport is shortest travel time
; trnwait=(mw[13][@AIRPORTTAZ@]+mw[14][@AIRPORTTAZ@])
; trnfare=mw[24][@AIRPORTTAZ@]
; trnrun=(mw[16][@AIRPORTTAZ@]+mw[17][@AIRPORTTAZ@]+mw[18][@AIRPORTTAZ@])
;endif

trntemp=mw[5][@AIRPORTTAZ@]
if (trntemp>0)
    trnrun=mw[5][@AIRPORTTAZ@]
    trntemp=mw[9][@AIRPORTTAZ@]+mw[10][@AIRPORTTAZ@]
    trnuse=1

    if (trntemp>0)
        trnrun=min(trnrun, mw[9][@AIRPORTTAZ@]+mw[10][@AIRPORTTAZ@])
        trndiff=trnrun-trntemp
        IF (trndiff=0)
            trnuse=2
        ENDIF

        ENDIF
        trntemp=mw[16][@AIRPORTTAZ@]+mw[17][@AIRPORTTAZ@]+mw[18][@AIRPORTTAZ@]
        if (trntemp>0)
            trnrun=min(trnrun, mw[16][@AIRPORTTAZ@]+mw[17][@AIRPORTTAZ@]+mw[18][@AIRPORTTAZ@])
            trndiff=trnrun-trntemp
            IF (trndiff=0)
                trnuse=3
            ENDIF

        ENDIF
        trntemp=0
    ENDIF

trntemp=mw[5][@AIRPORTTAZ@]
if (trntemp=0)
    trnrun=99999
    trntemp=mw[9][@AIRPORTTAZ@]+mw[10][@AIRPORTTAZ@]
    trnuse=1

    if (trntemp>0)
        trnrun=min(trnrun, mw[9][@AIRPORTTAZ@]+mw[10][@AIRPORTTAZ@])
        trndiff=trnrun-trntemp
        IF (trndiff=0)
            trnuse=2
        ENDIF

        ENDIF
        trntemp=mw[16][@AIRPORTTAZ@]+mw[17][@AIRPORTTAZ@]+mw[18][@AIRPORTTAZ@]
        if (trntemp>0)
            trnrun=min(trnrun, mw[16][@AIRPORTTAZ@]+mw[17][@AIRPORTTAZ@]+mw[18][@AIRPORTTAZ@])
            trndiff=trnrun-trntemp
            IF (trndiff=0)
                trnuse=3
            ENDIF

        ENDIF
        trntemp=0
    ENDIF

IF (trnuse=1) ; Bus Only
trnwait=(mw[2][@AIRPORTTAZ@]+mw[3][@AIRPORTTAZ@])/10.
trnfare=mw[22][@AIRPORTTAZ@]
trnrun=mw[5][@AIRPORTTAZ@]/10.
    trnwalk=mw[4][@AIRPORTTAZ@]/10.
ENDIF

IF (trnuse=2) ; Metrorail
trnwait=(mw[6][@AIRPORTTAZ@]+mw[7][@AIRPORTTAZ@])/10.
trnfare=mw[23][@AIRPORTTAZ@]
trnrun=(mw[9][@AIRPORTTAZ@]+mw[10][@AIRPORTTAZ@])/10.

```

```

trnwalk=mw[8][@AIRPORTTAZ@]/10.
ENDIF
IF (trnuse=3) ; Commuter Rail
trnwait=(mw[13][@AIRPORTTAZ@]+mw[14][@AIRPORTTAZ@])/10.
trnfare=mw[24][@AIRPORTTAZ@]
trnrun=(mw[16][@AIRPORTTAZ@]+mw[17][@AIRPORTTAZ@]+mw[18][@AIRPORTTAZ@])/10.
trnwalk=mw[15][@AIRPORTTAZ@]/10.
ENDIF

mw[200]=minwlktransitmode

hwytime=mw[21][@AIRPORTTAZ@]
hwydist=mw[1][@AIRPORTTAZ@]
hwycost=mw[1][@AIRPORTTAZ@]*@autocostmi@ ; cents per mile

;Parking cost: daily parking cost (long-term rate),
;multiplied by the business or pleasure average parking duration in days.
;http://www.bwiairport.com/directions_parking/parking_info/ Long Term Parking $8.00
; Long Term Parking
; BWI= $8.00 per day
PARKCOSTBWI = 8.00
; Dulles = $10.00 per day
PARKCOSTIAD = 10.00
; National = $12.00 per day
PARKCOSTDCA = 12.00

busparkdurBWI = 3.9 ; From 2009 Air Passenger Survey
plsparkdurBWI = 5.2 ; From 2009 Air Passenger Survey
busparkdurIAD = 4.0 ; Used to compute long-term parking cost (per "Air Passenger Model", page V-7)
plsparkdurIAD = 7.2 ; From 2009 Air Passenger Survey
busparkdurDCA = 3.9 ; From 2009 Air Passenger Survey
plsparkdurDCA = 5.6 ; From 2009 Air Passenger Survey

pcostb = parkcost@AIRPORT@ * busparkdur@AIRPORT@
pcostp = parkcost@AIRPORT@ * plsparkdur@AIRPORT@

;Taxi Fare and other assumptions
txrun = mw[21][@AIRPORTTAZ@] ;taxi run time is same as highway run time
txfare = taxifare ;taxi fare first 1/8 $2.90 thereafter $1.60 per mile

;shuttle fare

shfare=shuttlefare

shrun=mw[21][@AIRPORTTAZ@] * @circuitry@

;initialize holding variables

set val=0, vars=uds1, uds2, uds3, uds4,
udo1, udo2, udo3, udo4,
urc1, urc2, urc3, urc4,
utr1, utr2, utr3, utr4,
ush1, ush2, ush3, ush4,
utx1, utx2, utx3, utx4,

euds1, euds2, euds3, euds4,
eudo1, eudo2, eudo3, eudo4,
eurc1, eurc2, eurc3, eurc4,
eutr1, eutr2, eutr3, eutr4,
eush1, eush2, eush3, eush4,
eutx1, eutx2, eutx3, eutx4,

lsna1, lsna2, lsna3, lsna4,
lsau1, lsau2, lsau3, lsau4,

```

pds1, pds2, pds3, pds4,
 pdo1, pdo2, pdo3, pdo4,
 prc1, prc2, prc3, prc4,
 ptr1, ptr2, ptr3, ptr4,
 psh1, psh2, psh3, psh4,
 ptx1, ptx2, ptx3, ptx4

set val=0, vars=mshare11, mshare12, mshare13, mshare14, mshare15, mshare16,
 mshare21, mshare22, mshare23, mshare24, mshare25, mshare26,
 mshare31, mshare32, mshare33, mshare34, mshare35, mshare36,
 mshare41, mshare42, mshare43, mshare44, mshare45, mshare46,

uau1, uau2, uau3, uau4,
 euau1, euau2, euau3, euau4,
 una1, una2, una3, una4,
 euna1, euna2, euna3, euna4,

pau1, pau2, pau3, pau4,
 pna1, pna2, pna3, pna4,

cwalk, cwait, crun, cfare, ctxrun, ctxfare,
 caurun, caucost, cnanest, caunest, csrun, csfare,
 biasr, biasds, biasna, biassh, biasrc,

busresi1, busresi2, busresi3, busresi4,

busnresi1, busnresi2, busnresi3, busnresi4,
 nbusresi1, nbusresi2, nbusresi3, nbusresi4,
 nbusnresi1, nbusnresi2, nbusnresi3, nbusnresi4

array uds1=4, uds2=4, uds3=4, uds4=4,
 udo1=4, udo2=4, udo3=4, udo4=4,
 urc1=4, urc2=4, urc3=4, urc4=4,
 utr1=4, utr2=4, utr3=4, utr4=4,
 ush1=4, ush2=4, ush3=4, ush4=4,
 utx1=4, utx2=4, utx3=4, utx4=4,

euds1=4, euds2=4, euds3=4, euds4=4,
 eudo1=4, eudo2=4, eudo3=4, eudo4=4,
 eurc1=4, eurc2=4, eurc3=4, eurc4=4,
 eutr1=4, eutr2=4, eutr3=4, eutr4=4,
 eush1=4, eush2=4, eush3=4, eush4=4,
 eutx1=4, eutx2=4, eutx3=4, eutx4=4,

lsna1=4, lsna2=4, lsna3=4, lsna4=4,
 lsau1=4, lsau2=4, lsau3=4, lsau4=4,

pds1=4, pds2=4, pds3=4, pds4=4,
 pdo1=4, pdo2=4, pdo3=4, pdo4=4,
 prc1=4, prc2=4, prc3=4, prc4=4,
 ptr1=4, ptr2=4, ptr3=4, ptr4=4,
 psh1=4, psh2=4, psh3=4, psh4=4,
 ptx1=4, ptx2=4, ptx3=4, ptx4=4,

mshare11=4, mshare12=4, mshare13=4, mshare14=4, mshare15=4, mshare16=4,
 mshare21=4, mshare22=4, mshare23=4, mshare24=4, mshare25=4, mshare26=4,
 mshare31=4, mshare32=4, mshare33=4, mshare34=4, mshare35=4, mshare36=4,
 mshare41=4, mshare42=4, mshare43=4, mshare44=4, mshare45=4, mshare46=4,

uau1=4, uau2=4, uau3=4, uau4=4,
 euau1=4, euau2=4, euau3=4, euau4=4,
 una1=4, una2=4, una3=4, una4=4,
 euna1=4, euna2=4, euna3=4, euna4=4,

```

pau1=4, pau2=4, pau3=4, pau4=4,
pna1=4, pna2=4, pna3=4, pna4=4,

cwalk=4, cwait=4, crun=4, cfare=4, ctxrun=4, ctxfare=4,
caurun=4, caucost=4, cnanest=4, caunest=4, csrun=4, csfare=4,
biastr=4, biasds=4, biasna=4, biassh=4, biasrc=4,

; BusRes, BusNonRes, NonBusRes, NonBusNonRes - 4 income groups, 6 modes
      busresi1=6, busresi2=6, busresi3=6, busresi4=6,
      busnresi1=6, busnresi2=6, busnresi3=6, busnresi4=6,
nbusresi1=6, nbusresi2=6, nbusresi3=6, nbusresi4=6,
      nbusnresi1=6, nbusnresi2=6, nbusnresi3=6, nbusnresi4=6

; Bias coefficients: BIASmm(purpose1-4). mm = mode abbrev.

; Default coefficients, stratified by purpose (taken from the
; Washington, D.C. airport model):

cwalk[1]=-0.093
cwalk[2]=-0.089
cwalk[3]=-0.051
cwalk[4]=-0.045

cwait[1]=-0.107
cwait[2]=-0.096
cwait[3]=-0.077
cwait[4]=-0.071

crun[1]=-0.053
crun[2]=-0.050
crun[3]=-0.031
crun[4]=-0.029

cfare[1]=-0.00277
cfare[2]=-0.00256
cfare[3]=-1.04
cfare[4]=-0.973

ctxrun[1]=-0.071
ctxrun[2]=-0.068
ctxrun[3]=-0.044
ctxrun[4]=-0.039

ctxfare[1]=-0.00277
ctxfare[2]=-0.00256
ctxfare[3]=-1.04
ctxfare[4]=-0.973

caurun[1]=-0.071
caurun[2]=-0.068
caurun[3]=-0.044
caurun[4]=-0.039

caucost[1]=-0.00277
caucost[2]=-0.00256
caucost[3]=-1.04
caucost[4]=-0.973

cnanest[1]=0.30
cnanest[2]=0.30
cnanest[3]=0.30
cnanest[4]=0.30

caunest[1]=0.30
caunest[2]=1.00
caunest[3]=0.30

```

caunest[4]=1.00

;In the original model, shuttle was only available for non-residents.
 ;In the Baltimore application, shuttle is being extended to residents
 ;using the same system coefficients as non-residents.

csrun[1]=-0.071
 csrun[2]=-0.089
 csrun[3]=-0.044
 csrun[4]=-0.045

csfare[1]=-0.00277
 csfare[2]=-0.00256
 csfare[3]=-1.04
 csfare[4]=-0.973

biastrBWI1=-0.55+8.1 ; Bus Res
 biastrBWI2=-2.19+5.4 ; Bus NonRes
 biastrBWI3=-0.10+8.1 ; NonBus Res
 biastrBWI4=-0.38+5.4 ; NonBus NonRes
 biastrDCA1=-0.55+5.7 ; Bus Res
 biastrDCA2=-2.19+6.8 ; Bus NonRes
 biastrDCA3=-0.10+5.7 ; NonBus Res
 biastrDCA4=-0.38+6.8 ; NonBus NonRes
 biastrIAD1=-0.55+5.7 ; Bus Res
 biastrIAD2=-2.19+6.8 ; Bus NonRes
 biastrIAD3=-0.10+5.7 ; NonBus Res
 biastrIAD4=-0.38+6.8 ; NonBus NonRes
 biastradjBWI = 1.0
 biastradjDCA = 1.0
 biastradjIAD = 1.0

biasdsBWI1=0.27+0.0
 biasdsBWI2=0.00
 biasdsBWI3=0.22+0.0
 biasdsBWI4=0.00
 biasdsDCA1=0.27+1.9
 biasdsDCA2=0.00
 biasdsDCA3=0.22+1.9
 biasdsDCA4=0.00
 biasdsIAD1=0.27+0.5
 biasdsIAD2=0.00
 biasdsIAD3=0.22+0.5
 biasdsIAD4=0.00
 biasadjdsBWI = 1.0
 biasadjdsDCA = 1.0
 biasadjdsIAD = 1.0

biasadjdoBWI = 0.0
 biasadjdoDCA = 0.0
 biasadjdoIAD = 0.0

biasshBWI1=3.52+2.9
 biasshBWI2=4.21+0.3
 biasshBWI3=2.29+2.9
 biasshBWI4=2.28+0.3
 biasshDCA1=3.52-1.5
 biasshDCA2=4.21+1.9
 biasshDCA3=2.29-1.5
 biasshDCA4=2.28+1.9
 biasshIAD1=3.52+0.0
 biasshIAD2=4.21+3.9
 biasshIAD3=2.29+0.0
 biasshIAD4=2.28+3.9

biastxBWI1=3.52-4.4


```

biastxBWI2=4.21-1.6
biastxBWI3=2.29-4.4
biastxBWI4=2.28-1.6
biastxDCA1=3.52+1.7
biastxDCA2=4.21+0.8
biastxDCA3=2.29+1.7
biastxDCA4=2.28+0.8
biastxIAD1=3.52+0.0
biastxIAD2=4.21-4.6
biastxIAD3=2.29+0.0
biastxIAD4=2.28-4.6

```

```

biastxadjBWI = 1.00
biastxadjDCA = 1.00
biastxadjIAD = 1.00
biasshadjBWI = 1.00
biasshadjDCA = 1.00
biasshadjIAD = 1.00

```

```

biasrcBWI1=0.00
biasrcBWI2=-0.80-4.1
biasrcBWI3=0.00
biasrcBWI4=-1.70-4.1
biasrcDCA1=0.00
biasrcDCA2=-0.80-5.1
biasrcDCA3=0.00
biasrcDCA4=-1.70-5.1
biasrcIAD1=0.00
biasrcIAD2=-0.80-8.45
biasrcIAD3=0.00
biasrcIAD4=-1.70-8.45

```

```

biasadjrcBWI = 1.00 ; rental car bias term
biasadjrcDCA = 1.00 ; rental car bias term
biasadjrcIAD = 1.00 ; rental car bias term

```

```

biasnaBWI1=1.46-2.77
biasnaBWI2=2.40-0.15
biasnaBWI3=0.96-2.77
biasnaBWI4=0.48-0.15
biasnaDCA1=1.46-0.1
biasnaDCA2=2.40+1.8
biasnaDCA3=0.96-0.1
biasnaDCA4=0.48+1.8
biasnaIAD1=1.46-1.25
biasnaIAD2=2.40+1.07
biasnaIAD3=0.96-1.25
biasnaIAD4=0.48+1.07

```

```

biasadjnaBWI = 1.00 ; non-auto bias
biasadjnaDCA = 1.00 ; non-auto bias
biasadjnaIAD = 1.00 ; non-auto bias

```

```

;Calculate utilities, by purpose and income, while checking
; modal availability. Compare exponentiated utilities to KONN
; to protect against impossibly tiny values.

```

```

;Start with the Non-Auto nest.
;Transit. Only if run time > 0.

```

```

konn = pow(10,-40) ; a very, very small positive number

```

```

if(trnrun>0)
  loop inc=1,4

```

```

; utr1[inc]=(cwalk[1] * trnwalk + cwait[1] * trnwait +
utr1[inc]=(0.00 * cwalk[1] * trnwalk + 0.5 * cwait[1] * trnwait +
  crun[1] * trnrun + cfare[1] * trnfare +

```

```

        biastradj@AIRPORT@ * biastr@AIRPORT@1)/cnanest[1]
    eutr1[inc]=max(exp(utr1[inc]),konn)

;   utr2[inc]=(cwalk[2] * trnwalk + cwait[2] * trnwait +
    utr2[inc]=(0.00 * cwalk[2] * trnwalk + 0.5 * cwait[2] * trnwait +
        crun[2] * trnrun + cfare[2] * trnfare +
        biastradj@AIRPORT@ * biastr@AIRPORT@2)/cnanest[2]
    eutr2[inc]=max(exp(utr2[inc]),konn)

;   utr3[inc]=(cwalk[3] * trnwalk + cwait[3] * trnwait +
    utr3[inc]=(0.00 * cwalk[3] * trnwalk + 0.5 * cwait[3] * trnwait +
        crun[3] * trnrun + cfare[3] * trnfare/finc[inc] +
        biastradj@AIRPORT@ * biastr@AIRPORT@3)/cnanest[3]
    eutr3[inc]=max(exp(utr3[inc]),konn)

;   utr4[inc]=(cwalk[4] * trnwalk + cwait[4] * trnwait +
    utr4[inc]=(0.00 * cwalk[4] * trnwalk + 0.5 * cwait[4] * trnwait +
        crun[4] * trnrun + cfare[4] * trnfare/finc[inc] +
        biastradj@AIRPORT@ * biastr@AIRPORT@4)/cnanest[4]
    eutr4[inc]=max(exp(utr4[inc]),konn)
endloop
endif

;Taxi - Always available

loop inc=1,4
    utx1[inc]=(ctxrun[1]*txrun + ctxfare[1] * txfare + biastxadj@AIRPORT@ * biastx@AIRPORT@1) / cnanest[1]
    eutx1[inc]=max(exp(utx1[inc]),konn)

    utx2[inc]=(ctxrun[2]*txrun + ctxfare[2] * txfare + biastxadj@AIRPORT@ * biastx@AIRPORT@2) / cnanest[2]
    eutx2[inc]=max(exp(utx2[inc]),konn)

    utx3[inc]=(ctxrun[3]*txrun + ctxfare[3] * txfare / finc[inc] + biastxadj@AIRPORT@ * biastx@AIRPORT@3) / cnanest[3]
    eutx3[inc]=max(exp(utx3[inc]),konn)

    utx4[inc]=(ctxrun[4]*txrun + ctxfare[4] * txfare / finc[inc] + biastxadj@AIRPORT@ * biastx@AIRPORT@4) / cnanest[4]
    eutx4[inc]=max(exp(utx4[inc]),konn)
endloop

;Shuttle - Always available

loop inc=1,4

    ush1[inc]=(csrunk[1] * shrunk + csfare[1] * shfare + biasshadj@AIRPORT@ * biassh@AIRPORT@1) / cnanest[1]
    eush1[inc]=max(exp(ush1[inc]),konn)

    ush2[inc]=(csrunk[2] * shrunk + csfare[2] * shfare + biasshadj@AIRPORT@ * biassh@AIRPORT@2) / cnanest[2]
    eush2[inc]=max(exp(ush2[inc]),konn)

    ush3[inc]=(csrunk[3] * shrunk + csfare[3] * shfare / finc[inc] + biasshadj@AIRPORT@ * biassh@AIRPORT@3) / cnanest[3]
    eush3[inc]=max(exp(ush3[inc]),konn)

    ush4[inc]=(csrunk[4] * shrunk + csfare[4] * shfare / finc[inc] + biasshadj@AIRPORT@ * biassh@AIRPORT@4) / cnanest[4]
    eush4[inc]=max(exp(ush4[inc]),konn)

endloop

;Finish the non-auto nest. First compute the probabilities within the nest

loop inc=1,4
    toteu1 = eutr1[inc] + eutx1[inc] + eush1[inc]
    if(toteu1>konn)
        ptr1[inc] = eutr1[inc]/toteu1
        ptx1[inc] = eutx1[inc]/toteu1
    endif
endloop

```

```

    psh1[inc] = eush1[inc]/toteu1
    lsna1[inc] = log(toteu1)
endif

toteu2 = eutr2[inc] + eutx2[inc] + eush2[inc]
if(toteu2>konn)
    ptr2[inc] = eutr2[inc]/toteu2
    ptx2[inc] = eutx2[inc]/toteu2
    psh2[inc] = eush2[inc]/toteu2
    lsna2[inc] = log(toteu2)
endif

toteu3 = eutr3[inc] + eutx3[inc] + eush3[inc]
if(toteu3>konn)
    ptr3[inc] = eutr3[inc]/toteu3
    ptx3[inc] = eutx3[inc]/toteu3
    psh3[inc] = eush3[inc]/toteu3
    lsna3[inc] = log(toteu3)
endif

toteu4 = eutr4[inc] + eutx4[inc] + eush4[inc]
if(toteu4>konn)
    ptr4[inc] = eutr4[inc]/toteu4
    ptx4[inc] = eutx4[inc]/toteu4
    psh4[inc] = eush4[inc]/toteu4
    lsna4[inc] = log(toteu4)
endif
endloop

;now do the auto nest
;drive self - only for purposes 1 and 3 (res). Always available.

loop inc=1,4
    uds1[inc] = (caurun[1] * hwytime + biasadjds@AIRPORT@ * biasds@AIRPORT@1 + caucost[1] * (hwycost + pcostb)) / caunest[1]
    euds1[inc] = max(exp(uds1[inc]),konn)

    uds3[inc] = (caurun[3] * hwytime + biasadjds@AIRPORT@ * biasds@AIRPORT@3 + caucost[3] * (hwycost + pcostp) / finc[inc]) / caunest[3]
    euds3[inc] = max(exp(uds3[inc]),konn)
endloop

;dropped off

loop inc=1,4

    udo1[inc] = (caurun[1] * hwytime + caucost[1] * hwycost + biasadjdo@AIRPORT@) / caunest[1]
    eudo1[inc] = max(exp(udo1[inc]), konn)

    udo2[inc] = (caurun[2] * hwytime + caucost[2] * hwycost + biasadjdo@AIRPORT@) / caunest[2]
    eudo2[inc] = max(exp(udo2[inc]), konn)

    udo3[inc] = (caurun[3] * hwytime + caucost[3] * hwycost / finc[inc] + biasadjdo@AIRPORT@) / caunest[3]
    eudo3[inc] = max(exp(udo3[inc]), konn)

    udo4[inc] = (caurun[4] * hwytime + caucost[4] * hwycost / finc[inc] + biasadjdo@AIRPORT@) / caunest[4]
    eudo4[inc] = max(exp(udo4[inc]), konn)
endloop

;Finish the Auto nest. First compute the probabilities within the nest.

loop inc=1,4
    toteu1 = euds1[inc] + eudo1[inc]
    if(toteu1>konn)
        pds1[inc] = euds1[inc]/toteu1
        pdo1[inc] = eudo1[inc]/toteu1
        lsau1[inc] = log(toteu1)
    endif
endloop

```

```

toteu2 = euds2[inc] + eudo2[inc]
if(toteu2>konn)
  pds2[inc] = euds2[inc]/toteu2
  pdo2[inc] = eudo2[inc]/toteu2
  lsau2[inc] = log(toteu2)
endif
toteu3 = euds3[inc] + eudo3[inc]
if(toteu3>konn)
  pds3[inc] = euds3[inc]/toteu3
  pdo3[inc] = eudo3[inc]/toteu3
  lsau3[inc] = log(toteu3)
endif
toteu4 = euds4[inc] + eudo4[inc]
if(toteu4>konn)
  pds4[inc] = euds4[inc]/toteu4
  pdo4[inc] = eudo4[inc]/toteu4
  lsau4[inc] = log(toteu4)
endif
endloop

```

; Back to the top level. Compute the Rental Car utile and
; finish the mode choice calculations. (Right now, the Rental Car
; utile consists only of a constant term. Thus, it is inefficient
; to re-calculate it for each O/D pair. But: this could change in
; the future, with the possible addition of some O/D- or D- specific
; terms. So, leave it as it is for now.)
; Top level is the choice among Non-Auto, Auto, and Rental Car (for
; non-residents). Zero out Rental Car for the Resident trips.

```

loop inc=1,4
;purpose 1 & 3
uau1[inc] = caunest[1] * lsau1[inc]
euau1[inc] = max(exp(uau1[inc]),konn)

una1[inc] = cnanest[1] * lsna1[inc] + biasadjna@AIRPORT@ * biasna@AIRPORT@1
euna1[inc] = max(exp(una1[inc]),konn)

eutot1 = euau1[inc] + euna1[inc]
if(eutot1>konn)
  pau1[inc] = euau1[inc]/eutot1
  pna1[inc] = euna1[inc]/eutot1
endif

uau3[inc] = caunest[3] * lsau3[inc]
euau3[inc] = max(exp(uau3[inc]),konn)

una3[inc] = cnanest[3] * lsna3[inc] + biasadjna@AIRPORT@ * biasna@AIRPORT@3
euna3[inc] = max(exp(una3[inc]),konn)

eutot3 = euau3[inc] + euna3[inc]
if(eutot3>konn)
  pau3[inc] = euau3[inc]/eutot3
  pna3[inc] = euna3[inc]/eutot3
endif
;purpose 2 & 4
uau2[inc] = caunest[2] * lsau2[inc]
euau2[inc] = max(exp(uau2[inc]),konn)

urc2[inc] = biasadjrc@AIRPORT@ * biasrc@AIRPORT@2
eurc2[inc] = max(exp(biasadjrc@AIRPORT@ * biasrc@AIRPORT@2),konn)

una2[inc] = cnanest[2] * lsna2[inc] + biasadjna@AIRPORT@ * biasna@AIRPORT@2
euna2[inc] = max(exp(una2[inc]),konn)

eutot2 = euau2[inc] + eurc2[inc] + euna2[inc]

```

```

if(eutot2>konn)
  pau2[inc] = euau2[inc]/eutot2
  prc2[inc] = eurc2[inc]/eutot2
  pna2[inc] = euna2[inc]/eutot2
endif

uau4[inc] = caunest[4] * lsau4[inc]
euau4[inc] = max(exp(uau4[inc]),konn)

urc4[inc] = biasadjrc@AIRPORT@ * biasrc@AIRPORT@4
eurc4[inc] = max(exp(biasadjrc@AIRPORT@ * biasrc@AIRPORT@4),konn)

una4[inc] = cnanest[4] * lsna4[inc] + biasadjna@AIRPORT@ * biasna@AIRPORT@4
euna4[inc] = max(exp(una4[inc]),konn)

eutot4 = euau4[inc] + eurc4[inc] + euna4[inc]
if(eutot4>konn)
  pau4[inc] = euau4[inc]/eutot4
  prc4[inc] = eurc4[inc]/eutot4
  pna4[inc] = euna4[inc]/eutot4
endif
endloop

;calculate final modal share
;mshare is by purpose [4], mode (6) and income [4]

loop inc=1,4
  mshare11[inc] = pau1[inc] * pds1[inc] ;drive self
  mshare12[inc] = pau1[inc] * pdo1[inc] ;dropped off
  mshare13[inc] = prc1[inc] ;rental car
  mshare14[inc] = pna1[inc] * ptr1[inc] ;transit
  mshare15[inc] = pna1[inc] * psh1[inc] ;shuttle
  mshare16[inc] = pna1[inc] * ptx1[inc] ;taxi

  mshare21[inc] = pau2[inc] * pds2[inc] ;drive self
  mshare22[inc] = pau2[inc] * pdo2[inc] ;dropped off
  mshare23[inc] = prc2[inc] ;rental car
  mshare24[inc] = pna2[inc] * ptr2[inc] ;transit
  mshare25[inc] = pna2[inc] * psh2[inc] ;shuttle
  mshare26[inc] = pna2[inc] * ptx2[inc] ;taxi

  mshare31[inc] = pau3[inc] * pds3[inc] ;drive self
  mshare32[inc] = pau3[inc] * pdo3[inc] ;dropped off
  mshare33[inc] = prc3[inc] ;rental car
  mshare34[inc] = pna3[inc] * ptr3[inc] ;transit
  mshare35[inc] = pna3[inc] * psh3[inc] ;shuttle
  mshare36[inc] = pna3[inc] * ptx3[inc] ;taxi

  mshare41[inc] = pau4[inc] * pds4[inc] ;drive self
  mshare42[inc] = pau4[inc] * pdo4[inc] ;dropped off
  mshare43[inc] = prc4[inc] ;rental car
  mshare44[inc] = pna4[inc] * ptr4[inc] ;transit
  mshare45[inc] = pna4[inc] * psh4[inc] ;shuttle
  mshare46[inc] = pna4[inc] * ptx4[inc] ;taxi
endloop

;Calculate modal trips.
;Store in output files busres.trp MW[41], mw[42], mw[43], mw[44], busnonres.trp MW[151], pleares.trp MW[161], pleanonres.trp MW[171]
;six purposes DA,DROFF,RC,Tran, Shuttle, and Taxi (1-6)
; These output trips are now stored in 16 arrays (4 trip purpose x 4 income groups) rather than matrices. Each array has six elements
; corresponding to the six mode options

;business residential income 1-4

;mw[41][@AIRPORTTAZ@]=attrbusresinc1*mshare11[1]
;mw[42][@AIRPORTTAZ@]=attrbusresinc1*mshare12[1]

```

```
;mw[43][@AIRPORTTAZ@]=attrbusresinc1*mshare13[1]
;mw[44][@AIRPORTTAZ@]=attrbusresinc1*mshare14[1]
;mw[45][@AIRPORTTAZ@]=attrbusresinc1*mshare15[1]
;mw[46][@AIRPORTTAZ@]=attrbusresinc1*mshare16[1]
```

```
busresi1[1] = attrbusresinc1*mshare11[1] ; Drive Alone
busresi1[2] = attrbusresinc1*mshare12[1] ; Drop Off
busresi1[3] = attrbusresinc1*mshare13[1] ; Rental Car
busresi1[4] = attrbusresinc1*mshare14[1] ; Transit
busresi1[5] = attrbusresinc1*mshare15[1] ; Shuttle
busresi1[6] = attrbusresinc1*mshare16[1] ; Taxi
```

```
;mw[47][@AIRPORTTAZ@]=attrbusresinc2*mshare11[2]
;mw[48][@AIRPORTTAZ@]=attrbusresinc2*mshare12[2]
;mw[49][@AIRPORTTAZ@]=attrbusresinc2*mshare13[2]
;mw[50][@AIRPORTTAZ@]=attrbusresinc2*mshare14[2]
;mw[51][@AIRPORTTAZ@]=attrbusresinc2*mshare15[2]
;mw[52][@AIRPORTTAZ@]=attrbusresinc2*mshare16[2]
```

```
busresi2[1] = attrbusresinc2*mshare11[2] ; Drive Alone
busresi2[2] = attrbusresinc2*mshare12[2] ; Drop Off
busresi2[3] = attrbusresinc2*mshare13[2] ; Rental Car
busresi2[4] = attrbusresinc2*mshare14[2] ; Transit
busresi2[5] = attrbusresinc2*mshare15[2] ; Shuttle
busresi2[6] = attrbusresinc2*mshare16[2] ; Taxi
```

```
;mw[53][@AIRPORTTAZ@]=attrbusresinc3*mshare11[3]
;mw[54][@AIRPORTTAZ@]=attrbusresinc3*mshare12[3]
;mw[55][@AIRPORTTAZ@]=attrbusresinc3*mshare13[3]
;mw[56][@AIRPORTTAZ@]=attrbusresinc3*mshare14[3]
;mw[57][@AIRPORTTAZ@]=attrbusresinc3*mshare15[3]
;mw[58][@AIRPORTTAZ@]=attrbusresinc3*mshare16[3]
```

```
busresi3[1] = attrbusresinc3*mshare11[3] ; Drive Alone
busresi3[2] = attrbusresinc3*mshare12[3] ; Drop Off
busresi3[3] = attrbusresinc3*mshare13[3] ; Rental Car
busresi3[4] = attrbusresinc3*mshare14[3] ; Transit
busresi3[5] = attrbusresinc3*mshare15[3] ; Shuttle
busresi3[6] = attrbusresinc3*mshare16[3] ; Taxi
```

```
;mw[59][@AIRPORTTAZ@]=attrbusresinc4*mshare11[4]
;mw[60][@AIRPORTTAZ@]=attrbusresinc4*mshare12[4]
;mw[61][@AIRPORTTAZ@]=attrbusresinc4*mshare13[4]
;mw[62][@AIRPORTTAZ@]=attrbusresinc4*mshare14[4]
;mw[63][@AIRPORTTAZ@]=attrbusresinc4*mshare15[4]
;mw[64][@AIRPORTTAZ@]=attrbusresinc4*mshare16[4]
```

```
busresi4[1] = attrbusresinc4*mshare11[4] ; Drive Alone
busresi4[2] = attrbusresinc4*mshare12[4] ; Drop Off
busresi4[3] = attrbusresinc4*mshare13[4] ; Rental Car
busresi4[4] = attrbusresinc4*mshare14[4] ; Transit
busresi4[5] = attrbusresinc4*mshare15[4] ; Shuttle
busresi4[6] = attrbusresinc4*mshare16[4] ; Taxi
```

;business non-residential

```
;mw[65][@AIRPORTTAZ@]=attrbusnresinc1*mshare21[1]
;mw[66][@AIRPORTTAZ@]=attrbusnresinc1*mshare22[1]
;mw[67][@AIRPORTTAZ@]=attrbusnresinc1*mshare23[1]
;mw[68][@AIRPORTTAZ@]=attrbusnresinc1*mshare24[1]
;mw[69][@AIRPORTTAZ@]=attrbusnresinc1*mshare25[1]
;mw[70][@AIRPORTTAZ@]=attrbusnresinc1*mshare26[1]
```

```
busnresi1[1] = attrbusnresinc1*mshare21[1] ; Drive Alone
busnresi1[2] = attrbusnresinc1*mshare22[1] ; Drop Off
busnresi1[3] = attrbusnresinc1*mshare23[1] ; Rental Car
busnresi1[4] = attrbusnresinc1*mshare24[1] ; Transit
```

busnresi1[5] = attrbusresinc1*mshare25[1] ; Shuttle
 busnresi1[6] = attrbusresinc1*mshare26[1] ; Taxi

;mw[71][@AIRPORTTAZ@]=attrbusresinc2*mshare21[2]
 ;mw[72][@AIRPORTTAZ@]=attrbusresinc2*mshare22[2]
 ;mw[73][@AIRPORTTAZ@]=attrbusresinc2*mshare23[2]
 ;mw[74][@AIRPORTTAZ@]=attrbusresinc2*mshare24[2]
 ;mw[75][@AIRPORTTAZ@]=attrbusresinc2*mshare25[2]
 ;mw[76][@AIRPORTTAZ@]=attrbusresinc2*mshare26[2]

busnresi2[1] = attrbusresinc2*mshare21[2] ; Drive Alone
 busnresi2[2] = attrbusresinc2*mshare22[2] ; Drop Off
 busnresi2[3] = attrbusresinc2*mshare23[2] ; Rental Car
 busnresi2[4] = attrbusresinc2*mshare24[2] ; Transit
 busnresi2[5] = attrbusresinc2*mshare25[2] ; Shuttle
 busnresi2[6] = attrbusresinc2*mshare26[2] ; Taxi

;mw[77][@AIRPORTTAZ@]=attrbusresinc3*mshare21[3]
 ;mw[78][@AIRPORTTAZ@]=attrbusresinc3*mshare22[3]
 ;mw[79][@AIRPORTTAZ@]=attrbusresinc3*mshare23[3]
 ;mw[80][@AIRPORTTAZ@]=attrbusresinc3*mshare24[3]
 ;mw[81][@AIRPORTTAZ@]=attrbusresinc3*mshare25[3]
 ;mw[82][@AIRPORTTAZ@]=attrbusresinc3*mshare26[3]

busnresi3[1] = attrbusresinc3*mshare21[3] ; Drive Alone
 busnresi3[2] = attrbusresinc3*mshare22[3] ; Drop Off
 busnresi3[3] = attrbusresinc3*mshare23[3] ; Rental Car
 busnresi3[4] = attrbusresinc3*mshare24[3] ; Transit
 busnresi3[5] = attrbusresinc3*mshare25[3] ; Shuttle
 busnresi3[6] = attrbusresinc3*mshare26[3] ; Taxi

;mw[83][@AIRPORTTAZ@]=attrbusresinc4*mshare21[4]
 ;mw[84][@AIRPORTTAZ@]=attrbusresinc4*mshare22[4]
 ;mw[85][@AIRPORTTAZ@]=attrbusresinc4*mshare23[4]
 ;mw[86][@AIRPORTTAZ@]=attrbusresinc4*mshare24[4]
 ;mw[87][@AIRPORTTAZ@]=attrbusresinc4*mshare25[4]
 ;mw[88][@AIRPORTTAZ@]=attrbusresinc4*mshare26[4]

busnresi4[1] = attrbusresinc4*mshare21[4] ; Drive Alone
 busnresi4[2] = attrbusresinc4*mshare22[4] ; Drop Off
 busnresi4[3] = attrbusresinc4*mshare23[4] ; Rental Car
 busnresi4[4] = attrbusresinc4*mshare24[4] ; Transit
 busnresi4[5] = attrbusresinc4*mshare25[4] ; Shuttle
 busnresi4[6] = attrbusresinc4*mshare26[4] ; Taxi

;pleasure residents

;mw[89][@AIRPORTTAZ@]=attrplsresinc1*mshare31[1]
 ;mw[90][@AIRPORTTAZ@]=attrplsresinc1*mshare32[1]
 ;mw[91][@AIRPORTTAZ@]=attrplsresinc1*mshare33[1]
 ;mw[92][@AIRPORTTAZ@]=attrplsresinc1*mshare34[1]
 ;mw[93][@AIRPORTTAZ@]=attrplsresinc1*mshare35[1]
 ;mw[94][@AIRPORTTAZ@]=attrplsresinc1*mshare36[1]

nbusresi1[1] = attrplsresinc1*mshare31[1] ; Drive Alone
 nbusresi1[2] = attrplsresinc1*mshare32[1] ; Drop Off
 nbusresi1[3] = attrplsresinc1*mshare33[1] ; Rental Car
 nbusresi1[4] = attrplsresinc1*mshare34[1] ; Transit
 nbusresi1[5] = attrplsresinc1*mshare35[1] ; Shuttle
 nbusresi1[6] = attrplsresinc1*mshare36[1] ; Taxi

;mw[95][@AIRPORTTAZ@]=attrplsresinc2*mshare31[2]
 ;mw[96][@AIRPORTTAZ@]=attrplsresinc2*mshare32[2]
 ;mw[97][@AIRPORTTAZ@]=attrplsresinc2*mshare33[2]
 ;mw[98][@AIRPORTTAZ@]=attrplsresinc2*mshare34[2]
 ;mw[99][@AIRPORTTAZ@]=attrplsresinc2*mshare35[2]
 ;mw[100][@AIRPORTTAZ@]=attrplsresinc2*mshare36[2]

nbusresi2[1] = attrplsresinc2*mshare31[2] ; Drive Alone
 nbusresi2[2] = attrplsresinc2*mshare32[2] ; Drop Off
 nbusresi2[3] = attrplsresinc2*mshare33[2] ; Rental Car
 nbusresi2[4] = attrplsresinc2*mshare34[2] ; Transit
 nbusresi2[5] = attrplsresinc2*mshare35[2] ; Shuttle
 nbusresi2[6] = attrplsresinc2*mshare36[2] ; Taxi

;mw[101][@AIRPORTTAZ@]=attrplsresinc3*mshare31[3]
 ;mw[102][@AIRPORTTAZ@]=attrplsresinc3*mshare32[3]
 ;mw[103][@AIRPORTTAZ@]=attrplsresinc3*mshare33[3]
 ;mw[104][@AIRPORTTAZ@]=attrplsresinc3*mshare34[3]
 ;mw[105][@AIRPORTTAZ@]=attrplsresinc3*mshare35[3]
 ;mw[106][@AIRPORTTAZ@]=attrplsresinc3*mshare36[3]

nbusresi3[1] = attrplsresinc3*mshare31[3] ; Drive Alone
 nbusresi3[2] = attrplsresinc3*mshare32[3] ; Drop Off
 nbusresi3[3] = attrplsresinc3*mshare33[3] ; Rental Car
 nbusresi3[4] = attrplsresinc3*mshare34[3] ; Transit
 nbusresi3[5] = attrplsresinc3*mshare35[3] ; Shuttle
 nbusresi3[6] = attrplsresinc3*mshare36[3] ; Taxi

;mw[107][@AIRPORTTAZ@]=attrplsresinc4*mshare31[4]
 ;mw[108][@AIRPORTTAZ@]=attrplsresinc4*mshare32[4]
 ;mw[109][@AIRPORTTAZ@]=attrplsresinc4*mshare33[4]
 ;mw[110][@AIRPORTTAZ@]=attrplsresinc4*mshare34[4]
 ;mw[111][@AIRPORTTAZ@]=attrplsresinc4*mshare35[4]
 ;mw[112][@AIRPORTTAZ@]=attrplsresinc4*mshare36[4]

nbusresi4[1] = attrplsresinc4*mshare31[4] ; Drive Alone
 nbusresi4[2] = attrplsresinc4*mshare32[4] ; Drop Off
 nbusresi4[3] = attrplsresinc4*mshare33[4] ; Rental Car
 nbusresi4[4] = attrplsresinc4*mshare34[4] ; Transit
 nbusresi4[5] = attrplsresinc4*mshare35[4] ; Shuttle
 nbusresi4[6] = attrplsresinc4*mshare36[4] ; Taxi

;pleasure non-residents
 ;mw[113][@AIRPORTTAZ@]=attrplsresinc1*mshare41[1]
 ;mw[114][@AIRPORTTAZ@]=attrplsresinc1*mshare42[1]
 ;mw[115][@AIRPORTTAZ@]=attrplsresinc1*mshare43[1]
 ;mw[116][@AIRPORTTAZ@]=attrplsresinc1*mshare44[1]
 ;mw[117][@AIRPORTTAZ@]=attrplsresinc1*mshare45[1]
 ;mw[118][@AIRPORTTAZ@]=attrplsresinc1*mshare46[1]

nbusnresi1[1] = attrplsresinc1*mshare41[1] ; Drive Alone
 nbusnresi1[2] = attrplsresinc1*mshare42[1] ; Drop Off
 nbusnresi1[3] = attrplsresinc1*mshare43[1] ; Rental Car
 nbusnresi1[4] = attrplsresinc1*mshare44[1] ; Transit
 nbusnresi1[5] = attrplsresinc1*mshare45[1] ; Shuttle
 nbusnresi1[6] = attrplsresinc1*mshare46[1] ; Taxi

;mw[119][@AIRPORTTAZ@]=attrplsresinc2*mshare41[2]
 ;mw[120][@AIRPORTTAZ@]=attrplsresinc2*mshare42[2]
 ;mw[121][@AIRPORTTAZ@]=attrplsresinc2*mshare43[2]
 ;mw[122][@AIRPORTTAZ@]=attrplsresinc2*mshare44[2]
 ;mw[123][@AIRPORTTAZ@]=attrplsresinc2*mshare45[2]
 ;mw[124][@AIRPORTTAZ@]=attrplsresinc2*mshare46[2]

nbusnresi2[1] = attrplsresinc2*mshare41[2] ; Drive Alone
 nbusnresi2[2] = attrplsresinc2*mshare42[2] ; Drop Off
 nbusnresi2[3] = attrplsresinc2*mshare43[2] ; Rental Car
 nbusnresi2[4] = attrplsresinc2*mshare44[2] ; Transit
 nbusnresi2[5] = attrplsresinc2*mshare45[2] ; Shuttle
 nbusnresi2[6] = attrplsresinc2*mshare46[2] ; Taxi

;mw[125][@AIRPORTTAZ@]=attrplsresinc3*mshare41[3]


```

;mw[126][@AIRPORTTAZ@]=attrplsresinc3*mshare42[3]
;mw[127][@AIRPORTTAZ@]=attrplsresinc3*mshare43[3]
;mw[128][@AIRPORTTAZ@]=attrplsresinc3*mshare44[3]
;mw[129][@AIRPORTTAZ@]=attrplsresinc3*mshare45[3]
;mw[130][@AIRPORTTAZ@]=attrplsresinc3*mshare46[3]

nbusnresi3[1] = attrplsresinc3*mshare41[3] ; Drive Alone
nbusnresi3[2] = attrplsresinc3*mshare42[3] ; Drop Off
nbusnresi3[3] = attrplsresinc3*mshare43[3] ; Rental Car
nbusnresi3[4] = attrplsresinc3*mshare44[3] ; Transit
nbusnresi3[5] = attrplsresinc3*mshare45[3] ; Shuttle
nbusnresi3[6] = attrplsresinc3*mshare46[3] ; Taxi

;mw[131][@AIRPORTTAZ@]=attrplsresinc4*mshare41[4]
;mw[132][@AIRPORTTAZ@]=attrplsresinc4*mshare42[4]
;mw[133][@AIRPORTTAZ@]=attrplsresinc4*mshare43[4]
;mw[134][@AIRPORTTAZ@]=attrplsresinc4*mshare44[4]
;mw[135][@AIRPORTTAZ@]=attrplsresinc4*mshare45[4]
;mw[136][@AIRPORTTAZ@]=attrplsresinc4*mshare46[4]

nbusnresi4[1] = attrplsresinc4*mshare41[4] ; Drive Alone
nbusnresi4[2] = attrplsresinc4*mshare42[4] ; Drop Off
nbusnresi4[3] = attrplsresinc4*mshare43[4] ; Rental Car
nbusnresi4[4] = attrplsresinc4*mshare44[4] ; Transit
nbusnresi4[5] = attrplsresinc4*mshare45[4] ; Shuttle
nbusnresi4[6] = attrplsresinc4*mshare46[4] ; Taxi

;output travel time and cost for transit, shuttle and taxi

print file=look.txt, list=z(5), trnrun, trnfare, shrun, shfare, txrun, txfare,
      trnuse, trnwait, trnwalk,

      hwytime, hwydist, hwycost

print file=busres.txt, list=z(5), busresi1[1], busresi1[2], busresi1[3], busresi1[4], busresi1[5], busresi1[6],
      busresi2[1], busresi2[2], busresi2[3], busresi2[4], busresi2[5], busresi2[6],

      busresi3[1], busresi3[2], busresi3[3], busresi3[4], busresi3[5], busresi3[6],

      busresi4[1], busresi4[2], busresi4[3], busresi4[4], busresi4[5], busresi4[6]
print file=busnres.txt, list=z(5), busnresi1[1], busnresi1[2], busnresi1[3], busnresi1[4], busnresi1[5], busnresi1[6],
      busnresi2[1], busnresi2[2], busnresi2[3], busnresi2[4], busnresi2[5], busnresi2[6],

      busnresi3[1], busnresi3[2], busnresi3[3], busnresi3[4], busnresi3[5], busnresi3[6],

      busnresi4[1], busnresi4[2], busnresi4[3], busnresi4[4], busnresi4[5], busnresi4[6]
print file=nbusres.txt, list=z(5), nbusresi1[1], nbusresi1[2], nbusresi1[3], nbusresi1[4], nbusresi1[5], nbusresi1[6],
      nbusresi2[1], nbusresi2[2], nbusresi2[3], nbusresi2[4], nbusresi2[5], nbusresi2[6],

      nbusresi3[1], nbusresi3[2], nbusresi3[3], nbusresi3[4], nbusresi3[5], nbusresi3[6],

      nbusresi4[1], nbusresi4[2], nbusresi4[3], nbusresi4[4], nbusresi4[5], nbusresi4[6]
print file=nbusnres.txt, list=z(5), nbusnresi1[1], nbusnresi1[2], nbusnresi1[3], nbusnresi1[4], nbusnresi1[5], nbusnresi1[6],
      nbusnresi2[1], nbusnresi2[2], nbusnresi2[3], nbusnresi2[4], nbusnresi2[5], nbusnresi2[6],

      nbusnresi3[1], nbusnresi3[2], nbusnresi3[3], nbusnresi3[4], nbusnresi3[5], nbusnresi3[6],

      nbusnresi4[1], nbusnresi4[2], nbusnresi4[3], nbusnresi4[4], nbusnresi4[5], nbusnresi4[6]

endrun

,*****
,*****
;step 4 Person trip to Vehicle trips (DA, DO, TX, SH)

```

```

,*****
,*****
run pgm=matrix

ZONES = @ZONESIZE@

;mati[1]=busres.trp ; business resident trip table (24 trip tables - 4 incomes DA, DO, RC, TRN, SH, TX)
;mati[2]=busnonres.trp ; business non-resident trip table (24 trip tables - 4 incomes DA, DO, RC, TRN, SH, TX)
;mati[3]=pleares.trp ; pleasure resident trip table (24 trip tables - 4 incomes DA, DO, RC, TRN, SH, TX)
;mati[4]=pleanonres.trp ; pleasure non-resident trip table (24 trip tables - 4 incomes DA, DO, RC, TRN, SH, TX)

zdati[1]=busres.txt, z=#1, busresi1DA=#2, busresi1DO=#3, busresi1RC=#4, busresi1TRN=#5, busresi1SH=#6, busresi1TX=#7,
        busresi2DA=#8, busresi2DO=#9, busresi2RC=#10, busresi2TRN=#11, busresi2SH=#12, busresi2TX=#13,
        busresi3DA=#14, busresi3DO=#15,
busresi3RC=#16, busresi3TRN=#17, busresi3SH=#18, busresi3TX=#19,
        busresi4DA=#20,
busresi4DO=#21, busresi4RC=#22, busresi4TRN=#23, busresi4SH=#24, busresi4TX=#25
zdati[2]=busnonres.txt, z=#1, busnresi1DA=#2, busnresi1DO=#3, busnresi1RC=#4, busnresi1TRN=#5, busnresi1SH=#6, busnresi1TX=#7,
        busnresi2DA=#8, busnresi2DO=#9, busnresi2RC=#10, busnresi2TRN=#11, busnresi2SH=#12, busnresi2TX=#13,
        busnresi3DA=#14, busnresi3DO=#15,
busnresi3RC=#16, busnresi3TRN=#17, busnresi3SH=#18, busnresi3TX=#19,
        busnresi4DA=#20,
busnresi4DO=#21, busnresi4RC=#22, busnresi4TRN=#23, busnresi4SH=#24, busnresi4TX=#25
zdati[3]=nbusres.txt, z=#1, nbusresi1DA=#2, nbusresi1DO=#3, nbusresi1RC=#4, nbusresi1TRN=#5, nbusresi1SH=#6, nbusresi1TX=#7,
        nbusresi2DA=#8, nbusresi2DO=#9, nbusresi2RC=#10, nbusresi2TRN=#11, nbusresi2SH=#12, nbusresi2TX=#13,
        nbusresi3DA=#14, nbusresi3DO=#15,
nbusresi3RC=#16, nbusresi3TRN=#17, nbusresi3SH=#18, nbusresi3TX=#19,
        nbusresi4DA=#20,
nbusresi4DO=#21, nbusresi4RC=#22, nbusresi4TRN=#23, nbusresi4SH=#24, nbusresi4TX=#25
zdati[4]=nbusnonres.txt, z=#1, nbusnresi1DA=#2, nbusnresi1DO=#3, nbusnresi1RC=#4, nbusnresi1TRN=#5, nbusnresi1SH=#6, nbusnresi1TX=#7,
        nbusnresi2DA=#8, nbusnresi2DO=#9, nbusnresi2RC=#10, nbusnresi2TRN=#11, nbusnresi2SH=#12, nbusnresi2TX=#13,
        nbusnresi3DA=#14, nbusnresi3DO=#15, nbusnresi3RC=#16,
nbusnresi3TRN=#17, nbusnresi3SH=#18, nbusnresi3TX=#19,
        nbusnresi4DA=#20,
nbusnresi4DO=#21, nbusnresi4RC=#22, nbusnresi4TRN=#23, nbusnresi4SH=#24, nbusnresi4TX=#25

mato[1]=vehbusres@AIRPORT@.trp, mo=1, name=BusResVeh
mato[2]=vehbusnonres@AIRPORT@.trp, mo=2, name=BusNonResVeh
mato[3]=vehplsres@AIRPORT@.trp, mo=3, name=PlsResVeh
mato[4]=vehplsrenonres@AIRPORT@.trp, mo=4, name=PlsNonResVeh

mato[5]=trnbusres@AIRPORT@.trp, mo=5, name=BusResTrn
mato[6]=trnbusnonres@AIRPORT@.trp, mo=6, name=BusNonResTrn
mato[7]=trnplsres@AIRPORT@.trp, mo=7, name=PlsResTrn
mato[8]=trnplsrenonres@AIRPORT@.trp, mo=8, name=PlsNonResTrn

;Vehicle trip conversion
;Trip tables generated above are in person trip format. Apply fixed factors to convert to vehicle trips

aofda=1.00 ;Drive alone - 1.00
aofdo=1.65 ;Drop off - 1.65
aofrc=1.10 ;Rental Car - 1.10
aoftrn=1.00 ;Transit - 1.00
aofsh=7.65 ;Shuttle - 7.65
aoftx=1.45 ;Taxi - 1.45

jloop

IF(I=@AIRPORTTAZ@) CONTINUE
IF(J<@AIRPORTTAZ@) CONTINUE
IF(J>@AIRPORTTAZ@) CONTINUE

;VEHICLE TRIPS
; mw[1]= mi.1.1/aofda + mi.1.2/aofdo + mi.1.3/aofrc + mi.1.5/aofsh + mi.1.6/aoftx +
; mi.1.7/aofda + mi.1.8/aofdo + mi.1.9/aofrc + mi.1.11/aofsh + mi.1.12/aoftx +

```

```

; mi.1.13/aofda + mi.1.14/aofdo + mi.1.15/aofrc + mi.1.17/aofsh + mi.1.18/aofxt +
; mi.1.19/aofda + mi.1.20/aofdo + mi.1.21/aofrc + mi.1.23/aofsh + mi.1.24/aofxt
mw[1] = busresi1DA/aofDA + busresi1DO/aofDO + busresi1RC/aofRC + busresi1SH/aofSH + busresi1TX/aofTX +
      busresi2DA/aofDA + busresi2DO/aofDO + busresi2RC/aofRC + busresi2SH/aofSH + busresi2TX/aofTX +
      busresi3DA/aofDA + busresi3DO/aofDO + busresi3RC/aofRC +
busresi3SH/aofSH + busresi3TX/aofTX +
      busresi4DA/aofDA + busresi4DO/aofDO + busresi4RC/aofRC +
busresi4SH/aofSH + busresi4TX/aofTX

; mw[2]= mi.2.1/aofda + mi.2.2/aofdo + mi.2.3/aofrc + mi.2.5/aofsh + mi.2.6/aofxt +
; mi.2.7/aofda + mi.2.8/aofdo + mi.2.9/aofrc + mi.2.11/aofsh + mi.2.12/aofxt +
; mi.2.13/aofda + mi.2.14/aofdo + mi.2.15/aofrc + mi.2.17/aofsh + mi.2.18/aofxt +
; mi.2.19/aofda + mi.2.20/aofdo + mi.2.21/aofrc + mi.2.23/aofsh + mi.2.24/aofxt
mw[2] = busresi1DA/aofDA + busresi1DO/aofDO + busresi1RC/aofRC + busresi1SH/aofSH + busresi1TX/aofTX +
      busresi2DA/aofDA + busresi2DO/aofDO + busresi2RC/aofRC + busresi2SH/aofSH + busresi2TX/aofTX +
      busresi3DA/aofDA + busresi3DO/aofDO + busresi3RC/aofRC +
busresi3SH/aofSH + busresi3TX/aofTX +
      busresi4DA/aofDA + busresi4DO/aofDO + busresi4RC/aofRC +
busresi4SH/aofSH + busresi4TX/aofTX

; mw[3]= mi.3.1/aofda + mi.3.2/aofdo + mi.3.3/aofrc + mi.3.5/aofsh + mi.3.6/aofxt +
; mi.3.7/aofda + mi.3.8/aofdo + mi.3.9/aofrc + mi.3.11/aofsh + mi.3.12/aofxt +
; mi.3.13/aofda + mi.3.14/aofdo + mi.3.15/aofrc + mi.3.17/aofsh + mi.3.18/aofxt +
; mi.3.19/aofda + mi.3.20/aofdo + mi.3.21/aofrc + mi.3.23/aofsh + mi.3.24/aofxt
mw[3] = nbusresi1DA/aofDA + nbusresi1DO/aofDO + nbusresi1RC/aofRC + nbusresi1SH/aofSH + nbusresi1TX/aofTX +
      nbusresi2DA/aofDA + nbusresi2DO/aofDO + nbusresi2RC/aofRC + nbusresi2SH/aofSH + nbusresi2TX/aofTX +
      nbusresi3DA/aofDA + nbusresi3DO/aofDO + nbusresi3RC/aofRC +
nbusresi3SH/aofSH + nbusresi3TX/aofTX +
      nbusresi4DA/aofDA + nbusresi4DO/aofDO + nbusresi4RC/aofRC +
nbusresi4SH/aofSH + nbusresi4TX/aofTX

; mw[4]= mi.4.1/aofda + mi.4.2/aofdo + mi.4.3/aofrc + mi.4.5/aofsh + mi.4.6/aofxt +
; mi.4.7/aofda + mi.4.8/aofdo + mi.4.9/aofrc + mi.4.11/aofsh + mi.4.12/aofxt +
; mi.4.13/aofda + mi.4.14/aofdo + mi.4.15/aofrc + mi.4.17/aofsh + mi.4.18/aofxt +
; mi.4.19/aofda + mi.4.20/aofdo + mi.4.21/aofrc + mi.4.23/aofsh + mi.4.24/aofxt
mw[4] = nbusresi1DA/aofDA + nbusresi1DO/aofDO + nbusresi1RC/aofRC + nbusresi1SH/aofSH + nbusresi1TX/aofTX +
      nbusresi2DA/aofDA + nbusresi2DO/aofDO + nbusresi2RC/aofRC + nbusresi2SH/aofSH + nbusresi2TX/aofTX +
      nbusresi3DA/aofDA + nbusresi3DO/aofDO + nbusresi3RC/aofRC +
nbusresi3SH/aofSH + nbusresi3TX/aofTX +
      nbusresi4DA/aofDA + nbusresi4DO/aofDO + nbusresi4RC/aofRC +
nbusresi4SH/aofSH + nbusresi4TX/aofTX

;TRANSIT TRIPS
; mw[5]= mi.1.4/aoftrn + mi.1.10/aoftrn + mi.1.16/aoftrn + mi.1.22/aoftrn
; mw[6]= mi.2.4/aoftrn + mi.2.10/aoftrn + mi.2.16/aoftrn + mi.2.22/aoftrn
; mw[7]= mi.3.4/aoftrn + mi.3.10/aoftrn + mi.3.16/aoftrn + mi.3.22/aoftrn
; mw[8]= mi.4.4/aoftrn + mi.4.10/aoftrn + mi.4.16/aoftrn + mi.4.22/aoftrn
mw[5] = busresi1TRN/aoftrn + busresi2TRN/aoftrn + busresi3TRN/aoftrn + busresi4TRN/aoftrn
mw[6] = busresi1TRN/aoftrn + busresi2TRN/aoftrn + busresi3TRN/aoftrn + busresi4TRN/aoftrn
mw[7] = nbusresi1TRN/aoftrn + nbusresi2TRN/aoftrn + nbusresi3TRN/aoftrn + nbusresi4TRN/aoftrn
mw[8] = nbusresi1TRN/aoftrn + nbusresi2TRN/aoftrn + nbusresi3TRN/aoftrn + nbusresi4TRN/aoftrn

```

endjloop

endrun

```

*****
/
*****
;step 5 TOD for Business & Pleasure Resident and Non-Resident Vehicle Trips
*****
/
*****

```

run pgm=matrix

ZONES = @ZONESIZE@

```

mati[1]=vehbusres@AIRPORT@.trp
mati[2]=vehbusnonres@AIRPORT@.trp
mati[3]=vehplsres@AIRPORT@.trp
mati[4]=vehplsrenonres@AIRPORT@.trp
mati[5]=trnbusres@AIRPORT@.trp
mati[6]=trnbusnonres@AIRPORT@.trp
mati[7]=trnplsres@AIRPORT@.trp
mati[8]=trnplsrenonres@AIRPORT@.trp

mato[1]=veham@AIRPORT@.trp, mo=1, name=AMVehAirPass ;am veh trip table
mato[2]=vehmd@AIRPORT@.trp, mo=2, name=MDVehAirPass ;mid day veh trip table
mato[3]=vehpm@AIRPORT@.trp, mo=3, name=PMVehAirPass ;pm veh trip table
mato[4]=vehnt@AIRPORT@.trp, mo=4, name=NTVehAirPass ;over night veh trip table

mato[5]=trnam@AIRPORT@.trp, mo=5, name=AMTrnAirPass ;am veh trip table
mato[6]=trnmd@AIRPORT@.trp, mo=6, name=MDTrnAirPass ;mid day veh trip table
mato[7]=trnpm@AIRPORT@.trp, mo=7, name=PMTrnAirPass ;pm veh trip table
mato[8]=trnnt@AIRPORT@.trp, mo=8, name=NTTrnAirPass ;over night veh trip table

;Calculate Vehicle trip tables by time period
;Business trips will use HTS - HBW splits and Pleasure trips will use HTS - HBO splits

; Time periods: (4)
; AM = 6:30a - 9:30a
; MD = 9:30a - 3:30p
; PM = 3:30p - 6:30p
; NT = 6:30p - 6:30a

; MWCOGTime periods: (4)
; AM = 6:00a - 9:00a
; MD = 9:00a - 3:00p
; PM = 3:00p - 7:00p
; NT = 7:00p - 6:00a

; do not balace transit riders - keep in P & A format to match peak transit service

jloop
; AM peak
; Business-Resident and Business-NonResident
; Use HBW Splits (Pur=1); Drive Alone (Mod=2); Dir=H/NH (Dir=1/2); AM (Period=1)
; 0.5788 / 0.0114
; NonBusiness-Resident and NonBusiness-NonResident
; Use HBO Splits (Pur=3); Drive Alone (Mod=2); Dir=H/NH (Dir=1/2); AM (Period=1)
; 0.1958 / 0.0507
; Source: todcomp_2008HTSnew.dbf
;
; Use 2009 Washington-Baltimore Regional Air Passenger Survey, September 2010
; Table 3: Number of Survey Flights by Airport and Flight Hours
;fac1am=0.5788
;fac2am=0.0114
;fac3am=0.1958
;fac4am=0.0507
fac1amBWI=0.269
fac2amBWI=0.269
fac3amBWI=0.269
fac4amBWI=0.269
fac1amDCA=0.322
fac2amDCA=0.322
fac3amDCA=0.322
fac4amDCA=0.322
fac1amIAD=0.234
fac2amIAD=0.234
fac3amIAD=0.234
fac4amIAD=0.234

; mw[1]=0.5 * (0.6330 * mi.1.1 + 0.0109 * mi.1.1.t +

```

```

;      0.6330 * mi.2.1 + 0.0109 * mi.2.1.t +
;      0.2035 * mi.3.1 + 0.0868 * mi.3.1.t +
;      0.2035 * mi.4.1 + 0.0868 * mi.4.1.t)
      mw[1]=0.5 * (fac1am@AIRPORT@ * mi.1.1 + fac2am@AIRPORT@ * mi.1.1.t +
      fac1am@AIRPORT@ * mi.2.1 + fac2am@AIRPORT@ * mi.2.1.t +
      fac3am@AIRPORT@ * mi.3.1 + fac4am@AIRPORT@ * mi.3.1.t +
      fac3am@AIRPORT@ * mi.4.1 + fac4am@AIRPORT@ * mi.4.1.t)

; mw[5]= (0.32195 * mi.5.1 +
;         0.32195 * mi.6.1 +
;         0.14515 * mi.7.1 +
;         0.14515 * mi.8.1 )
mw[5]= ((fac1am@AIRPORT@ + fac2am@AIRPORT@)/2.) * mi.5.1 +
        ((fac1am@AIRPORT@ + fac2am@AIRPORT@)/2.) * mi.6.1 +
        ((fac3am@AIRPORT@ + fac4am@AIRPORT@)/2.) * mi.7.1 +
        ((fac3am@AIRPORT@ + fac4am@AIRPORT@)/2.) * mi.8.1 )

;MD
; Business-Resident and Business-NonResident
;   Use HBW Splits (Pur=1); Drive Alone (Mod=2); Dir=H/NH (Dir=1/2); MD (Period=2)
;   0.2289 / 0.0873
; NonBusiness-Resident and NonBusiness-NonResident
;   Use HBO Splits (Pur=3); Drive Alone (Mod=2); Dir=H/NH (Dir=1/2); MD (Period=2)
;   0.4112 / 0.2641
; Source: todcomp_2008HTSnew.dbf
;fac1md=0.2289
;fac2md=0.0873
;fac3md=0.4112
;fac4md=0.2641
fac1mdBWI=0.233
fac2mdBWI=0.233
fac3mdBWI=0.233
fac4mdBWI=0.233
fac1mdDCA=0.221
fac2mdDCA=0.221
fac3mdDCA=0.221
fac4mdDCA=0.221
fac1mdIAD=0.217
fac2mdIAD=0.217
fac3mdIAD=0.217
fac4mdIAD=0.217
; mw[2]=0.5 * (0.1569 * mi.1.1 + 0.1500 * mi.1.1.t +
;             0.1569 * mi.2.1 + 0.1500 * mi.2.1.t +
;             0.3707 * mi.3.1 + 0.2926 * mi.3.1.t +
;             0.3707 * mi.4.1 + 0.2926 * mi.4.1.t)
      mw[2]=0.5 * (fac1md@AIRPORT@ * mi.1.1 + fac2md@AIRPORT@ * mi.1.1.t +
      fac1md@AIRPORT@ * mi.2.1 + fac2md@AIRPORT@ * mi.2.1.t +
      fac3md@AIRPORT@ * mi.3.1 + fac4md@AIRPORT@ * mi.3.1.t +
      fac3md@AIRPORT@ * mi.4.1 + fac4md@AIRPORT@ * mi.4.1.t)

; mw[6]= (0.15345 * mi.5.1 +
;         0.15345 * mi.6.1 +
;         0.33165 * mi.7.1 +
;         0.33165 * mi.8.1 )
mw[6]= ((fac1md@AIRPORT@ + fac2md@AIRPORT@)/2.) * mi.5.1 +
        ((fac1md@AIRPORT@ + fac2md@AIRPORT@)/2.) * mi.6.1 +
        ((fac3md@AIRPORT@ + fac4md@AIRPORT@)/2.) * mi.7.1 +
        ((fac3md@AIRPORT@ + fac4md@AIRPORT@)/2.) * mi.8.1 )

;PM peak
; Business-Resident and Business-NonResident
;   Use HBW Splits (Pur=1); Drive Alone (Mod=2); Dir=H/NH (Dir=1/2); PM (Period=3)
;   0.0376 / 0.5962
; NonBusiness-Resident and NonBusiness-NonResident
;   Use HBO Splits (Pur=3); Drive Alone (Mod=2); Dir=H/NH (Dir=1/2); PM (Period=3)

```

```

; 0.2111 / 0.2687
; Source: todcomp_2008HTSnew.dbf
;fac1pm=0.0376
;fac2pm=0.5962
;fac3pm=0.2111
;fac4pm=0.2687
fac1pmBWI=0.304
fac2pmBWI=0.304
fac3pmBWI=0.304
fac4pmBWI=0.304
fac1pmDCA=0.274
fac2pmDCA=0.274
fac3pmDCA=0.274
fac4pmDCA=0.274
fac1pmIAD=0.320
fac2pmIAD=0.320
fac3pmIAD=0.320
fac4pmIAD=0.320
; mw[3]=0.5 * (0.0341 * mi.1.1 + 0.6328 * mi.1.1.t +
;           0.0341 * mi.2.1 + 0.6328 * mi.2.1.t +
;           0.2074 * mi.3.1 + 0.2221 * mi.3.1.t +
;           0.2074 * mi.4.1 + 0.2221 * mi.4.1.t)
      mw[3]=0.5 * (fac1pm@AIRPORT@ * mi.1.1 + fac2pm@AIRPORT@ * mi.1.1.t +
      fac1pm@AIRPORT@ * mi.2.1 + fac2pm@AIRPORT@ * mi.2.1.t +
      fac3pm@AIRPORT@ * mi.3.1 + fac4pm@AIRPORT@ * mi.3.1.t +
      fac3pm@AIRPORT@ * mi.4.1 + fac4pm@AIRPORT@ * mi.4.1.t)

; mw[7]= (0.33345 * mi.5.1 +
;         0.33345 * mi.6.1 +
;         0.21475 * mi.7.1 +
;         0.21475 * mi.8.1)
mw[7]= ((fac1pm@AIRPORT@ + fac2pm@AIRPORT@)/2.) * mi.5.1 +
        ((fac1pm@AIRPORT@ + fac2pm@AIRPORT@)/2.) * mi.6.1 +
        ((fac3pm@AIRPORT@ + fac4pm@AIRPORT@)/2.) * mi.7.1 +
        ((fac3pm@AIRPORT@ + fac4pm@AIRPORT@)/2.) * mi.8.1

;NT
;Business-Resident and Business-NonResident
; Use HBW Splits (Pur=1); Drive Alone (Mod=2); Dir=H/NH (Dir=1/2); NT (Period=4)
; 0.1547 / 0.3051
; NonBusiness-Resident and NonBusiness-NonResident
; Use HBO Splits (Pur=3); Drive Alone (Mod=2); Dir=H/NH (Dir=1/2); NT (Period=4)
; 0.1819 / 0.4165
; Source: todcomp_2008HTSnew.dbf
;fac1nt=0.1547
;fac2nt=0.3051
;fac3nt=0.1819
;fac4nt=0.4165
fac1ntBWI=0.194
fac2ntBWI=0.194
fac3ntBWI=0.194
fac4ntBWI=0.194
fac1ntDCA=0.183
fac2ntDCA=0.183
fac3ntDCA=0.183
fac4ntDCA=0.183
fac1ntIAD=0.230
fac2ntIAD=0.230
fac3ntIAD=0.230
fac4ntIAD=0.230
; mw[4]=0.5 * (0.1760 * mi.1.1 + 0.2063 * mi.1.1.t +
;           0.1760 * mi.2.1 + 0.2063 * mi.2.1.t +
;           0.2184 * mi.3.1 + 0.3985 * mi.3.1.t +
;           0.2184 * mi.4.1 + 0.3985 * mi.4.1.t)
      mw[4]=0.5 * (fac1nt@AIRPORT@ * mi.1.1 + fac2nt@AIRPORT@ * mi.1.1.t +
      fac1nt@AIRPORT@ * mi.2.1 + fac2nt@AIRPORT@ * mi.2.1.t +

```

```

fac3nt@AIRPORT@ * mi.3.1 + fac4nt@AIRPORT@ * mi.3.1.t +
fac3nt@AIRPORT@ * mi.4.1 + fac4nt@AIRPORT@ * mi.4.1.t)

; mw[8]= (0.19115 * mi.5.1 +
;         0.19115 * mi.6.1 +
;         0.30845 * mi.7.1 +
;         0.30845 * mi.8.1)
mw[8]= ( ((fac1nt@AIRPORT@ + fac2nt@AIRPORT@)/2.) * mi.5.1 +
         ((fac1nt@AIRPORT@ + fac2nt@AIRPORT@)/2.) * mi.6.1 +
         ((fac3nt@AIRPORT@ + fac4nt@AIRPORT@)/2.) * mi.7.1 +
         ((fac3nt@AIRPORT@ + fac4nt@AIRPORT@)/2.) * mi.8.1 )

endjloop
endrun

```