



National Capital Region  
**Transportation Planning Board**

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

# User's Guide for the COG/TPB Travel Demand Forecasting Model, Version 2.3.66

---

*Volume 1 of 2: Main Report and Appendix A (Flowcharts)*

*February 13, 2017*



## **User's Guide for the COG/TPB Travel Demand Forecasting Model, Version 2.3.66: Volume 1 of 2**

February 13, 2017

### **About the TPB**

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

### **About COG**

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

### **Credits**

Director of DTP: Kanti Srikanth

Director, Travel Forecasting and Emissions Analysis, DTP: Ronald Milone

Report Authors: Mark Moran and Ron Milone

Contributors: Meseret Seifu, Hamid Humeida, Dzung Ngo, and Mary Martchouk

Oversight: COG/TPB Travel Forecasting Subcommittee

### **Acknowledgements**

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

### **Accommodations Policy**

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

### **Title VI Nondiscrimination Policy**

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

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

Copyright © 2017 by the Metropolitan Washington Council of Governments

## Table of Contents

<b>1</b>	<b><i>Introduction</i></b> .....	<b>1</b>
1.1	<b>Adoption of the model by the TPB</b> .....	<b>2</b>
1.2	<b>History of the Version 2.3 Travel Model</b> .....	<b>2</b>
1.3	<b>Recent changes to the model: Ver. 2.3.57a to Ver. 2.3.66</b> .....	<b>4</b>
1.3.1	Update 1: Migrate from Cube Voyager 6.1 SP1 (32-bit) to Cube Voyager 6.4.1 (64-bit).....	6
1.3.2	Update 2: Sort the areawalk.txt file by TAZ number .....	6
1.3.3	Update 3: Fix two bugs in autoacc4.s .....	7
1.3.4	Update 4: Updates to the local-bus, in-vehicle speed degradation model.....	7
1.3.5	Update 5: Add comments/annotation to the script use for transit skimming for the bus/Metrorail transit sub-mode .....	8
1.3.6	Update 6: Fix the issue of double counting bridge penalties in path building.....	8
1.3.7	Update 7: Modeling HOT-lanes/toll setting: Streamline the existing process.....	9
1.3.8	Update 8: Modeling HOT-lanes/toll setting: Implementing HOV Skim replacement procedure for HOV2+, not simply HOV3+.....	10
1.3.9	Update 9: Updates to the automated process to generate transit walksheds.....	11
<b>2</b>	<b><i>Overview of the model</i></b> .....	<b>13</b>
2.1	<b>Model inputs</b> .....	<b>14</b>
2.2	<b>Model outputs</b> .....	<b>17</b>
2.3	<b>Modeling steps and the speed feedback loop</b> .....	<b>18</b>
2.4	<b>Special modeling procedures</b> .....	<b>24</b>
2.4.1	Metrorail constraint through the regional core.....	25
2.4.2	Modeling high occupancy/toll (HOT) lanes.....	25
2.4.3	Select-link analyses.....	28
<b>3</b>	<b><i>Hardware and software requirements</i></b> .....	<b>29</b>
3.1	<b>Hardware</b> .....	<b>29</b>
3.2	<b>Software</b> .....	<b>30</b>
3.3	<b>Examples of computer hardware used at COG for modeling</b> .....	<b>31</b>
<b>4</b>	<b><i>Mechanics of the model application process</i></b> .....	<b>35</b>
<b>5</b>	<b><i>Preparing to run the model</i></b> .....	<b>47</b>
5.1	<b>Software installation and setting the Windows PATH environment variable</b> .....	<b>47</b>
5.2	<b>Preparing input files and calculating zonal percent-walk-to-transit values</b> .....	<b>56</b>
<b>6</b>	<b><i>Running the model</i></b> .....	<b>57</b>
6.1	<b>Updating the Windows PATH environment variable</b> .....	<b>57</b>
6.2	<b>Parent batch files</b> .....	<b>57</b>

6.2.1	Description of the "run model"/wrapper batch file .....	61
6.2.2	Description of the "run model steps" batch files.....	62
<b>6.3</b>	<b>Running the model: An example .....</b>	<b>74</b>
<b>7</b>	<b>Summarizing model output and other utilities.....</b>	<b>75</b>
7.1	Summary Scripts .....	75
7.2	Utilities .....	76
<b>8</b>	<b>Use of parallel processing to reduce model run times .....</b>	<b>79</b>
8.1	Model run times .....	79
8.2	Use of parallel processing to reduce model run times .....	80
8.2.1	Background and terminology .....	81
8.2.2	Effect of Cube Cluster on modeled results .....	82
8.2.3	History of adding parallelization to the Version 2.3 Travel Model .....	82
8.2.4	Implementation of parallelization in the Version 2.3.52 through 2.3.66 travel models.....	83
<b>9</b>	<b>Debugging cases where the model run stops prematurely or crashes .....</b>	<b>95</b>
<b>10</b>	<b>Known issues related to running the model .....</b>	<b>97</b>
10.1	Cube Cluster differences .....	97
10.2	Model run stops before finishing .....	97
10.3	Issues with traffic assignment convergence.....	97
10.4	Running multiple concurrent model runs one computer/server .....	97
<b>11</b>	<b>Building transit walksheds and calculating zonal walk percentages .....</b>	<b>101</b>
11.1	Overview .....	101
11.2	Application Details.....	101
11.3	Known issues .....	103
<b>12</b>	<b>Set-Up Programs and Highway Network Building.....</b>	<b>105</b>
12.1	Overview .....	105
12.2	Application Details.....	110
<b>13</b>	<b>Highway Skim File Development.....</b>	<b>117</b>
13.1	Overview .....	117
13.2	Application Details.....	118
<b>14</b>	<b>Auto Driver Trip Development .....</b>	<b>123</b>
14.1	Overview .....	123
14.2	Application Details.....	123

<b>15</b>	<b>Pre-Transit Network Processing</b>	<b>125</b>
15.1	Overview	125
15.2	Application Details	128
<b>16</b>	<b>Transit Skim File Development</b>	<b>131</b>
16.1	Overview	131
16.2	Application Details	132
16.2.1	Skim file names and list of transit skim tables in the skim files	132
16.2.2	Description of local bus, future time degradation factors	134
<b>17</b>	<b>Transit Fare Development</b>	<b>137</b>
17.1	Overview	137
17.2	Application Details	139
<b>18</b>	<b>Demographic Submodels</b>	<b>143</b>
<b>19</b>	<b>Trip Generation</b>	<b>145</b>
19.1	Control/Support File(s):	145
19.2	Application Details:	145
<b>20</b>	<b>Trip Distribution</b>	<b>155</b>
20.1	Overview	155
20.2	Application Details	156
<b>21</b>	<b>Mode Choice</b>	<b>159</b>
21.1	Travel modes represented in the mode choice model	159
21.1.1	Treatment of LRT, BRT, and streetcar	159
21.1.2	Other issues relating to travel modes	161
21.2	Metrorail constraint through the regional core	163
21.3	Control/Support Files	165
21.4	Market segmentation	167
21.4.1	Market segmentation by household income	167
21.4.2	Market segmentation by geography	167
21.4.3	Market segmentation by access to transit	171
21.5	Transit access coding	184
21.5.1	Station file	184
21.5.2	Sidewalk links and zonal walk links	185
21.5.3	Zonal auto-access links	187
21.5.4	Station transfer links	190
21.5.5	Zonal percent walk to transit calculations	191
21.6	Transit path-building procedures	192

<b>21.7</b>	<b>Treatment of parking costs and terminal times for non-transit-related trips .....</b>	<b>195</b>
21.7.1	Non-transit-related parking costs.....	195
21.7.2	Non-transit-related highway terminal time assumptions.....	196
<b>21.8</b>	<b>Auto Operating Costs .....</b>	<b>197</b>
<b>22</b>	<b>Time-of-Day Processing .....</b>	<b>199</b>
22.1	Overview .....	199
<b>23</b>	<b>Traffic Assignment.....</b>	<b>201</b>
23.1	Overview .....	201
23.2	Two-step assignment .....	202
23.3	Application details .....	204
<b>24</b>	<b>Transit Assignment, Including Summary Process (LineSum).....</b>	<b>209</b>
<b>24.1</b>	<b>Transit assignment process .....</b>	<b>209</b>
24.1.1	Inputs to the transit assignment.....	211
24.1.2	Outputs of the transit assignment.....	211
<b>24.2</b>	<b>Transit assignment summary process.....</b>	<b>218</b>
24.2.1	Consolidating transit assignment output and displaying results .....	220
24.2.2	Generating transit assignment summaries using LineSum .....	224



## List of Figures

Figure 1 Major inputs and outputs of the TPB Version 2.3 Travel Model .....	14
Figure 2 Major steps of the Version 2.3 Travel Model .....	19
Figure 3 Application process of the Version 2.3 Travel Model .....	23
Figure 4 Subdirectory structure for executing the Version 2.3 Travel Model .....	36
Figure 5 Number of cores and logical processors .....	48
Figure 6 Citilabs Cube 6.4, "About" message window .....	49
Figure 7 "About Voyager": Presence of Cube Cluster license .....	50
Figure 8 "Run model" batch file for 2016_base .....	58
Figure 9 "Run model steps" batch file for 2016_base .....	59
Figure 10 "Run model steps" batch file for 2020_base .....	64
Figure 11 "Run model steps" batch file for 2020_final .....	67
Figure 12 "Run model steps" batch file for 2040_base .....	69
Figure 13 "Run model steps" batch file for 2040_final .....	71
Figure 14 Schematic of IDP and MDP in the highway assignment process of the Ver. 2.3 Travel Model (Highway_Assignment_Parallel.s): Existing naming convention for nodes .....	90
Figure 15 Schematic of IDP and MDP in the highway assignment process of the Ver. 2.3 Travel Model (Highway_Assignment_Parallel.s): Proposed new naming convention for nodes (changes shown in red) .....	91
Figure 16 Excerpts from the Highway_Assignment_Parallel.s script (triple ampersand => code removed) .....	92
Figure 17 An excerpt from an example of the "full output" text file that is created during a model run	.96
Figure 18 An excerpt from the "search for errors" file that is created during a model run .....	96
Figure 19 Folder structure for the automated ArcPy walkshed process .....	103
Figure 20 Walkshed buffers for a typical base-year scenario .....	104
Figure 21 Consumer price index file (CPI_File.txt) .....	110
Figure 22 A local bus route before its RUNTIME value is updated .....	135
Figure 23 A local bus route after its RUNTIME value is updated to reflect road congestion predicted to occur in future years .....	135
Figure 24 Outline/pseudo code for trip_generation.s .....	146
Figure 25 Example of seven NAVTEQ street blocks within TAZ 283 (Union Station) .....	148
Figure 26 Nesting structure of the nested-logit mode choice model in the Version 2.3 travel model ....	159
Figure 27 Ring 0 (white trapezoid) and Ring 1 (gray polygon), which form the "core" area used in the Metrorail constraint through the regional core .....	164
Figure 28 Seven superdistricts used in the Version 2.3 nested-logit mode choice model .....	168
Figure 29 Nesting structure of the nested-logit mode choice model in the Version 2.3 Travel Model ...	177
Figure 30 Zonal walk access links and sidewalk links in downtown DC near Farragut Square (Ver. 2.3 NL MC model) .....	187
Figure 31 Kiss-and-ride (KNR) auto access links to Metrorail stations in Northern Virginia .....	189
Figure 32 Park-and-ride (PNR) auto access links to Metrorail stations in Northern Virginia .....	190
Figure 33 Non-transit-related, daily parking cost model used in the Version 2.3 Model .....	196

Figure 34 Excerpt from one of the transit node DBF files output from transit assignment (i4_WKMRAMnode.dbf) .....	212
Figure 35 Transit volumes from transit assignment using TRNBUILD: One-way route .....	213
Figure 36 Transit volumes from transit assignment using TRNBUILD: Two-way route .....	214
Figure 37 Excerpt from one of the transit link DBF files output from transit assignment (i4_WKMRAMlink.dbf) showing mode-16 links .....	216
Figure 38 Excerpt from one of the transit link DBF files output from transit assignment (i4_WKMRAMlink.dbf) showing mode-3 links .....	216
Figure 39 Excerpt from one of the transit link DBF files output from transit assignment (i4_WKMRAMlink.dbf) showing mode-12 links .....	217
Figure 40 An excerpt of tranSum.bat transit summary batch file .....	219
Figure 41 Consolidating peak and off-peak transit assignment volumes (LineSum_Volume.ctl) .....	220
Figure 42 Using the pk_vol.dbf file in Cube Base as the transit layer: All transit routes turned on, but non-transit links (modes 11-16) turned off .....	221
Figure 43 Using the pk_vol.dbf file in Cube Base as the transit layer: Only mode-3 (Metrorail) links turned on .....	221
Figure 44 Using the pk_vol.dbf file in Cube Base as the transit layer: Only mode-3 (Metrorail) links turned on; using multi-bandwidth to represent transit loads (ab_vol): Arlington and DC .....	222
Figure 45 Using the pk_vol.dbf file in Cube Base as the transit layer: Only mode-3 (Metrorail) links turned on; using multi-bandwidth to represent transit loads (ab_vol): Metrorail system .....	222
Figure 46 Using the multi-bandwidth option in Cube Base to show transit volumes on the Metrorail system. ....	223
Figure 47 Generating a Metrorail station access report (lineSum_MR_access.ctl) .....	224
Figure 48 Generating a Metrorail line summary (lineSum_MR_line.ctl) .....	224
Figure 49 An excerpt from the report file generated by lineSum_MR_access.ct .....	226
Figure 50 The report file generated by lineSum_MR_line.ctl .....	228

## List of Tables

Table 1 Summary of changes between the two most recent versions of the TPB travel demand model: Ver. 2.3.57a and Ver. 2.3.66.....	5
Table 2 Outputs of the travel model: Disaggregate-level output that is not validated versus aggregate- level output that is validated .....	18
Table 3 User equilibrium closure criterion (relative gap) varies by speed feedback iteration .....	21
Table 4 Comparison of computer specs between tms6 and tms7f .....	32
Table 5 Storage drives used for travel demand modeling .....	33
Table 6 Input files needed to run the Version 2.3 Travel Model, stored in the CONTROLS, INPUTS, and SUPPORT folders .....	39
Table 7 Fortran and C++ executable files and dynamic-link library files required for running the Version 2.3 travel model .....	42
Table 8 Location for Cube DLL files .....	42
Table 9 Child batch files used to run the Version 2.3 Travel Model .....	43
Table 10 Sequence of the batch files used to run the Version 2.3 Travel Model .....	45
Table 11 Example of when special modeling procedures are invoked for three modeled years/scenarios .....	63
Table 12 Summary of differences to the “run model steps” batch files for the years 2016, 2020, and 2040 .....	64
Table 13 Travel Model Summary Scripts .....	75
Table 14 Five traffic assignments in the Version 2.2 travel model becomes six traffic assignments in the Version 2.3 travel model .....	80
Table 15 Modeling steps where parallelization is used, including the maximum number of threads/cores used .....	85
Table 16 Current and alternate names for the two Windows environment variables that store information about the number of subnodes to use in IDP in the highway assignment .....	87
Table 17 Running the Version 2.3 Travel Model on computers with fewer than 8 cores: Changes that need to be made to the “run model”/wrapper batch file .....	88
Table 18 Compatibility between Cube Base and ArcGIS, in terms of the ability to run the automated transit walkshed process .....	104
Table 19 Inputs to the set-up and highway network building process.....	105
Table 20 Land Use File Format Description (zone.dbf) .....	105
Table 21 Node Coordinate File Format Description (node.dbf).....	106
Table 22 Base Highway Link File Format Description (link.dbf) .....	106
Table 23 Toll Parameter File (Toll_esc.dbf) .....	107
Table 24 Outputs of the set-up and highway network building process.....	107
Table 25 Zonal Area Type File (AreaType_File.dbf) .....	108
Table 26 Unloaded binary highway network file (Zonehwy.net).....	109
Table 27 Elements of the highway network .....	111
Table 28 Area type codes, based on population and employment density .....	113
Table 29 zonehwy.net file Variables description .....	114

Table 30 Inputs to the highway skim file development.....	118
Table 31 Outputs of the highway skim file development.....	118
Table 32 Inputs to auto driver trip development .....	123
Table 33 Outputs of auto driver trip development .....	123
Table 34 Inputs to pre-transit network processing .....	125
Table 35 Variables in the transit station file (Station.dbf).....	126
Table 36 Interpretation of transit access distance codes (NCT): Metrorail, light rail, and bus PNR access distance codes and their meaning for the .....	127
Table 37 HBW zonal parking costs/terminal time file (HBWV2a1.dbf) .....	127
Table 38 Walk Access Links (WalkAcc_Lnks.dbf) .....	128
Table 39 Outputs of pre-transit network processing.....	128
Table 40 Overview of Version 2.3 Transit Network Elements .....	129
Table 41 Inputs to transit skim file development .....	131
Table 42 Outputs of transit skim file development .....	132
Table 43 Skim files developed by the transit skimming process .....	133
Table 44 Skim tables contained in each transit skim file .....	133
Table 45 Job accessibility by transit file format description (<ITER>_<Prd>_<AA>_[BM MR]_JobAcc.dbf) .....	135
Table 46 Inputs to transit fare development.....	137
Table 47 Outputs of transit fare development.....	138
Table 48 TAZ / Bus Fare Zone Equivalency File Format Description (TAZFRZN.ASC).....	139
Table 49 Inputs to the Demographic Models .....	143
Table 50 Outputs of the Demographic Models.....	144
Table 51 Inputs to trip generation .....	145
Table 52 External Production and Attraction File (Ext_PsAs.dbf).....	149
Table 53 Consolidated Zonal Land Use File.....	150
Table 54 Computed zonal trip productions file (<iter>_Trip_Gen_Productions_Comp.dbf) .....	150
Table 55 Computed zone trip attractions file (<iter>_Trip_Gen_Attractions_Comp.dbf) .....	151
Table 56 Truck and commercial vehicles trip ends (<iter>_ComVeh_Truck_Ends.dbf) .....	153
Table 57 Inputs to trip distribution.....	155
Table 58 Outputs of trip distribution.....	155
Table 59 Time Valuation (Minutes/2007\$) by Purpose and Income Level.....	157
Table 60 Transit sub-modes represented in the Version 2.3 travel model .....	162
Table 61 Transit Access and Transfer Links.....	162
Table 62 Transfer Prohibitions (No Transfer or NOX).....	163
Table 63 Inputs to the AEMS mode choice application program .....	165
Table 64 Outputs from the AEMS mode choice application program.....	166
Table 65 Household income quartiles computed from the ACS.....	167
Table 66 Equivalency between nested-logit mode choice superdistricts and TPB TAZ 3,722 .....	169
Table 67 Production and attraction market segments used in the TPB Version 2.3 NLMC model .....	170
Table 68 20 geographic market segments used in the TPB nested-logit mode choice model .....	170
Table 69 Equivalency between seven super-districts and the 20 geographic market segments.....	170

Table 70 Eleven examples showing how zonal percent-walk-to-transit values translate into probabilities of being in three transit-access markets: can walk, must drive, and no transit .....	174
Table 71 Probability of being in the “can walk” market segment for a zone-to-zone interchange, based on the production and attraction percent-walk-to-transit values .....	175
Table 72 Probability of being in the “must drive” market segment for a zone-to-zone interchange, based on the production and attraction percent-walk-to-transit values .....	175
Table 73 Probability of being in the “no transit” market segment for a zone-to-zone interchange, based solely on the attraction percent-walk-to-transit values .....	176
Table 74 Four “can walk” sub-markets .....	181
Table 75 Two “must drive” sub-markets .....	182
Table 76 Application of the seven transit-access segments to travel modes represented in the Ver. 2.3 mode choice model .....	183
Table 77 Mode codes used in the consolidated station file/database (station.dbf) .....	185
Table 78 Maximum link distances for drive-access-to-transit links: Ver. 2.3 NL MC model .....	188
Table 79 Path-specific parameters used in transit path building .....	194
Table 80 Non-transit-related highway terminal time as a function of employment density .....	197
Table 81 Inputs to time-of-day process .....	199
Table 82 Outputs of time-of-day process .....	199
Table 83 User equilibrium closure criterion (relative gap) varies by speed feedback iteration.....	202
Table 84 Traffic assignment in the Version 2.2 Travel Model prior to fall 2008: Three multiclass assignments.....	203
Table 85 Traffic assignment in the Version 2.2 Travel Model prior to fall 2008: Five multiclass assignments.....	203
Table 86 Traffic assignment in the Version 2.3.66 Travel Model: Six multiclass assignments .....	204
Table 87 Time Valuation by Vehicle Type and Time Period (minutes/dollar, in year-2007 prices).....	205
Table 88 Inputs to traffic assignment process .....	205
Table 89 Outputs of traffic assignment process .....	206
Table 90 Variables included in the final iteration, loaded highway network (i4_Assign_output.net) .....	206
Table 91 Inputs to transit assignment process .....	210
Table 92 Outputs of transit assignment process .....	210
Table 93 Mapping/concatenation of trip tables by trip purposes into peak and off-peak period trip tables prior to transit assignment.....	211

## **List of Equations**

Equation 1 Converting tolls into time-equivalent minutes of impedance.....	119
Equation 2 Composite time .....	156
Equation 3 Maximum walk distance formula, used for generating walk-access-to-transit links .....	186
Equation 4 PNR-to-station time/impedance .....	191
Equation 5 Daily non-transit-related parking cost for area types 1-3 .....	195

## **Appendices**

- A. Flowcharts
- B. Batch Files (in a separate volume)
- C. Cube Voyager Scripts (in a separate volume)
- D. AEMS Fortran Control Files (in a separate volume)

## **Colophon**

This report was created using Microsoft Word 2016 and Visio 2016, and was converted to a PDF file using Adobe Acrobat Pro DC. This report consists of two sections: the main report and the appendices. The files for these two sections can be found in a folder on COG's internal file server (I:\ateam\docum\fy17\model\_doc\_user\_guide).

## 1 Introduction

The National Capital Region Transportation Planning Board (NC RTPB or simply TPB) is the federally designated Metropolitan Planning Organization (MPO) for the Washington, D.C. metropolitan area and is also one of several policy boards that operate at the Metropolitan Washington Council of Governments (M WCOG or simply COG). The TPB is staffed by COG's Department of Transportation Planning (DTP). The COG/TPB staff develops and maintains, with consultant assistance, a series of regional travel demand models that are used for the regional transportation planning process in the Washington, D.C. area. These regional travel demand models are developed under the guidance of the Travel Forecasting Subcommittee (TFS), a subcommittee of TPB's Technical Committee. At any given time, the COG/TPB staff maintains at least two regional travel demand models: an adopted, production-use model and a developmental model. The production-use model is the one that is used planning studies conducted by COG/TPB and is made available to outside parties.<sup>1</sup> The developmental model is the one that is currently under development by COG/TPB staff, and is generally not made available to outside parties, since it is not yet considered a finished product.

This report explains how to setup and run the TPB Travel Demand Forecasting Model, **Version 2.3.66, which is the latest in a series of adopted, regional, production-use travel demand models, within the "Version 2.3" model family**, developed by the COG/TPB staff for regional transportation planning work in the Washington, D.C. metropolitan area. The previous adopted, regional, production-use travel demand model was the Ver. 2.3.57a model. All of the model versions in the Version 2.3 model family are aggregate, trip-based, four-step travel demand models.

Between 2008 and 2011, the TPB Version 2.3 travel model was calibrated to year-2007 conditions and this work was documented in a calibration report.<sup>2</sup> In 2013, the Version 2.3 Travel Model was validated to year-2010 conditions,<sup>3</sup> with an emphasis on validating the model's highway assignment results. Updates to the model resulting from this validation work were part of Build 52 of the Version 2.3 Travel Model (a.k.a., Ver. 2.3.52). Although the model was validated to year-2010 conditions, it was not recalibrated, so the January 20, 2012 calibration report remains the latest documentation for calibration

---

<sup>1</sup> The procedures for requesting the model can be found on the "Data Requests" webpage (<https://www.mwcog.org/transportation/data-and-tools/modeling/data-requests/>).

<sup>2</sup> Ronald Milone et al., "Calibration Report for the TPB Travel Forecasting Model, Version 2.3, on the 3,722-Zone Area System," Final Report (Washington, D.C.: National Capital Region Transportation Planning Board, January 20, 2012), <http://www.mwcog.org/transportation/activities/models/documentation.asp>.

<sup>3</sup> Ronald Milone to Files, "2010 Validation of the Version 2.3 Travel Demand Model," Memorandum, (June 30, 2013).

work. The previous model version, 2.3.57a, is documented in a user's guide.<sup>4 5</sup> This current user's guide is derived from the previous user's guide.

## 1.1 Adoption of the model by the TPB

The TPB does not *explicitly* adopt a particular version of the regional travel demand model. Instead, the adoption is made *implicitly* when the TPB adopts both 1) a given version of the Constrained, Long-Range Transportation Plan (CLRP) and 2) the findings from an air quality conformity analysis that determines that implementation of the given CLRP would result in mobile/vehicle emissions that are consistent with (conform to) the state air quality implementation plans (SIPs). The implicit adoption of a particular model version occurs by virtue of the fact that that model version was used for the analysis that was adopted by the TPB. Consequently, the Ver. 2.3.66 TPB travel demand model became the adopted, production-use model on Nov. 16, 2016, when the TPB adopted the following two resolutions:

- R3-2017: Resolution finding that the 2016 Constrained, Long-Range Plan Amendment and the FY 2017-2022 Transportation Improvement Program conform with the requirements of the Clean Air Act Amendments of 1990, as represented by the state air quality implementation plans (SIPs).
- R4-2017: Resolution approving of the 2016 Amendment to the Constrained, Long-Range Transportation Plan for the National Capital Region.

## 1.2 History of the Version 2.3 Travel Model

The Version 2.3 travel model represents a family of model versions. The first instance of the Version 2.3 model became the adopted regional travel model for the Washington, D.C. metropolitan area on November 16, 2011. In 2012, a newer version of the model, known as Build 39 of the Version 2.3 Travel Model, or Ver. 2.3.39, was used for the air quality conformity analysis of the 2012 Constrained Long-Range Plan and the FY 2013-2018 Transportation Improvement Plan. In 2013, Build 52 of the model was used for the air quality conformity analysis of the 2013 CLRP and FY 2013-2018 TIP. In 2015, Build 57a of the Version 2.3 model (Ver. 2.3.57a) became the production-use model. Lastly, as noted above, in 2016, the Ver. 2.3.66 model became the adopted model. Below is a list of milestones in the development of the TPB regional travel demand model from 2008 to 2015:

- March 1, 2008: TPB Version 2.2 Travel Model was released.<sup>6</sup>

---

<sup>4</sup> Ronald Milone, Mark Moran, and Meseret Seifu, "User's Guide for the COG/TPB Travel Demand Forecasting Model, Version 2.3.57a: Volume 1 of 2: Main Report and Appendix A (Flowcharts)" (Washington, D.C.: Metropolitan Washington Council of Governments, National Capital Region Transportation Planning Board, October 29, 2015), <https://www.mwcog.org/transportation/data-and-tools/modeling/model-documentation/>.

<sup>5</sup> Ronald Milone, Mark Moran, and Meseret Seifu, "User's Guide for the COG/TPB Travel Demand Forecasting Model, Version 2.3.57a: Volume 2 of 2: Appendices B (Batch Files), C (Cube Voyager Scripts), and D (AEMS Fortran Control Files)" (Washington, D.C.: Metropolitan Washington Council of Governments, National Capital Region Transportation Planning Board, October 29, 2015).

<sup>6</sup> Ronald Milone et al., *TPB Travel Forecasting Model, Version 2.2: Specification, Validation, and User's Guide* (Washington, D.C.: Metropolitan Washington Council of Governments, National Capital Region Transportation Planning Board, March 1, 2008), <http://www.mwcog.org/transportation/activities/models/documentation.asp>.



- The Version 2.2 travel model was developed on the 2,191-TAZ area system and most of its component models were estimated and calibrated with data from the COG/TPB 1994 Household Travel Survey (HTS), which included about 4,800 households.
- June 30, 2008: Draft TPB Version 2.3 Travel Model was released.<sup>7</sup>
  - At the time when the Version 2.2 Travel Model was released, a parallel effort was also underway to combine a nested-logit mode choice (NL MC) model and revised truck models into the Version 2.2 framework. This development effort proved to be viable and resulted in a release of what was then called the “draft Version 2.3 travel model” in June of 2008. The draft Version 2.3 model, like Version 2.2, was developed on the 2,191-TAZ area system.
  - The draft Version 2.3 model was not brought into production given that two related events were in motion during 2008. First, a new round of travel data collection was underway, including a major regional household travel survey (the COG/TPB 2007/2008 Household Travel Survey, which included about 11,000 households) and two transit on-board surveys (a bus on-board survey and a Metrorail passenger survey). Second, a new TAZ system was in development. The new zone system was envisioned to be developed over the same geographic area as the 2,191-TAZ system (6,800 square miles), but with smaller average zone sizes. TPB staff ultimately decided that the draft Version 2.3 Travel Model should not become the approved regional travel model until it incorporated the new zone system and the new data from the 2007/2008 Household Travel Survey (HTS) and the on-board transit surveys.
- February 28, 2011: TPB Version 2.3 Travel Model, Build 9, was released.<sup>8</sup>
  - From 2008 to 2012, TPB staff conducted the following activities:
    - Compiling and cleaning new survey data.
    - Preparing calibration files based on the new 3,722 TAZ system
    - Estimating and calibrating various sub-models in the regional travel model.
- November 11, 2011: TPB Version 2.3 Travel Model, Build 36, was released.<sup>9</sup> This is the model that became the adopted regional travel model for the for the Washington, D.C. metropolitan area on November 16, 2011.
- December 21, 2011: TPB Version 2.3 Travel Model, Build 38, was released. This model was documented in January 2012.<sup>10</sup>

---

<sup>7</sup> Ronald Milone et al., “TPB Travel Forecasting Model, Version 2.3: Specification, Validation, and User’s Guide,” Draft report (Washington, D.C.: Metropolitan Washington Council of Governments, National Capital Region Transportation Planning Board, June 30, 2008).

<sup>8</sup> Ronald Milone et al., “TPB Version 2.3 Travel Forecasting Model for the 3,722-Zone Area System: Calibration Report,” Draft report (Washington, D.C.: National Capital Region Transportation Planning Board, February 28, 2011).

<sup>9</sup> Ronald Milone et al., “Calibration Report for the TPB Travel Forecasting Model, Version 2.3.36, on the 3,722-Zone Area System,” Draft report (Washington, D.C.: National Capital Region Transportation Planning Board, November 18, 2011), <http://www.mwcog.org/uploads/committee-documents/aF1fV1xW20111118131827.pdf>.

<sup>10</sup> Ronald Milone et al., “User’s Guide for the TPB Travel Forecasting Model, Version 2.3, Build 38, on the 3,722-Zone Area System,” Final Report (Washington, D.C.: National Capital Region Transportation Planning Board, January 20, 2012), <http://www.mwcog.org/transportation/activities/models/documentation.asp>.

- July 17, 2013: TPB Version 2.3 Travel Model, Build 52 became the production-use travel model.
- October 15, 2014: The Version 2.3.57 model became the production-use travel model.
- October 21, 2015: The Version 2.3.57a model became the production-use travel model.

### **1.3 Recent changes to the model: Ver. 2.3.57a to Ver. 2.3.66**

There have been nine updates to the regional travel demand model since the previously adopted model (Ver. 2.3.57a), which is indicated by the fact that the third index in the model number, sometimes known as the model build, has been incremented by 9. The updates are listed Table 1 and are described in more detail later in this report. There are five types of updates:

- Bug fix (3 updates)
- Updated vendor/Citilabs software (1 update)
- New functionality (2 updates)
- Enhanced usability (4 updates)
- Improved documentation, i.e., added or corrected annotation/comments in scripts (2 updates)

Some model updates are classified as belonging to more than one type of update (as shown in Table 1), which is why there are 12 updates in the above list, not simply 9. Bug fixes are the most important updates and have the highest priority in terms of being incorporated into the travel demand model code. These are cases where someone (TPB staff or outside party) has identified a specific error in the model code. The TPB travel model makes use of Citilabs Cube Voyager software. So, an update to the Citilabs (vendor) software results in a de-facto update to the TPB model.

Five of these nine updates resulted in a change in the output of the model, as shown in Table 1, although the changes to model output were generally quite small. The next section of this report describes the nine model updates listed in Table 1.

Table 1 Summary of changes between the two most recent versions of the TPB travel demand model: Ver. 2.3.57a and Ver. 2.3.66

#	ID	Description	Type of update	Benefit(s)	Drawback(s)	Effect on model results
1	2.3.57b	Migrate from Cube Voyager 6.1 SP1 (32-bit) to Cube Voyager 6.4.1 (64-bit)	Updated vendor software	Faster model run times (12% decrease)	New licensing mechanism	Regional VMT increased 0.0042%
2	2.3.57c	Sort the areawalk.txt file by TAZ number	Enhanced usability	Easier to compare areawalk.txt files	None	None
3	2.3.57d	Fix two bugs in autoacc4.s (now called autoacc5.s)	Bug fix	1) Corrected TAZ representing center of the region; 2) Eliminated unintended drive-access links to center of the region	None	About 0.02% increase in regional VMT and 0.02% decrease in transit person trips
4	2.3.57e	Updates to the local-bus, in-vehicle speed degradation model	Enhanced usability, new functionality, & bug fix	1) Fewer updates for modeler to keep track of; 2) Updated transit (bus) run times stored in transit line files (mode*.tb), instead of simply applying a factor during skimming	Assumes RUNTIME keyword always occurs in specific columns	Total transit trips decreased by 0.2% and VMT remained the same in the modeled area; transit ridership on certain local bus routes decreased, which is expected.
5	2.3.57f	Add comments/annotation to Transit_Skims_BM.s	Improved documentation	More readable code/scripts	None	None
6	2.3.57g	Fix the issue of double counting bridge penalties in path building	Bug fix	Eliminated double counting of river-crossing time penalty	None	Small change
7	2.3.57i	Modeling HOT-lanes and toll setting: Streamline the existing process (Feng Xie)	Enhanced usability	Reduces model run time for toll setting	None	None
8	2.3.57k	Modeling HOT-lanes and toll setting: Implementing HOV Skim replacement (HSR) procedure for HOV2+, not simply HOV3+ (Anant Choudhary)	New functionality	Ability to model 2-occupant HOT lanes (not just the current 3-occupant HOT lanes).	None. Affects only 2017 & 2020 forecasts (HOT3 after that)	Regional VMT increases by 0.1%. Tolls increase from approx. \$10 to \$20 in peak period/peak direction from Beltway to Roosevelt Bridge
9	2.3.57m	Updates to the automated process to generate transit walksheds, MWCOG_ArcPy_Walkshed_Process.py	Enhanced usability & improved documentation	Enhanced readability of ArcPy walkshed generator script	None	None

### 1.3.1 Update 1: Migrate from Cube Voyager 6.1 SP1 (32-bit) to Cube Voyager 6.4.1 (64-bit)

**Update type:** Updated vendor (Citilabs) software

**Effect on model results?** Yes (see below)

**Description:** The TPB travel demand model makes use of Citilabs Cube Voyager software. The previous version of the TPB model, Ver. 2.3.57a, used Cube Voyager ver. 6.1 SP1, which is 32-bit software. Citilabs now makes a newer, 64-bit version of Cube Voyager, known as Cube Voyager 6.4. Most new computer workstations and servers have 64-bit architecture and run a 64-bit version of the Windows operating system. By moving to a 64-bit version of Cube, the model should be able to take advantage of more memory (though this is not currently a restricting factor for our model runs) and also faster model run times. In our tests conducted in early 2016, we found that simply switching from the 32-bit version of Cube (6.1 SP1) to the 64-bit version of Cube (6.4.1) resulted in a reduction of model run times of about 12%. For example, a model run that took 17 hours before would now take about 15 hours. This was one of the main motivations for making this switch. The other motivation is that software vendors are more likely to make updates and improvements to their newer software packages. One drawback to the switch was that TPB staff found the new licensing mechanism used in Cube 6.4 to be trickier to use than the older system, though this could be less of an issue for agencies that do not have multiple seats of the software. The change on model results/output was very modest. For example, “regional VMT increases 0.0042% and transit decreases 0.04% when switching to Cube 6.4.1.”<sup>11</sup> For agencies that have a current software license with Citilabs, there is no extra cost to make the upgrade.

### 1.3.2 Update 2: Sort the areawalk.txt file by TAZ number

**Update type:** Enhanced usability

**Effect on model results?** None

**Description:** A transit walkshed is defined as the area that is within walking distance to transit service. The transit walkshed is used to calculate the share of each zone that is within walking distance to transit. The current TPB travel demand model uses two distance thresholds for its transit walkshed: 0.5-mile (short walk) and 1.0-mile (long walk). In the past, a modeler had to conduct an off-line procedure to generate transit walksheds and calculate the share of each TAZ within walking distance to transit. Recently, we have been using an automated process, that is integrated into the first steps of the travel demand model. This automated process uses a Python/ArcPy script, which stores the area of each TAZ that is within a short-walk or a long-walk to transit in a text file (areawalk.txt). In the past, the areawalk.txt file was not sorted by TAZ number; now it is. This does not result in any change to the model results, and, for many users of the model, this will not make a difference. The primary benefit is

---

<sup>11</sup> Dzung Ngo and Mark S. Moran to Ronald Milone et al., “Benchmark Tests on Travel Model Server #7 (tms7) to Determine the Configuration for the Server’s Hard Drives and the Potential Use of Cube’s 64-Bit Version,” Memorandum, (February 2, 2016), 9.

for users who want to compare two or more areawalk.txt files (using programs such as Windiff or Winmerge). In this case, the sorted files are much more convenient to compare.

### 1.3.3 Update 3: Fix two bugs in autoacc4.s

**Update type:** Bug fix

**Effect on model results?** Yes (see below)

**Description:** The auto access script (autoacc4.s) is used to generate drive-access links between TAZs and transit stations, which are needed to build drive-access transit paths in the model. Kiss-and-ride (KNR) drive-access links connect directly to stations. Park-and-ride (PNR) drive-access links connect to the station's parking lot. This process began as a Fortran program developed by AECOM. Over time, the Fortran program was converted to a Cube TP+/Voyager script by Ron Milone. Although there has never been formal documentation for the program, one of the features of the program has always been the use of a backtracking penalty when generating drive-access links, which represented the fact that travelers do not like to backtrack when travelling to a destination. To model this aversion to backtracking, the program assumes that the dominant travel direction is toward downtown DC. Thus, one of the parameters in the program is the TAZ that represents downtown DC/the center of the region. This was originally set to be the TAZ that includes The Ellipse, the park just to the south of The White House. When the program was first developed, the zone system in use was the 2191-TAZ zone system and the Ellipse corresponded to TAZ #35. When COG moved to the newer 3722-TAZ system, the TAZ representing The Ellipse became TAZ #8, but this value was never updated in the autoacc program. In addition to updating this small bug, a second bug was identified and fixed. Although the Ellipse/CBD TAZ is used for the purposes of calculating whether one is backtracking, it was not intended to be designated as a transit station or PNR lot. Nonetheless, it was found that erroneous drive-access-to-transit links were being generated to the Ellipse/CBD TAZ. Both these bugs have been fixed and the program is now called autoacc5.s. It was found that the revised drive-access process caused some small changes to the modeled results. Specifically, after the change, the 2040\_final scenario exhibited a 0.02% increase in regional VMT and a 0.02% decrease in total transit person trips. This was in line with expectations, since it was expected that transit trips would drop slightly, due to the small reduction in drive-access links connecting various zones in DC with the CBD/Ellipse area.

### 1.3.4 Update 4: Updates to the local-bus, in-vehicle speed degradation model

**Update type:** Enhanced usability, new functionality, & bug fix

**Effect on model results?** Yes (see below)

**Description:** In the Version 2.3 model (both 2.3.57a and 2.3.66), it is assumed that the speeds of local bus routes will be affected by traffic congestion. Specifically, as congestion gets worse, the speeds of local bus routes are decreased, using what is known as the local-bus, in-vehicle speed degradation model (it is assumed that express bus and rail modes do not experience a speed degradation). Although one could develop a system that directly uses congested link speeds, this is not the approach that has been chosen by TPB staff, due to concerns that a hyper-congested link could result in an unrealistically

slow speed for one or more local bus routes that traverse that link. In the mid-2000s, TPB staff developed a process that would account for the fact that the speed of local bus (modes 1, 6, and 8) service would drop over time due to increasing congestion, but it was based on the region-level effects of congestion, not link level. In 2015, TPB staff identified two issues with the current process:

- The text-formatted lookup file containing bus speed factors must be manually adjusted to make sure that factors used in the skimming process reflect the appropriate modeled year. This manual adjustment can be cumbersome and prone to human error;
- While the transit skimming process, and subsequent mode choice process, accounts for future-year bus speed degradation, the transit assignment process does not currently account for bus speed degradation. This presents an inconsistency between modeling steps.

These two issues have been addressed with the revised process. In the revised process, instead of making adjustments to the bus in-vehicle travel time skims, the adjustments are made to the transit line files, by writing out new/adjusted RUNTIME values. In order to work correctly, the RUNTIME values must occur in specific columns (columns 32-29) in the transit line files. Two tests were performed to check the results on model output. The first measured the impact of all the changes, **minus** the inconsistency issue, where the bus speed degradation was not accounted for in transit assignment. In this test, there were no changes to model output. The second measured the impact of all the changes, **including** accounting for bus speed degradation in transit assignment. In the second test (forecast 2040 conditions), the total transit trips decreased by 0.2% and VMT remained the same in the modeled area, which was consistent with a-priori expectations of transit ridership decreasing on certain local bus routes.

### 1.3.5 Update 5: Add comments/annotation to the script use for transit skimming for the bus/Metrorail transit sub-mode

**Update type:** Improved documentation

**Effect on model results?** None

**Description:** The Transit\_Skims\_BM.s script is used for transit path-building and path-skimming associated with the bus/Metrorail transit sub-mode, one of four transit sub-modes in the current model. This change simply added new comments/annotation to the Transit\_Skims\_BM.s script to make it clearer to someone reading the program. Specifically, at about line 391, next to this statement (“IF ((MW[1] + MW[2] + MW[6] = 0 ) || (MW[3]+MW[5]=0))”) the following clarifying comment was added: “If the shortest path found in path building, which builds paths for all bus and Metro modes, has a Metro component = 0 or a bus component = 0, then this path will not be considered in the BM category and it will be equal to zero for this specific category, but will be included in AB or MR category.” This change made no material change to the script and had no effect on modeled results.

### 1.3.6 Update 6: Fix the issue of double counting bridge penalties in path building

**Update type:** Bug fix

**Effect on model results?** Yes (see below)

**Description:** Many regional travel demand models add time penalties to bridges, to account for the fact that people have a hesitation to make a river crossing, all other things being the same, due to the fact that routes that make use of bridges have fewer diversion options. In the TPB model there is an 11-minute perceived time penalty at the Potomac River from the Woodrow Wilson Bridge to the Harpers Ferry Bridge. This river crossing time penalty is introduced in the highway network building process and is used in both highway skimming and highway assignment. As noted in a recent memo, "TPB staff spotted a potential discrepancy with the implementation of this river crossing time penalty. Highway skimming accumulates highway skims along the shortest path built on highway impedances. While this time penalty is considered once and only once in highway skims, it seems to have been double-counted in developing highway impedances that precede the development of highway skims. As a result, this double-counting has a direct effect on path development, subsequently affecting model outputs."<sup>12</sup> This bug fix removes the double-counting of the river-crossing time penalty in the path-building phase, prior to path skimming. As would be expected, implementing this bug fix resulted in a small change in modeled results (however, this test was performed using the Ver. 2.3.52 model, not the Ver. 2.3.57a model). The following scripts were updated:

Highway\_Skims\_am.s, Highway\_Skims\_md.s, Highway\_Skims\_mod\_am.s, and  
Highway\_Skims\_mod\_md.s

### 1.3.7 Update 7: Modeling HOT-lanes/toll setting: Streamline the existing process

**Update type:** Enhanced usability

**Effect on model results?** None

**Description:** Many users of the TPB travel demand model make use of its ability to model HOV facilities and HOT-lane facilities. The current procedure for the treatment of tolls, HOV facilities, and HOT lanes is found in our current user's guide<sup>13</sup> and calibration report.<sup>14</sup> Prior to running the travel model, one must calculate the likely toll values for HOT lanes. TPB staff developed a semi-automated procedure for estimating toll values on HOT lane facilities (known as "toll setting"), which is described in a 2012 memo.<sup>15</sup> The update described in this section is about adding further automation to the toll setting routine. Since many of the users of the model do not conduct toll setting, this particular update will likely be of less interest to standard users of the model. The added automation is described in a recent memo.<sup>16</sup> One of the major benefit of using this update (for those who perform the toll setting process) is that it reduced the amount of time necessary to estimate tolls. This has no effect on the values of the

---

<sup>12</sup> Feng Xie to Dusan Vuksan, Ronald Milone, and Mark Moran, "Potomac River Crossing Time Penalty Sensitivity Analysis," Memorandum, (March 15, 2016).

<sup>13</sup> Milone, Moran, and Seifu, "User's Guide for the COG/TPB Travel Demand Forecasting Model, Version 2.3.57a: Volume 1 of 2: Main Report and Appendix A (Flowcharts)."

<sup>14</sup> Milone et al., "Calibration Report for the TPB Travel Forecasting Model, Version 2.3."

<sup>15</sup> Jinchul Park to Files, "HOT Lane Modeling Process of MWCOG/TPB (Draft)," Memorandum, (October 12, 2012).

<sup>16</sup> Feng Xie to Files, "An Integrated Toll Setting Procedure for COG/TPB's Version 2.3 Travel Demand Model," Draft Memorandum, (December 8, 2015).

estimated HOT lanes that come out of the toll setting process, so it does not have any effect on modeled results.

### 1.3.8 Update 8: Modeling HOT-lanes/toll setting: Implementing HOV Skim replacement procedure for HOV2+, not simply HOV3+

**Update type:** New functionality

**Effect on model results?** Yes (see below)

**Description:** COG/TPB staff has been using the regional travel demand model for modeling high-occupancy toll (HOT) lanes for a number of years. Up to now, the only type of HOT lane existing in the region has been one where the threshold auto occupancy for free usage of the facility has been three or more people (HOT 3+). The process used in the model to address this issue is called the HOV3+ skim substitution/replacement (HSR), or the multi-run traffic assignment.<sup>17</sup> However, for the air quality conformity analysis of the 2016 CLRP, we needed to update this procedure, since VDOT has recommended managing the I-66 HOT lanes inside the Beltway as a HOT 2+ facility beginning in 2017. As this was the first proposed occurrence of a HOT 2+ facility in the region, staff tested the impacts of the new HOV2 skim replacement process to assess:

1. The impact of HOV 2 skim replacement on regional modeling results (in addition to HOV 3+ replacement)
2. The estimated level of tolls on a HOT 2+ facility

The first test was designed to evaluate any anomalies that may result from implementing two skim replacement procedures (both HOV 2 and HOV 3+), while the second test was designated to assess how high the tolls might get on a HOT 2+ facility. The 2017 analysis showed that:

1. HOV2 skim replacement process did not have a significant impact on regional modeling results.
2. Managing I-66 Inside the Beltway as a HOT 2+ facility doubled the peak period and peak direction tolls from approximately \$10, end-to-end, to \$20.

With respect to the finding #2 above, given that both the model (2015 analysis year from the 2015 CLRP) and the observed data<sup>18</sup> show that there is already current congestion on this facility when it operates as HOV 2+ (for example, the model shows estimated V/C >1 on portions of the facility), the high tolls in the HOT 2+ case essentially serve as a deterrent to single-occupant vehicles (SOVs) from buying their way in. As a result, some of the links in the Falls Church area show zero SOVs using the facility in peak direction. Therefore, these findings are logical based on the way that the model operates.

---

<sup>17</sup> See, for example, Milone, Moran, and Seifu, "User's Guide for the COG/TPB Travel Demand Forecasting Model, Version 2.3.57a: Volume 1 of 2: Main Report and Appendix A (Flowcharts)," 23–25.

<sup>18</sup> Skycomp, Inc., "Traffic Quality on the Metropolitan Washington Area Freeway System, Spring 2014 Report," Draft Report (Washington, D.C.: National Capital Region Transportation Planning Board, Metropolitan Washington Council of Governments, November 18, 2014), <http://www.mwcog.org/uploads/committee-documents/YF1XV1db20150227142340.pdf>.



As there was minimal impact on regional model output, and since the toll-setting algorithm worked as expected, staff included this update in the regional travel demand model.

### **1.3.9 Update 9: Updates to the automated process to generate transit walksheds**

**Update type:** Enhanced usability & improved documentation

**Effect on model results?** None

**Description:** This updated involved making minor updates to the ArcPy script (MWCOG\_ArcPy\_Walkshed\_Process.py) used to generate transit walkshed and calculate the share of each TAZ within walking distance to transit. Updates included the following:

1. Incorporating the sorting of the AreaWalk.txt file.
2. Program comments/annotation: Added a table showing compatibility of various versions of Cube Voyager with various versions of ArcGIS or ArcGIS Engine Runtime (see Table 18 on p. 104).
3. Added path information allowing the process to run on a computer with ArcGIS 10.3 or ArcGIS Engine Runtime 10.3. The latter is included with Cube 6.4.
4. Re-formatted the script to shorten the length of its lines. Most lines are now 80 characters or fewer. This helps make the code more readable and more maintainable.



## 2 Overview of the model

The TPB Version 2.3 family of travel models is a classic, aggregate, “four-step,” trip-based, regional travel demand model. The four steps in a classic travel demand model are

- Trip generation
- Trip distribution
- Mode choice
- Traffic assignment<sup>19</sup>

The first three steps deal with estimating current-year or future-year demand for travel. The last step, traffic assignment, is where the demand for travel is assigned to a transportation network. This final step represents an equilibration between the transportation demand and the transportation supply. In many models, traffic assignment includes only a highway assignment, where private-use motor vehicles are assigned to a roadway network. In larger urban areas with extensive transit systems, there is often also a transit assignment, in addition to the highway assignment. The TPB travel model includes both a highway assignment and a transit assignment. So-called “four-step” models are trip based, meaning that trips are the basic unit of analysis, and are also “aggregate,” meaning that the model represents aggregate person flows and aggregate vehicle flows between transportation analysis zones (TAZs). In other words, these models do not model trips that happen within an individual TAZ (intra-zonal trips) and they do not simulate the movement of individual people or individual vehicles.

A highway assignment can be conducted at one of three different scales: microscopic, mesoscopic, or macroscopic. The TPB Version 2.3 Travel Model highway assignment is a **macroscopic**, static traffic assignment. This is the standard practice for almost every four-step model used in the United States. To better understand the meaning of a **macroscopic** traffic assignment, it is useful to understand the two other scales of assignment: microscopic and mesoscopic. In a **microscopic** traffic assignment, individual vehicles are modeled, using a small time step, such as every second. In a **mesoscopic** traffic assignment, platoons of vehicles are modeled, with a demand that varies though the assignment period (e.g., the AM peak hour demand is higher than the demand found in the shoulder hours of the AM peak period). By contrast, in a **macroscopic** traffic assignment, all traffic moving from one zone to another zone is modeled, but demand does not vary within the assignment period (e.g., a constant demand is assumed for all three hours in the AM peak period). Although it would seem appealing to use a microscopic or mesoscopic assignment in a regional travel demand model, these fine-grained assignments are almost never used in regional travel demand models since they would take too long to run and would require, at the regional level, too much input data (e.g., information about the traffic control devices and signal timings at every intersection). Thus, a macroscopic traffic assignment is usually the norm for regional travel demand models. As noted in a recent TRB report, “While there is much ongoing research into the

---

<sup>19</sup> The Version 2.3 family of travel models actually has six major steps. The two additional steps are “demographic sub-models” and the “time-of-day model.” All six of these steps are described in section 2.3 of this report, beginning on page 6.

use of dynamic assignment and traffic simulation procedures, the state of the practice for regional travel models remains static equilibrium assignment.”<sup>20</sup>

## 2.1 Model inputs

The major inputs and outputs of the regional travel demand model are shown in Figure 1. The travel model requires three major inputs:

- Zone-level land activity forecasts for year/scenario X;
- Transportation networks (both highway and transit) for year/scenario X; and
- Transportation policy assumptions for year/scenario X.

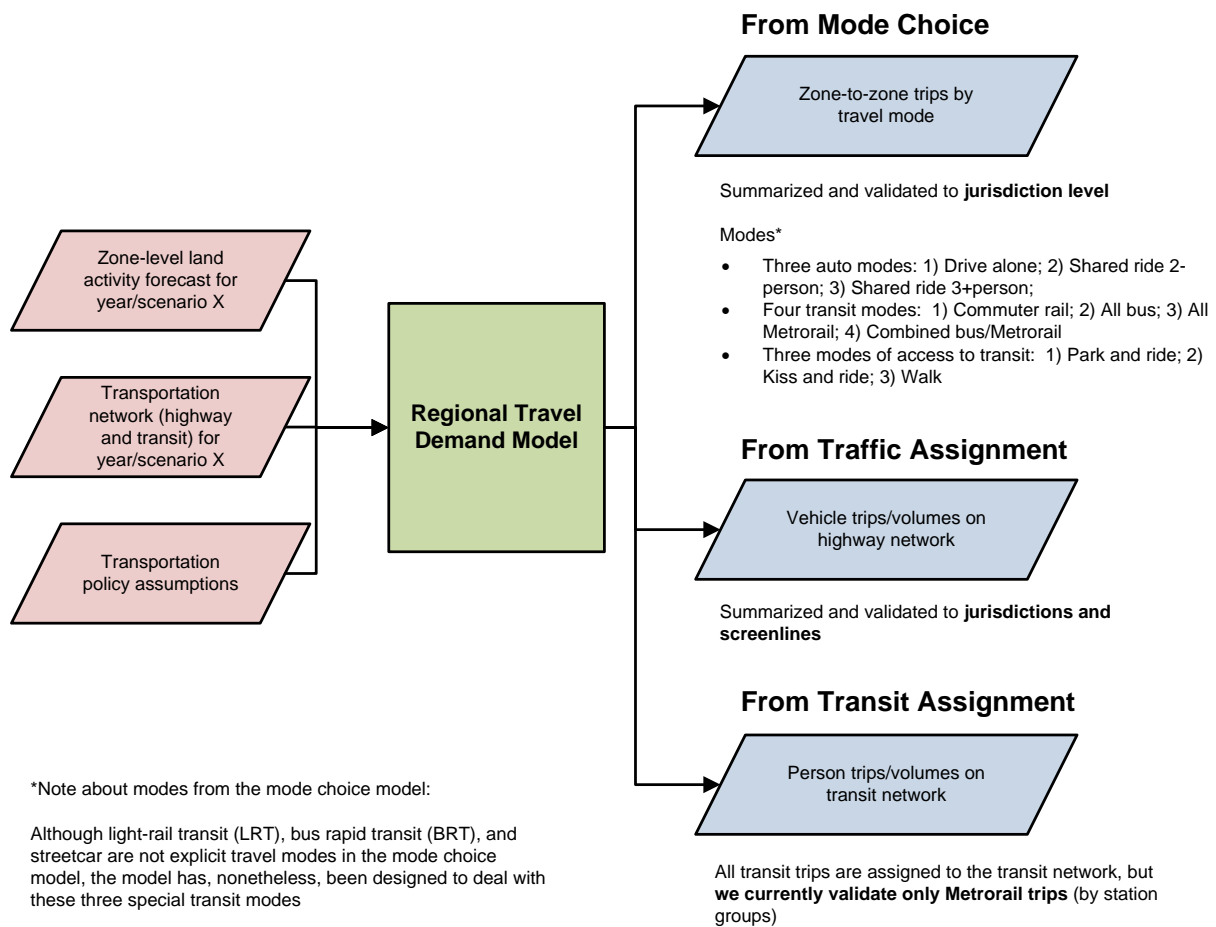


Figure 1 Major inputs and outputs of the TPB Version 2.3 Travel Model

Ref: travel\_model\_flowchart\_overview\_v3.vsd

<sup>20</sup> Cambridge Systematics, Inc. et al., *NCHRP Report 716: Travel Demand Forecasting: Parameters and Techniques*, National Cooperative Highway Research Program (Washington, D.C.: Transportation Research Board of the National Academies, 2012), 74, <http://www.trb.org/Main/Blurbs/167055.aspx>.

The zone-level land activity forecasts are developed by COG's Cooperative Forecasting Program, working through its Cooperative Forecasting and Data Subcommittee.<sup>21</sup> COG does not use a formal land use model. In the early 1970s, COG tried using a land use model called EMPIRIC,<sup>22</sup> but COG staff was not satisfied with its performance, and later abandoned its use.<sup>23</sup> Instead of a land use model, like many MPOs and regional planning agencies, COG uses a process, often known as a "modified Delphi process," which involves reconciling top-down and bottom-up land activity forecasts.<sup>24</sup> The top-down forecasts are regional econometric projections of employment, population, and households. The bottom-up forecasts are also projections of employment, population, and households, but made at the zone level and are based on information from the local governments. These bottom-up forecasts are derived from both building permits (providing short-term information) and comprehensive land use plans (providing long-term information). Each update of the zone-level, land activity forecasts in the Cooperative Forecasting program is called a "round" and the latest update is Round 9.0.

Before the zone-level land activity data can be used as an input to the travel model, it must undergo an adjustment process, known as the CTPP-based employment adjustment, which ensures that a consistent employment definition is used by all counties and jurisdictions in the modeled area. The reason for this adjustment is that different jurisdictions in the modeled area, which covers DC, Maryland, Virginia, and one county in West Virginia, use different definitions of employment. For example, jurisdictions in the Baltimore region and several other Maryland jurisdictions develop their base-year employment estimates using data from Bureau of Economic Analysis (BEA). By contrast, most of the jurisdictions in the Washington region develop their base-year employment estimates using data from the Quarterly Census of Employment and Wages (QCEW) collected by the Bureau of Labor Statistics (BLS).<sup>25</sup> The QCEW is a joint federal/state cooperative arrangement between the BLS and state employment security agencies (ESAs). According to Spear, "In lieu of using the publicly available QCEW database, some state DOTs (and even some MPOs) have entered into formal agreements with their state ESAs to obtain access to the enhanced QCEW microdata files that are used by BLS to develop the QCEW... [The QCEW files] are more commonly known in the transportation community as ES-202 data, but this terminology

---

<sup>21</sup> "Cooperative Forecasting and Data Subcommittee," *Metropolitan Washington Council of Governments*, 2016, <https://www.mwcog.org/committees/cooperative-forecasting-and-data-subcommittee/>.

<sup>22</sup> Peat, Marwick, Mitchell and Company, "EMPIRIC Activity Allocation Model: Application to the Washington Metropolitan Region" (Metropolitan Washington Council of Governments, 1972).

<sup>23</sup> Reid Ewing and Keith Bartholomew, "Comparing Land Use Forecasting Methods: Expert Panel Versus Spatial Interaction Model," *Journal of the American Planning Association* 75, no. 3 (2009): 347.

<sup>24</sup> Paul DesJardin, "Round 8.4 Cooperative Forecasts" (March 18, 2015 meeting of the National Capital Region Transportation Planning Board, held at the Metropolitan Washington Council of Governments, Washington, D.C., March 18, 2015).

<sup>25</sup> Robert E. Griffiths to Ronald Milone, "Travel Model Employment Data Adjustment Factors for Round 7.0," Memorandum, (August 10, 2005), 1.

is no longer used by BLS.”<sup>26</sup> The most recent CTPP-based employment adjustment was conducted on Round 9.0 land activity data.<sup>27</sup>

As for the transportation networks, COG/TPB staff develops a series of highway and transit networks for the air quality conformity analysis, and these networks are often used as the starting point for other planning studies. The highway network consists of all freeways, expressways, and major arterials in the modeled area. It also includes many minor arterials and some collectors, but almost no local roads (centroid connectors represent local roads, but one centroid connector may represent many local roads, so there is not a one-to-one representation like one finds for other link types in the highway network). The highway network forms the base layer for the transit network, since buses mostly make use of the highway network. In addition to the highway network, the transit network includes the following elements:

- Transit infrastructure: Transit-only links
- Transfer links
- Transit service
- Transit fares

The latest full-scale documentation of the transportation networks was done in 2016.<sup>28</sup>

Transportation policy assumptions include the following:

- Assumptions about how transportation costs will increase over time, e.g.,
  - Will transit fares rise at the same rate as inflation or a different rate?
  - How will auto operating costs change over time?
- Cost of parking;
  - For drive-access transit trips, the cost of parking is stored in the station file. For park-and-ride (PNR)-to-station transfer links, the walk time is a function of parking capacity and parking cost,<sup>29</sup> but parking cost is not used as part of the transit path-building.
  - For driving trips not involving transit, a parking cost model is used, where parking cost is a function of employment density (see section 21.7.1 “Non-transit-related parking costs”).

---

<sup>26</sup> Bruce D. Spear, “NCHRP 08-36, Task 098: Improving Employment Data for Transportation Planning” (Washington, D.C.: American Association of State Highway and Transportation Officials (AASHTO), Standing Committee on Planning, September 2011), ES-7, [http://onlinepubs.trb.org/onlinepubs/nchrp/docs/NCHRP08-36\(98\)\\_FR.pdf](http://onlinepubs.trb.org/onlinepubs/nchrp/docs/NCHRP08-36(98)_FR.pdf).

<sup>27</sup> Dzung Ngo to Mark Moran and Ronald Milone, “Developing Land Use Input Files for the Version 2.3 Travel Model Using Round 9.0 Cooperative Forecasts and the CTPP-Based Employment Adjustment Factors,” Memorandum, (May 10, 2016).

<sup>28</sup> Ronald Milone et al., “Highway and Transit Networks for the Version 2.3.57a Travel Model, Based on the 2015 CLRP and FY 2015-2020 TIP,” Final Report (Washington, D.C.: Metropolitan Washington Council of Governments, National Capital Region Transportation Planning Board, March 18, 2016), <https://www.mwcog.org/transportation/data-and-tools/modeling/model-documentation/>.

<sup>29</sup> Manish Jain to Ronald Milone and Mark Moran, “MWCOC Network Coding Guide for Nested Logit Model (First Draft: September 20, 2007; Updated February 2008 and October 2010),” Memorandum, (October 2010), 6.

- Amount of in-commuting from areas outside the modeled cordon.

## 2.2 Model outputs

The travel model produces a large number of outputs. **Each model run produces about 25 GB of output files.** Since many of these are intermediate files, a clean-up process has been added to the model that moves these intermediate/temporary files to a folder where they can be easily deleted. **Once these are deleted, the amount of output files per model run is about 10 GB.** As stated earlier, the travel model is an “aggregate” model meaning that the model represents aggregate person flows and aggregate vehicle flows between transportation analysis zones (TAZs). Nonetheless, the model produces many fine-grained outputs. These include link-level outputs, such as the number of vehicles traveling on a particular link in the AM peak period, and zone-interchange-level outputs, such as the number of bus person trips traveling from TAZ X to TAZ Y. However, although the model *produces* these fine-grained outputs, **the model has not been validated to these fine-grained levels, so it is not recommended that one use these fine-grained outputs from the travel model.** A general rule is that, before using or reporting any model outputs, they should be summarized or aggregated to the same, or a higher, level as was used in model validation. For example, although the model produces link-level traffic volumes, this information should be aggregated to the screenline level, jurisdiction level, or regional level, before it is used or reported. Despite this rule, these fine-grained outputs are sometimes used in corridor-level or project-level planning studies, but typically only after the outputs have undergone post-processing (see, for example the classic report NCHRP 255,<sup>30</sup> or its recent update, NCHRP 765<sup>31</sup>). Given the regional nature of most of the transportation planning studies conducted for the TPB, the COG/TPB staff rarely conducts this type of post-processing work (it is more commonly conducted by consultants working for local governments or state DOTs). In conclusion, when using outputs of the regional travel demand model, one should generally use outputs that have been aggregated or summarized to the following levels:

- Region level, e.g.,
  - The modeled area,
  - The TPB planning area,
  - The metropolitan statistical area (MSA), or
  - One of the air quality non-attainment areas, which can vary by pollutant.<sup>32</sup>
- Jurisdiction level
- Jurisdiction-to-jurisdiction level
- For highway assignments: Regional screenlines

---

<sup>30</sup> Neil J. Pedersen and D. R. Samdahl, *NCHRP Report 255: Highway Traffic Data for Urbanized Area Project Planning and Design*, National Cooperative Highway Research Program (NCHRP) (Transportation Research Board, National Research Council, 1982), <http://trid.trb.org/view/1982/M/188432>.

<sup>31</sup> CDM Smith et al., *NCHRP Report 765: Analytical Travel Forecasting Approaches for Project-Level Planning and Design*, National Cooperative Highway Research Program (NCHRP) (Transportation Research Board of the National Academies, 2014).

<sup>32</sup> The modeled area is the largest of these regional areas.

- For transit assignments: Metrorail station groups

Figure 1 shows the three major outputs of the travel model, listing both the disaggregate-level output and the more aggregate-level output that is recommended for use. Table 2 adds some detail to the information found in Figure 1.

**Table 2 Outputs of the travel model: Disaggregate-level output that is not validated versus aggregate-level output that is validated**

<b>Model producing the output</b>	<b>Disaggregate-level output (Produced by the model, but not recommended for use)</b>	<b>Aggregate-level output (recommended for use)</b>
Mode choice	Zone-to-zone trips by travel mode	<ul style="list-style-type: none"> <li>• Jurisdiction-to-jurisdiction flows</li> <li>• Jurisdiction-level mode splits</li> <li>• Region-level mode splits</li> </ul>
Traffic assignment	Vehicle trips/volumes on the road links	<ul style="list-style-type: none"> <li>• Jurisdiction-level metrics, such as VMT by jurisdiction</li> <li>• Screenline-level metrics, such as total number of vehicles crossing screenline</li> </ul>
Transit assignment	Transit person trips/volumes on transit links	Although all transit person trips are assigned to the transit network, we currently validate only Metrorail trips, and these are validated only by station groups (generally three to four stations per group).

### 2.3 Modeling steps and the speed feedback loop

The major steps of the Version 2.3 Travel Model, including major inputs and outputs, can be found in Figure 2. As mentioned earlier, the major inputs are the transportation networks, the zonal land use data, and the transportation policy assumptions. The model itself, which is delineated in Figure 2 by a gray, dashed-line forming a box, begins with demographic models and ends with traffic assignment and transit assignment. Each of the steps of the travel model is discussed in subsequent chapters of the user's guide.



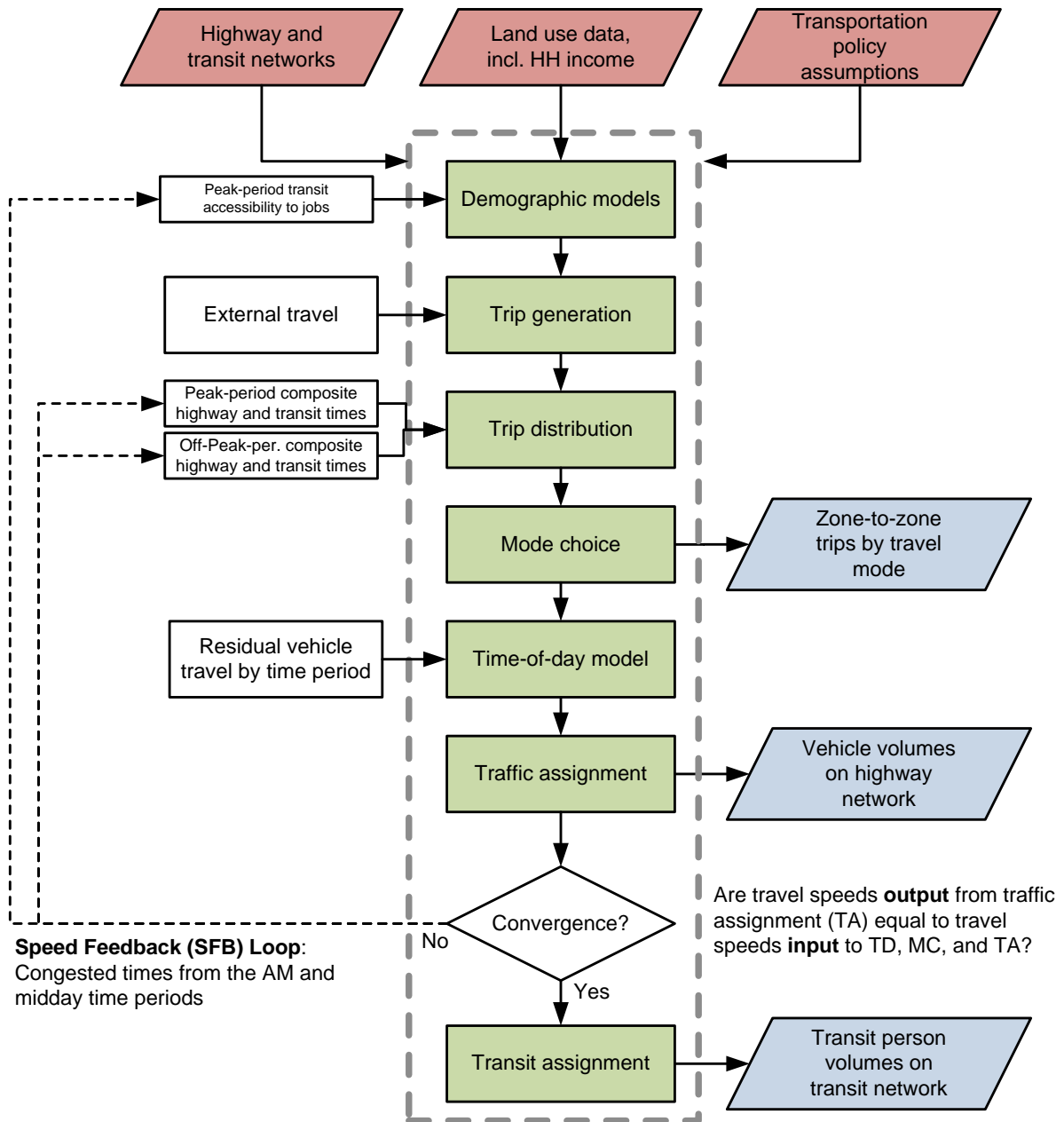


Figure 2 Major steps of the Version 2.3 Travel Model

Ref: six\_step\_model\_ver2.3\_v2.vsd

As can be seen in Figure 2, the Version 2.3 Travel Model uses a speed feedback (SFB) loop to ensure that the travel times (and hence speeds) coming out of traffic assignment are consistent with those used as inputs to trip distribution and mode choice. In theory, at the end of each SFB loop, the model would have a test to determine whether convergence has been met. In other words, one could continue running iterations of the speed feedback loop until a convergence stopping criterion has been met. Currently, however, we do not use a convergence-based stopping criterion. Instead, based on past tests with the regional model, we have determined that the model is sufficiently converged after four SFB

iterations,<sup>33</sup> so we simply use a fixed number of SFB loop iterations (five iterations, including the initialization iteration, known as the “pump prime” iteration). In the future, we may use a more formal convergence-based stopping criterion for the SFB loop, such as the percent root-mean-square error (%RMSE) of the travel skims.<sup>34</sup> Nonetheless, in 2011, Cambridge Systematics was unable to find any MPOs that used a formal stopping criterion for the SFB loop.<sup>35</sup> The SFB loop and the volume averaging method used in the SFB loop are discussed in more detail later in this section.

Traffic assignment is discussed both here and in its own chapter (Chapter 23). Like most travel models in the U.S., the Version 2.3 Travel Model uses a user-equilibrium (UE) traffic assignment, which is the generally accepted method for static traffic assignments. Furthermore, the assignment process is a multi-class UE assignment, meaning that separate user classes can be assigned at the same time. The Version 2.3 model includes six user classes:

1. Single-occupant vehicle (SOV)
2. High-occupant vehicle with two persons (HOV2)
3. High-occupant vehicle with three+ persons (HOV3+)
4. Medium and heavy trucks
5. Commercial vehicles
6. Airport passengers traveling to/from the three commercial airports

Additionally, the Version 2.3 model includes four time-of-day periods for traffic assignment:

- AM peak period (3 hours: 6:00 AM to 9:00 AM)
- Midday period (6 hours: 9:00 AM to 3:00 PM)
- PM peak period (4 hours: 3:00 PM to 7:00 PM)
- Night/early morning period (11 hours: 7:00 PM to 6:00 AM)

Most MPOs use a UE traffic assignment that relies on an optimization algorithm known as the Frank-Wolfe (FW) algorithm.<sup>36</sup> The FW algorithm is essentially a series of all-or-nothing traffic assignments where flows are combined using weights from an optimization process whose goal is to minimize an objective function. The process stops when a stopping criterion is met. Previously, the Version 2.3 Travel Model used the following UE stopping criterion: When the relative gap  $\leq 10^{-3}$  OR the number of UE

---

<sup>33</sup> Ron Milone, “TPB Models Development Status Report” (January 23, 2009 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., January 23, 2009), [http://www.mwcog.org/transportation/committee/committee/archives.asp?COMMITTEE\\_ID=43](http://www.mwcog.org/transportation/committee/committee/archives.asp?COMMITTEE_ID=43).

<sup>34</sup> See, for example, Caliper Corporation, “Traffic Assignment and Feedback Research to Support Improved Travel Forecasting,” Final Report (Washington, D.C.: Federal Transit Administration, Office of Planning and Environment, July 31, 2015), pages 3-2 and 3-4, <http://www.fta.dot.gov/documents/traffic-assignment-and-feedback-research-to-support-improved-travel-forecasting.pdf>.

<sup>35</sup> Cambridge Systematics, Inc., “Fiscal Year 2010 Task Reports,” Final Report (Washington, D.C.: National Capital Region Transportation Planning Board, November 16, 2010), 1-20-21, <http://www.mwcog.org/transportation/activities/models/review.asp>.

<sup>36</sup> Marguerite Frank and Philip Wolfe, “An Algorithm for Quadratic Programming,” *Naval Research Logistics Quarterly* 3, no. 1–2 (1956): 95–110, doi:10.1002/nav.3800030109.

iterations  $\geq 300$ . The relative gap threshold was always intended to be the primary stopping criterion, with the number of UE iterations functioning as a backup criterion. Now, however, we have moved to what we call a “progressive” relative gap stopping criterion. The idea is that, in the early SFB iterations, the UE closure criterion will be relatively loose, but, in the later SFB iterations, the UE closure criterion will tighten, as shown in Table 3.

**Table 3 User equilibrium closure criterion (relative gap) varies by speed feedback iteration**

Speed feedback iteration	Primary closure criterion for UE traffic assignment	Secondary closure criteria for UE traffic assignment
Pump prime	Relative gap $\leq 10^{-2}$ (i.e., 0.01)	Number of UE iterations $\geq 1000$
1	Relative gap $\leq 10^{-2}$ (i.e., 0.01)	Number of UE iterations $\geq 1000$
2	Relative gap $\leq 10^{-2}$ (i.e., 0.01)	Number of UE iterations $\geq 1000$
3	Relative gap $\leq 10^{-3}$ (i.e., 0.001)	Number of UE iterations $\geq 1000$
4	Relative gap $\leq 10^{-4}$ (i.e., 0.0001)	Number of UE iterations $\geq 1000$

By using the higher value for UE iterations (1000 vs. 300), we were able to ensure that this secondary criterion is unlikely to be used as the stopping criterion. Based on a series of sensitivity tests,<sup>37</sup> we found that the new progressive relative gap scheme results in a relatively converged traffic assignment, without the extremely lengthy model run times that would be needed if one were to use a high threshold (e.g.,  $10^{-4}$  relative gap) for each of the five SFB iterations. The Version 2.3 Travel Model uses a slight variation of the FW algorithm, called the *bi-conjugate* Frank-Wolfe algorithm, which converges marginally faster than the classic FW algorithm.

Regarding data inputs, the zonal land use data that is input to the travel model (Figure 2) includes information about average household wealth, in the form of an average household income index. This index is the ratio of the zonal median household income to the regional median household income, in year-2007 dollars. So-called “residual vehicle” or exogenous trips are added to the modeling stream at the time-of-day model stage. These trips include

- Through trips (auto and truck);
- Taxi trips;
- School trips;
- Visitor/tourist trips;
- Airport passenger trips (i.e., trips by air passengers destined to the three commercial airports in the region).

Figure 3 is another view of the Version 2.3 Travel Model, but with an emphasis on which steps occur *before* the speed feedback (SFB) loop and which steps occur *within* the SFB loop. Before the loop is

---

<sup>37</sup> Mark S. Moran and Ronald Milone, “Status Report on the Version 2.3 Travel Model: Updates to the Model and Year-2010 Validation” (March 22, 2013 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., March 22, 2013), 7–11.

begun, there is an initialization phase, known as the “pump-prime” iteration. In the pump prime iteration, a first pass of the travel model is performed using *initial* AM and off-peak highway speeds, and *initial* mode choice percentages (i.e., the mode choice model is not executed in the pump prime iteration). The “skimmed” highway times are used to develop drive-access-to-transit (zone-to-PNR-lot) links as part of the transit network. After the transit network is built and skimmed, trip generation and trip distribution are executed. The resulting person trips are converted to vehicle trips on the basis of default zone-level mode choice and car occupancy percentages, and are assigned to the highway network.

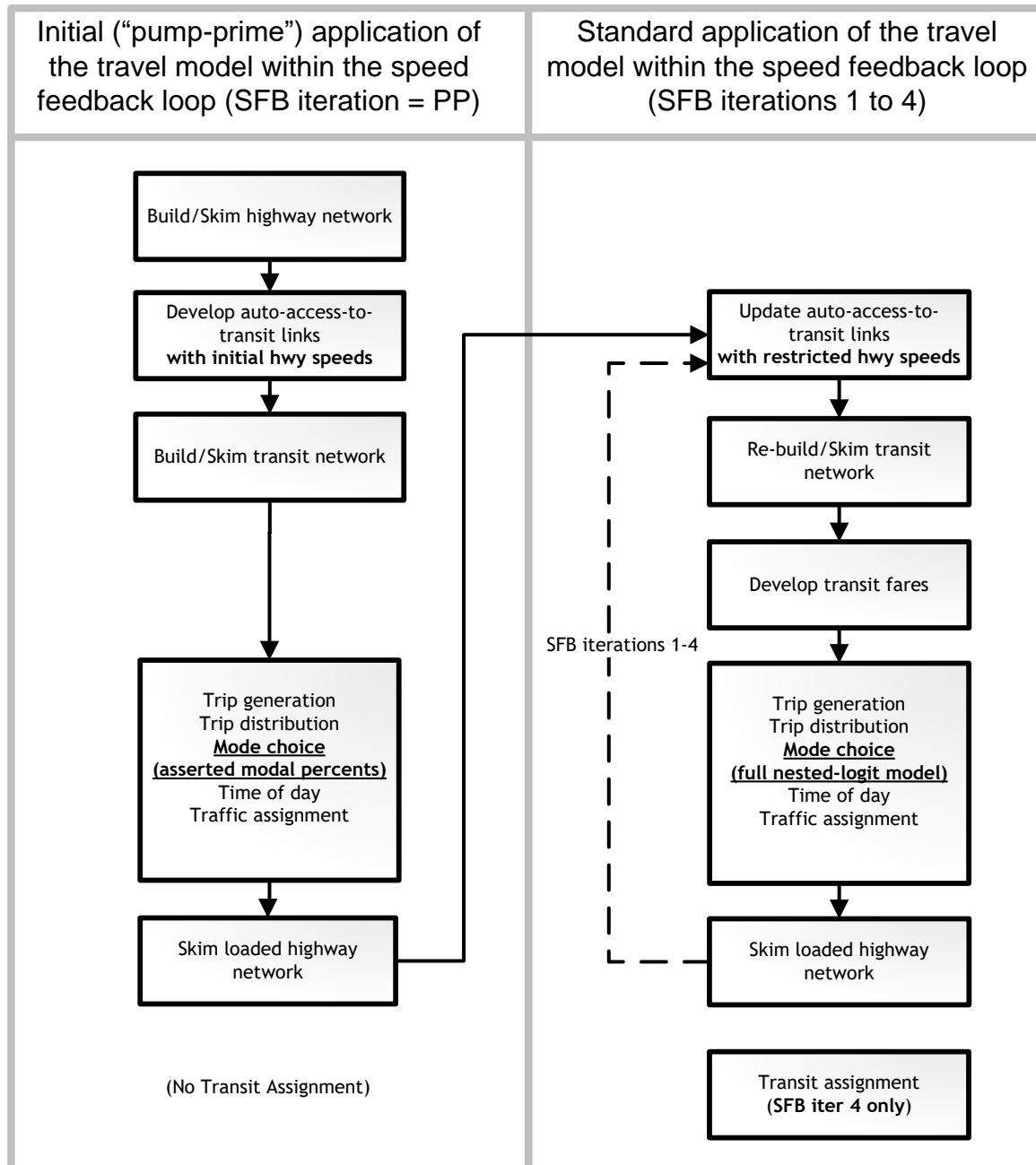


Figure 3 Application process of the Version 2.3 Travel Model

Ref: pumpPrime\_vs\_other\_sfb\_iter\_v5.vsd

The next series of “standard” SFB iterations (1 through 4) involve the execution of the complete travel model which includes: 1) a mode choice model execution and 2) the use of recycled traffic assignment-based speeds as input. The AM peak and off-peak restrained highway times are used to update the zone-to-PNR link speeds, and the transit network is re-built and skimmed. The highway and transit time skims are used as inputs to the mode choice model. The auto driver trips produced from the mode choice model are processed through the time-of-day model, which apportions the auto drivers among four time-of-day periods: the AM peak period (6 - 9 AM), the midday period (9 AM - 3 PM), the PM peak

period (3 - 7 PM), and the night/early morning period (7 PM - 6 AM). The four time-of-day trip tables are subsequently loaded onto the highway network in separate traffic assignment procedures. The loaded-link volumes are successively averaged using the method of successive averages (MSA) to facilitate the convergence of the final link speeds. The averaging occurs individually for each of the four time-of-day periods at the link level, as follows:

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

In both the Version 2.2 and 2.3 travel models, a fixed number of speed-feedback (SFB) iterations is used. The Version 2.2 model used six speed feedback iterations (in addition to the pump prime iteration). By contrast, the Version 2.3 model uses four speed feedback iterations (in addition to the pump prime iteration). The Version 2.3 model produces **two final loaded network files** called **i4\_Assign\_Output.net** and **i4\_HWY.NET** (not i6hwy.net, as was the case with the Version 2.2 Travel Model). The first file (**i4\_Assign\_Output.net**) is based on the direct trip table output, while **i4\_HWY.NET** is based on volume averaging. TPB staff use **i4\_Assign\_Output.net** to compute the emissions as a part of air quality conformity and SIP work.

As shown in both Figure 2 and Figure 3, transit assignment is not conducted within each speed feedback loop, but is instead conducted once, after the final feedback iteration is complete. Transit assignment is conducted for two time-of-day periods (peak and off-peak) using trip tables in production/attraction (P/A) format (not origin/destination format, as is the case for highway assignment) and, unlike highway assignment, transit assignment is not capacity constrained. Although both these aspects of transit assignment may seem like shortcomings, the state of the practice for regional transit assignments is P/A assignment without capacity constraint.<sup>38</sup>

## 2.4 Special modeling procedures

COG/TPB staff has developed two special modeling procedures, which are part of the Version 2.3 Travel Model, to handle two specific phenomena: The Metrorail constraint through the regional core, and modeling High Occupancy/Toll (HOT) lanes. Since, up to now, all HOT lanes in this region have been designed such that HOVs with 3+ persons travel for free, the special procedure for modeling HOT lanes, up to now, has been focused on the HOV3+ occupancy requirement, and an HOV skim replacement (HSR) procedure, described below, which is geared to this occupancy requirement. However, VDOT has recently introduced a proposed improvement to I-66 inside the Beltway that set the HOV occupancy requirement for free use as HOV2+ (not HOV3+ like the other facilities). This would begin in 2017 and

---

<sup>38</sup> See, for example, Cambridge Systematics, Inc. et al., *NCHRP 716*, 77.

last through 2020. Consequently, the Ver. 2.3.66 model now includes three special modeling procedures, which are described below.

#### **2.4.1 Metrorail constraint through the regional core**

The Metrorail constraint through the regional core (sometimes referred to using the less precise term “transit constraint through the regional core”) is a technical adjustment to the trip tables coming out of the mode choice process designed to reflect a WMATA policy assumption that, during peak periods, the Metrorail system may have insufficient capacity to serve all the demand traveling to and through the regional core. Typically, it is assumed that the Metrorail system will be able to handle all of the peak-period demand to and through the regional core in the near term, but, since demand is growing through time, the system might not be able to handle all the peak-period demand at some future time, depending on the amount of growth in demand and the number of rail cars available in a given year. The assumed year at which the Metrorail system will be at its peak capacity during the peak periods to and through the regional core is known as the “binding year.” For years beyond the binding year, it is assumed that any growth in peak-period Metrorail demand to and through the regional core will be forced to switch to other travel modes (specifically, auto person trips). The Metrorail constraint was initiated by WMATA in 2000 to address funding shortfalls restricting the expansion of the rail fleet.<sup>39</sup> WMATA policy sets the binding year, which is currently set at 2020. This means that, for any forecast year past 2020, the Metrorail constraint is applied, i.e., any forecasted peak-period Metrorail trips that exceed the 2020 demand to and through the regional core are shifted to other travel modes (specifically, auto person trips). Details about the Metrorail constraint through the regional core, including a definition of the extent of the regional core can be found in section 21.2 on page 163, which is part of the “mode choice” section of this report.

#### **2.4.2 Modeling high occupancy/toll (HOT) lanes**

According to a recent FHWA report, “ ‘Managed lanes’ are defined as highway facilities or a set of lanes where operational strategies are proactively implemented and managed in response to changing conditions....Examples of operating managed lane projects include high-occupancy vehicle (HOV) lanes, value priced lanes, high-occupancy toll (HOT) lanes, or exclusive or special use lanes.”<sup>40</sup> Most HOT lane facilities exist on freeways which include one or more adjacent general purpose (GP) lanes that are not managed.

---

<sup>39</sup> Ronald Milone, “TPB Version 2.3 Travel Model on the 3,722-TAZ area system: Status report” (presented at the September 23, 2011 meeting of the Travel Forecasting Subcommittee of the Technical Committee of the National Capital Region Transportation Planning Board, held at the Metropolitan Washington Council of Governments, Washington, D.C., September 23, 2011).

<sup>40</sup> FHWA, “Managed Lanes: A Primer” (Washington, D.C.: U.S. Department of Transportation, Federal Highway Administration, 2008), 5, [http://www.ops.fhwa.dot.gov/publications/managelanes\\_primer/managed\\_lanes\\_primer.pdf](http://www.ops.fhwa.dot.gov/publications/managelanes_primer/managed_lanes_primer.pdf).

#### 2.4.2.1 HOT lanes which allow free use with 3+ occupants per vehicle

On Nov. 17, 2012, HOT lanes, known as the I-495 Express Lanes, opened on I-495 in Virginia. On this facility, vehicles with three or more occupants (HOV3+) may use the facility for free, but single-occupant vehicles (SOVs) and two-occupant vehicles (HOV2) must pay a toll to use the facility. The toll is dynamically set based on congestion levels. The toll is set such that the HOT lanes will remain free flowing. In December 2014, HOT lanes, known as the I-95 Express Lanes, opened on I-95. This second facility also had the same HOV restriction (HOV3+), so both facilities are HOT3+.

COG/TPB staff has developed a procedure for using the regional travel demand model to model HOT lanes. Since the first facilities were HOT3+, the modelling procedures were designed to represent HOT3+ facilities. Before the opening of the Express Toll Lanes, the Virginia Department of Transportation (VDOT) stated that the toll paying traffic in the HOT lanes should not impede the flow of HOV traffic in the HOT lanes, as per federal guidance.<sup>41</sup> To accommodate the VDOT/federal guidance, COG/TPB staff developed a modeling procedure where the travel model is run twice. The “**base run**” captures the travel time for unimpeded flow of HOV traffic on HOT lanes, consistent with VDOT/federal guidance. The “**final run**” of the travel model uses the HOV skims from the “base run” and uses all other (non-HOV) skims from the “final run.” Under this framework, the “base run” serves solely as a means for measuring times for HOV traffic on HOT facilities. This procedure is referred to as the “HOV 3+ skim substitution option,” the “**HOV3+ skim replacement (HSR)**,” or the “multi-run procedure” for modeling HOT lanes. Below is a more detailed description of the process from the Version 2.2 Travel Model documentation:<sup>42</sup>

---

A second procedure, known as the ‘HOV-3+ skims substitution’ option, is related to the modeling of planned HOT (High Occupancy Toll) lanes in Northern Virginia. Modeling this type of facility has proven to be especially challenging given:

1. The special operating characteristics of the facility (i.e., the tolls will change in real time to maintain a service speed at or near free-flow levels);
2. The stipulation that SOV’s and 2-occupant HOVs will pay for the HOT lane access while 3+ occupant HOVs will be allowed access at no charge; and
3. The stipulation that the HOV travel market will suffer no service degradation from the HOT lane operation.

---

<sup>41</sup> “Federal-Aid Highway Program Guidance on High Occupancy Vehicle (HOV) Lanes,” *U.S. Department of Transportation, Federal Highway Administration*, November 2012, <http://ops.fhwa.dot.gov/freewaymgmt/hovguidance/chapter3.htm>.

<sup>42</sup> Ronald Milone et al., “TPB Travel Forecasting Model, Version 2.2: Specification, Validation, and User’s Guide” (Washington, D.C.: Metropolitan Washington Council of Governments, National Capital Region Transportation Planning Board, March 1, 2008), 1-10-12.



Obviously, a central modeling objective in representing HOT lanes is to specify detailed toll rates that will result in demand levels that do not degrade the prevailing speed on the HOT facility. Another modeling objective is to ensure that HOV 3+ service levels will remain unaffected by the HOT operation. To achieve these two objectives, the following four steps are currently undertaken on a year-by-year basis to simulate HOT lanes in Virginia.

- 1) The travel model is fully executed whereby all Virginia HOT lanes in the highway network are coded as HOV 3+-priority lanes. The resulting HOV 3+ LOS skim files corresponding to each iteration (pump-prime, iteration 1,..., iteration 6) are preserved for later use. This step is known as the 'base' execution.
- 2) The travel model is fully executed again. This time, the Virginia HOT lanes are coded as mixed use lanes (i.e., allowing access by both SOVs and HOVs). The HOT lanes are assigned a 'straw' toll level of 20 cents per mile during the peak periods and 15 cents per mile during the off-peak. The toll is included into the overall highway impedance during path building in the traffic assignment step. Therefore, the toll level impacts the loading in the HOT lane facility.
- 3) The final (iteration 6) highway assignment process (including three traffic assignments for each time period) resulting from Step 2 is run iteratively, on a trial-and-error basis, to identify HOT lane toll rates (cents/mile) which yield optimum speeds on the facility. The toll rates are developed on a freeway segment by segment basis. The queuing delay function is disabled on the HOT lane facility during this particular step as such delay will presumably be minimal given the special operating conditions of the facility. At the end of each assignment execution, the segment level volume and speed is evaluated and the toll rate is adjusted incrementally. The toll level is increased when the desired segment speed is too low. The result of this process is a file containing 'final' toll rates for each HOT lane freeway segment, by time period.
- 4) The model is again executed as in Step 2, such that: 1) the final HOT lane toll rates developed in Step 3 are invoked and 2) the HOV 3+ skims developed in Step 1 are used as 'overrides' to the HOV 3+ skims that would be normally developed as part of the modeling process. The result of this execution produces the final loaded links (this step is referred to as the 'final' run).

In summary, TPB travel forecasts involving HOT-lane scenarios are developed using two separate model executions: 1) the 'base' run from which HOV 3+ skims are developed, and the 'final' run which uses specially developed HOT lane toll rates and the HOV 3+ skims from the base run.

Highway skims used by the mode choice model normally consist of time, distance and toll matrix tables. The TPB mode choice model distinguishes tolls on variably priced facilities from tolls charged on fixed price facilities. Variably priced facilities are those that levy differential toll charges by time period (such as the planned ICC electronic toll lanes (ETLs) in Maryland or the planned HOT lanes in Virginia). Fixed toll facilities are those that levy toll charges that do not fluctuate during the day. The TPB model expresses toll values on variably priced facilities as equivalent minutes that are added to the highway time, rather than as monetary values expressed in the toll matrix. Toll charges on fixed price facilities, however, are expressed as monetary values.

---

The HOT3+ modeling procedure makes use of the following batch file: HSR\_Highway\_Skims.bat.

#### 2.4.2.2 HOT lanes which allow free use with 2+ occupants per vehicle

VDOT is recommending managing the I-66 HOT lanes inside the Beltway as a HOT2+ facility beginning in 2017 and continuing through 2020. Before this point, the only HOT lane facilities in the region were HOT3+. COG/TPB staff has developed a new procedure to deal with HOT2+ lanes. Since this applies to only the years from 2017 to 2020, there are only two model years that are affected by this in the most recent air quality conformity analysis (2017 and 2020), which used networks from the 2016 CLRP. This results in a new batch file (HSR\_Highway\_Skims\_2017HOV2.bat), which is discussed later in this report.

#### 2.4.3 Select-link analyses

A select-link analysis (SLA) and a select-link assignment are common procedures in travel demand modeling, but these are not part of our standard modeling procedures. There are theoretical reasons why SLAs should not be performed,<sup>43</sup> but we often get requests for help with running SLAs. COG/TPB staff has developed some SLA procedures<sup>44</sup> for the Ver. 2.3 travel model, which may be requested by outside parties in the normal fashion (see <https://www.mwcog.org/transportation/data-and-tools/modeling/data-requests/>).

---

<sup>43</sup> See, for example, Hillel Bar-Gera and Amos Luzon, "Non-Unique Solutions of User-Equilibrium Assignments and Their Practical Implications," in *Compendium of Papers CD-ROM* (Transportation Research Board 85th Annual Meeting, held January 22-26, 2006, Washington, D.C., 2006).

<sup>44</sup> Feng Xie to Files, "Select Link Analysis for TPB's Version 2.3.57 Travel Demand Model," Draft Memorandum, (September 21, 2015).

### 3 Hardware and software requirements

This section of the report describes the hardware and software requirements for running the Version 2.3 family of travel models, with an emphasis on the Ver. 2.3.66 model. It also includes a section discussing the hardware used for modeling at COG. In addition to requirements, this section of the report also discusses any recommendations regarding hardware and software.

#### 3.1 Hardware

- Processor/central processing unit (CPU)/chip:
  - Intel or Intel-like processor, e.g., Intel, AMD, with 64-bit architecture (“x64”).
  - Number of cores: The Version 2.3.66 Travel Model has been designed to run some steps in parallel, using Cube Cluster.
    - During the highway assignment step, there can be up to 8 concurrent program threads running at once, which means that it is recommended that you have a computer with 8 or more cores. Nonetheless, you can run the regional travel model on a computer with only 4 or 2 cores (see instructions found in Table 17 on p. 88), however, due to rounding issues in Cube Cluster, running with fewer than the recommended 8 cores may result in slight differences in modeled results.
    - The two biggest chip manufacturers are Intel and AMD. Some Intel chips feature a technology known as Hyper-Threading. When Hyper-Threading technology is enabled on the chip, the operating system sees double the number of cores. So, if your computer has four cores and Hyper-Threading is enabled, the operating system will see eight virtual cores, thus doubling your CPU capacity. See the section 8.2.1 for more details. COG/TPB staff has executed the Version 2.3 family of travel models on only computers running Intel chips, but the model should run equally well on computers running AMD chips.
  - Chip/CPU speed: While there is no minimum chip speed, we have found that model run time scales inversely with chip speed, so a faster chip/CPU is always preferred. We recommend a chip speed of around 3 GHz.
- Memory: 64-bit versions of Windows can a large amount of memory (e.g., from 128 GB on Windows 10 Home to 2 TB on Windows 10 Pro). However, model runs are not typically memory intensive, so 3 to 4 GB of RAM should suffice.
- Storage space: We recommend you have at least 500 GB of free space on your computer storage (hard disk drive [HDD] or solid-state drive [SSD]). One modeling scenario/year generates about 25 GB of files (1,600 files) before the clean-up procedure is run, and about 10 GB of files after the clean-up procedure is run. A solid-state drive (SSD) could provide shorter model run times, but in one test we performed on a new travel model server, the SSD performed no better

than the hard drive.<sup>45</sup> This result was unexpected, since one would generally expect an SSD to out-perform an HDD. In this test, the data drive was an SSD and the operating system (O/S) drive was a HDD. We did not, however, have time to test the case where both the data drive and the O/S drive were an SSD.

## 3.2 Software

- Operating system: Microsoft Windows (64-bit version), such as Windows 10, Windows Server 2008, or Windows Server 2012. To our knowledge, the Ver. 2.3.66 model has not been tested at COG using Windows 10, but it should work.
- The Version 2.3.66 Travel Model: This is provided for free to those who request it. The procedures for requesting the model can be found on the “Data Requests” webpage (<https://www.mwcog.org/transportation/data-and-tools/modeling/data-requests/>). In short, one needs to send an e-mail or a signed letter to Ronald Milone, Director of Travel Forecasting and Emissions Analysis, in COG’s Department of Transportation Planning. The e-mail/letter should indicate how you intend to use the model, naming the specific study or research project, if applicable. When someone is given the TPB travel model, this includes the model inputs (principally transportation networks and land use data), batch files, Cube Voyager scripts, and the Fortran program used to run the mode choice model. Although Cube Voyager scripts are part of the model transmittal package, you cannot run these scripts without having a copy of Cube Voyager software (see below). Please note that the COG/TPB staff does not have the resources to staff a “help desk” for the regional travel model, so it is expected that individuals who request the travel model will have the knowledge and skills to use the model with minimal assistance from COG/TPB staff.
- Citilabs Cube software: The TPB Version 2.3.66 Travel Model is implemented using Citilabs Cube software, a proprietary software package, which is produced, licensed, and marketed by Citilabs, Inc. Thus, to run the regional travel model, you will need to purchase the Cube software from Citilabs ([www.citilabs.com](http://www.citilabs.com)). COG/TPB staff cannot provide copies of the Citilabs Cube software. The Version 2.3.66 Travel Model has been developed and applied by COG/TPB staff using Cube version 6.4.1, so it is recommended you have this version of the software. This is a 64-bit version of Cube.
  - Cube Base: Cube Base is the graphical user interface (GUI) for editing transportation networks, matrices, and scripts. In theory, Cube Base can also be used for managing network scenarios (Scenario Manager) and running travel models (Application Manager), but that is not how the Version 2.3 family of travel model have been implemented. Instead, the Version 2.3 model is implemented using a command-line interface (CLI), as described later in this report. **Note that Cube Base is actually 32-bit software (Cube Voyager is 64 bit).**

---

<sup>45</sup> Ngo and Moran to Milone et al., “Benchmark Tests on Travel Model Server #7 (tms7) to Determine the Configuration for the Server’s Hard Drives and the Potential Use of Cube’s 64-Bit Version,” 8.

- Cube Voyager: Cube Voyager is the numerical engine that powers the Cube suite of software and includes its own proprietary scripting language. This is 64-bit software.
- Cube Cluster: Cube Cluster is Citilabs' implementation of distributed processing, which is a technique for distributing computing jobs across multiple computers or processors, thus reducing model run times by allowing two or more processes to run in parallel. Strictly speaking, Cube Cluster is not required to run the Version 2.3 Travel Model. But it is strongly recommended, in order to keep model run times to a minimum, and, if you choose not to use it, you will have to modify the model setups that are supplied by COG/TPB staff (this is described later in this report).
- A text editor (optional, but recommended): The choice of which text editor to use is a personal one. Cube Base includes its own text editor, optimized, obviously, for editing Cube Voyager scripts. In addition to the Cube Base built-in text editor, COG staff uses one or more of the following: Notepad++ (free and open source), PSPad (free, but not open source), KEDIT (for purchase, though the software may no longer be supported by the vendor).
- Cygwin (<http://www.cygwin.com/>) is a Linux-like environment for Windows that provides a series of Unix-like command-line tools, such as head, tail, and which. This free and open source software is not required to run the travel model, but some functionality of the travel model summary process will not work properly without this software installed.
- ArcGIS Engine Runtime 10.1 or ArcGIS 10.1. ArcGIS Engine Runtime 10.1 comes with Cube 6.4.1. There is one Cube installation file (), which presumably checks to see if your computer already has ArcGIS installed. See Table 18 on p. 104 for more information.
- Software for comparing or diffing text files (not required, but recommended). COG staff uses both WinDiff, which is older, and WinMerge, which is newer and has more functionality. Both are available for free.

### 3.3 Examples of computer hardware used at COG for modeling

COG/TPB staff performs most modeling runs on computer servers that are dedicated for this task, though one can also run the travel model on a standard, desktop computer. COG/TPB staff typically accesses a travel model server (TMS) using a Remote Desktop Connection. We currently have four travel model servers, named tms4, tms5, tms6, and tms7. COG's Models Applications Group makes use of tms4 and tms6, and COG's Models Development Group makes use of tms5 and tms7.

Table 4 compares the computer specifications ("specs") of the latest travel model servers used by the Models Development Group (tms7) and the Models Applications Group (tms6). Both computers are running 64-bit versions of Windows. Tms7 is running Windows Server 2012 R2, and tms6 is running Windows Server 2008 Enterprise R2.

Table 4 Comparison of computer specs between tms6 and tms7f

Host Name:	Tms6	Tms7
OS Name:	MS Windows Server 2008 R2 Enterprise	MS Windows Server 2012 R2 Standard
OS Version:	6.1.7601 Service Pack 1 Build 7601	6.3.9600 Build 9600
System Manufacturer:	HP	HP
System Model:	ProLiant DL380p Gen8	ProLiant DL380 Gen9
System Type:	64-bit	64-bit
Number of processors:	2	2
Processor name(s):	Intel Xeon E5-2690	Intel Xeon E5-2687W V3
Clock speed of processor (GHz):	2.90	3.10
No. of cores/processor:	8	10
No. of threads/processor:	16	20
Total number of cores:	16	20
Total number of threads:	32	40
Hyper-Threading Technology:	Yes	Yes
Total Physical Memory (MB):	32,733 MB	32,640 MB
Available Physical Memory (MB):	30,421 MB	30,504 MB
Hard drives for data storage:		
	Local Disk (C:), 838 GB RAID 0	Local Disk (C:), 1 TB, RAID 1 (2 disks)
	<b>ATeam (D:), 2 TB RAID 0</b>	<b>Data (E:), 3.27 TB, RAID 5 (4 disks)</b>
	<b>BTeam (E:), 2 TB RAID 0</b>	SSD (F:), 186 GB, RAID 0 (for testing)
	ATeamArray (J:), 10 TB RAID 5 DAS	
	BTeamArray (K:), 10 TB RAID 5 DAS	

Ref: "I:\ateam\docum\fy17\memos\travel\_model\_server\_tms\_specs\_2016\_v2.xlsx"

On a 64-bit computer with a 64-bit version of windows, some applications are 32-bit applications whereas others are native 64-bit applications, and each type of application has its own installation folder, as shown below:

- Installation location for 64-bit applications: "C:\Program Files"
  - Example: Cube Voyager
- Installation location for 32-bit applications: "C:\Program Files (x86)"
  - Examples: Cube Base, WinMerge

Although both tms6 and tms7 have two processors, tms6 has 8 cores per processor, resulting in a total of 16 physical cores. By contrast, tms7 has 10 cores per processor, resulting in a total of 20 physical cores. Because of Intel's Hyper Threading Technology, each server appears to have double the number of cores. Thus, tms6 appears to the operating system as 32 virtual cores (which can handle 32 threads of instruction) and tms7 appears to the operating system as 40 virtual cores (which can handle 40 threads of instruction).

The processor clock speed for tms6 is 2.9 GHz. By contrast, tms7 has a processor clock speed of 3.1 GHz, which is about 7% faster.

One can use the total number of cores in a computer to determine the maximum number of concurrent model runs that can be conducted. Since the Version 2.3.66 model is set up to use a maximum of 8 threads/cores, three concurrent model runs require the simultaneous use of 24 (= 3 x 8) cores. Four concurrent model runs could require up to 32 (= 4 x 8) cores. In tests conducted on tms6, which has 32 virtual cores ("threads"), TPB staff found that we could run four concurrent model runs of the Ver. 2.3.57 model (the results should apply to the Ver. 2.3.66 model as well).<sup>46</sup> However, we also found that, if two or more users tried to launch concurrent model runs, even if it was only two users, each with one model run, one of the two model runs would often crash. Thus, when conducting concurrent model runs, it is better to make sure that they are launched by only one user. Also, based on recent communications with Citilabs (personal communication, 2/6/17), it is better not to overload the processor, so, although a 32-core computer should be able to run 4 concurrent model runs (4 x 8 = 32), it would be better to limit this computer to 3 concurrent model runs. It is hoped that further information about this issue will be added to future Cube documentation.

Travel model servers often have two logical disk drives: one containing the software, usually called "C:", and one used to store data, such as the model runs. Each one of these logical disks could actually be one or more disks, storage arrays, or, conceivably, solid state drives. The data drives associated with the travel model servers at COG are shown in Table 5.

**Table 5 Storage drives used for travel demand modeling**

Server	UNC Path	Mapped Drive		
		Letter	Size	Setup
nas	\\nas\TMSARCHIVE\MODELAPP	N:	13.9 TB	RAID 5
nas	\\nas\TMSARCHIVE\MODELDEV	O:	13.9 TB	RAID 5
sas	\\sas\ntp_sas\$	S:	649 GB	VM. Gets storage from the SAN.
tms4	\\tms4\D	Y:	2.4 TB	RAID 5
tms5	\\tms5\E	X:	4.5 TB	RAID 5
tms6	\\tms6\ateam	L:	2 TB	RAID 0
tms6	\\tms6\bteam	P:	2 TB	RAID 0
tms6	\\tms6\ateamarray	T:	10 TB	RAID 5 DAS
tms6	\\tms6\bteamarray	V:	10 TB	RAID 5 DAS
tms7	\\tms7\Data	M:	3.3 TB	RAID 5 (4 disks)

Ref: "I:\ateam\docum\fy17\model\_doc\_user\_guide\mapped\_drives\_cog\_2016.xlsx"

<sup>46</sup> Mark S. Moran and Dzung Ngo to Ronald Milone et al., "Stress Tests of Travel Model Server #6 (tms6) to Determine the Maximum Number of Model Runs That Can Run Concurrently," Memorandum, (October 29, 2014).

For example, tms4 has one data drive with a capacity of 2.4 TB. This drive is mapped as the D drive when logged on to tms4 and is mapped to the Y drive when not logged on to the server. By contrast, tms6 has four data drives. The UNC path for each data drive indicates both the server name (e.g., tms6) and the share name (e.g., ateam). Logical drives that are made of storage arrays consist of multiple physical disk drives, which can be configured in different ways to allow redundancy (using RAID, which stands for Redundant Array of Inexpensive Disks or Redundant Array of Independent Disks). RAID 0 provides no redundancy, but it can often be the fastest configuration. For example, ateamarray and bteamarray have been set up with RAID 5, so they have redundancy in the case of a hard drive crash.



## 4 Mechanics of the model application process

The Version 2.3 family of travel models, including Ver. 2.3.66, is applied using a command-line interface (CLI), not a graphical user interface (GUI). The model is launched via a single command that is typed or pasted in a single command window (this is covered in the section about running the model). The Version 2.3 Travel Model makes use of the following:

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

An example subdirectory structure for applying the Version 2.3 model is shown in Figure 4. The “root” subdirectory appears at the top of the structure. The root subdirectory may exist anywhere on the computer hard drive and may be arbitrarily named by the analyst, but **it is recommended that the name of the root subdirectory include information about both the travel model being used (e.g. Ver2.3.66) and the modeling project being undertaken.** For example, an analyst performing model runs to support the Air Quality Conformity Determination of the 2016 Constrained Long-Range Transportation Plan (CLRP) might name the root subdirectory as follows:

```
C:\modelRuns\fy17\Ver2.3.66_aqc_2016clrp
```

Note that the root subdirectory need not be located directly off the root of the C drive (or D drive, etc.). In the example above, the root subdirectory is below the “fy16” subdirectory. On the left side of Figure 4, there are five specially designated subdirectories under the root that are established:

- SOFTWARE: Fortran executable files and dynamic-link library (DLL) files
- CONTROLS: Control files that are required by the Fortran programs
- SCRIPTS: Cube Voyager scripts
- SUPPORT: General parameter files used by the scripts or other programs, such as AEMS (Fortran) and LineSum (C++)
- SUMMARY: Summary scripts, which are used to summarize the model run

The first four subdirectories are required, but the fifth subdirectory is optional. The SUPPORT subdirectory is reserved for parameter files that generally do not change by modeled scenario such as K-factors, F-factors, and the like. The four required subdirectories must exist under the root, and must be named as shown, although the names are not case sensitive. The optional summary subdirectory may be given any name. Furthermore, the files residing in these four required subdirectories should generally not be altered or renamed.

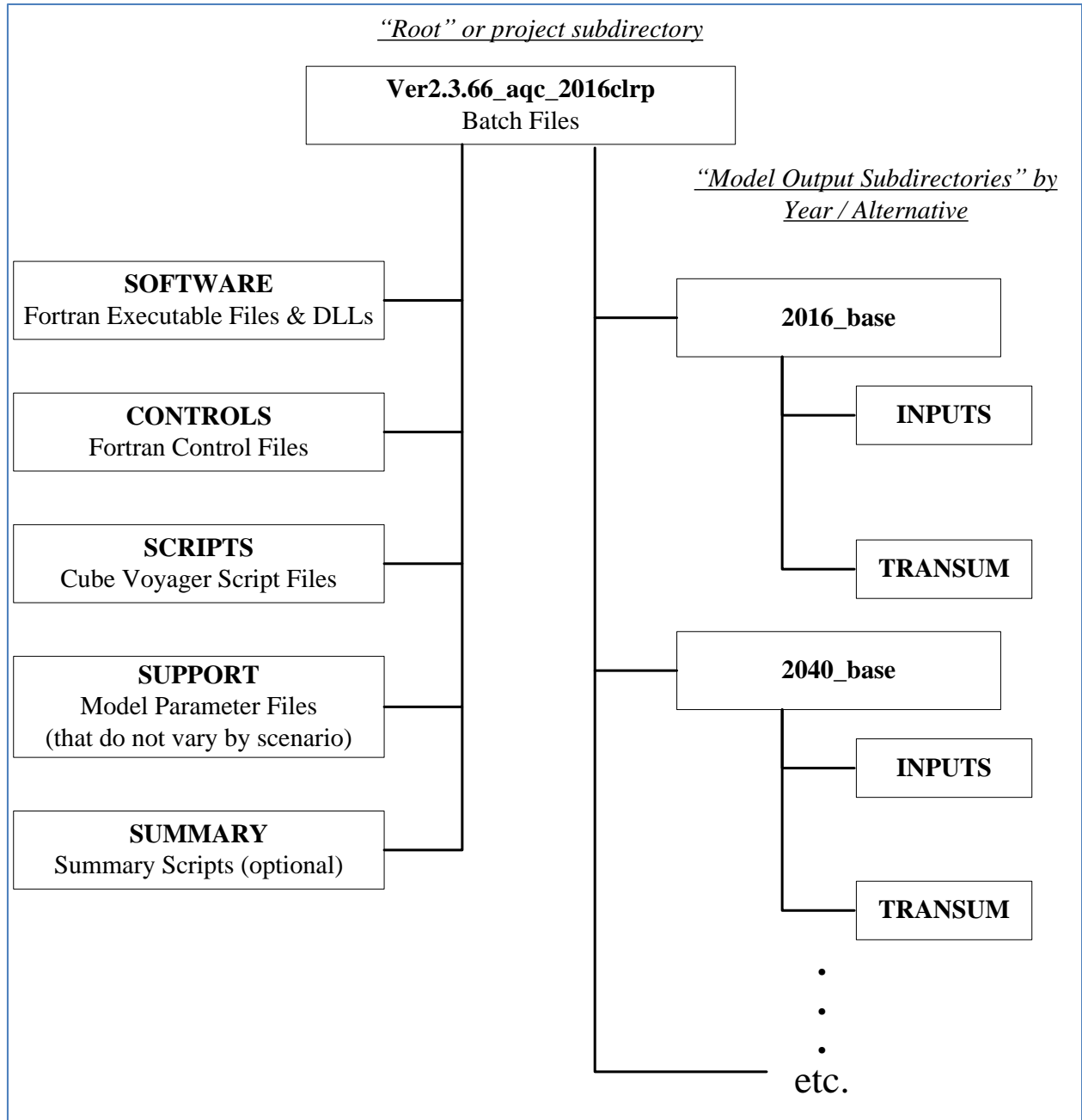


Figure 4 Subdirectory structure for executing the Version 2.3 Travel Model

Ref: directoryStruct\_v23\_model\_v8.vsd

The right side of Figure 4 shows two subdirectories, named “2015\_base” and “2040\_base.” These two subdirectories are the output subdirectories (a.k.a. the scenario subdirectories). The user is free to choose any name for output subdirectories. Since travel demand models are best used in a comparative manner, a travel demand modeling project would typically have two or more scenarios or alternatives. Each alternative would get its own output subdirectory for scenario-specific outputs from the travel model.

Under each scenario-specific subdirectory that exists, there need to be two subdirectories, one named "inputs" and the other named "transum." These exact names must be used, but, as stated previously, names are case insensitive. The "transum" subdirectory is for storing summary information about the transit assignment summary. At the beginning of the model run, the "transum" subdirectory will be completely empty, but, at the end of the run, the subdirectory will contain reports from the process (LineSum) that summarizes the transit assignment. The "inputs" subdirectory is where one stores all necessary model inputs that area specific to a modeled scenarios (see Table 6). Note that some "inputs" that are common to all modeled scenarios are stored in the "support" subdirectory (see Table 6, which also includes input files stored in the CONTROLS subdirectory). Input files in the "inputs" folder are named generically (e.g., land use data is stored in a file named zone.dbf; network link data is stored in a file named link.dbf, etc.). The user may establish an unlimited number of output subdirectories, as long as each one contains one "inputs" subdirectory and one "transum" subdirectory. Neither the inputs nor transum subdirectories can be shared among more than one alternative. After a model has been run, if the automated transit walkshed process was run with the model (the default setup), then the "inputs" folder will contain a new subfolder called "Transit\_Walksheds\_GIS."

One change to Table 6 that is associated with the Ver. 2.3.57 model and later models, the areawalk.txt file is no longer a necessary input file, since this file is generated during the automated transit walkshed process. If a copy of areawalk.txt already exists in the "inputs" folder, the old copy will be renamed as AreaWalk\_Old.txt, before the new areawalk.txt file is created.<sup>47</sup>

The actual structure of the folders/subdirectories storing the travel model, its input files, and output folders, will be a function of the years/scenarios analyzed and the way the Metrorail constraint and HOT-lane modeling procedures are applied, but the list below is representative:

```
+---2016_base
|   +---inputs
|   \---transum
+---2016_final
|   +---Inputs
|   \---transum
+---2040_base
|   +---Inputs
|   \---transum
+---2040_final
|   +---Inputs
|   \---transum
+---Controls
+---Docs
+---Scripts
+---Software
+---Summary
```

---

<sup>47</sup> AECOM, "FY 2014 Final Report, COG Contract 12-006: Assistance with Development and Application of the National Capital Region Transportation Planning Board Travel Demand Model" (National Capital Region Transportation Planning Board, Metropolitan Washington Council of Governments, August 18, 2014), 9–2, <http://www.mwcog.org/uploads/committee-documents/Y11YWFZd20140922110646.pdf>.

+---Support

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

- Run number/ID (a unique sequence number to quickly name a model run)
- Parent run number/ID (indicates the run number of the run that formed the basis for the current run). Useful in figuring which run was derived from which other runs.
- Subdirectory name (i.e., the name of the root folder/subdirectory)
- Key modeling assumption parameters, such as the network year, land use year, land use round (e.g., Round 8.4), WMATA tariff number, etc.
- Key modeling output parameters, such as model run time, regional VMT, total transit, etc.

Table 6 Input files needed to run the Version 2.3 Travel Model, stored in the CONTROLS, INPUTS, and SUPPORT folders

No.	Folder	Filename	Description	File Type	Category	Modeling step where file is used	Transit/ Non-Transit Mode(s)
1	controls	HBO_NL_MC.ct1	HBO nested-logit mode choice mode control file	Text	Model	Mode choice (MC_Purp.bat & AEMS.EXE)	
2	controls	HBS_NL_MC.ct1	HBS nested-logit mode choice mode control file	Text	Model	Mode choice (MC_Purp.bat & AEMS.EXE)	
3	controls	HBW_NL_MC.ct1	HBW nested-logit mode choice mode control file	Text	Model	Mode choice (MC_Purp.bat & AEMS.EXE)	
4	controls	lineSum_MR_access.ct1	Summary of Metrorail riders by access mode	Text	Summary	LineSum.exe	
5	controls	lineSum_MR_line.ct1	Summary of Metrorail boardings, alightings, and ridership	Text	Summary	LineSum.exe	
6	controls	LineSum_Volume.ct1	Consolidate peak & off-peak vols from transit assignment	Text	Summary	LineSum.exe	
7	controls	NHO_NL_MC.ct1	NHO nested-logit mode choice mode control file	Text	Model	Mode choice (MC_Purp.bat & AEMS.EXE)	
8	controls	NHW_NL_MC.ct1	NHW nested-logit mode choice mode control file	Text	Model	Mode choice (MC_Purp.bat & AEMS.EXE)	
9	controls	station_names.dbf	Contains rail station names (derived from station.dbf)	DBF	Metadata	Created by set_factors.s; Used by LineSum	
10	inputs	airpax.adr	Air Passenger Auto Driver Trips	Binary	Assumptions	Miscellaneous time of day	
11	inputs	AM_Tfac.dbf	AM Toll Factors by Vehicle Type	DBF	Assumptions	Highway skimming and assignment	
12	inputs	areadef3722.prn	Input TAZ-Mode choice district equivalence	Text	Assumptions	Transit fare development (prefarv23.s)	
13	inputs	AreaWalk.txt	Optional. This file is now generated/re-generated by model	Text	Transit network	Generated by automated ArcPy process	
14	inputs	AT_override.TXT	Cases where zones have area-type override values	Text	Assumptions	AreaType_File.s	
15	inputs	Bus_Factor_File.dbf	Local Bus Time Degradation Factors	DBF	Assumptions	transit_skims_??s	
16	inputs	bus_pnrn.tb	Bus PNR lots	Text	Transit network	transit_skims_??s	1,2,6-9
17	inputs	BUSFARAM.ASC	AM Bus Fare matrix (Bus fares zones '1' to '21')	Text	Assumptions	mfare2.s	
18	inputs	BUSFAROP.ASC	OP Bus Fare matrix (Bus fares zones '1' to '21')	Text	Assumptions	mfare2.s	
19	inputs	com_bus.tb	Transfer link (walk) between commuter rail station and bus & LRT stop	Text	Transit network	transit_skims_??s	12
20	inputs	com_link.tb	Commuter rail links	Text	Transit network	transit_skims_??s	4
21	inputs	com_node.tb	Commuter rail stations	Text	Transit network	transit_skims_??s	4
22	inputs	com_pnrn.tb	Commuter rail PNR lots	Text	Transit network	transit_skims_??s	4
23	inputs	CPI_File.txt	Assumed rate of inflation, based on historical CPI	Text	Assumptions	Set_CPI.s	
24	inputs	Ext_PsAs.dbf	External Productions and Attractions	DBF	Observed data	trip_generation.s	
25	inputs	GIS_Variables.dbf	GIS variables used to calculate non-motorized trips	DBF	Observed data	trip_generation.s	
26	inputs	HBO_NL_MC.MTT	Pre-existing mode choice model output	Binary	Assumptions	pp_auto_drivers.s	
27	inputs	HBS_NL_MC.MTT	Pre-existing mode choice model output	Binary	Assumptions	pp_auto_drivers.s	
28	inputs	HBW_NL_MC.MTT	Pre-existing mode choice model output	Binary	Assumptions	pp_auto_drivers.s	
29	inputs	Jur.dbf	Equiv. between juris and river superdistricts: Disallows river crossings for PNR	DBF	Land use	Autoacc5.s	
30	inputs	Link.dbf	Highway network links	DBF	Highway network	V2.3_Highway_Build.s	
31	inputs	lrt_bus.tb	Transfer link (walk) between LRT station and bus stop	Text	Transit network	transit_skims_??s	12
32	inputs	lrt_link.tb	LRT links	Text	Transit network	transit_skims_??s	5
33	inputs	lrt_node.tb	LRT stations/stops	Text	Transit network	transit_skims_??s	5
34	inputs	lrt_pnrn.tb	LRT PNR lots	Text	Transit network	transit_skims_??s	5
35	inputs	MD_Tfac.dbf	MD Toll Factors by Vehicle Type	DBF	Assumptions	Highway skimming and assignment	
36	inputs	met_bus.tb	Transfer link (walk) between Metrorail station and bus stop	Text	Transit network	transit_skims_??s	12
37	inputs	met_link.tb	Metrorail links	Text	Transit network	transit_skims_??s	3
38	inputs	met_node.tb	Metrorail stations	Text	Transit network	transit_skims_??s	3
39	inputs	met_pnrn.tb	Metrorail PNR lots	Text	Transit network	transit_skims_??s	3
40	inputs	metlnkm1.tb	Metrorail Links	Text	Transit network	metrorail_skims.s	
41	inputs	metnodm1.tb	Metrorail Nodes	Text	Transit network	metrorail_skims.s	
42	inputs	mfare1.al	Metrorail Sta XYs scaled to 1/100ths of miles	Text	Transit network	mfare1.s	
43	inputs	mfare1_sta_Disc.ASC	Metrorail Sta fare discount array in cents	Text	Assumptions	mfare1.s	
44	inputs	MODE1AM,... MODE10AM.tb	AM Transit Line Files	Text	Transit network	transit_skims_??s	
45	inputs	MODE10P,... MODE10OP.tb	OP Transit Line Files	Text	Transit network	transit_skims_??s	
46	inputs	new_bus.tb	Transfer link (walk) between BRT/streetcar stop and bus stop	Text	Transit network	transit_skims_??s	12
47	inputs	new_link.tb	BRT/streetcar links	Text	Transit network	transit_skims_??s	10
48	inputs	new_node.tb	BRT/streetcar stations/stops	Text	Transit network	transit_skims_??s	10
49	inputs	new_pnrn.tb	BRT/streetcar PNR lots	Text	Transit network	transit_skims_??s	10
50	inputs	NHO_NL_MC.MTT	Pre-existing mode choice model output	Binary	Assumptions	pp_auto_drivers.s	
51	inputs	NHW_NL_MC.MTT	Pre-existing mode choice model output	Binary	Assumptions	pp_auto_drivers.s	
52	inputs	Node.dbf	XY coordinates of nodes in highway network	DBF	Highway network	AreaType_File.s	
53	inputs	NT_Tfac.dbf	NT Toll Factors by Vehicle Type	DBF	Assumptions	Highway skimming and assignment	
54	inputs	Pen.dbf	List of TAZs considered to be in the "slugging" shed of the Pentagon	DBF	Assumptions	Autoacc5.s	
55	inputs	PM_Tfac.dbf	PM Toll Factors by Vehicle Type	DBF	Assumptions	highway_assignment.s	
56	inputs	schl.adr	School Auto Driver Trips	Binary	Assumptions	misc_time-of-day.s	

No.	Folder	Filename	Description	File Type	Category	Modeling step where file is used	Transit/ Non-Transit Mode(s)
57	inputs	StaAcc.dbf	Lookup table: Maximum drive-access-to-transit distances	DBF	Assumptions	Autoacc5.s	
58	inputs	station.dbf	Station file: Metrorail, commuter rail, LRT stations/PNR lots and bus PNR lots	DBF	Transit network	parker.s	
59	inputs	station_names.dbf*	Names of rail stations and express bus stops	DBF	Summary	LineSum.exe	
60	inputs	tariff.txt	WMATA tariff policy	Text	Assumptions	mfare1.s	
61	inputs	taxi.adr	Taxi Auto Driver Trips	Binary	Assumptions	misc_time-of-day.s	
62	inputs	tazfrzn.asc	Fare Zone File	Text	Assumptions	prefarv23.s	
63	inputs	Toll_Esc.dbf	Toll escalation assumptions: Highway tolls & deflators	DBF	Assumptions	V2.3_Highway_Build.s	
64	inputs	trnpen.dat	Turn Penalty file to ensure correct Metrorail fares	Text	Assumptions	metrorail_skims.s	
65	inputs	visi.adr	Visitor Auto Driver Trips	Binary	Assumptions	misc_time-of-day.s	
66	inputs	xtrawalk.dbf	Extra walk links that the analyst wishes to include	DBF	Transit network	walkacc.s	13
67	inputs	xxaut.vtt	Auto Driver Through Trips	Binary	Assumptions	misc_time-of-day.s	
68	inputs	XXCVT.vtt	Com/Mtk/Htk through Trips	Binary	Calculated data	misc_time-of-day.s	
69	inputs	Zone.dbf	Land use/land activity data at zonal level, 3722 TAZ	DBF	Land use	AreaType_File.s	
70	support	AM_SPD_LKP.txt	Initial lookup speeds used for highway links, AM period	Text	Highway network	V2.3_Highway_Build.s	
71	support	AttrRates.dbf	Trip Attractions	DBF	Calculated data	trip_generation.s	
72	support	cvdelta_3722.trp	Calibration matrix, or "delta table" for commercial vehicles	Binary	Assumptions	misc_time-of-day.s	
73	support	equiv_toll_min_by_inc.s	Equivalent minutes (min/'07\$) by period & income level	Text	Assumptions	trip_distribution.s	
74	support	HBincRat.dbf	HB Income Shares	DBF	Calculated data	trip_generation.s	
75	support	hwy_assign_capSpeedLookup.s	FT x AT Speed & Capacity lookup	Text	Highway network	highway_assignment.s	
76	support	hwy_assign_Conical_VDF.s	Volume Delay Functions file	Text	Highway network	highway_assignment.s	
77	support	MD_SPD_LKP.txt	Initial lookup speeds used for highway links, midday	Text	Highway network	V2.3_Highway_Build.s	
78	support	NMArates.dbf	Non-motorized Trip Attractions	DBF	Calculated data	trip_generation.s	
79	support	NMPrates.dbf	Non-motorized Trip Productions	DBF	Calculated data	trip_generation.s	
80	support	TAZ3722_to_7Mrkts.txt	Equivalency between TAZs and mode choice superdistricts	Text	Assumptions	PP_Auto_Drivers.s	
81	support	tkdelta_3722.trp	Calibration matrix, or "delta table" for med and hvy truck	Binary	Assumptions	misc_time-of-day.s	
82	support	todcomp_2008HTS.dbf	Time of day model/factors	Binary	Assumptions	time-of-day.s	
83	support	toll_minutes.txt	Toll minutes equivalence file by Vehicle Type	Text	Assumptions	Highway_skims.s	
84	support	TPBMod_Jur_Boundary.shp	Jurisdictional boundaries	SHP	Network	Network editing with Cube Base	
85	support	Truck_Com_Trip_Rates.dbf	Truck and Commercial Vehicle Trip Rates	DBF	Calculated data	truck_com_trip_generation.s	
86	support	True_Shape_2040_Nov20.shp	Used to display highway network with True Shape	SHP	Highway network	Network editing with Cube Base	
87	support	Ver23_f_factors.dbf	F-factors for trip distribution	DBF	Calculated data	trip_distribution.s	
88	support	weighted_trip_rates.dbf	Trip Productions	DBF	Calculated data	trip_generation.s	

\* This file is created automatically by set\_factors.s from the station.dbf file.

Ref: v23\_inputs\_v13.xls

Pre-established “parent” and “child” batch files for executing the model reside in the root subdirectory. Typically, “parent” batch files are edited to correspond to each modeled scenario, while “child” batch files remain unaltered. The parent batch files can be named as the user likes. The two main parent batch files are the “wrapper” batch file and the “run model steps” batch file (the latter file used to be called the “run all” batch file). Details about these two files can be found in section 6.2 (“Parent batch files”) on page 57. The child batch files are the ones that actually execute individual modeling steps, such as the trip generation step (e.g., Trip\_Generation.bat) or the traffic assignment step (e.g., Highway\_Assignment\_Parallel.bat). Child batch files generally call the Cube Voyager scripts and/or Fortran programs. The child batch files also assign names to report files that result from each model step. Listing files are typically assigned file extensions of RPT or TAB. The former refers to Cube Voyager report or listing files, while the latter refers to a subset tabulation of the report file containing only trip table totals or jurisdictional summaries. Parent batch files are used to string child batch files together so that the entire model execution can be initiated with a single command or batch file. The parent batch files also establish Windows environment variables that are used in the child batch files and Cube Voyager scripts, such as the iteration number, the model year, and the model description.

As stated earlier, all of the input files located in the “inputs,” “controls,” and “support” folders are listed in Table 6. It is the user’s responsibility to make sure that the generically named files are appropriate for the modeled scenario and are in the prescribed format (described later). Additionally, almost all of the files shown in Table 6 must exist for the model run to complete successfully, with the exception of some files such as the shapefiles used for displaying a highway network in True Shape mode (True\_Shape\_2040\_Nov20.shp). The advantage of using generic filenames is that the input and output filenames referenced in each Cube Voyager script and control file do not need to be tailored to match the different scenarios that are run. The disadvantage of using generic filenames is that, when moving or sharing files, two files with the same name could be quite different (e.g., zone.dbf for the year 2007 has the same name as zone.dbf for the year 2040). Thus, the metadata that describes the scenario name is stored in the name of the output subdirectory (e.g., “2016\_base”), not in the filenames themselves.

The SOFTWARE folder contains two Fortran executable programs (AEMS.exe and extrtab.exe), one C++ executable program (LineSum.exe),<sup>48</sup> and several dynamic-link library (DLL) files, as shown in Table 7.

---

<sup>48</sup> In the future, if we replace AEMS with ModeChoice, this folder will also include the C++ mode choice application program ModeChoice.exe.

Table 7 Fortran and C++ executable files and dynamic-link library files required for running the Version 2.3 travel model

Executable Name	Ver	Date	Size (bytes)	Program Function	Requires a control file?
AEMS.exe		2/13/2012	195,900	Mode choice application program (Fortran, 32-bit)	yes
cw3240.dll		2/13/2012	827,392	Dynamic-link library file associated w/ AEMS.exe	no
DFORMD.dll		2/13/2012	425,984	Dynamic-link library file associated w/ AEMS.exe	no
Tppdlibx.dll		**		Dynamic-link library file associated w/ AEMS.exe	no
Tputlibc.dll		**		Dynamic-link library file associated w/ AEMS.exe	no
extrtab.exe		2/13/2012	464,559	Extracts sections from Cube Voyager report files (Fortran, 32-bit)	no
Linesum.exe	6.0.2	3/26/2014	697,344	Creates reports summarizing transit loaded link files (C++, 32-bit)	yes

\*\* Note: There are two Cube DLL files needed for running AEMS.exe: Tppdlibx.dll and Tputlibc.dll.<sup>49</sup> These two files come with Cube. When AEMS runs, it needs to “see” these two files. This can be accomplished by either 1) placing a copy of these two files in the folder where AEMS runs (the SOFTWARE folder under the root folder of the model run); OR 2) **setting the Windows PATH environment variable to point to the location where these DLL files exist. It has been found that the second option is generally the best one.** One complicating factor is the fact that Cube Base is 32-bit and Cube Voyager is 64-bit, and each comes with a version of these two files (see Table 8). **AEMS needs the 32-bit version** (which is stored here: C:\Program Files (x86)\Citilabs\CubeVoyager). For more information about setting the Windows PATH environment variable, see section 5.1 (“Software installation”).

Table 8 Location for Cube DLL files

Cube DLL File	Location for 32-bit version	Location for 64-bit version
TPPDLIBX.DLL	C:\Program Files (x86)\Citilabs\Cube\ C:\Program Files (x86)\Citilabs\CubeVoyager\	none C:\Program Files\Citilabs\CubeVoyager\
TPUTLIBC.DLL	C:\Program Files (x86)\Citilabs\Cube\ C:\Program Files (x86)\Citilabs\CubeVoyager\	none C:\Program Files\Citilabs\CubeVoyager\

A listing of child batch files is provided in Table 9. The table also indicates the programs and/or Cube Voyager scripts that are invoked and the purpose of each batch file. Given the iterative application process of the model, most of the batch files are called multiple times during a model run. The sequence of batch file applications, by iteration, is shown in Table 10. The table indicates that there are 48 batch file steps called during a standard application of the model. Some of the batch files are called once, while others (e.g., trip\_generation.bat) are called during the pump-prime and all four standard iterations. A parent batch file (“run\_ModelSteps\*.bat”) is used to string each of the child batch files together during a typical model execution. The parent batch files, like child batch files, reside in the root subdirectory. Two parent batch files are typically prepared for each individual model run. The process

<sup>49</sup> In earlier versions of Cube, the filename of the second file omitted the letter “c”: Tputlib.dll



for executing a model is addressed in the next section. The remaining chapters address the specific details of each modeling step.

**Table 9 Child batch files used to run the Version 2.3 Travel Model**

Batch File	Scripts/Programs	Purpose
set_up_model_run_folders.bat	None	Not used in the running of the mode, but can be used to set up folders for a new model run.
ArcPy_Walkshed_Process.bat	MWCOG_Prepare_Inputs_to_Walkshed_Proc ss_PT.s MWCOG_Prepare_Inputs_to_Walkshed_Proc ss_TRNBUILD.s	Run the automated/integrated ArcPy/Python transit walkshed process
Set_CPI.bat	Set_CPI.s Set_Factors.s	Create highway and transit cost deflators. Create K factors and time penalties. Create station_names.dbf file from station.dbf file.
PP_Highway_Build.bat	AreaType_File.s V2.3_higway_build.s	Build highway networks.
PP_Highway_Skims.bat (see also Highway_Skims.bat)	Highway_skims_am.s Highway_skims_md.s Modnet.s Highway_skims_mod_am.s Highway_skims_mod_md.s Joinskims.s Remove_PP_Speed.s	Create AM/off-peak highway skims.
Transit_Skim_All_Modes_Parallel.bat	parker.s walkacc.s autoacc5.s transit_Accessibility.s Transit_Skim_LineHaul_Parallel.bat Transit_Skims_AB.s Transit_Skims_BM.s Transit_Skims_CR.s Transit_Skims_MR.s	Create the transit network: <ul style="list-style-type: none"> <li>• Create transit access links</li> <li>• Create transit network</li> <li>• Skim the four transit submodes</li> </ul> Also runs the transit accessibility process.
Transit_Fare.bat	prefarV23.s Metrorail_skims.s MFARE1.s MFARE2.s Assemble_Skims_MR.s Assemble_Skims_BM.s Assemble_Skims_AB.s Assemble_Skims_CR.s	Create transit fares for the current speed feedback iteration.
Trip_Generation.bat	Demo_Models.s Trip_Generation.s Trip_Generation_Summary.s Truck_Com_Trip_Generation.s	Execute daily trip generation.
Trip_Distribution.bat	Prepare_Ext_Auto_Ends.s Prepare_Ext_ComTruck_Ends.s Trip_Distribution_External.s Prepare_Internal_Ends.s Trip_Distribution_Internal.s	Execute daily trip distribution.

Batch File	Scripts/Programs	Purpose
Mode_Choice_Parallel.bat (or Mode_Choice_TC_V23_Parallel.bat)	MC_purp.bat => AEMS.EXE mc_NL_summary.s (mc_constraint_v23.s)	Execute the daily mode choice model (in P/A format). Optionally execute the mode choice model with the transit constraint (TC), which is more specifically a Metrorail constraint through the regional core.
copyBaseMC_to_final_inputs.bat	None	Not currently called as part of a model run. Could potentially be used by modeler to copy pre-existing NL mode choice model output into the input folder for new run.
Auto_Driver.bat	mc_Auto_Drivers.s	Generate initial auto driver trips after mode choice.
PP_Auto_Drivers.bat	PP_Auto_Drivers.s	Generate initial auto driver trips without the use of the mode choice model.
Time-of-Day.bat	Time-of-Day.s Misc_Time-of-Day.s Prepare_Trip_Tables_for_Assignment.s	Convert daily modeled trips to AM, PM, midday, and night. Convert trip tables from P/A format to O/D format.
Highway_Assignment_Parallel.bat	Highway_Assignment_Parallel.s	Execute user equilibrium highway assignment for four time-of-day periods
Average_Link_Speeds.bat	Average_Link_Speeds.s	Compute average link speeds. Run for only speed feedback iterations 2-4
Highway_Skims.bat (or HSR_Highway_Skims.bat or HSR_Highway_Skims_2017HOV2.bat)	Highway_Skims_am.s Highway_Skims_md.s modnet.s Highway_Skims_mod_am.s Highway_Skims_mod_md.s joinskims.s	Build zone-to-zone paths on the highway network and skim the times and costs on each path. Store the skimmed times and paths in matrix files. (Highway skim replacement [HSR]: 1) Do the same, but with the HOV3+ skim substitution/replacement technique used for modeling HOT lanes; or 2) Do the same, but with with the HOV2+ skim substitution technique.)
Transit_Assignment_Parallel.bat	Combine_Tables_For_TrAssign_Parallel.s Transit_Assignment_LineHaul_Parallel.bat Transit_Assignment_AB.s Transit_Assignment_BM.s Transit_Assignment_CR.s Transit_Assignment_MR.s	Execute the transit assignment (P/A format) for peak and off-peak periods
TranSum.bat	LineSum_*.ctl (such as LineSum_Volume.ctl, or lineSum_MR_access.ctl)	Summarize the transit assignment
dateName.bat	None (used by searchForErrs.bat)	
searchForErrs.bat	None	Searches through log and print files for possible error codes
move_temp_files_v6.bat	None	Moves temporary files to a location where they can be later deleted manual by the modeler.
updating_tpp_dll_files.bat	None	Not used for a model run. In the past, this batch file could be used to put the TP+ DLL files in the correct location, but this file is no longer used.

Table 10 Sequence of the batch files used to run the Version 2.3 Travel Model

Batch File	Scripts/Programs	Speed Feedback Iteration				
		PP	1	2	3	4
ArcPy_Walkshed_Process.bat	MWCOG_Prepare_Inputs_to_Walkshed_Process_PT.s MWCOG_Prepare_Inputs_to_Walkshed_Process_TRNBU ILD.s	1				
Set_CPI.bat	Set_CPI.s Set_Factors.s	2				
PP_Highway_Build.bat	AreaType_File.s V2.3_higway_build.s	3				
PP_Highway_Skims.bat (see also Highway_Skims.bat)	Highway_skims_am.s Highway_skims_md.s Modnet.s Highway_skims_mod_am.s Highway_skims_mod_md.s Joinskims.s Remove_PP_Speed.s	3				
Transit_Skim_All_Modes_Parallel.bat	parker.s walkacc.s autoacc5.s transit_Accessibility.s Transit_Skim_LineHaul_Parallel.bat Transit_Skims_AB.s Transit_Skims_BM.s Transit_Skims_CR.s Transit_Skims_MR.s	5	12	21	30	39
Transit_Fare.bat	prefarV23.s Metrorail_skims.s MFARE1.s MFARE2.s Assemble_Skims_MR.s Assemble_Skims_BM.s Assemble_Skims_AB.s Assemble_Skims_CR.s		13	22	31	39
Trip_Generation.bat	Demo_Models.s Trip_Generation.s Trip_Generation_Summary.s Truck_Com_Trip_Generation.s	6	14	23	32	41
Trip_Distribution.bat	Prepare_Ext_Auto_Ends.s Prepare_Ext_ComTruck_Ends.s Trip_Distribution_External.s Prepare_Internal_Ends.s Trip_Distribution_Internal.s	7	15	24	33	42
Mode_Choice_Parallel.bat (or Mode_Choice_TC_V23_Parallel.bat)	MC_purp.bat => AEMS.EXE mc_NL_summary.s (mc_constraint_v23.s)		16	25	34	43
Auto_Driver.bat	mc_Auto_Drivers.s		17	26	35	44
PP_Auto_Drivers.bat	PP_Auto_Drivers.s	8				
Time-of-Day.bat	Time-of-Day.s Misc_Time-of-Day.s Prepare_Trip_Tables_for_Assignment.s	9	18	27	36	45
Highway_Assignment_Parallel.bat	Highway_Assignment_Parallel.s	10	19	28	37	46

Batch File	Scripts/Programs	Speed Feedback Iteration				
		PP	1	2	3	4
Highway_Skims.bat (or HSR_Highway_Skims.bat)	Highway_Skims_am.s Highway_Skims_md.s modnet.s Highway_Skims_mod_am.s Highway_Skims_mod_md.s joinskims.s	11	20	29	38	47
Transit_Assignment_Parallel.bat	Combine_Tables_For_TrAssign_Parallel.s Transit_Assignment_LineHaul_Parallel.bat Transit_Assignment_AB.s Transit_Assignment_BM.s Transit_Assignment_CR.s Transit_Assignment_MR.s					48
TranSum.bat	LineSum_*.ctl (such as LineSum_Volume.ctl, or lineSum_MR_access.ctl)					49

## 5 Preparing to run the model

Before the travel model can be run, one must install the necessary software, as described in the next section.

### 5.1 Software installation and setting the Windows PATH environment variable

**Step 1:** Make sure you are logged on to your computer with administrator privileges, so you can install software (or ask your IT department to perform the installation).

At COG, you will need to be a member of these two groups: "Administrators" and "SophosAdministrator" (the second group is associated with antivirus software). In other agencies, there may be other requirements. Additionally, based on testing done at COG, the mode choice application program (AEMS.EXE) may not work if you are not part of the Administrators group.

**Step 2:** Verify that your computer is running a 64-bit version of Windows, since this is needed to install the 64-bit version of Cube Voyager (Cube Base is still 32-bit software).

<Windows key><Pause/Break> will bring up the System Properties window. The "System Type" should be listed as "64-bit Operating System." Alternatively, if you prefer using the command prompt, you can run the command "systeminfo | more":

- If you are running a 32-bit version of Windows, you will see "System Type: X86-based PC". This will not allow you to install the 64-bit version of Cube Voyager, so you will not be able to run the Ver. 2.3.66 model.
- If you are running a 64-bit version of Windows, you will see "System Type: x64-based PC".

On 64-bit versions of Windows

- 64-bit software, such as Cube Voyager, is stored here: "C:\Program Files".
- 32-bit software, such as Cube Base, is stored here: "C:\Program Files (x86)".

**Step 3:** Determine the number of cores on your computer.

Again, <Windows key><Pause/Break> will bring up the System Properties window. Here you can see the CPU type (e.g., "Intel Core i5-4590"). You can perform an internet search with this information to find the number of cores that are contained in your processor.

Also, if you open up the Task Manager (keyboard combination <CTRL><SHIFT><ESC>) and select the Performance tab, you can see the number of cores that the Windows operating system sees, as well as the number of logical procesors (see Figure 5).

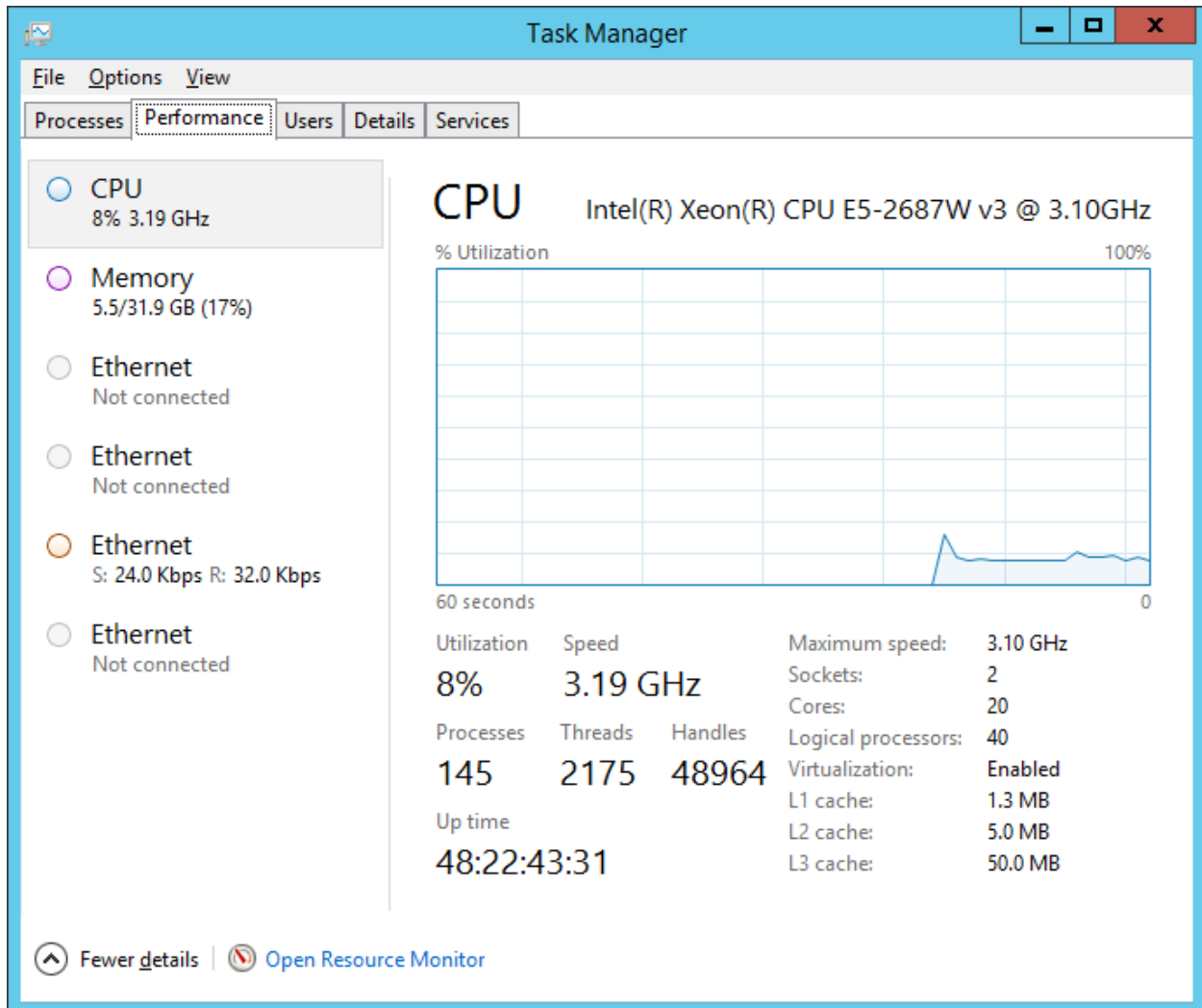


Figure 5 Number of cores and logical processors

In this example, Windows sees 20 cores and 40 logical processors. If the number of logical processors is double the number of cores, this means that Intel's Hyper-Threading Technology is turned on. Intel's Hyper-Threading Technology allows each core to handle two threads, so the operating system (OS) will see twice as logical processors as the actual number of physical cores.

**Step 4:** If you do not already have one, install the text editor of your choice, such as Notepad++ or PSPad. Notepad++ is free and open source. PSPad is free, but is not open source. Each of these text editors has its pluses and minuses, in terms of syntax highlighting, code folding, and other features, such as diffing two text files. Some non-open source text editors may include bloat-ware, which you may not want, so, when performing the installation, you need to be vigilant and de-select any options you do not want.

**Step 5:** Install Citilabs Cube Base and Cube Voyager software, according to the vendor's instructions. If you have purchased the license for Cube Cluster, this will also be installed at this point.

Once you have installed Cube Base, you can open it and click on help (“ ? ”) and “About...”, which should bring up a window like the following:

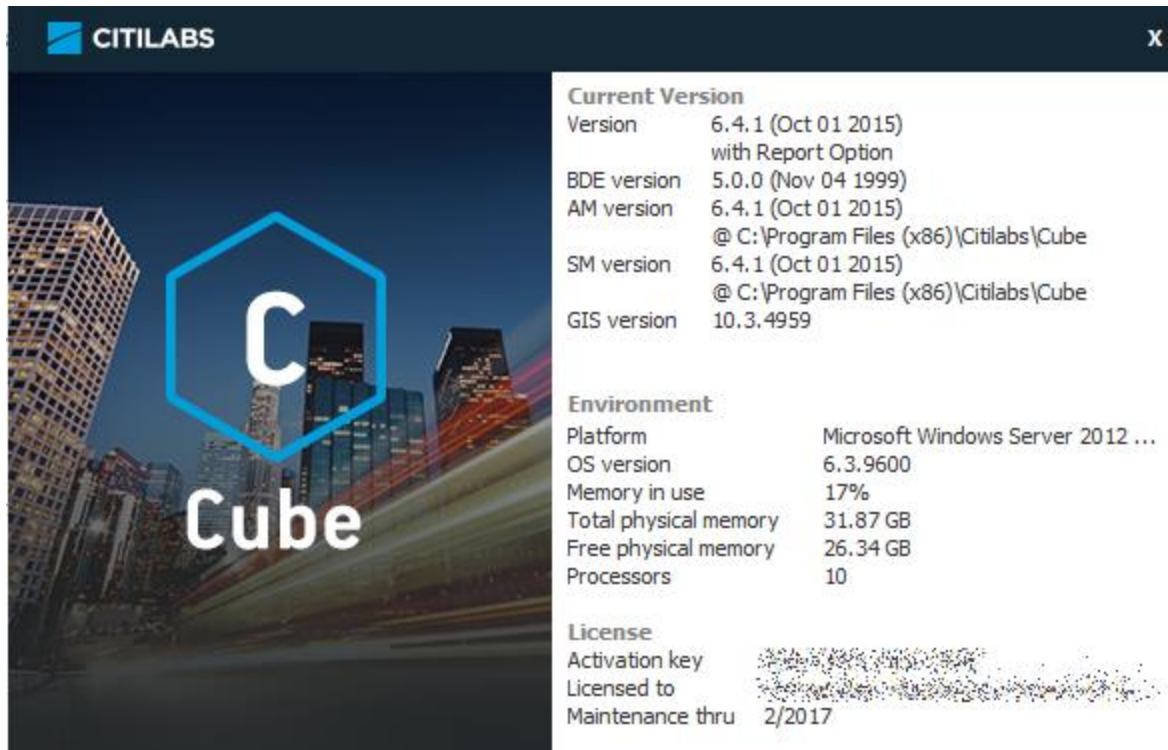


Figure 6 Citilabs Cube 6.4, “About” message window

From this window, we can see that we have installed Cube Base, version 6.4.1. The maintenance license expires in Feb. 2017. Cube sees 10 processors, even though this is on a computer with 20 cores and 40 logical processors.

If you are running the Ver. 2.3 Travel Model on a computer with fewer than 8 logical processors, you will need to follow the instructions in Table 17 (p. 88) before running the Ver. 2.3 Travel Model.

**Determining if you have Cube Cluster:** If you wish to determine whether a given computer includes a Cube Cluster license, open up Cube Voyager and click the “About Voyager” button. If your computer has a Cube Cluster license, you should see “with Cluster License” (as shown in Figure 7).

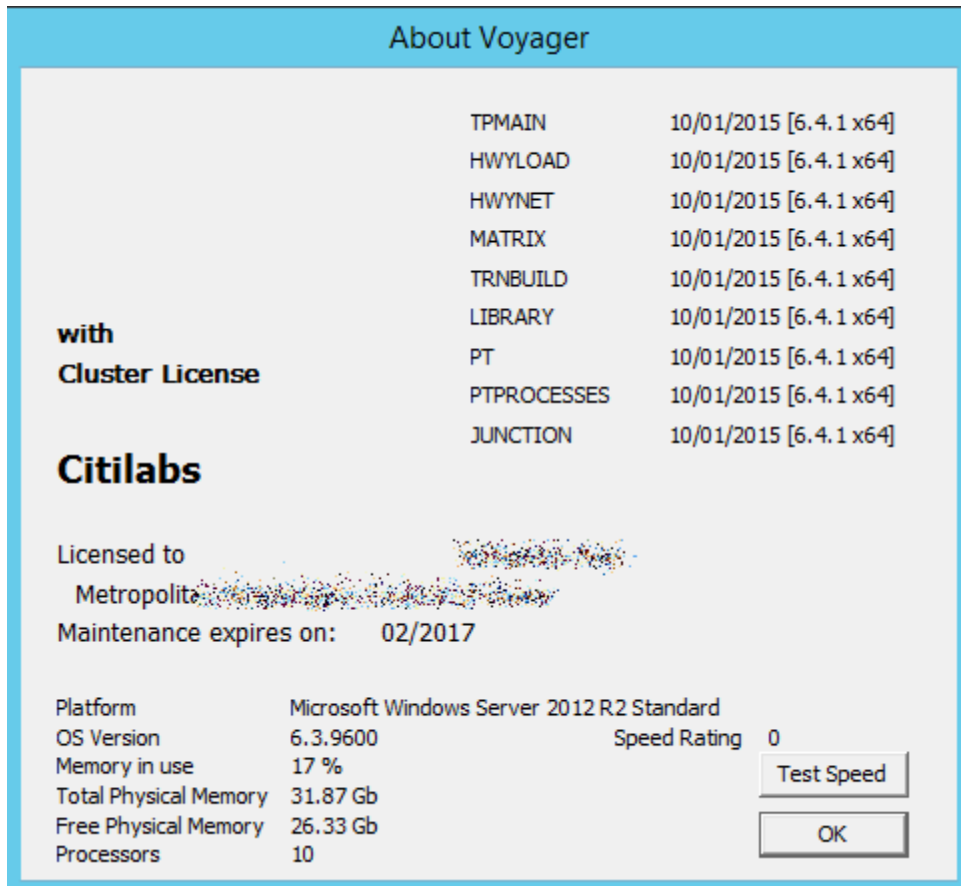


Figure 7 "About Voyager": Presence of Cube Cluster license

According to Citilabs, the "Test Speed" button is no longer active. This window also shows 10 processors, even though this computer has 20 cores and 40 logical processors.

**Step 6:** Install Cygwin (<http://www.cygwin.com/>). This is a free, open source software package that provides a Linux-like environment for Windows. It provides a series of Unix-like command-line tools, such as head, tail, and which. **This is not required to run the model**, but some commands, such as tail, are used in the runModel batch file, to extract information from various report files. Cygwin comes in two versions: a 32-bit version (setup-x86.exe) and a 64-bit version (setup-x86\_64.exe). Normally, we would advise you to install the version that is appropriate for your computer. However, we have found that, in the 64-bit version of Cygwin, the head and tail commands are not working correctly.

**Consequently, we recommend that you install the 32-bit version of the software.**

1. Download the 32-bit version of Cygwin: setup-x86.exe.
2. Double click the setup file to run. It will install a default set of packages. You can always add more in the future by rerunning the setup file.

Do not forget where this file is, since you might need to run it in the future to add or remove components from Cygwin. The recommended location is to place the file in a folder in your "downloads" folder (e.g., C:\Users\\downloads\cygwin). Once you have done this, you



should create a shortcut to the setup file on the Windows Desktop so that you can find this file easily in the future. As an alternative, you can also store the setup file directly on the Windows Desktop. The disadvantage with this second location is that, during the installation procedure, Cygwin will place a folder of downloaded files on the Desktop, and this folder may have an odd name, such as “ftp%3a%2f%2fftp.gtlib.gatech.edu%2fpub%2fcygwin%2f”.

Although the model will run without having Cygwin installed, if it is not installed, you may receive a window noting that Cygwin or one of its commands could not be found. As a future change to the model, we are considering removing the need to install Cygwin and replacing its functionality with other scripts.

**Step 7:** Set the Windows PATH environment variable.

Among other files, the 32-bit software folder (C:\Program Files (x86)\Citilabs\CubeVoyager) contains the following files:

```
10/01/2015 07:11 AM      3,416,528 CLUSTER.EXE
10/01/2015 07:44 AM      111,056 RUNTPP.EXE
10/01/2015 07:44 AM      415,744 TPPDLIBX.DLL
10/01/2015 07:44 AM      152,576 TPUTLIBC.DLL
```

By contrast, the 64-bit software folder (C:\Program Files\Citilabs\CubeVoyager) contains the following files:

```
10/01/2015 07:11 AM      4,206,544 CLUSTER.EXE
10/01/2015 07:45 AM      150,480 RUNTPP.EXE
10/01/2015 07:45 AM      373,712 VOYAGER.EXE
10/01/2015 07:44 AM      511,488 TPPDLIBX.DLL
10/01/2015 07:44 AM      178,688 TPUTLIBC.DLL
```

AEMS.EXE requires the use of the two TP DLL files in the 32-bit folder (C:\Program Files (x86)\Citilabs\CubeVoyager), so the Windows PATH variable should point to that folder. However, the 32-bit folder contains the wrong version of Cluster and does not contain Voyager.exe at all. Luckily, when Voyager.exe is called using the “start /w” command, Windows knows how to find the correct version of Voyager (in this case, the only version of Voyager). Thus, as a minimum, you will want to add the following two paths to your Windows PATH environment variable:

64-bit version of Windows	Reason
C:\cygwin\bin	Needed to run Cygwin from the command line
C:\Program Files (x86)\Citilabs\CubeVoyager	Needed so that AEMS can find the two TP DLL files (Tppdlibx.dll and Tputlibc.dll)

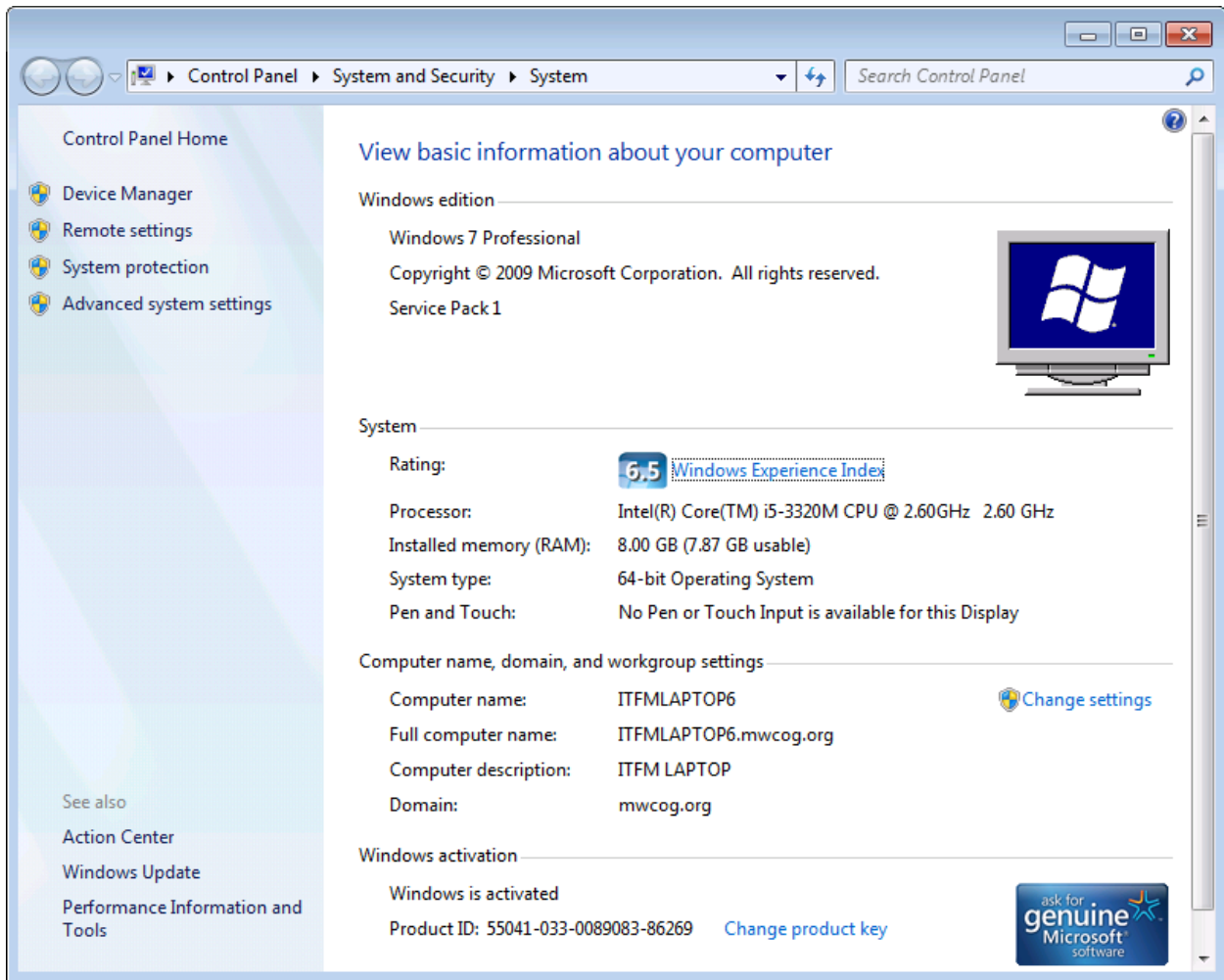
Additionally, the following paths might also be useful additions to your Windows PATH environment variable:

64-bit version of Windows	Reason
C:\Program Files (x86)\PSPad editor	To be able to open the PSPad text editor from the command line
C:\Program Files (x86)\WinMerge	To be able to open WinMerge from the command line

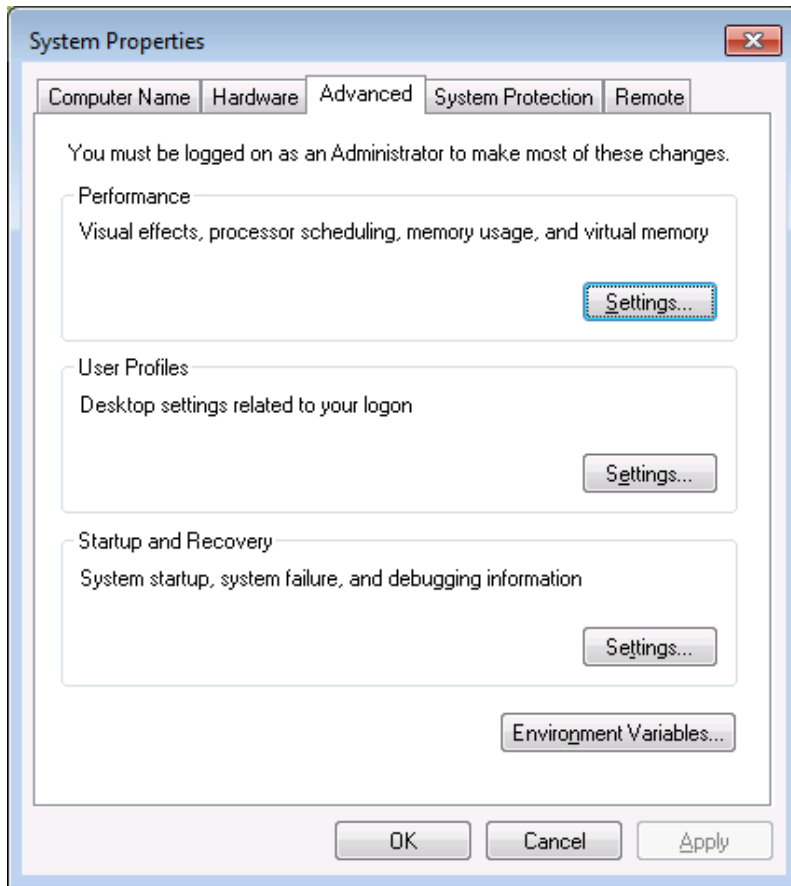
Here are instructions for updating the Windows PATH environment variable:

Hold down these two keys simultaneously to bring up the Windows System Properties window:

<Windows key><Pause/Break key>

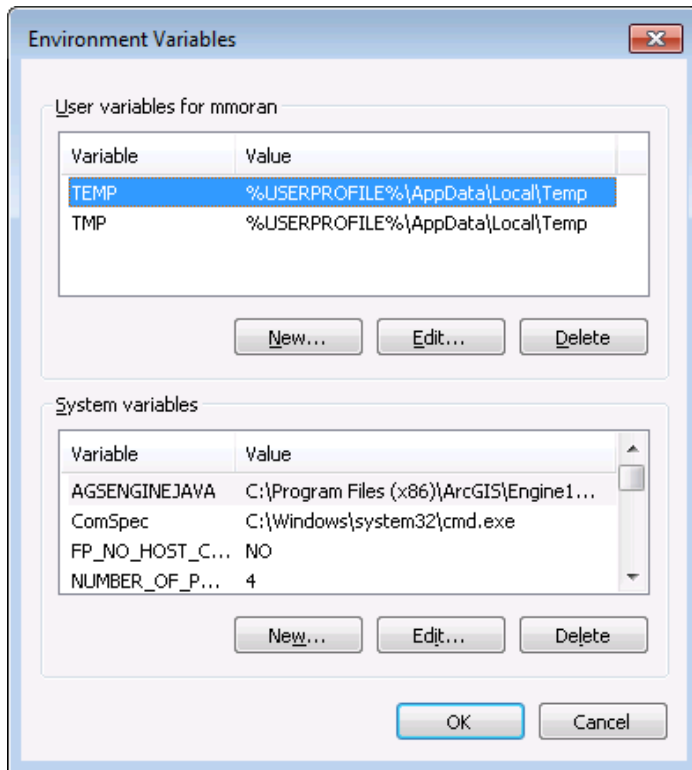


Click "Advanced system settings." Click the "Advanced" tab.

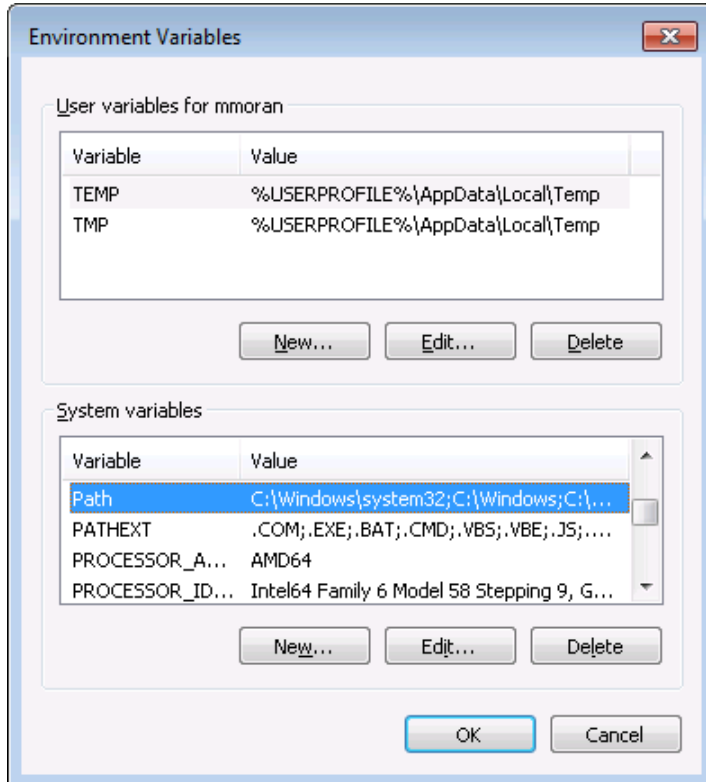


Click the “Environment Variables” button.

The lower half of this window contains “system variables.”



Find the "Path" environment variable in the lower half of this window.



Click "Edit."

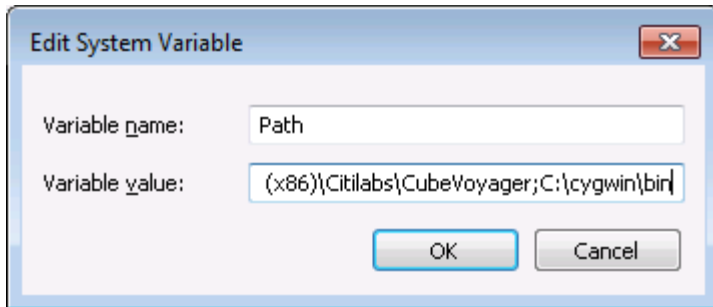
Add the Windows paths that you want. Add these to the end of the Path variable, using a semicolon (“;”) as the separator.

For example, this path:

```
%SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\Wbem;%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
```

Would become this path:

```
%SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\Wbem;%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\;C:\Program Files (x86)\Citilabs\CubeVoyager;C:\cygwin\bin
```



Click “OK” three times.

To test whether Cygwin is working correctly, open a command window and type a Cygwin command, such as

```
which ls
```

Or

```
tail --help
```

To test Cube Voyager, type:

```
start /w voyager
```

Or

```
cluster
```

**Step 8:** Useful, but not essential: Install Winmerge and update the PATH environment variable to include:

```
C:\Program Files\WinMerge
```

**Step 9:** You may want to associate \*.net files with Cube.exe. This will allow the file i4\_assign\_output.net to be opened in Cube automatically at the completion of a model run. You may

also want to associate \*.txt and \*.rpt files with your preferred text editor. At the completion of a model run, the “run model” batch file tries to open several of these files (such as i4\_Highway\_Assignment.rpt). By setting up the desired file association, these files will be opened at the conclusion of the model run using the desired text editor (versus the default Windows text editor, which is Notepad).

## **5.2 Preparing input files and calculating zonal percent-walk-to-transit values**

After a person has requested the COG/TPB travel model from COG/TPB staff (<https://www.mwcog.org/transportation/data-and-tools/modeling/data-requests/>), he or she will be sent a transmittal memo and the actual travel model, including its inputs. If the user wants to simply run the travel model for the years/scenarios that have been supplied by COG/TPB staff, then there is no need to make any changes to the model inputs (This also pre-supposes that the user has required hardware and software, as specified in this user's guide).

In the past, prior to running the travel demand model, if the user made changes to the transportation network that would affect walking distances to transit, one would have needed to re-do the transit walkshed process to calculate the percent of each TAZ within walking distance to transit. This process was a semi-automated process that was run before a model run. Now, however, with the new automated/integrated transit walkshed process, this step is accomplished when one runs the model (the first step in the “pump-prime” speed feedback iteration), so there is no need to make a special effort to re-run the transit walkshed process. The new transit walkshed process is discussed in section 11 (“Building transit walksheds and calculating zonal walk percent”) of this report, beginning on p. 101.

## 6 Running the model

As noted in the “Hardware and software ” section, the Version 2.3 Travel Model is implemented using Citilabs Cube software. Cube Base is the graphical user interface (GUI) for the Cube suite of software. Cube base can be used for editing Cube Voyage scripts, editing transportation networks, viewing matrix files, managing network scenarios (Scenario Manager), and running travel models (Application Manager). However, the Version 2.3 Travel Model is not launched using Cube Base’s Application Manager. Instead, the Version 2.3 Travel Model is implemented using a command-line interface (CLI) that is initiated from a Windows command window (also called a DOS command window by some).

### 6.1 Updating the Windows PATH environment variable

**It is important to update the Windows PATH environment variable**, as described in section 5.1 (“Software installation and setting the Windows PATH environment variable”).

The next section describes a simple example of how to run the travel model. Following that are two sections that describe the wrapper batch file and the “run model steps” batch file in more detail.

### 6.2 Parent batch files

To run the Version 2.3 Travel Model (including Ver. 2.3.66), the user must edit two batch files and then run one of the batch files, which, in turn, will call the other file.<sup>50</sup> These two batch files are known as the parent batch files, since they call a series of other batch files (known as the child batch files). The first parent batch file is called the “wrapper” batch file or the “run model” batch file (an example can be seen in Figure 8). The second parent batch file is called the “run model steps” (formerly “run all”) batch file (an example can be seen in Figure 9). In computer programming, the term “wrapper function” is used for a function whose main purpose is to call a second function and set up a computing environment for that second function. We are using this term in a similar vein, since the main purpose of our wrapper batch file is to call a second batch file (the “run model steps” batch file) and set up the running environment for the model run. Once the user has edited the two parent batch files with a text editor, the user launches the model run by launching the wrapper batch file within a command prompt window that is pointing to the root directory. For example, if the root directory is “C:\modelRuns\fy16\Ver2.3.66\_aqc\_2016clrp”, then **the user would open a command prompt window at this location and type the name of the “run model”/wrapper batch file and press Enter to execute it**. This process is described in more detail below, along with some preliminary information needed to make the model run correctly.

There is typically a “run model” batch file and a “run model steps” batch file for each scenario/year that is modeled, e.g.,:

```
run_Model1_2016_base.bat
```

---

<sup>50</sup> Note: Be careful not to double click the batch file when viewing its icon in Windows Explorer/File Explorer, since this will launch the batch file, instead of editing it. The proper way to launch the batch file is described later in this chapter.

```
run_Model_2016_final.bat
run_Model_2017_base.bat
run_Model_2017_Final.bat
... etc. ...
run_Model_2040_base.bat
run_Model_2040_final.bat

run_ModelSteps_2016_base.bat
run_ModelSteps_2016_final.bat
run_ModelSteps_2017_base.bat
run_ModelSteps_2017_Final.bat
... etc. ...
run_ModelSteps_2040_base.bat
run_ModelSteps_2040_final.bat
```

**To launch a model run**, one needs to open a Windows command window that points to the location where you have placed the parent batch files (the so-called “root” folder). One way to do this is to open Windows Explorer (File Explore in some versions of Windows) and navigate to the root folder, and then select the root folder by clicking it once. Next, in most versions of Windows, one selects the folder in the left pane, and then, with nothing selected in the right pane, one uses the mouse to **shift-right-click** in the right pane, selecting “Open Command Window Here.”

Figure 8 “Run model” batch file for 2016\_base

```
1 :: tms7 E:\Data\fy17\CGV2_3_66_Conformity2016CLRP_Xmittal\run_Model_2016_base.bat
2 :: Version 2.3.66
3 :: 10/24/2016 11:13:10 AM
4
5 set root=.
6 set scenar=2016_base
7 set runbat=run_ModelSteps_2016_base.bat
8 :: Environment variables for (multistep) distributed processing:
9 :: Environment variables for (intrastep) distributed processing:
10 :: use MDP = t/f (for true or false)
11 :: use IDP = t/f (for true or false)
12 :: Number of subnodes: 1-3 => 3 subnodes and one main node = 4 nodes in total
13 set useIdp=t
14 set useMdp=t
15 :: AMsubnode & MDsubnode are used in highway_assignment_parallel.bat/s
16 set AMsubnode=1-4
17 set MDsubnode=2-4
18 :: subnode used in transit fare and transit assignment
19 :: We no longer use IDP in transit skimming, since it would require 16 cores
20 set subnode=1-3
21
22 :: This command will
23 :: 1) time the model run (using timethis.exe and the double quotes)
24 :: 2) redirect standard output and standard error to a file
25 :: 3) Use the tee command so that stderr & stdout are sent both to the file and the screen
26
27 timethis "%runbat% %scenar%" 2>&1 | tee %root%\%scenar%\%scenar%_fulloutput.txt
28
29 :: Open up the file containing the stderr and stdout
30 if exist %root%\%scenar%\%scenar%_fulloutput.txt start %root%\%scenar%\%scenar%_fulloutput.txt
31
32 :: Look four errors in the reports and output files
33 call searchForErrs.bat %scenar%
34 :: Open up the file containing any errors found
```



```

35 if exist %root%\%scenar%\%scenar%_searchForErrs.txt start %root%\%scenar%\%scenar%_searchForErrs.txt
36
37 :: Open up other report files
38 if exist %root%\%scenar%\i4_Highway_Assignment.rpt start %root%\%scenar%\i4_Highway_Assignment.rpt
39 if exist %root%\%scenar%\i4_mc_NL_summary.txt start %root%\%scenar%\i4_mc_NL_summary.txt
40 if exist %root%\%scenar%\i4_Assign_Output.net start %root%\%scenar%\i4_Assign_Output.net
41 cd %scenar%
42 start cmd /k ..\tail -n1 i4_ue*AM_nonHov*txt i4_ue*AM_hov*txt i4_ue*PM_nonHov*txt i4_ue*PM_hov*txt i4_ue*MD*txt
43 i4_ue*NT*txt
44 cd ..
45 move_temp_files_v6.bat %scenar%
46
47 :: Cleanup
48 set root=
49 set scenar=
50 set runbat=
51 set useIdp=
52 set useMdp=
53 set AMsubnode=
54 set MDsubnode=
55 set subnode=

```

Figure 9 “Run model steps” batch file for 2016\_base

```

1  :: tms7 E:\Data\fy17\CGV2_3_66_Conformity2016CLRP_Xmittal\run_ModelSteps_2016_base.bat
2  :: Version 2.3.66
3  :: 10/24/2016 11:18:23 AM
4
5  :: Version 2.3 TPB Travel Model on 3722 TAZ System
6
7  set _year_=2016
8  set _alt_=Ver2.3.66_2016_base
9  :: Maximum number of user equilibrium iterations used in traffic assignment
10 :: User should not need to change this. Instead, change _relGap_ (below)
11 set _maxUeIter_=1000
12
13 :: Set transit constraint path and files
14 :: Current year used to set the constraint = 2020
15 :: For years before constraint year: set _tcpath_=<blank>
16 :: For years after constraint year: set _tcpath_=..\2020_final
17 set _tcpath_=
18
19 :: Location of substitute HOV3+ skims
20 set _HOV3PATH_=
21
22 :: UE relative gap threshold: Progressive (10^-2 for pp-i2, 10^-3 for i3, & 10^-4 for i4)
23 :: Set the value below
24
25 rem ===== Pump Prime Iteration =====
26
27 set _iter_=pp
28 set _prev_=pp
29 set _relGap_=0.01
30
31 call ArcPy_Walkshed_Process.bat %1
32 call Set_CPI.bat %1
33 call PP_Highway_Build.bat %1
34 call PP_Highway_Skims.bat %1
35 call Transit_Skim_All_Modes_Parallel.bat %1
36 call Trip_Generation.bat %1
37 call Trip_Distribution.bat %1
38 call PP_Auto_Drivers.bat %1
39 call Time-of-Day.bat %1
40 call Highway_Assignment_Parallel.bat %1
41 call Highway_Skims.bat %1
42
43 :: rem ===== Iteration 1 =====
44

```

```

45 set _iter_=i1
46 set _prev_=pp
47
48 call Transit_Skim_All_Modes_Parallel.bat %1
49 call Transit_Fare.bat %1
50 call Trip_Generation.bat %1
51 call Trip_Distribution.bat %1
52 call Mode_Choice_Parallel.bat %1
53 call Auto_Driver.bat %1
54 call Time-of-Day.bat %1
55 call Highway_Assignment_Parallel.bat %1
56 call Highway_Skims.bat %1
57
58 :: rem ===== Iteration 2 =====
59
60 set _iter_=i2
61 set _prev_=i1
62
63 call Transit_Skim_All_Modes_Parallel.bat %1
64 call Transit_Fare.bat %1
65 call Trip_Generation.bat %1
66 call Trip_Distribution.bat %1
67 call Mode_Choice_Parallel.bat %1
68 call Auto_Driver.bat %1
69 call Time-of-Day.bat %1
70 call Highway_Assignment_Parallel.bat %1
71 call Average_Link_Speeds.bat %1
72 call Highway_Skims.bat %1
73
74 :: rem ===== Iteration 3 =====
75
76 set _iter_=i3
77 set _prev_=i2
78 set _relGap_=0.001
79
80 call Transit_Skim_All_Modes_Parallel.bat %1
81 call Transit_Fare.bat %1
82 call Trip_Generation.bat %1
83 call Trip_Distribution.bat %1
84 call Mode_Choice_Parallel.bat %1
85 call Auto_Driver.bat %1
86 call Time-of-Day.bat %1
87 call Highway_Assignment_Parallel.bat %1
88 call Average_Link_Speeds.bat %1
89 call Highway_Skims.bat %1
90
91 :: rem ===== Iteration 4 =====
92
93 set _iter_=i4
94 set _prev_=i3
95 set _relGap_=0.0001
96
97 call Transit_Skim_All_Modes_Parallel.bat %1
98 call Transit_Fare.bat %1
99 call Trip_Generation.bat %1
100 call Trip_Distribution.bat %1
101 call Mode_Choice_Parallel.bat %1
102 call Auto_Driver.bat %1
103 call Time-of-Day.bat %1
104 call Highway_Assignment_Parallel.bat %1
105 call Average_Link_Speeds.bat %1
106 call Highway_Skims.bat %1
107
108 :: rem ===== Transit assignment =====
109 @echo Starting Transit Assignment Step
110 @date /t & time/t
111
112 call Transit_Assignment_Parallel.bat %1

```

```

113 call TranSum.bat %1
114
115 @echo End of batch file
116 @date /t & time/t
117 :: rem ===== End of batch file =====
118
119 REM cd %1
120 REM copy *.txt MDP_%useMDP%\*.txt
121 REM copy *.rpt MDP_%useMDP%\*.rpt
122 REM copy *.log MDP_%useMDP%\*.log
123 REM CD..
124
125 set _year_=
126 set _alt_=
127 set _iter_=
128 set _prev_=
129 set _maxUeIter_=
130 set _relGap_=

```

### 6.2.1 Description of the “run model”/wrapper batch file

The first three lines of the “run model” batch file shown in Figure 8 are simply comments. Comments in batch files can be indicated using either a double colon (“:”) or the word REM at the start of the line.<sup>51</sup> In line #5, we define a Windows environment variable called “root” and set its value to “.”, which simply means the current directory location (i.e., the current directory where one has opened a command prompt). In line #6, we define an environment variable called “scenar” (scenario) and set its value to the model scenario/year we want to run (in this case, 2016\_base, but any string may be used, such as “2030\_lowGrowth”). In line #7, we define an environment variable named “runbat” which is used to store the name of the “run model steps” batch file that we will use for the year-2016 model run. Lines 13-20 is where one sets the environment variables that control distributed processing. Distributed processing is covered in more detail later in this report.

**Line 27 is the actual line that runs the model.** The “timethis” command is used to time how long the command takes to run. In this case, the command being timed is the entire model run. The “2>&1” and “tee” sections of line 27 are explained next. When a program is run in a command-line interface, such as the Windows command window, there are two streams of output information: standard output and standard error. Standard output is information that the program supplies to a user while the program is running, such as messages about finishing a step, or the current TAZ number that is being processed. Standard error is information about errors that occur while running a program, for example, “file not found.” Normally, both the standard output stream and the standard error stream are sent to the screen (in this case, the Windows command window). However, since model run last many hours, it is not practical for a model user to watch the screen to see what messages occur during the model run. One solution is to redirect these two information streams to a file, instead of the screen, which allows one to review the contents of the file after the model run is completed. The “2>&1” keyword redirects both standard error and standard output to one file (in this case, the file ending with “\_fulloutput.txt”). However, the drawback to this approach is that the model user will not see any real-time information on the screen, since all the information is being sent to a file. An alternate approach is to combine the use of

<sup>51</sup> A single colon (“:”) before a word indicates a label, which is often the target of a GOTO statement.

"2>&1" with the "tee" command, which splits any stream of information into two streams of identical information. The result of using these two keywords together is that the standard output and standard error streams are sent both to the screen and to a file at the same time. Line 30 simply opens up, at the conclusion of the model run, the file containing the standard output and standard error information. The Tee.exe utility program is part of the Windows 2000 Resource Kit.

Line 33 calls a batch file that searches reports and output files for certain errors. Line 34 simply opens up this file containing the listing of errors. It should be noted that this file was mainly used for model development, **so it contains little useful information for the average model user. For the average model users, the key file to review is the one that combines the standard output and error information ("\_fulloutput.txt").**

Lines 38 through 40 contain commands which open up other report files, after the model run has completed. Line 42 contains a command that opens up a command window showing some summary convergence metrics for traffic assignment. In order for the "tail" command to work, you should have installed Cygwin, as was described in section 5.1 ("Software installation and setting the Windows PATH environment variable"). **Lastly, line 45 runs the cleanup process**, which divides model output files into two sets: files to keep and temporary files that can be deleted. At the completion of a model run, there are about 26 GB of output files, many of which are temporary or non-final versions of files. The move\_temp\_files\_v6.bat batch file creates the folder "temp\_files" and moves about 16 GB of the 26 GB of files to the temp\_files folder. **To save disk space, the user can then either delete the temp\_files folder or the contents of the temp\_files folder** (such as using Windows Explorer). The advantage of deleting the *contents* of the temp\_files folder, but not the folder itself, is that, in multi-user environments, it will be apparent to other model users that the cleanup process has already been run.

### 6.2.2 Description of the "run model steps" batch files

As explained in section 2.4, COG/TPB staff has developed three special modeling procedures, which are part of the Ver. 2.3.66 model:

1. Metrorail constraint through the regional core.
2. HOT lanes with free access for HOV3+ (e.g., I-495 and I-95 Express Lanes).
3. HOT lanes with free access for HOV2+ (e.g., I-66 inside the Beltway for 2017-2020).

As stated earlier, there is a "run model steps" batch file for each model run scenario/year. The set up of these "run model steps" batch files is influenced by the three special modeling procedures mentioned above and described in section 2.4. As an example, suppose we have a project, such as an air quality conformity analysis, that includes three years: 2016, 2020, and 2040. Table 11 shows under which circumstances one or more of the three special modeling procedures are invoked. For the model run representing year-2016 conditions, only one of the three special procedures is invoked (the HOV3+ skim replacement procedure). For the year-2020 model run, two of the three special procedures are invoked (the HOV2+ skim replacement and the HOV3+ skim replacement). For the year-2040 model run, two of the three special procedures are invoked (but this time, it is the Metrorail constraint and the HOV3+

skim replacement). The HOV2+ procedure is not invoked because, after 2020, the I-66 HOT lanes would make use of the HOV3+ condition.

**Table 11 Example of when special modeling procedures are invoked for three modeled years/scenarios**

Year	Metrorail constraint through the regional core	HOT2+ facilities	HOV 3+ facilities
2016	Prior to the binding year (2020), so procedure is <b>not</b> invoked	No HOT2+ facilities, so procedure is <b>not</b> invoked	HOT3+ facilities on I-495 & I-95, so procedure <b>is invoked</b>
2020	2020 is the binding year, so procedure is <b>not</b> invoked	HOT2+ facility on I-66 inside the Beltway, so procedure <b>is invoked</b>	HOT3+ facilities on I-495 & I-95, so procedure <b>is invoked</b>
2040	After the binding year (2020), so procedure <b>is invoked</b>	No HOT2+ facilities (they are all HOT3+), so procedure is <b>not</b> invoked	HOT3+ facilities on I-495 & I-95, so procedure <b>is invoked</b>

To correctly model HOT lane facilities, either HOT2+ or HOT3+, one would need to run both a “base” and a “final” scenario for all modeled years following 2012, since that is the first year that HOT lane facilities began in the DC area. that (in our case, all three years: 2016, 2020, and 2040). Table 12 provides additional details about how the three special modeling procedures described in Table 11 are implemented in the model.

Note: Regarding Table 12, select lines in the relevant “run model steps” batch files (Figure 9 through Figure 13) have been highlighted in yellow to call attention to key changes between batch files, such as the lines containing the text “set \_tcpath\_ =”, “set \_HOV3PATH\_ =”, “rem call Highway\_Skims.bat”, “call HSR\_Highway\_Skims.bat”, and “call HSR\_Highway\_Skims\_2017HOV2.bat”. Also, in Figure 9 through Figure 13, though not highlighted in yellow, note that the Average\_Link\_Speeds.bat batch file is called in speed feedback iterations 2-4, but not in the pump-prime or iteration #1 iterations.

Table 12 shows that the year 2016 is represented using two scenarios (2016\_base and 2016\_final). The Metrorail constraint is not an issue, since the constraining year is 2020, so, in the 2016\_base and 2016\_final “run model steps” batch files, the environment variable “\_tcpath\_” (transit constraint path) is set to blank/null, since no Metrorail constraint path is needed to model the year 2016 (see line 17 in Figure 9). Since there are HOT lane facilities in the 2016 network, two scenarios must be modeled (2016\_base and 2016\_final). For the “base” run, the HOV3+ skim substitution/replacement technique is not used. In the 2016\_base “run model steps” batch file, this means that the “\_HOV3PATH\_” environment variable is set to blank/null (see line 20 in Figure 9).

Table 12 Summary of differences to the “run model steps” batch files for the years 2016, 2020, and 2040

Year / Scenario	Metrorail constraint through regional core?	HOV2+ skim substitution technique for modeling HOT lanes?	HOV3+ skim substitution technique for modeling HOT lanes?
2016_base	Not used (2020 is constraining year)  i.e., “set_tspath_=”	N/A	No (Base HOV3+ skims are estimated)  i.e., “set_HOV3PATH_=”
2016_final	Not used (2020 is constraining year)  i.e., “set_tspath_=”	N/A	<b>Yes</b> (Base HOV3+ skims are used from the “base” run)  i.e., “set_HOV3PATH_=. \2016_base”
2020_base	2020 is the year used to set the constraint, but no change is made to batch file  i.e., “set_tspath_=”	No (Base HOV2&3+ skims are estimated)  i.e., “set_HOV3PATH_=”	No (Base HOV3+ skims are estimated)  i.e., “set_HOV3PATH_=”
2020_final	2020 is the year used to set the constraint, but no change is made to batch file  i.e., “set_tspath_=”	<b>Yes</b> (Base HOV2&3+ skims are used from the “base” run)  i.e., “set_HOV3PATH_=. \2020_base”	<b>Yes</b> (Base HOV3+ skims are used from the “base” run)  i.e., “set_HOV3PATH_=. \2020_base”
2040_base	<b>Yes</b>  e.g., “set_tspath_=. \2020_final”	N/A	No (Base HOV3+ skims are estimated)  i.e., “set_HOV3PATH_=”
2040_final	<b>Yes</b>  e.g., “set_tspath_=. \2020_final”	N/A	<b>Yes</b> (Base HOV3+ skims are used from the “base” run)  i.e., “set_HOV3PATH_=. \2040_base”

Table 12 shows that the year 2020 is represented with two scenarios: 2020\_base and 2020\_final. 2020 is the year used to set the Metrorail constraint, but this does not require any change to the “run model steps” batch file shown in Figure 10. Thus, the environment variable “\_tspath\_” (transit constraint path) is still set to blank/null (see line 18 in Figure 10 and Figure 11). As for HOT lane modeling, since 2020 is a year with both HOT2+ lanes and HOT3+ lanes, we need to run a “base” scenario (2020\_base) which is used to get the travel time skims for HOV traffic using the HOT lanes (and this information will be used in the 2020\_final run). However, for the 2020\_base run, since we are applying neither the HOV2+ nor HOV3+ skim substitution/replacement technique, we do not need to designate the location of the HOV2/3+ baseline skims, so the “\_HOV3PATH\_” environment variable is set to blank/null (see line 21 in Figure 10), as was the case for 2016\_base. Also, in Figure 10, we have highlighted in yellow the lines that contain “call Highway\_Skims.bat” (lines 42, 58, 75, 93, and 111) since these lines will be changed in the 2020\_final “run model steps” batch file so that they call the HOV2/3+ skim substitution/replacement technique.

Figure 10 “Run model steps” batch file for 2020\_base

```

1 :: E:\Data\fy17\CGV2_3_66_Conformity2016CLRP_Xmittal\run_ModelSteps_2017_base.bat
2 :: Version 2.3.66
3 :: 10/31/2016 11:18:23 AM
4

```

```

5
6 :: Version 2.3 TPB Travel Model on 3722 TAZ System
7
8 set _year_=2020
9 set _alt_=Ver2.3.66_2020_base
10 :: Maximum number of user equilibrium iterations used in traffic assignment
11 :: User should not need to change this. Instead, change _relGap_ (below)
12 set _maxUeIter_=1000
13
14 :: Set transit constraint path and files
15 :: Current year used to set the constraint = 2020
16 :: For years before constraint year: set _tcpath_=<blank>
17 :: For years after constraint year: set _tcpath_=..\2020_final
18 set _tcpath_=
19
20 :: Location of substitute HOV3+ skims
21 set _HOV3PATH_=
22
23 :: UE relative gap threshold: Progressive (10^-2 for pp-i2, 10^-3 for i3, & 10^-4 for i4)
24 :: Set the value below
25
26 rem ===== Pump Prime Iteration =====
27
28 set _iter_=pp
29 set _prev_=pp
30 set _relGap_=0.01
31
32 call ArcPy_Walkshed_Process.bat %1
33 call Set_CPI.bat %1
34 call PP_Highway_Build.bat %1
35 call PP_Highway_Skims.bat %1
36 call Transit_Skim_All_Modes_Parallel.bat %1
37 call Trip_Generation.bat %1
38 call Trip_Distribution.bat %1
39 call PP_Auto_Drivers.bat %1
40 call Time-of-Day.bat %1
41 call Highway_Assignment_Parallel.bat %1
42 call Highway_Skims.bat %1
43 REM call HSR_Highway_Skims_2017HOV2.bat %1
44
45 :: rem ===== Iteration 1 =====
46
47 set _iter_=i1
48 set _prev_=pp
49
50 call Transit_Skim_All_Modes_Parallel.bat %1
51 call Transit_Fare.bat %1
52 call Trip_Generation.bat %1
53 call Trip_Distribution.bat %1
54 call Mode_Choice_Parallel.bat %1
55 call Auto_Driver.bat %1
56 call Time-of-Day.bat %1
57 call Highway_Assignment_Parallel.bat %1
58 call Highway_Skims.bat %1
59 REM call HSR_Highway_Skims_2017HOV2.bat %1
60
61 :: rem ===== Iteration 2 =====
62
63 set _iter_=i2
64 set _prev_=i1
65
66 call Transit_Skim_All_Modes_Parallel.bat %1
67 call Transit_Fare.bat %1
68 call Trip_Generation.bat %1
69 call Trip_Distribution.bat %1
70 call Mode_Choice_Parallel.bat %1
71 call Auto_Driver.bat %1
72 call Time-of-Day.bat %1

```

```

73 call Highway_Assignment_Parallel.bat %1
74 call Average_Link_Speeds.bat %1
75 call Highway_Skims.bat %1
76 REM call HSR_Highway_Skims_2017HOV2.bat %1
77
78 :: rem ===== Iteration 3 =====
79
80 set _iter_=i3
81 set _prev_=i2
82 set _relGap_=0.001
83
84 call Transit_Skim_All_Modes_Parallel.bat %1
85 call Transit_Fare.bat %1
86 call Trip_Generation.bat %1
87 call Trip_Distribution.bat %1
88 call Mode_Choice_Parallel.bat %1
89 call Auto_Driver.bat %1
90 call Time-of-Day.bat %1
91 call Highway_Assignment_Parallel.bat %1
92 call Average_Link_Speeds.bat %1
93 call Highway_Skims.bat %1
94 REM call HSR_Highway_Skims_2017HOV2.bat %1
95
96 :: rem ===== Iteration 4 =====
97
98 set _iter_=i4
99 set _prev_=i3
100 set _relGap_=0.0001
101
102 call Transit_Skim_All_Modes_Parallel.bat %1
103 call Transit_Fare.bat %1
104 call Trip_Generation.bat %1
105 call Trip_Distribution.bat %1
106 call Mode_Choice_Parallel.bat %1
107 call Auto_Driver.bat %1
108 call Time-of-Day.bat %1
109 call Highway_Assignment_Parallel.bat %1
110 call Average_Link_Speeds.bat %1
111 call Highway_Skims.bat %1
112 REM call HSR_Highway_Skims_2017HOV2.bat %1
113
114 :: rem ===== Transit assignment =====
115 @echo Starting Transit Assignment Step
116 @date /t & time/t
117
118 call Transit_Assignment_Parallel.bat %1
119 call TranSum.bat %1
120
121 @echo End of batch file
122 @date /t & time/t
123 :: rem ===== End of batch file =====
124
125 REM cd %1
126 REM copy *.txt MDP_%useMDP%\*.txt
127 REM copy *.rpt MDP_%useMDP%\*.rpt
128 REM copy *.log MDP_%useMDP%\*.log
129 REM CD..
130
131 set _year_=
132 set _alt_=
133 set _iter_=
134 set _prev_=
135 set _maxUeIter_=
136 set _relGap_=

```

Regarding the 2020\_final scenario, whose “run model steps” batch file is shown in Figure 11, Table 12 shows that, just as was the case for 2020\_base, the environment variable “\_tspath\_” (transit constraint



path) is still set to blank/null (see line 18 in Figure 11), since 2020 is the constraining year (i.e., not a post constraining year, such as 2040). However, in contrast to what was done for 2020\_base, for the 2020\_final scenario, we do apply the HOV2/3+ skim substitution/replacement technique, as indicated in Table 12, so we do need to designate the location of the HOV2/3+ baseline skims. Thus, the “\_HOV3PATH\_” environment variable is set to “..\2020\_base” (the double dots means move up one level in the folder structure; see line 21 in Figure 11). Also, in Figure 11, we have highlighted in yellow the lines where we have commented out (via “REM”) one line (“call Highway\_Skims.bat”) and added in a new line (“call HSR\_Highway\_Skims\_2017HOV2.bat”), which calls the HOV2/3+ Skim substitution/Replacement (HSR) process (lines 43, 59, 76, 94, and 112). This process is described in section 2.4.2 of this report and in Chapter 8 of the calibration report.

Figure 11 “Run model steps” batch file for 2020\_final

```

1  :: E:\Data\fy17\CGV2_3_66_Conformity2016CLRP_Xmittal\run_ModelSteps_2020_Final.bat
2  :: Version 2.3.66
3  :: 10/31/2016 3:59:28 PM
4
5
6  :: Version 2.3 TPB Travel Model on 3722 TAZ System
7
8  set _year_=2020
9  set _alt_=Ver2.3.66_2020_Final
10 :: Maximum number of user equilibrium iterations used in traffic assignment
11 :: User should not need to change this. Instead, change _relGap_ (below)
12 set _maxUeIter_=1000
13
14 :: Set transit constraint path and files
15 :: Current year used to set the constraint = 2020
16 :: For years before constraint year: set _tcpath_=<blank>
17 :: For years after constraint year: set _tcpath_=..\2020_final
18 set _tcpath_=
19
20 :: Location of substitute HOV3+ skims
21 set _HOV3PATH_=..\2020_base
22
23 :: UE relative gap threshold: Progressive (10^-2 for pp-i2, 10^-3 for i3, & 10^-4 for i4)
24 :: Set the value below
25
26 rem ===== Pump Prime Iteration =====
27
28 set _iter_=pp
29 set _prev_=pp
30 set _relGap_=0.01
31
32 call ArcPy_Walkshed_Process.bat %1
33 call Set_CPI.bat %1
34 call PP_Highway_Build.bat %1
35 call PP_Highway_Skims.bat %1
36 call Transit_Skim_All_Modes_Parallel.bat %1
37 call Trip_Generation.bat %1
38 call Trip_Distribution.bat %1
39 call PP_Auto_Drivers.bat %1
40 call Time-of-Day.bat %1
41 call Highway_Assignment_Parallel.bat %1
42 REM rem call Highway_Skims.bat %1
43 call HSR_Highway_Skims_2017HOV2.bat %1
44
45 :: rem ===== Iteration 1 =====
46
47 set _iter_=i1
48 set _prev_=pp
49

```

```

50 call Transit_Skim_All_Modes_Parallel.bat %1
51 call Transit_Fare.bat %1
52 call Trip_Generation.bat %1
53 call Trip_Distribution.bat %1
54 call Mode_Choice_Parallel.bat %1
55 call Auto_Driver.bat %1
56 call Time-of-Day.bat %1
57 call Highway_Assignment_Parallel.bat %1
58 rem call Highway_Skims.bat %1
59 call HSR_Highway_Skims_2017HOV2.bat %1
60
61 :: rem ===== Iteration 2 =====
62
63 set _iter_=i2
64 set _prev_=i1
65
66 call Transit_Skim_All_Modes_Parallel.bat %1
67 call Transit_Fare.bat %1
68 call Trip_Generation.bat %1
69 call Trip_Distribution.bat %1
70 call Mode_Choice_Parallel.bat %1
71 call Auto_Driver.bat %1
72 call Time-of-Day.bat %1
73 call Highway_Assignment_Parallel.bat %1
74 call Average_Link_Speeds.bat %1
75 rem call Highway_Skims.bat %1
76 call HSR_Highway_Skims_2017HOV2.bat %1
77
78 :: rem ===== Iteration 3 =====
79
80 set _iter_=i3
81 set _prev_=i2
82 set _relGap_=0.001
83
84 call Transit_Skim_All_Modes_Parallel.bat %1
85 call Transit_Fare.bat %1
86 call Trip_Generation.bat %1
87 call Trip_Distribution.bat %1
88 call Mode_Choice_Parallel.bat %1
89 call Auto_Driver.bat %1
90 call Time-of-Day.bat %1
91 call Highway_Assignment_Parallel.bat %1
92 call Average_Link_Speeds.bat %1
93 rem call Highway_Skims.bat %1
94 call HSR_Highway_Skims_2017HOV2.bat %1
95
96 :: rem ===== Iteration 4 =====
97
98 set _iter_=i4
99 set _prev_=i3
100 set _relGap_=0.0001
101
102 call Transit_Skim_All_Modes_Parallel.bat %1
103 call Transit_Fare.bat %1
104 call Trip_Generation.bat %1
105 call Trip_Distribution.bat %1
106 call Mode_Choice_Parallel.bat %1
107 call Auto_Driver.bat %1
108 call Time-of-Day.bat %1
109 call Highway_Assignment_Parallel.bat %1
110 call Average_Link_Speeds.bat %1
111 rem call Highway_Skims.bat %1
112 call HSR_Highway_Skims_2017HOV2.bat %1
113
114 :: rem ===== Transit assignment =====
115 @echo Starting Transit Assignment Step
116 @date /t & time/t
117

```

```

118 call Transit_Assignment_Parallel.bat %1
119 call TranSum.bat %1
120
121
122 @echo End of batch file
123 @date /t & time/t
124 :: rem ===== End of batch file =====
125
126 REM cd %1
127 REM copy *.txt MDP_%useMDP%\*.txt
128 REM copy *.rpt MDP_%useMDP%\*.rpt
129 REM copy *.log MDP_%useMDP%\*.log
130 REM CD..
131
132 set _year_=
133 set _alt_=
134 set _iter_=
135 set _prev_=
136 set _maxUeIter_=
137 set _relGap_=

```

Regarding the 2040\_base scenario, whose “run model steps” batch file is shown in Figure 12, Table 12 shows that, yes, we are using the Metrorail constraint process. Thus, there are two changes to the batch file. First, the “\_tspath\_” environment variable is set to the location containing the Metrorail trips for the constraint year (2020), thus “set \_tspath\_=..\2020\_final” (see line 18 in Figure 12). Second, the lines that had been used to apply the mode choice process in previous runs (“call Mode\_Choice\_Parallel.bat”) have now been changed to call the Metrorail constraint mode choice process (“call Mode\_Choice\_TC\_V23\_Parallel.bat”). These changes have been highlighted in yellow in Figure 12 (e.g., line 53). Regarding the modeling of HOT lanes, Table 12 shows us that, since this is a “base” scenario (i.e., since we are not yet applying the HOV3+ skim substitution/replacement technique), we do not need to designate the location of the HOV2/3+ baseline skims, so the “\_HOV3PATH\_” environment variable is set to blank/null (see line 21 in Figure 12).

Figure 12 “Run model steps” batch file for 2040\_base

```

1  :: E:\Data\fy17\CGV2_3_66_Conformity2016CLRP_Xmittal\run_ModelSteps_2040_base.bat
2  :: Version 2.3.66
3  :: 11/7/2016 11:18:23 AM
4
5
6  :: Version 2.3 TPB Travel Model on 3722 TAZ System
7
8  set _year_=2040
9  set _alt_=Ver2.3.66_2040_base
10 :: Maximum number of user equilibrium iterations used in traffic assignment
11 :: User should not need to change this.  Instead, change _relGap_ (below)
12 set _maxUeIter_=1000
13
14 :: Set transit constraint path and files
15 :: Current year used to set the constraint = 2020
16 :: For years before constraint year: set _tspath_=<blank>
17 :: For years after constraint year: set _tspath_=..\2020_final
18 set _tspath_=..\2020_final
19
20 :: Location of substitute HOV3+ skims
21 set _HOV3PATH_=
22
23 :: UE relative gap threshold: Progressive (10^-2 for pp-i2, 10^-3 for i3, & 10^-4 for i4)
24 :: Set the value below
25
26 rem ===== Pump Prime Iteration =====

```

```

27
28 set _iter_=pp
29 set _prev_=pp
30 set _relGap_=0.01
31
32 call ArcPy_Walkshed_Process.bat %1
33 call Set_CPI.bat %1
34 call PP_Highway_Build.bat %1
35 call PP_Highway_Skims.bat %1
36 call Transit_Skim_All_Modes_Parallel.bat %1
37 call Trip_Generation.bat %1
38 call Trip_Distribution.bat %1
39 call PP_Auto_Drivers.bat %1
40 call Time-of-Day.bat %1
41 call Highway_Assignment_Parallel.bat %1
42 call Highway_Skims.bat %1
43
44 :: rem ===== Iteration 1 =====
45
46 set _iter_=i1
47 set _prev_=pp
48
49 call Transit_Skim_All_Modes_Parallel.bat %1
50 call Transit_Fare.bat %1
51 call Trip_Generation.bat %1
52 call Trip_Distribution.bat %1
53 call Mode_Choice_TC_V23_Parallel.bat %1
54 call Auto_Driver.bat %1
55 call Time-of-Day.bat %1
56 call Highway_Assignment_Parallel.bat %1
57 call Highway_Skims.bat %1
58
59 :: rem ===== Iteration 2 =====
60
61 set _iter_=i2
62 set _prev_=i1
63
64 call Transit_Skim_All_Modes_Parallel.bat %1
65 call Transit_Fare.bat %1
66 call Trip_Generation.bat %1
67 call Trip_Distribution.bat %1
68 call Mode_Choice_TC_V23_Parallel.bat %1
69 call Auto_Driver.bat %1
70 call Time-of-Day.bat %1
71 call Highway_Assignment_Parallel.bat %1
72 call Average_Link_Speeds.bat %1
73 call Highway_Skims.bat %1
74
75 :: rem ===== Iteration 3 =====
76
77 set _iter_=i3
78 set _prev_=i2
79 set _relGap_=0.001
80
81 call Transit_Skim_All_Modes_Parallel.bat %1
82 call Transit_Fare.bat %1
83 call Trip_Generation.bat %1
84 call Trip_Distribution.bat %1
85 call Mode_Choice_TC_V23_Parallel.bat %1
86 call Auto_Driver.bat %1
87 call Time-of-Day.bat %1
88 call Highway_Assignment_Parallel.bat %1
89 call Average_Link_Speeds.bat %1
90 call Highway_Skims.bat %1
91
92 :: rem ===== Iteration 4 =====
93
94 set _iter_=i4

```

```

95 set _prev_=i3
96 set _relGap_=0.0001
97
98 call Transit_Skim_All_Modes_Parallel.bat %1
99 call Transit_Fare.bat %1
100 call Trip_Generation.bat %1
101 call Trip_Distribution.bat %1
102 call Mode_Choice_TC_V23_Parallel.bat %1
103 call Auto_Driver.bat %1
104 call Time-of-Day.bat %1
105 call Highway_Assignment_Parallel.bat %1
106 call Average_Link_Speeds.bat %1
107 call Highway_Skims.bat %1
108
109 :: rem ===== Transit assignment =====
110 @echo Starting Transit Assignment Step
111 @date /t & time/t
112
113 call Transit_Assignment_Parallel.bat %1
114 call TranSum.bat %1
115
116 @echo End of batch file
117 @date /t & time/t
118 :: rem ===== End of batch file =====
119
120 REM cd %1
121 REM copy *.txt MDP_%useMDP%\*.txt
122 REM copy *.rpt MDP_%useMDP%\*.rpt
123 REM copy *.log MDP_%useMDP%\*.log
124 REM CD..
125
126 set _year_=
127 set _alt_=
128 set _iter_=
129 set _prev_=
130 set _maxUeIter_=
131 set _relGap_=

```

Regarding the 2040\_final scenario, whose “run model steps” batch file is shown in Figure 13, Table 12 shows that, yes, again, we are using the Metrorail constraint process. Thus, again, there are two changes to the batch file. First, the “\_tspath\_” environment variable is set to the location containing the Metrorail trips for the constraint year (2020), thus “set \_tspath\_=..\2020\_final” (see line 18 in Figure 13). Second, the lines that had been used to apply the mode choice process in previous runs (“call Mode\_Choice\_Parallel.bat”) have been changed to call the Metrorail constraint mode choice process (“call Mode\_Choice\_TC\_V23\_Parallel.bat”). These changes have been highlighted in yellow in Figure 13 (e.g., line 54). Regarding the modeling of HOT lanes, Table 12 shows us that, since this is a “final” scenario (i.e., we are applying the HOV3+ skim substitution/replacement technique, but not the HOV2+ skim substitution/replacement technique), we do need to designate the location of the HOV3+ baseline skims. Thus, the “\_HOV3PATH\_” environment variable is set to “..\2040\_base” (see line 21 in Figure 13). Also, in Figure 13, we have highlighted in yellow the lines where we have commented out (via “REM”) one line (“call Highway\_Skims.bat”) and added in a new line (“call HSR\_Highway\_Skims.bat”), which calls the HOV3+ Skim substitution/Replacement (HSR) process.

Figure 13 “Run model steps” batch file for 2040\_final

```

1 :: E:\Data\fy17\CGV2_3_66_Conformity2016CLRP_Xmittal\run_ModelSteps_2040_Final.bat
2 :: Version 2.3.66
3 :: 11/8/2016 3:59:28 PM
4

```

```

5
6 :: Version 2.3 TPB Travel Model on 3722 TAZ System
7
8 set _year_=2040
9 set _alt_=Ver2.3.66_2040_Final
10 :: Maximum number of user equilibrium iterations used in traffic assignment
11 :: User should not need to change this. Instead, change _relGap_ (below)
12 set _maxUeIter_=1000
13
14 :: Set transit constraint path and files
15 :: Current year used to set the constraint = 2020
16 :: For years before constraint year: set _tcpath_=<blank>
17 :: For years after constraint year: set _tcpath_=..\2020_final
18 set _tcpath_=..\2020_final
19
20 :: Location of substitute HOV3+ skims
21 set _HOV3PATH_=..\2040_base
22
23 :: UE relative gap threshold: Progressive (10^-2 for pp-i2, 10^-3 for i3, & 10^-4 for i4)
24 :: Set the value below
25
26 rem ===== Pump Prime Iteration =====
27
28 set _iter_=pp
29 set _prev_=pp
30 set _relGap_=0.01
31
32 call ArcPy_Walkshed_Process.bat %1
33 call Set_CPI.bat %1
34 call PP_Highway_Build.bat %1
35 call PP_Highway_Skims.bat %1
36 call Transit_Skim_All_Modes_Parallel.bat %1
37 call Trip_Generation.bat %1
38 call Trip_Distribution.bat %1
39 call PP_Auto_Drivers.bat %1
40 call Time-of-Day.bat %1
41 call Highway_Assignment_Parallel.bat %1
42 rem call Highway_Skims.bat %1
43 call HSR_Highway_Skims.bat %1
44
45 :: rem ===== Iteration 1 =====
46
47 set _iter_=i1
48 set _prev_=pp
49
50 call Transit_Skim_All_Modes_Parallel.bat %1
51 call Transit_Fare.bat %1
52 call Trip_Generation.bat %1
53 call Trip_Distribution.bat %1
54 call Mode_Choice_TC_V23_Parallel.bat %1
55 call Auto_Driver.bat %1
56 call Time-of-Day.bat %1
57 call Highway_Assignment_Parallel.bat %1
58 rem call Highway_Skims.bat %1
59 call HSR_Highway_Skims.bat %1
60
61 :: rem ===== Iteration 2 =====
62
63 set _iter_=i2
64 set _prev_=i1
65
66 call Transit_Skim_All_Modes_Parallel.bat %1
67 call Transit_Fare.bat %1
68 call Trip_Generation.bat %1
69 call Trip_Distribution.bat %1
70 call Mode_Choice_TC_V23_Parallel.bat %1
71 call Auto_Driver.bat %1
72 call Time-of-Day.bat %1

```

```

73 call Highway_Assignment_Parallel.bat %1
74 call Average_Link_Speeds.bat %1
75 rem call Highway_Skims.bat %1
76 call HSR_Highway_Skims.bat %1
77
78 :: rem ===== Iteration 3 =====
79
80 set _iter_=i3
81 set _prev_=i2
82 set _relGap_=0.001
83
84 call Transit_Skim_All_Modes_Parallel.bat %1
85 call Transit_Fare.bat %1
86 call Trip_Generation.bat %1
87 call Trip_Distribution.bat %1
88 call Mode_Choice_TC_V23_Parallel.bat %1
89 call Auto_Driver.bat %1
90 call Time-of-Day.bat %1
91 call Highway_Assignment_Parallel.bat %1
92 call Average_Link_Speeds.bat %1
93 rem call Highway_Skims.bat %1
94 call HSR_Highway_Skims.bat %1
95
96 :: rem ===== Iteration 4 =====
97
98 set _iter_=i4
99 set _prev_=i3
100 set _relGap_=0.0001
101
102 call Transit_Skim_All_Modes_Parallel.bat %1
103 call Transit_Fare.bat %1
104 call Trip_Generation.bat %1
105 call Trip_Distribution.bat %1
106 call Mode_Choice_TC_V23_Parallel.bat %1
107 call Auto_Driver.bat %1
108 call Time-of-Day.bat %1
109 call Highway_Assignment_Parallel.bat %1
110 call Average_Link_Speeds.bat %1
111 rem call Highway_Skims.bat %1
112 call HSR_Highway_Skims.bat %1
113
114 :: rem ===== Transit assignment =====
115 @echo Starting Transit Assignment Step
116 @date /t & time/t
117
118 call Transit_Assignment_Parallel.bat %1
119 call TranSum.bat %1
120
121 @echo End of batch file
122 @date /t & time/t
123 :: rem ===== End of batch file =====
124
125 REM cd %1
126 REM copy *.txt MDP_%useMDP%\*.txt
127 REM copy *.rpt MDP_%useMDP%\*.rpt
128 REM copy *.log MDP_%useMDP%\*.log
129 REM CD..
130
131 set _year_=
132 set _alt_=
133 set _iter_=
134 set _prev_=
135 set _maxUeIter_=
136 set _relGap_=

```

All five of the “run model steps” batch files (Figure 9 to Figure 13) apply the progressive relative gap procedure by using the “\_relGap\_” environment variable. The relGap variable starts at a value of 0.01

( $10^{-2}$ ) on line 29 or 30 (depending on the batch file), and then changes to 0.001 ( $10^{-3}$ ) on lines 77-78, before attaining the final value of 0.0001 ( $10^{-4}$ ) in speed-feedback iteration 4 (lines 94-95). This is covered in more detail in the text surrounding both Table 3 and Table 83.

### 6.3 Running the model: An example

This section provides an example of how to run the travel model for the year 2015 (2015\_base) using the travel model package that is typically transmitted to external users. It is assumed that the user has copied the transmitted model into the folder where it will be run and did not modify any input files or folder names. The top-level folder is referred to as the “root” folder and typically has a name referring to the travel model version and the particular modeling project (e.g. “Ver2.3.66\_aqc\_2016clrp”). The root folder contains all of the batch files and modeling folders (shown in Figure 4). The folder/subdirectory called “2016\_base” is referred to as the “outputs” folder or the “scenario-specific” folder. It should also be noted that, in order to follow the steps below, one should have followed all the steps in Chapter 5 (“Preparing to run the model”). It is also assumed that you are not making any changes to the default parallel processing setup in the model, which requires a computer with eight cores. Otherwise, see instructions found in Table 17 on p. 88.

1. Ensure that the root folder has the two parent batch files:
  - a. run\_Model\_2016\_base.bat, which is known as the wrapper batch file or the “run model” batch file.
  - b. run\_ModelSteps\_2016\_base.bat, which is known as the “run model steps” batch file.
2. These two batch files are ready to go and should not need any editing for a normal model run. However, the general practice would be to open the wrapper file in a text editor (do not double click it, since this will launch the model run) and check the following:
  - a. The “root” environment variable should be set equal to “.” (which means the current working directory, i.e., the current folder in your command window)
  - b. The “scenar” environment variable should be set equal to “2016\_base”
  - c. The “runbat” environment variable should be set equal to “run\_ModelSteps\_2016\_base.bat” (which is the relevant “run model steps” batch file)
  - d. Other environment variables, such as those used for distributed processing, are explained in the chapter on parallel processing.

3. Open a command window and navigate to the root folder OR

Use Windows Explorer/File Explorer to navigate to the root folder, deselect all files, shift-right-click in the right pane of Windows Explorer/File Explorer, and select “open command window here.” (Explained in more detail on page in the section entitled “Parent batch files” which begins on page 57).

4. Type “run\_Model\_2016\_base.bat” (without the quotes) and hit Enter.

The model run should begin and the user should see numerous commands scrolling in the command window.



## 7 Summarizing model output and other utilities

### 7.1 Summary Scripts

In addition to the model, the user is provided with a number of summary scripts, which may be helpful in analyzing the model output. These are listed in Table 13.

Table 13 Travel Model Summary Scripts

Summary script	Description	Folder
Compare_mode_choice.s	Compares <b>estimated</b> mode choice results to HTS <b>observed</b> mode choice data	summary
Compare_mode_choice_adj.s	Compares <b>estimated</b> mode choice results to adjusted HTS and transit surveys <b>observed</b> mode choice data	summary
COMPARE_NL_MC.S	Compares <b>estimated</b> mode choice results between two different model runs.	summary
Compare_Trip_Distribution.s	Compares estimated trip distribution to observed trip distribution from HTS	summary
view_from_space_v2.3.66.s	Creates global summary of demographic info, trips, and VMT	summary
COMPARE_NL_MC_Expanded_Alt_minus_Base_V23_52.S	Compares two scenarios developed with Ver. 2.3.52 or later	summary
COMPARE_NL_MC_Expanded_Alt_V23_52_minus_Base_V23_39.S	Compares alternative developed with V2.3.52 and base developed with V.2.3.39. Such a script is needed because naming conventions for output files changed between Build 39 and Build 52.	summary
Summarize_Est_Obs_Volume_Daily.s	Compares estimated daily volumes on select links to observed counts. Also compares estimated and observed daily screenline volumes.	assignment_summary
Summarize_Est_Obs_Volume_Period.s	Compares estimated AM, MD, PM, and NT volumes on select links to observed counts	assignment_summary
Summarize_2007_2040_Screenlines.s	Compares estimated screenlines volumes in 2007 and 2040	assignment_summary
Plot_Volume_Differences.s	Plots volume differences between two input networks	assignment_summary

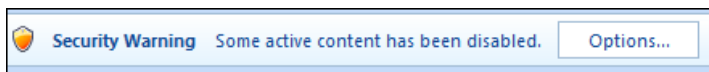
Additionally, the program LineSum.exe is used for summarizing the transit assignment (see Chapter 24 (“Transit Assignment, Including Summary Process (LineSum)”)).

## 7.2 Utilities

The Version 2.3 Travel Model requires a number of input files in various file formats. One of the file formats is dBase or DBF. Compared to space-delimited text files, DBF files have a number of advantages (e.g., fields do not mistakenly run together when values become large), but DBF files can also have some drawbacks, e.g., they can be difficult to create and it can be difficult to compare two DBF files. On this second issue, there are a number of utilities for comparing or “diffing” text files (such as the Unix/Linux `diff` command, WinDiff, WinMerge, PSPad, and Notepad++), but it is more difficult to find programs that allow one to compare DBF files. To facilitate such comparison, a member of the TPB staff, Feng Xie, has developed a utility, known as the DBF Converter (DBF\_Converter\_v3.2.xls) that enables the user to convert DBF files to text files in comma-separated variable (CSV) format. This conversion can also be done within Cube (using File > Export). Once the files are converted to CSV format, it is easier to compare or “diff” them using other existing utilities. TPB staff is making this DBF converter available to users of the regional travel model to aid in checking/comparing input files.

Using the DBF converter, the user has the option of converting all DBF files in a directory or a select subset of the files. This utility requires Microsoft Office Excel software.

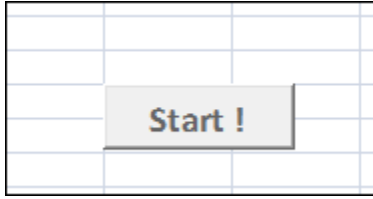
To begin the process, the user double clicks on the converter file/icon (DBF\_Converter\_v3.2.xls), which will open an Excel spreadsheet. In the center of the spreadsheet, there is a “Start” button. Before clicking on this button, the user has to enable the button by clicking on the “Options...” button:



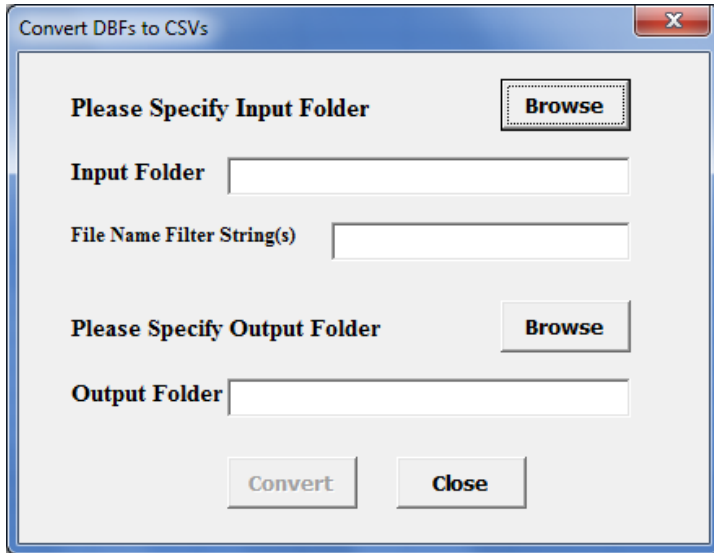
When prompted, the user will have to click “Enable this content” and “OK”:



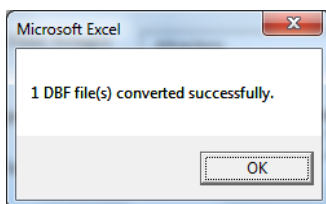
Now, the user can click on the “Start” button:



This will result in the following pop-up window prompting the user to enter the input folder, output folder, and the file name filter string(s):



The input folder must contain the DBF files that the user wishes to convert. The output folder is the folder where the newly created CSV files will be placed. The Input/Output folders can be specified by either by clicking the "Browse" button or by typing/pasting in the text boxes. Once the user has selected the input and output folders, he or she may wish to specify a file name filter string. The filter string textbox allows multiple filter strings, separated by spaces. If the user would like to convert all DBF files in the input folder, then the "File Name Filter String(s)" field should be left blank.<sup>52</sup> However, if the user wishes to convert only one DBF file or only a subset of DBF files in the input folder, he or she should specify either a full or partial file name. When the fields are filled, the user needs to press the "Convert" button. Once the conversion process is complete, the user will see a pop-up window stating that the DBF file(s) were converted successfully.



---

<sup>52</sup> Note, however, that this can take several minutes, since there are over 100 files.

Once the converter has run, the user can find the newly created CSV file(s) in the specified output folder.

## 8 Use of parallel processing to reduce model run times

### 8.1 Model run times

In the period from 2008 to 2011, when COG/TPB staff had first transitioned from the Version 2.2 Travel Model to the Version 2.3 Travel Model, we noticed that the Version 2.3 model required much longer run times. For example, using a server bought in 2009 (such as COG's travel model server 3, or tms3), a run of the Ver. 2.2 Travel Model took 15-20 hours, whereas a run of the Version 2.3 Travel Model required about 80-90 hours initially (a factor of 4.5 times or 350%), which was later reduced to about 30 hours in 2012 by using Cube Cluster, Citilabs' implementation of distributed processing. On a newer travel model server, such as tms6 or tms7, the model run time is about 18-20 hours.

There are several reasons why the Version 2.3 Travel Model, when it was first developed, had such long run times, compared to its predecessor, the Version 2.2 Travel Model. First, the number of transportation analysis zones (TAZs) increased from 2,191 to 3,722. This represents a 70% increase in the number of TAZs and a 189% increase in matrix sizes used to store trip tables and travel time skims ( $3,722^2/2,191^2$ ). The other factors causing longer run times are associated mainly with refinements to the Version 2.2 traffic assignment process:

- The number of time-of-day periods went from three (AM, PM, and off peak) to four (AM, midday, PM, night/early morning)
- The number of user classes went from five to six (an explicit commercial-vehicle user class has been added);
- The number of traffic assignments has increased. The Version 2.2 Travel Model had originally used three traffic assignments, one for each time-of-day period (AM, PM, and off peak). Later versions of the Version 2.2 Travel Model split the peak assignments into two groups (HOV3+ and non-HOV3+, the so called "two step traffic assignment"), resulting in the five assignments shown in the left-hand column of Table 14. In the Version 2.3 Travel Model, the off-peak period has been further split into two parts: midday and night/early morning. So, the number of traffic assignments has increased from five in Version 2.2 to six in Version 2.3.
- Higher convergence thresholds
  - In the Version 2.2 model, all five traffic assignments were run with 60 user equilibrium (UE) iterations. This resulted in a range of relative gaps values, from a low value of  $1.10 \times 10^{-4}$  (0.0001) for the AM HOV3+ assignment to a high of  $1.19 \times 10^{-2}$  (0.0119) for the AM non-HOV3+ assignment.<sup>53 54</sup>
  - In the Version 2.3 model, prior to Build 52, all six traffic assignments were run to either a relative gap of 0.001 ( $1 \times 10^{-3}$ ) or 300 user equilibrium iterations, whichever came first.

---

<sup>53</sup> From a model run representing year-2002 conditions from the air quality conformity determination of the 2009 CLRP/FY 2010-2015 TIP.

<sup>54</sup> The modeler can check the relative gap by consulting the highway assignment report file for the final speed feedback iteration (i.e., i4\_Highway\_Assignment.rpt). The variable is called RELGAP.

For the Version 2.3.52, 57, 57a, and 66, we use a progressively tightening relative gap procedure, which is described in more detail later in this chapter.

**Table 14** Five traffic assignments in the Version 2.2 travel model becomes six traffic assignments in the Version 2.3 travel model

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

## 8.2 Use of parallel processing to reduce model run times

One way to reduce model run times is to buy quicker hardware. However, there are limits to this approach, given the recent trend of chip makers, such as Intel, to focus less on increasing clock speeds and focus more on increasing the number of cores (i.e., the capacity) of computer processors. As evidence of this trend, one of COG's newer travel model servers, tms6, has a processor whose clock speed is 16% *slower* than that of its predecessor (travel model server #5, or tms5). By contrast, the number of cores has gone from 12 physical cores (24 virtual cores with Hyper-Threading) in tms5 to 16 physical cores (32 virtual cores with Hyper-Threading) in tms6. Consequently, we have focused on achieving run time reductions via the software side of the equation. COG's newest travel model server, tms7, has a clock speed of 3.1 GHz, has two processors, each with 10 physical cores, which, with Hyper-Threading turned on, appears to the operating system as 40 logical processors (virtual cores), as noted in Table 4.

We use the term "parallelization" to mean running two or more processes or threads in parallel. By running two or more steps in parallel, one can reduce model run time. A common way to achieve this parallelization is by using distributed processing, which essentially distributes the computing load across multiple computer processors or cores. These computer processors/cores could be in separate computers (linked by a local area network or LAN) or could be on one computer that has multiple cores. Citilabs has its own implementation of distributed processing called Cube Cluster, which is an add-on component of Cube Voyager. There are two forms of distributed processing available in Cube Cluster:

- "Intrastep distributed processing (IDP): This type of distributed processing works by breaking up zone based processing in a single step into zone groups that can be processed concurrently on multiple computing nodes. Currently only the Matrix and the Highway programs are available for IDP."<sup>55</sup>
- "Multistep distributed processing (MDP): This type of distributed processing works by breaking up blocks of one or more modeling steps and distributes them to multiple computing nodes to

<sup>55</sup> Citilabs, Inc., "Cube Voyager Reference Guide, Version 6.4.1" (Citilabs, Inc., September 30, 2015), 1124–25.

process. This can be used for any program in Cube Voyager as well as user-written programs with the caveat that the distributed blocks and the mainline process must be logically independent of each other.”<sup>56</sup>

The Version 2.3. Travel Model uses both IDP and MDP, and also uses a third method of parallelization that is already part of the Windows operating system: Running programs in parallel using multiple concurrent command windows.

### 8.2.1 Background and terminology

A computer contains a central processing unit (CPU), which is also known as a chip or processor. Modern CPUs are often divided into two to ten. A core functions as a separate processor, so, to an operating system, a computer with two CPUs is the same as a computer with one CPU divided into two cores. The two biggest chip manufacturers for computers running the Microsoft Windows operating system are Intel and AMD. COG/TPB staff has run the Version 2.3 Travel Model on only computers with Intel chips, but the model should run on computers with any Intel-like chip, such as AMD. Some Intel chips feature a technology known as Hyper-Threading. When Hyper-Threading technology is enabled on the chip, the operating system sees double the number of cores. So, if your computer has four cores and Hyper-Threading is enabled, the operating system will see eight virtual cores (or “logical processors”), thus doubling your CPU capacity. **Thus, a computer with one CPU that contains four cores and has Hyper-Threading enabled, should be able to run the Version 2.3 Travel Model “out of the box” without making changes to the “run model”/wrapper batch file, since it has eight virtual cores.**<sup>57</sup> When a computer executes a task, it uses a process or “thread.” In general, one process or thread runs on one processor or core. The operating system (Microsoft Windows) chooses the actual physical core to use when running a process, and, by opening up Windows Task Manager, one can see that the operating system appears to randomly move the task from one core to the next until the process completes, but the user need not focus on this detail. Cube Base documentation does briefly discuss Hyper-Threading.<sup>58</sup>

In Cube Cluster parlance, a set of processors that can be used for a computing task, whether they exist in one computer or a network of computers is called a “cluster.” Any individual processor or core is called a “computing node” or simply a “node.” Cube Cluster, which is a part of Cube Voyager, allows the nodes in the cluster to communicate, so that they can work together, essentially running in parallel, to accomplish a computing task. Citilabs originally wrote Cube Cluster with the idea that users would want to harness the power of multiple, run-of-the-mill PCs that were networked together using a local area network (LAN). However, COG/TPB staff has not used Cube Cluster in that way. Instead, COG/TPB staff has harnessed the power of Cube Cluster by running on one computer (server) at a time, by virtue of the fact that the computer contained multiple cores. If you are running Cube Cluster across multiple

---

<sup>56</sup> Ibid., 1125.

<sup>57</sup> According to one external user who had a computer with only four cores (it was not clear whether these were physical cores or virtual cores) found that the model crashed at the mode choice step. This was likely due to the fact that the default configuration of the model is designed to run five concurrent mode choice runs. However, this user was able to follow the procedures listed in Table 17 to get the model to run on the four-core computer.

<sup>58</sup> Citilabs, Inc., “Cube Base Reference Guide, Version 6.4.1” (Citilabs, Inc., September 30, 2015), 10–11.

computers, you would have a main computer, known as the “main node,” and also one or more helper computers, known as “sub-nodes” (or “subnodes”). When running Cube Cluster in a single computer with multiple cores, the “main node” and “sub-nodes” would then exist within the same CPU. So, continuing with the single-computer scenario, the user can think of a model run as occurring on a “main node” (which is simply one of the cores on the CPU), and the main node can then call upon one or more sub-nodes (other cores on the CPU) to make use of IDP or MDP.

### 8.2.2 Effect of Cube Cluster on modeled results

It should be noted that **using Cube Cluster can result in numerical rounding which can affect model results**. For instance, COG/TPB staff found that the use of IDP resulted in a very small change in the estimated VMT coming out of the travel model.<sup>59</sup> As part of a series of test conducted in 2011, COG/TPB staff conducted two model runs: 1) a year-2007 traffic assignment with IDP using 4 cores; and 2) a year-2007 traffic assignment without IDP (i.e., one core). COG/TPB staff then calculated the VMT difference between the two runs at the regional level, the jurisdiction level, and the link level. At the regional level, the use of IDP had almost no effect on modeled results – it resulted in only a 1/100th to 3/100ths of a percent drop in estimated VMT (slide 25). At the jurisdiction level, the use of IDP also resulted in almost no difference in estimated VMT – the difference was as large as 9/100ths of a percent for some jurisdictions (slide 27). At the link level, however, the use of IDP resulted in a number of cases where the VMT difference was above 20% (slide 29). Fortunately, the links with the largest volume differences were the lower-class facilities (e.g., not freeways). Both runs were done as part of the full travel model and both were done using Cube Voyager/Cluster version 5.1.2. Newer versions of Cube Voyager/Cluster are now available (e.g., COG is now using 6.4.1), but COG/TPB staff have not re-tried the sensitivity test with the newer versions of Cube Voyager. COG/TPB staff shared these results with Citilabs and, in 2012, Citilabs updated its documentation to note this rounding phenomenon: “Use of Cluster can have a very small effect on volumes generated by the HIGHWAY program. During the ADJUST phase, when iteration volumes are combined, the final assigned volumes might vary slightly over different numbers of cluster nodes.”<sup>60</sup>

### 8.2.3 History of adding parallelization to the Version 2.3 Travel Model

In Build 16 of the Version 2.3 Travel Model (Ver. 2.3.16), COG/TPB staff added IDP to the highway assignment script. Staff set the travel model up to use four cores, and, based on the findings of the aforementioned tests, staff recommended that users who wanted to replicate COG results also use four cores. In Builds 20 through 24 of the Version 2.3 Travel Model, COG/TPB staff added IDP to other modeling steps, such as MFARE2.s, Time-of-Day.s, and the transit skimming scripts. In 2012, COG asked for AECOM’s assistance to further reduce model run times. AECOM suggested model changes that

---

<sup>59</sup> See slides 25-32 of Ronald Milone and Mark S. Moran, “TPB Version 2.3 Travel Model on the 3,722-TAZ Area System: Status Report” (May 20, 2011 meeting of the COG/TPB Travel Forecasting Subcommittee, held at the Metropolitan Washington Council of Governments, Washington, D.C., May 20, 2011).

<sup>60</sup> See, for example, Citilabs, Inc., “Cube Voyager Reference Guide, Version 6.0.2” (Citilabs, Inc., July 26, 2012), 1009; or Citilabs, Inc., “Cube Voyager Reference Guide, Version 6.4.1,” 1129.



introduced MDP to the travel model.<sup>61</sup> Now, in addition to using four cores for IDP traffic assignment, the use of MDP allowed two traffic assignments to run in parallel (thus, 8 cores would be in use, but only 4 in each of the two IDP sessions). COG/TPB staff incorporated these AECOM recommendations into Build 40 of the Version 2.3 Travel Model (Ver. 2.3.40), and these same parallelization enhancements, such as the use of both IDP and MDP, also exist in the Version 2.3.52 Travel Model and later models.

#### 8.2.4 Implementation of parallelization in the Version 2.3.52 through 2.3.66 travel models

The Version 2.3.52 Travel Model (and later models, including Ver. 2.3.66) has three types of parallelization to help minimize run times:

- Cube Cluster intra-step distributed processing (IDP)
- Cube Cluster multi-step distributed processing (MDP)
- Windows operating system: Running programs in parallel using multiple concurrent command windows

IDP is used in three modeling steps:

- Highway assignment (Highway\_Assignment\_Parallel.s)
- Transit fare development (MFARE2.S)
- Transit assignment (Combine\_Tables\_For\_TrAssign\_Parallel.s)

By contrast, MDP is used for only one step: Highway assignment (Highway\_Assignment\_Parallel.s). In other words, highway assignment uses both IDP and MDP. The model is set up to use four cores in IDP, and, using MDP, there are two concurrent IDP sessions: A main node, which uses four cores via IDP, and a sub-node, which also uses four cores via IDP. **This combination of IDP and MDP means that highway assignment uses 8 cores concurrently for processing.**

Lastly, running programs in parallel by using multiple concurrent command windows is used for three modeling steps:

- Transit skimming (command windows invoked by Transit\_Skim\_All\_Modes\_Parallel.bat)
- Mode choice (command windows invoked by Mode\_Choice\_Parallel.bat or Mode\_Choice\_TC\_V23\_Parallel.bat)
- Transit assignment (command windows invoked by Transit\_Assignment\_Parallel.bat)

**IDP**, which works only for the MATRIX or HIGHWAY modules of Cube Voyager, is implemented in a Cube Voyager script using a **single line of code**, such as this from the Highway\_Assignment\_Parallel.s script:

---

<sup>61</sup> For more details, see AECOM and Stump/Hausman Partnership, "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" (National Capital Region Transportation Planning Board, Metropolitan Washington Council of Governments, July 13, 2012), chap. 5, <http://www.mwcog.org/uploads/committee-documents/aV1dWVhb20120720132722.pdf>.

```
distributeIntrastep processId='AM', ProcessList=%AMsubnode%
```

By contrast, **MDP** is implemented in a Cube Voyager script using **an MDP block of code**. The code block begins and ends with code such as the following (from the Highway\_Assignment\_Parallel.s script):

```
DistributeMULTISTEP ProcessID='AM', ProcessNum=1  
    (various lines of code)  
ENDDistributeMULTISTEP
```

The IDP statement above can be used on its own or within an MDP block. Examples of both of these cases can be found in the Highway\_Assignment\_Parallel.s script.

Table 15 shows the five modeling steps where parallelization is used, noting the method of parallelization (e.g., IDP, MDP, or batch file); the names of the batch files that call each step; the names of the tokens (variables) used to store the number of processing nodes/subnodes to use for IDP/MDP; and the maximum number of simultaneous threads/cores used by the step. For example, in the case of transit skimming, parallelization is achieved by calling multiple simultaneous batch files. The batch file that actually initiates the multiple Windows command windows is Transit\_Skim\_All\_Modes\_Parallel.bat, and, since this step uses neither IDP nor MDP, there are no associated IDP or MDP tokens. The transit skimming process uses 4 cores. In the highway assignment step, both MDP and IDP are used. The tokens used for IDP are AMsubnode and MDsubnode (more on this later in this chapter). Although MDP is used, no tokens are used for MDP. Instead, the subnode for MDP is labeled using a fixed name, "AM1". The highway assignment step can use up to 8 simultaneous nodes/cores, since IDP is implemented with four cores and there are two concurrent IDP sessions, run using MDP. As can be seen in Table 15, modeling steps with parallelization use 4, 5, or 8 cores. Modeling steps without parallelization use only one core at a time.

#### 8.2.4.1 Parallel processing in the "Run model"/wrapper batch file

This section of the report describes how the code in the "run model"/wrapper batch files affects parallel processing implemented in the Version 2.3.52 Travel Model. This section uses the 2010 "run model" batch file (Figure 8 on page 58) as an example. It also describes changes that can be made to the "run model"/wrapper batch files in order run the model on a computer with fewer than the standard 8 cores. The next section of the report, 8.2.4.3 on page 89, describes how parallel processing (specifically IDP and MDP) has been implemented in one script: Highway\_Assignment\_Parallel.s. As noted in Table 15, the number of cores used in each of the parallelized modeling steps varies from 4 to 8 cores. Those steps that do not contain parallelization use only one core at a time. **Thus, to run the Version 2.3.52 Travel Model "out of the box," without making any changes, one needs a computer with eight or more cores,** as was explained in the section 3.1 of the report.

Table 15 Modeling steps where parallelization is used, including the maximum number of threads/cores used

Modeling Step	First-Level "Child" Batch File	Second-Level "Child" Batch File	Method of Parallelization (batch file or script which calls parallel process)	Tokens Used for IDP**	Max. No. of Cores
Transit skimming	Transit_Skim_All_Modes_Parallel.bat	TransitSkim_LineHaul_Parallel.bat	<b>Batch file</b> (Transit_Skim_All_Modes_Parallel.bat)		4
Highway assignment	Highway_Assignment_Parallel.bat	None	<b>MDP &amp; IDP</b> (Highway_Assignment_Parallel.s)	AMsubnode MDsubnode	8
Transit fare development	Transit_Fare.bat	None	<b>IDP</b> (MFARE2.s)	subnode	4
Mode choice	Mode_Choice_Parallel.bat	MC_purp.bat	<b>Batch file</b> (Mode_Choice_Parallel.bat <u>or</u> Mode_Choice_TC_V23_Parallel.bat)		5
Transit assignment	Transit_Assignment_Parallel.bat	TransitAssignment_LineHaul_Parallel.bat	<b>Batch file</b> (Transit_Assignment_Parallel.bat) <b>IDP</b> (Combine_Tables_For_TrAssign_Parallel.s)	subnode	4

\*\* MDP as implemented in Highway\_Assignment\_Parallel.s does not use a token. Instead, the subnode name designation is done using a hard-coded value in the script, i.e.. "AM1", as is explained later in this chapter.

The “run model”/wrapper batch file makes use of a number of user-defined Windows environment variables. Those environment variables that do not deal with distributed processing were discussed in section 6.2.1 (“Description of the “run model”/wrapper batch file”) on page 61. By contrast, those environment variables that do deal with distributed processing are discussed in this chapter (Chapter 8).

It is possible to have IDP-related or MDP-related statements in a script, but not use them. Consequently, one of the first Cube Cluster statements in any script that uses Cube Cluster will be a statement that indicates whether Cube Cluster should be used or not. An example of such a statement is shown below:

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

The above statement indicates that IDP will be used (since its flag has been set to a value of TRUE or T) and MDP will not be used (since its flag has been set to a value of FALSE or F). In this example, even if there is code for MDP in the Cube Voyager script, the MDP processing will not be executed, since it has been set to FALSE. In our scripts and batch files, we generally use user-defined, Windows environment variables to set these two values. Thus, the statement above appears like this, using two “tokens” or variables to hold the true/false flags:

```
distribute intrastep=%useIdp% multistep=%useMdp%
```

In lines 12 and 13 of the “run model”/wrapper batch file (shown in Figure 8), these two “set” statements simply set the IDP and MDP usage flags to a value of TRUE:

```
set useIdp=t
set useMdp=t
```

The statement “distribute intrastep=%useIdp% multistep=%useMdp%” is used in both `Combine_Tables_For_TrAssign_Parallel.s` and `Highway_Assignment_Parallel.s`.<sup>62</sup> By contrast, in `MFARE2.s`, which uses IDP, but not MDP, the MDP flag has been hard-coded to FALSE, instead of using the token value set in the wrapper batch file: “distribute intrastep=%useIdp% multistep=f”.

The “useidp” environment variable is used in the three steps shown in Table 15 that make use of IDP. As one would expect, the “usemdp” environment variable is used in the highway assignment step, since this step makes use of MDP. However, the “usemdp” environment variable **is also used in Mode\_Choice\_Parallel.bat and Mode\_Choice\_TC\_V23\_Parallel.bat**. Specifically, if the “usemdp” flag is set to TRUE, then parallel processing is used in the mode choice step (via concurrent batch files, not MDP), which means that the five mode choice models (HBW, HBS, HBO, NHW, and NHO) are run in parallel command windows. If the “usemdp” flag is set to FALSE, then the mode choice process assumes that there is only one core available and runs the five mode choice models in sequence.

---

<sup>62</sup> This same statement is also currently found in the four transit skimming scripts (`Transit_Skims_AB|BM|CR|MR.s`), but it is no longer being used, so it should eventually be removed. At one point, we had used IDP in transit skimming, but, for the Ver. 2.3.40 model, when parallelization via concurrent batch files was added, the parallelization via IDP was dropped, so that the model would not use more than 8 concurrent cores.

The next two environment variables dealing with distributed processing can be found on in lines 15 and 16 of the “run model”/wrapper batch file (Figure 8):

```
set AMsubnode=1-4
set MDsubnode=2-4
```

As shown in Table 15, the AMsubnode and MDsubnode are used for IDP in traffic assignment. The names AMsubnode and MDsubnode would seem to indicate a subnode for processing the AM peak period and one for processing the midday period. Originally, when AECOM first proposed adding MDP to various steps of the model, it was added to both highway skimming and highway assignment. In highway skimming, there is a peak period skim, represented by the AM peak period, and an off-peak period skim, represented by the midday (MD) period. So, for skimming, the processing of the AM skims was sent off to a subnode, using MDP, and the AMsubnode variable/token was used to define the number of subnodes to use in IDP for the processing of the AM skims. And for the processing of the midday (MD) skims, this work was retained on the main processing node, with the MDsubnode variable/token used to define the number of subnodes to use in IDP for the processing of the MD skims.<sup>63</sup> For reasons of expediency, the same variable names (AMsubnode and MDsubnode) were used for the MDP in the highway assignment step. This meant that the AM peak period highway assignment was transferred, via MDP, to a subnode called “AM1.” But, in the case of the PM traffic assignment, it was processed on the main node, but it used the “MD” token for naming its IDP subnodes.<sup>64</sup> What’s more, the MD period was processed on branch/sub-node delineated “AM” (from the AMsubnode, not “MD”) and the NT period was processed by the main node, but was delineated “MD” (from MDsubnode, not “NT”). When COG/TPB staff chose which of the AECOM suggested parallelization enhancements to implement, it chose not to implement MDP in the highway skimming, just in highway assignment.<sup>65</sup> The end result was that the naming convention used in the highway assignment step is somewhat confusing, even though, strictly speaking, the code functions normally. This is explained in more detail in the next section of the report. Consequently, given the way that MDP is used in only the highway assignment step, one could come up with better names for the two variables that are currently called AMsubnode and MDsubnode, as shown in Table 16.

**Table 16 Current and alternate names for the two Windows environment variables that store information about the number of subnodes to use in IDP in the highway assignment**

<b>Current name of environment variable</b>	<b>Usage</b>	<b>Alternate name for environment variable</b>
AMsubnode	Number of IDP subnodes used within an MDP block	idp_for_mdp_branch
MDsubnode	Number of IDP subnodes used w/in main processing branch	idp_for_main_branch

<sup>63</sup> See, for example, AECOM and Stump/Hausman Partnership, “FY 2012 Report,” figs. 5–2.

<sup>64</sup> See, for example, *ibid.*, figs. 5–6.

<sup>65</sup> See Mary Martchouk and Mark S. Moran to Ronald Milone, “Reducing Model Run Times: Results from the TPB Staff Tests of AECOM’s Proposed Parallelization Enhancements to the Travel Model,” Memorandum, (September 17, 2012), 6.

Given that we generally have decided upon using four cores for IDP processing in the model (to maintain consistency), one might expect that AMsubnode = MDsubnode = 1-4. In other words, one might expect that we would provide Cube Cluster with a list of four nodes (1-4) for both the main branch of IDP processing and the MDP branch of IDP. According to AECOM, the reason for delineating only three subnodes (i.e., "MDsubnode=2-4") and not four, is that "only three slave threads [sub-nodes] are launched since the master uses itself as one of the threads to process the PM highway assignment."<sup>66</sup> Thus, despite the appearance of 3 nodes for MD and 4 nodes for AM, both IDP sessions use 4 nodes. To further clarify this issue, the IDP and MDP processes running in Highway\_Assignment\_Parallel.s have been diagrammed in Figure 14 and Figure 15 in section 8.2.4.3.

### 8.2.4.2 Changing the "run model"/wrapper batch file for computers with fewer than 8 cores

The Version 2.3 Travel Model (Ver. 2.3.52 and later) is designed to run on a computer that has 8 or more cores. Table 17 shows the changes that a user should make in order to run the Version 2.3 model on computers with fewer than 8 cores. See section 8.2.2 ("Effect of Cube Cluster on modeled results") on p. 82 for a discussion about how modeled results can change slightly with the number of cores used.

**Table 17 Running the Version 2.3 Travel Model on computers with fewer than 8 cores: Changes that need to be made to the "run model"/wrapper batch file**

Number of cores in your computer	Changes needed in the "run model"/wrapper batch file	Result
8 or more	<ul style="list-style-type: none"> <li>No changes need be made</li> </ul>	The model will run using between 1 and 8 cores, depending on the modeling step. Eight cores are used in highway assignment, due to the use of both IDP and MDP.
4	<ul style="list-style-type: none"> <li>Change "useMdp=t" to "useMdp=f"</li> </ul>	This change will mean that highway assignment no longer uses MDP, only four cores with IDP. Also, in the mode choice model, sequential processing will be done (i.e., the five models will no longer run in parallel).
2	<ul style="list-style-type: none"> <li>Change "useMdp=t" to "useMdp=f"</li> <li>Change "set AMsubnode=1-4" to "set AMsubnode=1-2"</li> <li>Change "set MDsubnode=2-4" to "set MDsubnode=2"</li> </ul>	This should result in only 2 cores being used in IDP.
1	<ul style="list-style-type: none"> <li>Change "useMdp=t" to "useMdp=f"</li> <li>Change "useldp=t" to "useldp=f"</li> </ul>	This will disable IDP and MDP and will also result in disabling the parallel processing in the mode choice step.

Note that the information in Table 17 is based on testing done by COG/TPB staff using a virtual computer with Cube 6.0.2 installed.

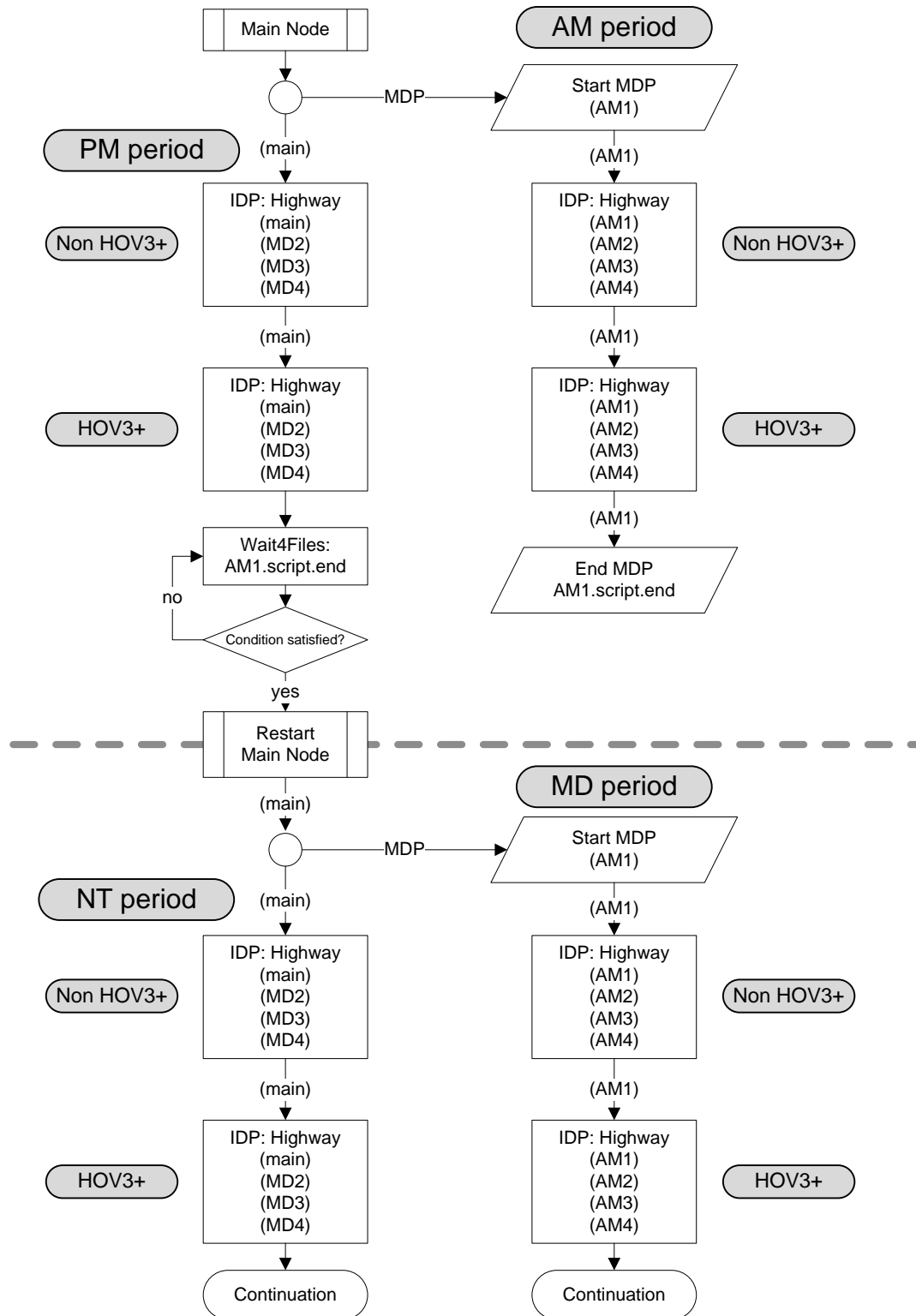
<sup>66</sup> AECOM and Stump/Hausman Partnership, "FY 2012 Report," 5-9-10.

### 8.2.4.3 Parallel processing in the highway assignment script

The previous section of the report, section 8.2.4.1, described how the code in the “run model”/wrapper batch files affects parallel processing implemented in the Version 2.3 Travel Model (Ver. 2.3.52 and later). This section of the report describes how parallel processing (specifically IDP and MDP) have been implemented in one script: Highway\_Assignment\_Parallel.s. Figure 14 shows a schematic of how IDP and MDP have been implanted in the highway assignment script. This figure shows the sub-node naming conventions that are used in the Version 2.3 Travel Model, keeping in mind some of the issues related to the naming of sub-nodes in section 8.2.4.1. Figure 15 is a revision of Figure 14, which shows a schematic of how IDP and MDP have been implanted in the highway assignment script, but with more logical naming conventions (the changed sub-node names are indicated by using red font).

The highway assignment script relating to Figure 14 can be found in Appendix C. Since the script has over 2,000 lines of code, we have created an excerpt of the Highway\_Assignment\_Parallel.s script (about 150 lines), shown in Figure 16, that focuses on the lines that are most relevant to IDP and MDP. Locations where code has been removed are indicated in Figure 16 by a triple ampersand (“&&&”). On line 4 of Figure 16, the statement “distribute intrastep=%useIdp% multistep=%useMdp%” either turns IDP and MDP on or off, based on the value of the two tokens. The code in Figure 16 contains two MDP blocks. Each MDP block begins with the keyword “DistributeMULTISTEP” and ends with the keyword “ENDDistributeMULTISTEP.” The beginning and ending of each of the two MDP blocks has been highlighted in green. IDP does not require a block of statements – it simple uses a single statement begun with the keyword “distributeIntrastep.” Lines containing this keyword have been highlighted in yellow. For example, the first MDP block includes two IDP statements, but the next two IDP statements occur outside of an MDP block (in other words, they are run from the main node, not a sub-node). As shown in Figure 14, since we have two parallel streams of processes (e.g., one for the AM period and one for the PM period), we need to use a “Wait4Files” keyword, which ensures that the main line of processing stops until the MDP branch completes it work. The Wait4Files keywords have been highlighted in blue in Figure 16. So, as indicated in Figure 14, when the AM period processing is finished a file called AM1.script.end is generated. The Wait4Files tells the main line of processing to stop until it detects that the AM1.script.end file has been generated.

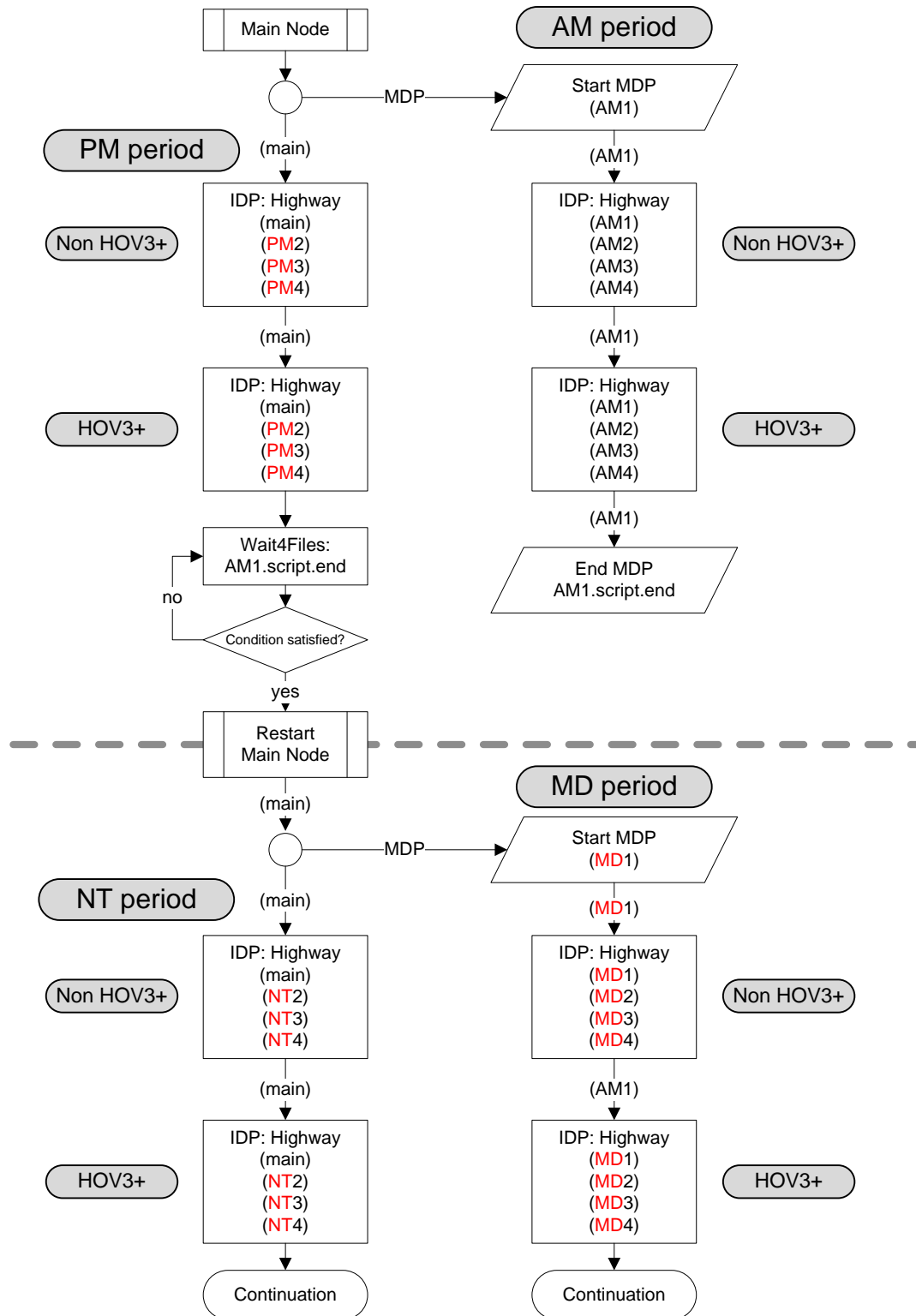
Figure 14 Schematic of IDP and MDP in the highway assignment process of the Ver. 2.3 Travel Model (Highway\_Assignment\_Parallel.s): Existing naming convention for nodes



Ref: ver2.3.52\_hwy\_assign\_mdp\_idp.vsd



Figure 15 Schematic of IDP and MDP in the highway assignment process of the Ver. 2.3 Travel Model (Highway\_Assignment\_Parallel.s): Proposed new naming convention for nodes (changes shown in red)



Ref: ver2.3.52\_hwy\_assign\_mdp\_idp.vsd

Figure 16 Excerpts from the Highway\_Assignment\_Parallel.s script (triple ampersand => code removed)

```

1  &&&
2  /* **** Set up tokens in Voyager Pilot step **** */
3  ; useIdp = t (true) or f (false); this is set in the wrapper batch file
4  distribute intrastep=%useIdp% multistep=%useMdp%
5  &&&
6  ;;*****
7  ;;; Step 1: Execute peak-period traffic assignments (AM & PM)
8  ;;;      AM nonHOV, HOV and PM nonHOV and HOV Assignemnts
9  ;;;*****
10
11  itr = '%_iter_%'  ;;
12  &&&
13  INPNET = 'ZONEHWY.NET'
14
15  DistributeMULTISTEP ProcessID='AM', ProcessNum=1
16
17  PRD      = 'AM'      ;
18  PCTADT   = 41.7      ; %_AMPF_% AM PHF (% of traffic in pk hr of period)
19  CAPFAC=1/(PCTADT/100) ; Capacity Factor = 1/(PCTADT/100)
20  &&&
21  in_capSpd = '..\support\hwy_assign_capSpeedLookup.s'      ;; FT x AT Speed & Capacity lookup
22  VDF_File  = '..\support\hwy_assign_Conical_VDF.s'        ;; Volume Delay Functions file
23
24  ;;*****
25  ;;; Step 1.1: Assign AM NonHOV3+ trip tables only
26  ;;;      (SOV, HOV2, CV, TRUCK & AIRPORT PASSENGER TRIPS)
27  ;;;*****
28
29  RUN PGM=HIGHWAY ; NonHOV3+ traffic assignment
30  distributeIntrastep processId='AM', ProcessList=%AMsubnode%
31  FILEI NETI      = @INPNET@ ; TP+ Network
32  &&&
33  ENDRUN
34  ;;*****
35  ;;; Step 1.2: Assign AM HOV3+ only
36  ;;;*****
37
38  RUN PGM=HIGHWAY ; HOV3+ traffic assignment
39  distributeIntrastep processId='AM', ProcessList=%AMsubnode%
40  FILEI NETI      = TEMP1_@PRD@.NET ; TP+ Network
41  &&&
42  ENDRUN
43  ENDDistributeMULTISTEP
44
45  PRD      = 'PM'      ;
46  PCTADT   = 29.4      ; %_AMPF_% AM PHF (% of traffic in pk hr of period)
47
48  &&&
49  ;;*****
50  ;;; Step 1.3: Assign PM NonHOV3+ trip tables only
51  ;;;      (SOV, HOV2, CV, TRUCK & AIRPORT PASSENGER TRIPS)
52  ;;;*****
53
54  RUN PGM=HIGHWAY ; NonHOV3+ traffic assignment
55  distributeIntrastep processId='MD', ProcessList=%MDsubnode%
56  FILEI NETI      = @INPNET@ ; TP+ Network
57  &&&
58  ENDRUN
59  ;;*****
60  ;;; Step 1.4: Assign PM HOV3+ only
61  ;;;*****
62
63  RUN PGM=HIGHWAY ; HOV3+ traffic assignment
64  distributeIntrastep processId='MD', ProcessList=%MDsubnode%
65  FILEI NETI      = TEMP1_@PRD@.NET ; TP+ Network
66  &&&

```

```

67 ENDRUN
68
69 Wait4Files Files=AM1.script.end, CheckReturnCode=T, PrintFiles=Merge, DelDistribFiles=T
70
71 ;;*****
72 ;;; Step 2: Execute off-peak-period traffic assignments (midday/MD & night/NT)
73 ;;; All 6 trip tables are assigned together.
74 ;;;*****
75
76 DistributeMULTISTEP ProcessID='AM', ProcessNum=1
77 ; Off-Peak Period
78 PRD = 'MD' ;
79 PCTADT = 17.7 ; %_MDPF_% Midday PHF (% of traffic in pk hr of period)
80 CAPFAC=1/(PCTADT/100) ; Capacity Factor = 1/(PCTADT/100)
81 ; Turnpen = 'inputs\turnpen.pen' ; Turn penalty
82
83 RUN PGM=HIGHWAY ; Off-peak (midday & evening) traffic assignment
84 distributeIntrastep processId='AM', ProcessList=%AMsubnode%
85 FILEI NETI = @INPNET@ ; TP+ Network
86 &&&
87 ENDRUN
88
89 ENDDistributeMULTISTEP
90
91 PRD = 'NT' ;
92 PCTADT = 15.0 ; %_NTPF_% NT PHF (% of traffic in pk hr of period)
93 CAPFAC=1/(PCTADT/100) ; Capacity Factor = 1/(PCTADT/100)
94
95 RUN PGM=HIGHWAY ; Off-peak (midday & evening) traffic assignment
96 distributeIntrastep processId='MD', ProcessList=%MDsubnode%
97 FILEI NETI = @INPNET@ ; TP+ Network
98 &&&
99 ENDRUN
100
101 Wait4Files Files=AM1.script.end, CheckReturnCode=T, PrintFiles=Merge, DelDistribFiles=T
102
103 ; END OF MIDDAY and OFF PEAK ASSIGNMENT
104
105 ;;*****
106 ;;; Step 3: Calculate restrained final Volumes, speeds, V/Cs (No MSA)
107 ;;;*****
108 ;;; Step 3.1: Loop thru 1 (AM) and 2 (PM)
109 ;;;*****
110
111 LOOP PERIOD = 1,2 ; Loop thru 1 (AM) and 2 (PM); Each pk per. includes NonHOV3+ and HOV3+
112
113 IF (PERIOD==1)
114     PRD = 'AM' ;
115     PCTADT = 41.7 ;
116 ELSE
117     PRD = 'PM' ;
118     PCTADT = 29.4 ;
119 ENDIF
120 CAPFAC=1/(PCTADT/100) ; Capacity Factor = 1/(PCTADT/100)
121
122 RUN PGM=HWYNET ; Calculate restrained speed/perform MSA volume averaging
123 &&&
124 ENDRUN
125 ENDLOOP ; Loop thru 1 (AM) and 2 (PM); Each pk per. includes NonHOV3+ and HOV3+
126
127 ;;*****
128 ;;; Step 3.2: Loop thru 3 (MD) and 4 (OP)
129 ;;;*****
130
131 LOOP PERIOD = 3,4 ; Loop thru 1 (midday, MD) and 2 (evening/off-peak, OP)
132 IF (PERIOD==3)
133     PRD = 'MD' ;
134     PCTADT = 17.7 ;

```

```

135 ELSE
136     PRD = 'NT' ;
137     PCTADT = 15.0
138 ENDIF
139 CAPFAC=1/(PCTADT/100) ; Capacity Factor = 1/(PCTADT/100)
140
141 RUN PGM=HWYNET ; Calculate restrained speed/perform MSA volume averaging
142 &&&
143 ENDRUN
144 ENDLOOP ; Loop thru 1 (midday, MD) and 2 (evening/off-peak, OP)
145
146 ;;*****
147 ;; Step 4: Summarize 24-hour VMT of current AM, PM, MD & NT assignments
148 ;;*****
149
150 RUN PGM=HWYNET ; Summarize 24-hour VMT of current AM, PM, MD & OP assignments
151 &&&
152 ENDRUN

```

Ref: Highway\_Assignment\_Parallel\_excerpt2.s

To conclude the discussion of Figure 16, we note here some cases where process sub-nodes have somewhat misleading names. The four periods being processed are AM, PM, MD, and NT, and these are indicated in Figure 16 by pink/purple highlighting. For the AM assignment, there are no issues with misleading names for the sub-nodes. For example, we can see that on line 17, the AM processing starts, and the four IDP sub-nodes for the non-HOV3+ assignment are named AM1, AM2, AM3, and AM4 (line 30 of Figure 16), since %AMsubnode% equals "1-4". After the non-HOV3+ assignment is complete, then HOV3+ assignment occurs, and the four IDP sub-nodes for the HOV assignment are also named AM1, AM2, AM3, and AM4 (line 39 of Figure 16). The mislabeled naming of sub-nodes begins with the PM period assignment, which begins on line 45. For example, for the PM non-HOV3+ assignment, the sub-nodes are named MD2, MD3, and MD4 (line 55 of Figure 16), since %MDsubnode% equals "2-4". It would be less confusing if these sub-nodes had been named PM2, PM3, and PM4. The same misleading naming convention is used for the sub-node names in the PM HOV3+ assignment: MD2, MD3, and MD4 -- instead of PM2, PM3, and PM4 (line 64 of Figure 16). A similar issue occurs for the midday assignment (beginning on line 78 of Figure 16) and the nighttime assignment (beginning on line 91 of Figure 16). **At any rate, it should be noted that the code works, despite the misnamed sub-nodes.** In a future version of the model, it is possible that we will clean up the sub-node naming.

## 9 Debugging cases where the model run stops prematurely or crashes

If a model run stops prematurely or crashes, one can use the “full output” text file to determine:

- The speed feedback iteration (e.g., pump prime, iteration 1, ..., iteration 4) that was underway when the model stopped
- The particular modeling step, within a given speed feedback iteration, that was underway when the model stopped (e.g., network building, trip distribution, mode choice, traffic assignment).
- Possible error messages, returned by any programs that crash.

An excerpt from one of the “full output” text files can be seen in Figure 17. Additionally, when debugging a model run crash, one should find the latest print file (\*.prn) to see any relevant error or warning messages. One can search this file using regular expressions to find any warnings or errors.<sup>67</sup>

In some cases, it is sufficient to review the “full output” text file and the latest print file to determine why a model run stops. As an additional tool, however, one can also scan the “search for errors” text file (e.g. 2010\_Final\_searchForErrs.txt), which is created by the searchForErrs.bat batch file. An example of the “search for errors” text file can be found in Figure 18.

One of the most common causes for a model run crash is a sharing violation, which typically occurs when one launches two or more concurrent model runs in the same root directory at about the same time. **One way to protect against this happening is to ensure there is a time delay (ca. 1 hour) between the start of two model runs that share the same root directory.** Additionally, there is now a second reason to offset model runs by about an hour: As described in Chapter 11 (“Building transit walksheds and calculating zonal walk percent”), with the new process for generating transit walksheds and calculating the percent of each zone within walking distance to transit, it is imperative to use a 45- to 60-minute gap in the start times of two or more model runs on the same computer.

---

<sup>67</sup> For example, using the text editor PSPad, one can use this regular expression (regex) to find warnings or errors: `F\([0-9]*\);|W\([0-9]*\):`

Figure 17 An excerpt from an example of the “full output” text file that is created during a model run

```

1 E:\Data\fy17\CGV2_3_66_Conformity2016CLRP_Xmittal>set _year_=2016
2
3 E:\Data\fy17\CGV2_3_66_Conformity2016CLRP_Xmittal>set _alt_=Ver2.3.66_2016_base
4
5 E:\Data\fy17\CGV2_3_66_Conformity2016CLRP_Xmittal>set _maxUeIter_=1000
6
7 E:\Data\fy17\CGV2_3_66_Conformity2016CLRP_Xmittal>set _tcpath_=
8
9 E:\Data\fy17\CGV2_3_66_Conformity2016CLRP_Xmittal>set _HOV3PATH_=
10
11 E:\Data\fy17\CGV2_3_66_Conformity2016CLRP_Xmittal>rem ===== Pump Prime Iteration
12 =====
13
14 E:\Data\fy17\CGV2_3_66_Conformity2016CLRP_Xmittal>set _iter_=pp
15
16 E:\Data\fy17\CGV2_3_66_Conformity2016CLRP_Xmittal>set _prev_=pp
17
18 E:\Data\fy17\CGV2_3_66_Conformity2016CLRP_Xmittal>set _relGap_=0.01
19
20 E:\Data\fy17\CGV2_3_66_Conformity2016CLRP_Xmittal>call ArcPy_Walkshed_Process.bat 2016_base
21     Searching for Python in Path C:\Python27\ArcGIS10.3
22     Found Python in Path C:\Python27\ArcGIS10.3
23
24
25     Using Python from Directory = C:\Python27\ArcGIS10.3
26
27
28
29 1) Creating Subdirectories ...
30
31
32 2) Preparing Inputs ...
33
34     using TRNBUILD line files
35
36 3) Launching ArcPy-based Walkshed Process ...

```

Ref: M:\fy17\CGV2\_3\_66\_Conformity2016CLRP\_Xmittal\2016\_Base\2016\_base\_fulloutput.txt

Figure 18 An excerpt from the “search for errors” file that is created during a model run

```

1 ***** Searching for errors and anomalies after a travel model run *****
2 Program name: searchForErrs.bat
3
4 ***** Searching *fulloutput.txt
5
6     *** Searching for cases where a file could not be found
7
8
9 ***** Searching report files (*.rpt)
10     *** Searching for evidence that TP+ (TPMAIN) is running, instead of Voyager (PILOT)
11     *** Searching for evidence of LINKO nodes that do not have XY values
12 2016_base\i1_TRANSIT_SKIMS_AB.RPT:W(693): The following LINKO nodes do not have XY values:
13 2016_base\i1_TRANSIT_SKIMS_AB.RPT:W(693): The following LINKO nodes do not have XY values:
14 2016_base\i1_TRANSIT_SKIMS_AB.RPT:W(693): The following LINKO nodes do not have XY values:
15 2016_base\i1_TRANSIT_SKIMS_AB.RPT:W(693): The following LINKO nodes do not have XY values:
16 2016_base\i1_TRANSIT_SKIMS_AB.RPT:W(693): The following LINKO nodes do not have XY values:
17 2016_base\i1_TRANSIT_SKIMS_AB.RPT:W(693): The following LINKO nodes do not have XY values:

```

Ref: M:\fy17\CGV2\_3\_66\_Conformity2016CLRP\_Xmittal\2016\_Base\2016\_base\_searchForErrs.txt

## 10 Known issues related to running the model

### 10.1 Cube Cluster differences

When using Cube Cluster, the estimated VMT coming from the model can change slightly, depending on how many cores/nodes are used. See section 8.2.2 (“Effect of Cube Cluster on modeled results”) on page 82 for more details.

### 10.2 Model run stops before finishing

We have experienced some cases where a model run will prematurely stop (“crash”) for no apparent reason. Sometimes the exact same model run will complete successfully if run on a different computer. While we are still trying to determine the cause of these crashes, we do, however, have a pragmatic way for dealing with these events. Determine where the model run crashed. Re-launch the model run, but comment out all the steps in the “run model steps” that have completed successfully, so that the model runs only the step that crashed and the steps that follow it. This procedure will typically result in a normal model run, even though it requires the analyst to intervene midstream. Please see Chapter 9 (“Debugging cases where the model run stops prematurely or crashes”) on page 95.

### 10.3 Issues with traffic assignment convergence

In the past, we have identified some cases where the gap (but not relative gap) for a given user equilibrium iteration in traffic assignment is equal to exactly zero, as opposed to a small, but non-zero value. We have reported this issue to Citilabs, which began an investigation into the matter. However, since the Version 2.3.66 model (like its predecessors, e.g., 2.3.57 & 2.3.57a) uses the *relative* gap and the number of user equilibrium iterations as stopping criteria, this issue should not affect the running of the model. Nonetheless, a model user could experience convergence issues if they change the model to use a really tight stopping criterion. For example, in one test conducted by TPB staff, a stopping criterion of  $10^{-6}$  was used, but the traffic assignment continued on, going past  $10^{-7}$ , even though  $10^{-6}$  was specified. This happened for a for a future-year scenario that had variably-priced facilities. At this point, we do not have any definitive answers, but we contacted Citilabs, whose staff thought that the difficulty reaching convergence was due to large toll values that dominate the link-cost function. Again, this should not be an issue for standard runs of the travel model, which use progressively tightening relative gap tolerances of  $10^{-2}$ ,  $10^{-3}$ , and  $10^{-4}$ .

### 10.4 Running multiple concurrent model runs one computer/server

A user may wish to run two or more travel model runs on one computer or server at the same time. To compute the maximum number of concurrent model runs that may be run on a given computer, divide the number of cores (real or virtual, whichever is greater) by the number of cores needed per model run (currently 8, in the traffic assignment step). For example, on a computer like COG’s travel model server #6 (tms6), which has 16 physical cores or 32 virtual cores due to Intel’s Hyper-Threading Technology, the calculation would be:

$(32 \text{ virtual cores}) / (8 \text{ cores needed per model run}) = 4 \text{ concurrent model runs (maximum)}$

However, based on our experience at COG, the actual number of concurrent model runs that you can run on a given computer may be less than the maximum number, depending on factors such as the following:

- The number of users launching the model runs. So, for example, four concurrent runs by one user may not be the same as two runs by two users, even though both cases result in four concurrent runs.
- The version of Cube used.
- Whether one runs the automated ArcPy walkshed process. This is turned on by default, but it can also be commented out by the user if the areawalk.txt file is already been developed and exists in the "inputs" folder.
- Whether one introduces a time delay (lag time) between model runs. For example, **two model runs can be launched at exactly the same time**, or the **modeler can choose to offset the two launch times by a certain amount of time**. Thus, "**concurrent**" can mean that all of the runs were started at exactly the same time or that there was some offset between the start times of the model runs.

In late October 2014, COG/TPB staff conducted a series of stress tests to determine the maximum number of model runs that can be run concurrently on a given server.<sup>68</sup> Below are some of the findings from those tests, which were conducted on a travel model server ("tms6") with 16 physical cores and 32 virtual cores (threads), due to Hyper-Threading:

- One user is able to launch four model runs concurrently (either at the same time if the models have the walkshed process disabled or using the 45-minute offset between model launch times with the walkshed process enabled).
- When two or more users make concurrent model runs, it is very likely that one or more runs will crash. Unfortunately, this seems to be true even for the case where there are two users each launching only one run (resulting in two concurrent model runs). The point at which the run crashes and the amount of time that the model runs both seem to be somewhat random, but the crash is often accompanied by the message "Voyage has stopped working" notification. This typically occurs at one of the early SFB iterations, such as the "pump prime", i1, or i2 SFB iteration.
- It is still not clear what causes these crashes. To investigate this more, COG/TPB may need to further test some other scenarios, such as varying the version of Cube Voyager, the travel model version, the number of users, and the number of concurrent runs. However, due to other competing needs for travel model server tms6, such as air quality conformity work, those tests have not yet occurred.

---

<sup>68</sup> Moran and Ngo to Milone et al., "Stress Tests of Travel Model Server #6 (tms6) to Determine the Maximum Number of Model Runs That Can Run Concurrently."



Finally, as noted in Section 3.3, based on recent communications with Citilabs (personal communication, 2/6/17), it is better not to overload the processor, so, although a 32-core computer should be able to run 4 concurrent model runs ( $4 \times 8 = 32$ ), it would be better to limit this computer to 3 concurrent model runs. It is hoped that further information about this issue will be added to future Cube documentation.



## 11 Building transit walksheds and calculating zonal walk percentages

### 11.1 Overview

One of the inputs to the travel demand model is the percentage of each zone that is within walking distance to transit. Conceptually, one develops a series of transit walksheds, which are then combined geographically with zone boundaries to calculate the percentage of each zone that is within walking distance to transit. This procedure creates point buffers around transit stop nodes and then overlays these point buffers with TAZ boundaries. The process is made more complicated by the fact that two walking distances are differentiated: a short walk (0.5 miles) and a long walk (1.0 miles). See Section 21.4.3 (“Market segmentation by access to transit”) beginning on p. 171 for more details.

The model assumes that the area of each TAZ that is within a short-walk or a long-walk to transit is stored in a text file (areawalk.txt). This file is used by the walk access script (walkacc.s) to calculate the zonal walk *percentages*, which are then stored in a second text file (NLWalkPCT.txt).<sup>69</sup> This second file is then an input to both the transit fare process (prefarv23.s) and the mode choice process.

### 11.2 Application Details

The Ver. 2.3.57 travel model introduced an automated/integrated transit walkshed process, which was developed by AECOM (the Ver. 2.3.66 model continues to use this feature). The process is automated in the sense that it is run using a Python/ArcPy script, so it does not require manual intervention from the user. The process is integrated in the sense that it is built into the travel model run: It is now the first step in the “run model steps” batch file (“call ArcPy\_Walkshed\_Process.bat %1”). The new process was developed by AECOM in FY 2014, and is discussed both in AECOM’s FY 2014 report.<sup>70</sup> Since the process is integrated in the model run stream, the modeler does not have to remember to run it if network changes are made. As noted by AECOM, “ArcPy was chosen as the basis for development because it provides convenient and powerful access to the GIS functionalities in a (Python) programming environment that is transparent and relatively easy to modify.”<sup>71</sup>

Another advantage of the new process is that it does not require one to have/purchase ArcGIS. One needs only to have purchased Cube, which is already a requirement to run the model. To run the new integrated walkshed process, one must have the following:

- One of the following two software packages:
  - ArcGIS, version 10.1: Available for purchase from Esri. Some modelers may already have this software; some may not.
  - ArcGIS engine runtime, version 10.3: Available for free, if you have purchased Cube. All modelers will have purchased Cube (in order to run the model). Cube version 6.4.1

---

<sup>69</sup> See the modeling flowchart in Appendix A.

<sup>70</sup> AECOM, “FY 2014 Final Report, COG Contract 12-006: Assistance with Development and Application of the National Capital Region Transportation Planning Board Travel Demand Model,” chap. 3.

<sup>71</sup> *Ibid.*, 3–2.

comes with ArcGIS engine runtime 10.3.4959. If you do not have a full installation of ArcGIS 10.1, you will want to install Cube, including the ArcGIS engine runtime.

- Python: This is free, open-source software. One way to get it automatically is to install Cube “with ArcGIS engine runtime.” It may also be included when one installs the full version of ArcGIS.

Notes:

1. As of Ver. 2.3.66, the areawalk.txt file created in the new automated walkshed process is sorted by TAZ.
2. In testing, if two model runs that incorporate the new transit walkshed process were started at the same time, one of the two runs will stop prematurely in the walkshed process. This is likely due to a license restriction with the ArcGIS runtime engine. Consequently, it is recommended that multiple model runs on the same computer not be launched at exactly the same time. Instead, **it is recommended that the start times be staggered/offset by 45 to 60 minutes.**<sup>72</sup> Based on a series of recent “stress tests” to see how many concurrent model runs could be completed on one server, it was also found that the 45-60-minute time offset is useful for minimizing the chance of a model run crash (irrespective of whether the new walkshed process is run).

The new walkshed process appears to give identical results to the previous walkshed process, providing the inputs, such as the transit line files, are identical and correctly coded.<sup>73</sup>

The automated ArcPy walkshed process is called by a line in the “run model steps” batch file:

```
call ArcPy_Walkshed_Process.bat %1
```

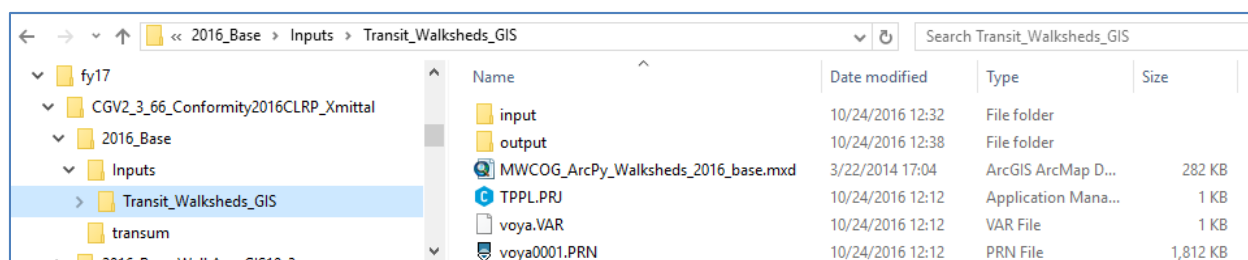
This is done at the start of the pump-prime (PP) speed feedback loop. Once the walkshed process has been run, there a new folder will be created in the “inputs” folder, named “Transit\_Walkshed\_GIS,” as shown in Figure 19. Within this folder, one can find an ArcGIS map document file (MWCOG\_ArcPy\_Walksheds\_\*.mxd) which can be used to visualize the walkshed buffers, as shown in Figure 20.

---

<sup>72</sup> Ibid., 3–8.

<sup>73</sup> See page 2-3 of Mark S. Moran and Dzung Ngo to David Roden and Krishna Patnam, “Comments on Your Cube/ArcPy-Based Transit Walkshed Process and Its Associated Memo Dated March 25, 2013,” Memorandum, (May 15, 2014).

Figure 19 Folder structure for the automated ArcPy walkshed process



Note that the Transit\_Walkshed\_GIS folder includes two subfolders, “input” and “output”, which should not be confused with the “inputs” folder that is stored within the scenario-specific folder (which, in this case is called 2016\_Base). The ArcPy walkshed process creates two files: areawalk.txt and PercentWalk.txt, but only the first file is used by the travel model. If a copy of areawalk.txt already exists in the “inputs” folder, the old copy will be renamed as AreaWalk\_Old.txt, before the new areawalk.txt file is created.<sup>74</sup>

### 11.3 Known issues

Although the new automated transit walkshed generation process has been a benefit to most users of the regional travel model, it has also been the source of many technical assistance calls to the COG/TPB staff when the process crashes for one reason or another. To minimize the likelihood of the automated ArcPy transit walkshed process from crashing, we recommend you use versions of Cube and ArcGIS that are compatible. Table 18 shows which versions of Cube Base are **compatible** with ArcGIS, **in terms of the ability to run the automated transit walkshed process**. We have tested four different versions of ArcGIS with Cube Base ver. 6.4.1. Two of these worked (ArcGIS Engine Runtime 10.3 and ArcGIS 10.1) and two of these did not (ArcGIS 10.3.1 and ArcGIS 10.4.1). The table also shows one other case, tested by an external user, where the two software packages were incompatible. Note that ArcGIS Engine Runtime comes with Cube. The full, stand-alone versions of ArcGIS do not come with Cube, so they must be purchased separately.

Note: According to recent Citilabs documentation covering Cube 6.4.2 and 6.4.1, “Cube 6.4.1 includes support for ArcGIS versions 9.3 to 10.3.1.”<sup>75</sup> However, in our test #3, we found that the “support” was not such that it would allow the automated ArcPy transit walkshed process to run to completion, which is why we have noted that Cube Base 6.4.1 and ArcGIS 10.3.1 are not compatible for running the ArcPy automated walkshed process.

<sup>74</sup> AECOM, “FY 2014 Final Report, COG Contract 12-006: Assistance with Development and Application of the National Capital Region Transportation Planning Board Travel Demand Model,” 9–2.

<sup>75</sup> Citilabs, Inc., “Cube Base Release Summary, Version 6.4.2” (Tallahassee, Florida: Citilabs, Inc., April 24, 2013), 4, [http://citilabs-website-resources.s3.amazonaws.com/resources/RS\\_CubeBase.pdf](http://citilabs-website-resources.s3.amazonaws.com/resources/RS_CubeBase.pdf).

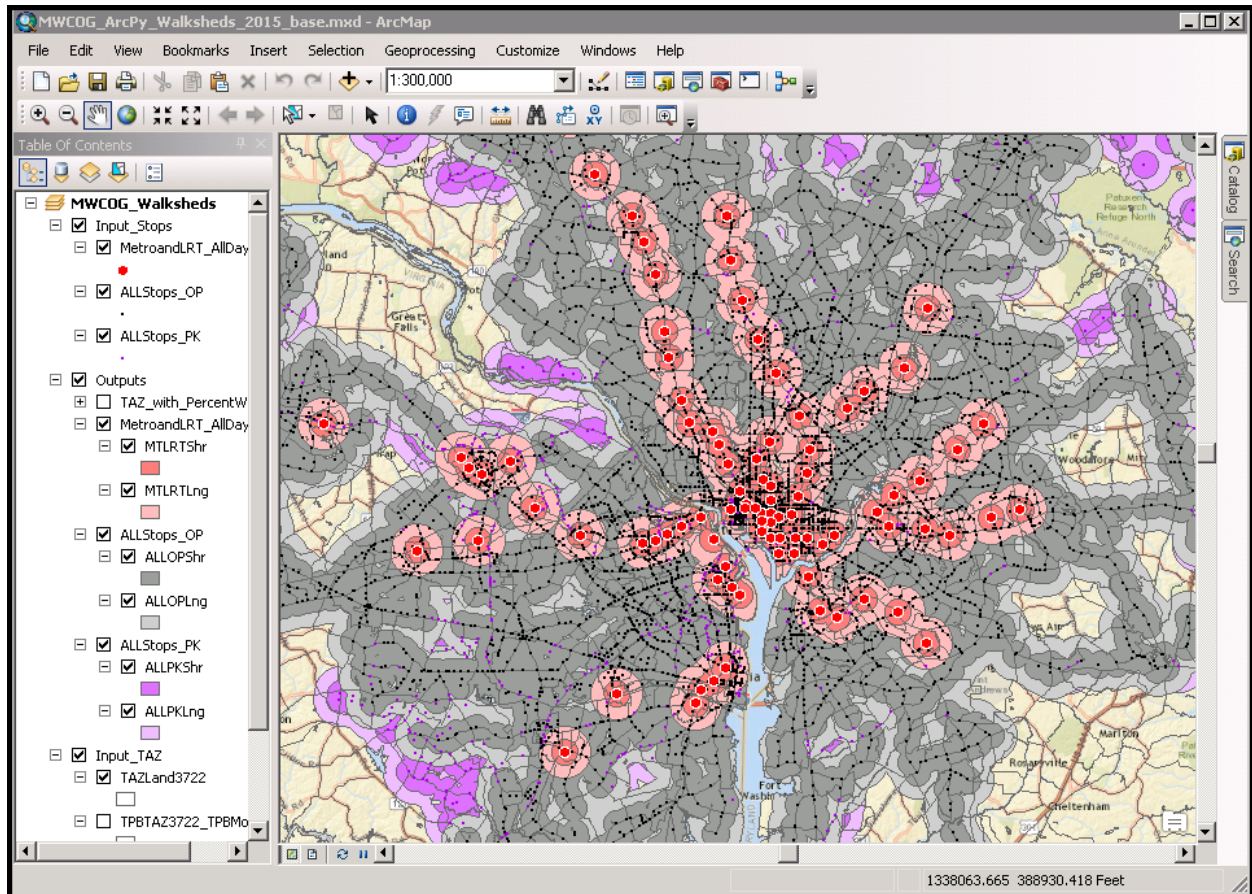
Table 18 Compatibility between Cube Base and ArcGIS, in terms of the ability to run the automated transit walkshed process

Version of Cube Base	Version of ArcGIS	Compatible? *	Test Conducted by	Test No.
6.4.1	10.3.4959 (ArcGIS Engine Runtime**)	Yes	COG/TPB	1
	10.1 (full version)	Yes	COG/TPB	2
	10.3.1 (full version)	No	COG/TPB	3
	10.4.1 (full version)	No	COG/TPB	4
6.4.2	10.4 (ArcGIS Engine Runtime**)	No	No. Va. Transportation Commission	5

\* "Compatible" means that the tester was able to run the automated ArcPy transit walkshed process using the noted version of Cube and ArcGIS.

\*\* ArcGIS Engine Runtime comes with Cube.

Figure 20 Walkshed buffers for a typical base-year scenario



## 12 Set-Up Programs and Highway Network Building

### 12.1 Overview

Following the generation of transit walksheds, the initial modeling steps of the Version 2.3 model are executed to establish basic modeling parameters to construct a binary (or “built”) highway network. The steps are executed using two batch files:

- SET\_CPI.BAT: The batch file calls two Cube Voyager scripts, SET\_CPI.S and SET\_FACTORS.S
- PP\_HIGHWAY\_BUILD.BAT: The batch file calls two Cube Voyager scripts, AreaType\_File.S and V2.3\_Highway\_Build.S

The modeling steps included in these two batch files are shown on pages A-2 and A-3 of the flowchart in Appendix A. Set\_CPI.s is used to establish deflation factors that are used in subsequent toll-related and transit fare-related steps. Set\_Factors.s is used to establish K-Factors used in trip distribution and is also used to create the file station\_names.dbf (used for the transit assignment summary process), which is developed using information pulled from station.dbf. The Area\_Type.s step establishes zonal area type codes based on land activity densities (see Table 28). The resulting area type file is subsequently used in the highway building step, V2.3\_Highway\_Build.s. These steps are not implemented within the speed feedback loop of the travel model; they are executed only once, in the “pump prime” stage of the travel model. The principal inputs to above modeling steps are listed in Table 19 and detailed in Table 20 through Table 23. The principal outputs are listed in Table 24, and are detailed in Table 25 and Table 26.

**Table 19 Inputs to the set-up and highway network building process**

File description	File name and location	Format
CPI schedule and parameter file	\\Inputs\CPI_File.txt	Text
Zonal land use file	\\Inputs\ZONE.DBF	DBF
Node coordinate file	\\Inputs\NODE.DBF	DBF
Zonal area type override file	\\Inputs\AT_override.txt	Text
Link file	\\Inputs\LINK.DBF	DBF
Initial AM and midday hwy. speed lookup files	\\Support\AM_SPD_LKP.TXT, \\Support\MD_SPD_LKP.TXT	Text
Toll parameter file	\\Inputs\Toll_Esc.dbf	DBF

**Table 20 Land Use File Format Description (zone.dbf)**

Variable name	Description
TAZ	TAZ (1-3722)
HH	Households
HHPOP	Household population
GQPOP	Group quarters population
TOTPOP	Total population
TOTEMP	Total employment

Variable name	Description
INDEMP	Industrial employment
RETEMP	Retail employment
OFFEMP	Office employment
OTHEMP	Other employment
JURCODE	Jurisdiction Code (0-23) <i>0/dc, 1/mtg, 2/pg, 3/alr/, 4/alx,5, ffx, 6/ldn, 7/ pw, 8/(unused), 9/ frd, 10/how, 11/aa, 12/chs, 13/(unused), 14/car, 15/cal, 16/stm, 17/ kg, 18/fbg, 19/stf, 20/spts, 21/fau, 22/clk, 23/jef</i>
LANDAREA	Gross land area (square miles)
HHINCIDX	Ratio of zonal HH median income to regional median HH income in tenths (i.e., 10 = 0.1) per the 2007 ACS
ADISTTOX	Airline distance to the nearest external station (whole miles)
TAZXCRD	TAZ X-coordinate (NAD83, whole feet)
TAZYCRD	TAZ Y-coordinate (NAD83, whole feet)

Table 21 Node Coordinate File Format Description (node.dbf)

Variable name	Description
N	Highway node number
X	X-coordinate (NAD83, whole feet)
Y	Y-coordinate (NAD83, whole feet)

Table 22 Base Highway Link File Format Description (link.dbf)

File Name	Variable Name	Description
Link.dbf	A	A-Node
	B	B_Node
	DISTANCE	Link distance (in 1/100 <sup>th</sup> s of miles)
	JUR	Jurisdiction Code (0-23) <i>0/dc, 1/mtg, 2/pg, 3/alr/, 4/alx,5, ffx, 6/ldn, 7/ pw, 8/(unused), 9/ frd, 10/how, 11/aa, 12/chs, 13/(unused), 14/car, 15/cal, 16/stm, 17/ kg, 18/fbg, 19/stf, 20/spts, 21/fau, 22/clk, 23/jef</i>
	SCREEN	Screenline Code
	FTYPE	Link Facility Type Code (0-6) <i>0/centroids, 1/Freeways, 2/Major Art., 3/Minor Art, 4/ Collector, 5/ Expressway, 6/ Ramp</i>
	TOLL	Toll Value in current year dollars
	TOLLGRP	Toll Group Code
	AMLANE	AM Peak No. of Lanes



File Name	Variable Name	Description
	AMLIMIT	AM Peak Limit Code (0-9)
	PMLANE	PM Peak No. of Lanes
	PMLIMIT	PM Peak Limit Code (0-9)
	OPLANE	Off-Peak No. of Lanes
	OPLIMIT	Off-Peak Limit Code (0-9)
	EDGEID	Geometry network link identifier
	LINKID	Logical network link identifier
	NETWORKYEARLink	Planning year of network link
	SHAPE_LENGLink	Geometry length of network link (in feet)
	PROJECTID	Project identifier
	CODE	Unused (place marker to flag network edits)

Notes:

- The mode choice model requires that all costs be in 2007 dollars.
- Limit Codes are 0,1 = General Use, 2 = HOV2,3+ only, 3 = HOV 3+ Only, 4 = Truck Prohibited, 5 = Non-Airport Vehicles Prohibited, 6-8 = (unused), 9 = "Transit Only" link (links used to depict coded transit routes more accurately, but are below the grain of the zone system; these links are not included in the highway assignment process).

Table 23 Toll Parameter File (Toll\_esc.dbf)

File Name	Variable Name	Description
Toll_Esc.dbf	Tollgrp	Toll group code 1 = Flat toll (pertains to most existing tolled facilities); 2 = Toll that varies by time of day (e.g. ICC), 3+= Tolls that change dynamically based on congestion level (e.g., VA HOT lanes/Express Lanes)
	Escfac	Deflation factor override. Can be used to group various toll policies.
	Dstfac	Distance (per mile) based toll factor in present year cents/dollar (optional)
	AM_Tftr	AM period Toll factor
	PM_Tftr	PM period toll factor
	OP_Tftr	Off-peak period toll factor
	AT_MIn	Area Type minimum override (optional)
	AT_Max	Area Type maximum override (optional)
	TollType	<i>Toll Type (1=operating in calibration year, 2= operating after calibration year)</i>

Table 24 Outputs of the set-up and highway network building process

Highway, transit deflator files	Trn_Deflator.txt, Hwy_Deflator.txt	Text
Summary text file of Fare CPI assumptions used	MFARE2_CPI.txt	Text
Zone centroid co-ordinates	TAZ_XYs.dbf	DBF
1-mile floating land use	Floating_LU.dbf	DBF

Area type file	AreaType_File.dbf	DBF
Unloaded/built highway network file	ZONEHWY.NET	Binary
Summary text file of Fare CPI assumptions used	MFARE2_CPI.txt	Text
Zonal K-factors	HBW_K.MAT, HBS_K.MAT, HBO_K.MAT, NHW_K.MAT, NHO_K.MAT	Binary

Table 25 Zonal Area Type File (AreaType\_File.dbf)

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

The one-mile floating density is calculated by using the TAZ centroids and a one-mile point buffer around these centroids.

Table 26 Unloaded binary highway network file (Zonehwy.net)

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

## 12.2 Application Details

The SET\_CPI.S script is used to produce deflation factor files (Trn\_Deflator.txt and Hwy\_Deflator.txt) which are used to convert present-year costs to constant-year (2007) costs. The deflation parameter files are inputs to the V2.3\_Highway\_Build.s and MFARE2.S scripts. This procedure has been established to ensure that cost deflation for highway tolls and transit fares are treated consistently.

The SET\_CPI.S script reads a preexisting look-up table (\INPUTS\CPI\_File.txt) containing historical annualized CPI figures published by the U.S. Bureau of Labor Statistics beginning with the model calibration year, 2007. The CPI figures are based on the U.S. city average of all urban consumers (100 = 1982-84). An example listing of the file appears in Figure 21.

Figure 21 Consumer price index file (CPI\_File.txt)

```

1  ;; - MWC0G V2.3 Travel Model - Cost deflation Table
2  ;; - 5/15/2016 - RJM
3  ;; Data from BLS / All Urban Consumers (CPI-U) US City Avg.1982-84=100.0
4  ;; http://www.usinflationcalculator.com/inflation/consumer-price-index-and-annual-percent-changes-from-1913-to-2008
5
6  InflationFTR      = 1.0    ; Inflation Assumption (DEFAULT IS 1.0)
7  Defl_Override    = 0.0    ; Deflation Override (DEFAULT IS 0.0) If Non-zero it is used as deflator
8                    ; Used as deflator IF NON-ZERO
9  BaseCPIYear      = 2007   ; Base year of the CPI Table
10 CurrCPIYear      = 2015   ; Current year on CPI table below (Year for which complete annual CPI data is available)
11 ;
12 ;=====
13 ; Establish historic CPI table and Deflation Factor =
14 ;=====
15 ;
16 LOOKUP Name=CPI_Table,
17     LOOKUP[1] = 1,Result = 2,          ;; CPI index (from US BLS)
18     LOOKUP[2] = 1,Result = 3,          ;; Compounded Growth Rate From Base Year
19     LOOKUP[3] = 1,Result = 4,          ;; Deflation Factor
20     Interpolate = N, FAIL=0,0,0,list=Y,
21     ;;
22     ;;          (((YrCPI/BsCPI)^(1/n))-1.0)*100   BsCPI/YrCPI)
23     ;;          Annual_Avg.                       Historic Deflation
24     ;; YEAR   CPI          Growth_Rate(%)         Factor
25     ;; ----   -
26 R=' 2007, 207.342,      0.00,      1.0000 ', ; <--- BaseCPIYear
27   ' 2008, 215.303,      3.84,      0.9630 ', ;
28   ' 2009, 214.537,      1.72,      0.9665 ', ;
29   ' 2010, 218.056,      1.69,      0.9509 ', ;
30   ' 2011, 224.939,      2.06,      0.9218 ', ;
31   ' 2012, 229.594,      2.06,      0.9031 ', ;
32   ' 2013, 232.957,      1.96,      0.8900 ', ;
33   ' 2014, 236.736,      1.91,      0.8758 ', ;
34   ' 2015, 237.017,      1.69,      0.8748 ', ; <--- Curr(ent)CPI Year
35 ; --- end of CPI File -----
36 ; -----

```

Ref: M:\fy17\CGV2\_3\_66\_Conformity2016CLRP\_Xmittal\2016\_Base\Inputs\CPI\_File.txt

The script computes a cost deflation factor using the CPI table and the parameters *BaseCPIYear*, *CurrCPIYear*, *InflationFTR* (all specified in the above text file), and the ***Year*** environment variable specified in the “Run\_ModelSteps” batch file. These parameters are defined as:

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

- **CurrCPIYear** = the most recent year for which historical CPI data exists (as reflected in the CPI table)
- **\_Year\_** = the year of the modeled scenario (as defined in the Run\_ModelSteps.bat file)
- **InflationFTR** = Factor reflecting special CPI growth assumptions beyond CurrCPIYear that might be considered in scenario testing. For example, a value of 1.0 indicates future cost escalation is assumed to remain constant with the historical rate of inflation; a value of 2.0 would indicate that the future cost escalation is assumed to be twice the historical rate of inflation; a value of 0.5 would indicate that the future cost escalation is assumed to be one half of the historical rate of inflation, etc. The default value is 1.0.

Under default conditions, if the modeled year (*\_Year\_*) is less than or equal to *CurrCPIYear*, the CPI factor will equal  $CPI_{2007} / CPI_{\_Year\_}$  from values provided in the CPI table. If the modeled year (*\_Year\_*) is greater than *CurrCPIYear*, the CPI factor will equal  $(CPI_{2007} / CPI_{CurrCPIYear})$  from values provided in the CPI table. The user may optionally invoke the *InflationFTR* parameter to arrive at a deflation factor that reflects something other than the “historical inflation rate” assumption. In addition to the output deflation factor files mentioned above, the script also writes a text file (Mfare2\_CPI.txt) that lists the input and output parameter values used.

The Set\_Factors.s script is used to generate a family of K-factors by modeled trip purpose, to be used subsequently by the trip distribution process. The K-factors are jurisdiction-based and have been formulated during the calibration and validation phase of the model development process. Separate K-factor files are produced by purpose as Cube/Voyager binary matrix files (zone-to-zone). These files are currently stored in the Support folder, though there has been some thought of moving these into the inputs folder to prevent sharing violations when multiple model runs are launched concurrently. As of the Ver. 2.3.57a model, Set\_Factors.s is also used to generate the station names file (station\_names.dbf), which is stored in the INPUTS folder, used in the transit assignment summary process, and is derived from information found in the station file (station.dbf, see Table 35).

It is useful to understand the basic elements of the highway and transit networks that are reflected in the highway link input file (link.dbf) to the Version 2.3 model. The highway elements are shown in Table 27.

Table 27 Elements of the highway network

Highway Network Element	What It Represents	Node No. Ranges	Notes
Zone centroid	Center of activity for the TAZ; Start and end point for trips	1-3722	3676-3722 allocated as external stations. 3723-5000 reserved for TAZ expansion. Established ranges for each jurisdiction. Some TAZs are unused
Station PNR centroid	Location of the station's park-and-ride lot. Used to develop congested highway times between each TAZ and each PNR lot.	5001-7999*	5001-5999 for Metrorail. 6000-6999 for commuter rail. 7000-7999 for LRT, BRT, and streetcar.
Highway node	Highway intersections or junctions, including where zone centroids connect to the highway network	20000-60000	Established ranges for each jurisdiction.

Highway Network Element	What It Represents	Node No. Ranges	Notes
Zone centroid connectors	Connection from zone centroid to the highway network. One zone centroid connector can represent multiple local roads.		Facility type (FTYPE) = 0
PNR lot connectors	Connection from PNR lot to the highway network		Facility type (FTYPE) = 4
Highway links	Road segments		0 = centroid connectors; 1 = freeways; 2 = major arterials; 3 = minor arterials; 4 = collectors; 5 = expressways; 6 = ramps on freeways and arterials; 9 = transit only;

Notes: \* Station PNR centroids (a.k.a. dummy station centroids) are not required for Mode 5 (LRT) or Mode 10 (BRT/streetcar).<sup>76</sup> For the sake of consistency, the current COG coding practice is to refrain from using station PNR centroids for LRT, BRT, and streetcar. In other words, in the station file, the STAC variable is coded with a value of zero.

The network includes two types of centroids: a zonal centroid, which represents the geographic center of land activity within a TAZ, and a park-and-ride (PNR) lot centroid (also known as a “station centroid,” “dummy PNR centroid”), which represents PNR lot locations at Metrorail and commuter rail stations. The PNR centroid represents a kiss-and-ride (KNR) drop-off point if no PNR lot exists at a given station. Within the station file (station.dbf), the PNR centroid/station centroid is denoted with the variable name STAC. Each Metrorail station and commuter rail station should have its own unique STAC. The two centroid types are assigned specific numbering ranges. TAZ centroids are numbered 1-3722 and PNR centroids are numbered 5001-7999. The numbering gap between the TAZ and PNR ranges, 3723-5000, are reserved for future TAZ assignments.<sup>77</sup> The two centroid types are employed so that highway level-of-service (LOS) matrices may be built, not only between TAZs, but also between TAZs and PNR lots.

Highway nodes representing intersections or highway access points from TAZs or PNR lots are assigned a number from the following range: 20000 to 60000. Network links (i.e., centroid connectors and highway links) are assigned facility type (“Ftype”) attributes ranging from 0 to 6.

The highway network building process -- i.e., the process for creating a binary highway network file which is used in subsequent modeling steps -- is undertaken with two scripts that are executed in sequence: AreaType\_File.s and V2.3\_Highway\_Build.s (page A-3). The AreaType\_File.s script, which reads a preexisting zonal land activity file (Zone.dbf) and a highway node coordinate file (Node.dbf), computes the area type code associated with each TAZ. Area type codes range from 1 to 6 and are based on population and employment density, as shown in Table 28.

<sup>76</sup> Jain to Milone and Moran, “MWCOC Network Coding Guide for Nested Logit Model (First Draft: September 20, 2007; Updated February 2008 and October 2010),” 6 and 10.

<sup>77</sup> The existing Version 2.3 scripts, inputs, and support files would need to be modified if additional TAZs were added to the highway network.

Table 28 Area type codes, based on population and employment density

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

The AreaType\_File.s script produces three files which are used as inputs to the v2.3\_highway\_build.s script:

- TAZ\_Xys.dbf (zonal coordinates),
- Floating\_LU.dbf (a zonal file containing the area, population, and employment within one mile),<sup>78</sup>
- Areatype\_file.dbf (a zonal file containing the associated area type, in accordance with the land activity file)

The V2.3\_Highway\_Build.S script reads the zonal area type file, along with a node file, a link attribute file, a zone file, and four parameter files. The parameter files include initial speed and capacity lookup files (AMSpd.lkp, MDSpd.lkp), both arrayed by facility type and area type. The deflation file created by the SET\_CPI.s script (Hwy\_Deflator.txt) is also read into the highway building script. Finally, a toll parameter/escalation file (Toll\_esc.dbf) is also used by the script. The file contains a number of toll-related parameters that are indexed by a tolled facility code (tollgrp) which is included as a link attribute.

The highway building process consists of the following steps:

- 1) Each highway link is evaluated against all TAZ centroids to determine its nearest zone (i.e., the TAZ centroid nearest to the airline mid-point of the link a-node and b-node). The nearest zone is then saved to a temporary link file containing the A-node, B-node, and nearest TAZ.

<sup>78</sup> TAZ-level floating density is calculated by using the TAZ centroids and creating a 1-mile point buffer around each centroid.

- 2) The link file, zonal area type file, and link-TAZ (from step 1) are merged to enable the zonal area type of the nearest TAZ to be assigned to each link. The link file contains basic link attributes, including distance, facility code, time-of-day-period-specific (AM, PM, OP) lanes and limit codes, coded tolls, toll group codes, jurisdiction, and screenline codes.
- 3) Toll parameters are merged to each link on the basis of the tollgrp code.
- 4) Speed and capacity classes are next defined as a two-digit integer, where the first digit is the facility type and the second digit is the area type.
- 5) Period-specific tolls (AM, PM, and OP) are computed. The general form of the toll computation is:

$$\langle \text{prd} \rangle \text{Toll} = (\text{Toll} + (\text{DstFac}_t * \text{Distance} * \langle \text{prd} \rangle \text{TFtr}_t)) * (\text{EscFac}_t \text{ if } > 0.0; \text{ Otherwise: Hdefl})$$

Where:

- $\langle \text{prd} \rangle \text{Toll}$  = period-specific toll coded on link in constant year dollars (e.g., Amtoll)
- Toll = link-coded "Toll" link attribute value
- $\text{DstFac}_t$  = distance factor (cents/mi) for toll group "t", as specified in Toll\_Esc.dbf
- Distance = link-coded distance (miles)
- $\langle \text{prd} \rangle \text{TFtr}_t$  = period-specific factor for toll group "t" as specified in Toll\_Esc.dbf
- Hdefl = Default highway deflation factor based on CPI assumptions (Set\_CPI.s)
- $\text{Esc\_Fac}_t$  = Hwy. deflation factor *override* for toll group "t", as specified in Toll\_Esc.dbf

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

The binary network file resulting from the highway network building process is named Zonehwy.net. Variables that are included in the zonehwy.net file are described in Table 29.

Table 29 zonehwy.net file Variables description

VariableName	Description
A	A-Node
B	B-Node
DISTANCE	Link Distance in miles (x.xx)
SPDC	Not used
CAPC	Not used
JUR	Jurisdiction Code (0-23) <i>0/dc, 1/mtg, 2/pg, 3/alr/, 4/alx,5, ffx, 6/ldn, 7/ pw, 8/(unused), 9/ frd, 10/how, 11/aa, 12/chs, 13/(unused), 14/car, 15/cal, 16/stm, 17/ kg, 18/fbg, 19/stf, 20/spts, 21/fau, 22/clk, 23/jef</i>
SCREEN	Screenline Code (1-38)
FTYPE	Link Facility Type Code (0-6) <i>0/centroids, 1/Freeways, 2/Major Art., 3/Minor Art, 4/ Collector, 5/ Expressway, 6/ Ramp</i>



<b>VariableName</b>	<b>Description</b>
TOLL	Toll Value in current year dollars
TOLLGRP	Toll Group Code (1-9999)
<Period>LANE	<Period> No. of Lanes
<Period>LIMIT	<Period> Limit Code (0-9)
EDGEID	Geometry network link identifier
LINKID	Logical network link identifier
NETWORKYEA	Planning year of network link
SHAPE_LEN	Geometry length of network link (in feet)
PROJECTID	Project identifier
TAZ	TAZ (1-3,722)
ATYPE	Area Type (1-6)
SPDCLASS	Speed Class
CAPCLASS	Capacity Class
DEFLATIONFTR	Factor for deflating current year tolls to constant year tolls
<Period>TOLL	<Period> Toll Value in current year dollars
<Period>TOLL_VP	<Period> Toll Value in current year dollars - Variably priced tolled facilities only
<Period> HTIME	<Period> Highway Time - based on initial highway lookup speeds
<b>KEY</b>	
<Period>= AM	AM Peak Period (6:00 AM - 9:00 AM)
MD	Midday (9:00 AM - 3:00 PM)
PM	PM Peak Period (3:00 PM - 7:00 PM)
NT	All remaining hours ("nighttime")



## 13 Highway Skim File Development

### 13.1 Overview

Highway skimming begins with path building, the process of building minimum-impedance paths from every TAZ to every other TAZ. After paths have been built, the paths can be “skimmed,” i.e., the paths are traversed and key variables are summed over the paths. The variables that are skimmed include travel times, distances, and tolls. The resultant zone-to-zone sums are saved in one or more skim matrices. The input to the skimming process is usually a loaded network with congested travel speeds, generated from a traffic assignment process. Although traffic assignment is conducted for four time-of-day periods (AM peak period, midday, PM peak period, and nighttime), the travel model is set up to use skims for only two time-of-day periods: a peak period (represented by the AM peak period) and an off-peak period (represented by the midday period). Highway skims in the Version 2.3 model are generated after each traffic assignment step.

Highway skims are generated by time period (AM and Midday), and by highway mode (SOV, HOV 2-occupant, HOV 3+occupant). In addition, truck skims are generated for the midday period only. Mode-specific paths are very important in the Washington, D.C. region, due to special operating restrictions, particularly during the AM peak period.

The TPB's highway skimming is done twice: once to develop zone-to-zone (3722 x 3722) skim matrices and then again to develop zone/PNR lot-to-zone/PNR lot (7999 x 7999) skim matrices. The latter set enables restrained highway speeds and distances to be calculated between zones and PNR lots, thus allowing transit auto-access links to be built. The entire highway skimming process is applied with the scripts named HIGHWAY\_SKIMS\_AM.S, HIGHWAY\_SKIMS\_MD.S, MODNET.S, Highway\_Skims\_mod\_am.s, Highway\_Skims\_mod\_md.s,<sup>79</sup> JOINSKIMS.S, and Remove\_PP\_Speed.S. These are invoked with the PP\_Highway\_Skims.bat file in the initial or pump-prime iteration (see page A-4 of Appendix A) and the Highway\_Skims.bat file (see page A-11) in the standard iterations. For analysis years involving HOT lanes, instead of using the Highway\_Skims.bat file, the HSR\_Highway\_Skims.bat (HOV3+ skim replacement) file is used instead (p. A-11). The Remove\_PP\_Speed.s script is executed in the pump-prime iteration only. The principal inputs and outputs are shown in Table 30 and Table 31, respectively.

---

<sup>79</sup> Prior to build 37, there was one script (Highway\_Skims\_mod.s), which had a loop covering the two time periods, AM and midday. However, it was found that this script would crash on some hardware configurations, when running Voyager 5.1.3, resulting in the following errors: 1) Voyager.exe, APPCRASH, TPPDLIBX.DLL; and 2) Voyager, APPCRASH, MSVCR90.DLL. The script was then split into two files (Highway\_Skims\_mod\_am.s and Highway\_Skims\_mod\_md.s), which eliminated this problem.

**Table 30 Inputs to the highway skim file development**

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

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

**Table 31 Outputs of the highway skim file development**

Total highway skims	<ITER>_SKIMTOT.TXT	Text
Truck skims	<ITER>_MD_TRK.SKM	Binary
SOV skims	<ITER>_<Prd>_SOV.SKM	Binary
HOV2 skims	<ITER>_<Prd>_HOV2.SKM	Binary
HOV3+ skims	<ITER>_<Prd>_HOV3.SKM	Binary
SOV skims (used by mode choice model)	<ITER>_<Prd>_SOV_MC.SKM	Binary
HOV2 skims (used by mode choice model)	<ITER>_<Prd>_HOV2_MC.SKM	Binary
HOV3+ skims (used by mode choice model)	<ITER>_<Prd>_HOV3_MC.SKM	Binary
AM highway skims	<ITER>_HWY_AM.SKM	Binary
Off peak highway skims	<ITER>_HWY_OP.SKM	Binary
Network with added station centroid connectors	<ITER>_HWYMOD.NET	Binary
Walk access links	WalkAcc_Links.dbf	DBF
	<ITER>_<Prd>_SOV_MOD.SKM	Binary
	<ITER>_<Prd>_HOV2_MOD.SKM	Binary
	<ITER>_<Prd>_HOV3_MOD.SKM	Binary
Highway network with PP speeds removed	ZoneHWY.NET	Binary

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

## 13.2 Application Details

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

Equation 1 Converting tolls into time-equivalent minutes of impedance

$$(\text{Impedance})_v = (\text{Restrained over-the-network time})_v + (\text{Toll-related time})_v$$

or

$$(\text{Impedance})_v = (\text{Restrained over-the-network time})_v + ([\text{Toll cost}]_v \times [\text{Time rate}]_v \times [\text{Vehicle factor}]_{vf})$$

where

$$(\text{Impedance})_v = \text{Restrained over-the-network time}_v + \text{Toll-related Time}_v$$

$$(\text{Restrained over-the-network time})_v$$

= Congested/restrained network travel time (min) for vehicle class "V"

$$[\text{Toll cost}]_v = \text{Tolls (2007 dollars) paid by vehicle class "V", if a tolled facility was used}$$

$$[\text{Time rate}]_v = \text{Time valuation (min/2007 dollar) of toll costs for vehicle class "V"}$$

$$[\text{Vehicle factor}]_{vf} = \text{Vehicle class factor for tolled facility "F"}$$

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

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

```

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

```

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

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

Information about the “toll setting” process that is used to estimate reasonable toll values can be found in two technical memos.<sup>80</sup>

The standard zone-to-zone highway skims are developed by the scripts Highway\_Skims\_am.s and Highway\_Skims\_md.s. The scripts produce skim files pertaining to two time periods (AM and midday) and to four mode/path types (SOV, HOV2, HOV3+, and truck). The truck skim file contains one table pertaining to travel time. The SOV and HOV skim files contain three tables: time (min), distance in implied tenths of miles, and tolls (year-2007 cents).

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

The *Joinskims.s* script is used to merge the six skim files used by the mode choice model into two files, <iter>\_HWY\_AM.skm and <iter>\_HWY\_MD.skm, which are read directly into the mode choice model.

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

The Highway\_Skims\_Mod\_am.s and Highway\_Skims\_Mod\_md.s scripts read the expanded network and create an expanded set of highway skims dimensioned 7999 by 7999, which includes highway skims between zone centroids as well as between zone and PNR lot pairs. The latter will be used subsequently to create auto access link attributes. As explained in footnote 79, prior to build 37, there was one script

---

<sup>80</sup> Jinchul Park to Team B Modelers, “Processes Related to Toll Setting in Version 2.3 Model (Draft),” Memorandum, (October 12, 2012); Park to Files, “HOT Lane Modeling Process of MWCOG/TPB (Draft).”

(Highway\_Skims\_mod.s), which had a loop covering the two time periods, AM and midday. However, it was found that this script would crash on some hardware configurations, but not on others. The script was then split into two files (Highway\_Skims\_mod\_am.s and Highway\_Skims\_mod\_md.s), which eliminated this problem.

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





## 14 Auto Driver Trip Development

### 14.1 Overview

The “auto drivers” step is used to convert daily auto person trip tables by occupant group (1, 2, and 3+) into auto driver trips by occupant group. This step occurs in between the mode choice model and the time-of-day model steps (see pages A-8 and A-14 of Appendix A). The auto driver step uses daily auto person trips estimated by the mode choice model and computes auto driver trips by occupant groups using matrix division. Because the mode choice file output includes only internal-to-internal movements, total external auto person trips produced in the trip distribution step are also used as a basis for developing external auto driver trips by occupant groups. External auto person trips produced by the trip distribution process are not stratified by occupant groups. The auto driver step uses modeled occupant disaggregation curves to develop external auto drivers by occupant groups.

The scripts used are PP\_AUTO\_DRIVERS.S, invoked by the PP\_AutoDrivers.bat file (see page A-8 of Appendix A) and MC\_AUTO\_DRIVERS.S, invoked by the Auto\_Drivers.bat file (see page A-14 of Appendix A). The inputs to this step are shown in Table 32. The outputs are shown in Table 33. The outputs consist of five purpose-specific auto driver files, each containing three tables (one for each occupant group). The output files contain both internal and external auto driver movements.

Note that the PP\_AUTO\_DRIVERS.S script uses a pre-existing mode choice model output file that resides in the \inputs subdirectory, while the MC\_AUTO\_DRIVERS.S script reads mode choice model output that is generated within the model execution stream. A pre-existing file must be used in the pump prime iteration in order to provide initial zonal mode choice percentages. TPB staff uses a pre-existing file that is as current and as reasonable for the modeled scenario as possible.

Table 32 Inputs to auto driver trip development

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

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

### 14.2 Application Details

Table 33 Outputs of auto driver trip development

Auto drivers trips by trip purpose (t1= 1-occ. auto drivers, t2= 2-occ. auto drivers, t3 =3+ occ. Auto drivers	<ITER>_??_ADR.MAT	Binary
--	-------------------	--------

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



## 15 Pre-Transit Network Processing

### 15.1 Overview

Prior to transit network building (shown on page A-5), a series of Cube Voyager scripts is executed to generate special transit-access links that are subsequently folded into the transit network, along with highway links, transit links, and transit lines. The scripts include PARKER.S (used to generate PNR-lot-to-rail-station links), WALKACC.S (used to develop zonal walk access links), Adjust\_Runtime (to update the RUNTIME values for local bus service to account for worsening congestion), and AUTOACC5.S (used to generate TAZ-to-station links, a.k.a., auto access links). The automated approach for generating these links has greatly streamlined the transit network coding process. Three of these programs (PARKER.S, WALKACC.S, AND AUTOACC5.S) were originally developed as stand-alone Fortran programs developed by AECOM Consult. TPB staff converted these three Fortran programs to Cube Voyager scripts to facilitate the implementation of future enhancements.

The inputs used by the above programs are list in Table 34. Specific file descriptions are shown in Table 35 through Table 38. The output files are shown in Table 39.

**Table 34 Inputs to pre-transit network processing**

Zonal land use file	Zone.dbf	DBF
Station file	Station.dbf	DBF
Highway node file	node.dbf	DBF
Supplemental walk link file	xtrawalk.dbf	DBF
Sidewalk network links	WalkAcc_Links.dbf	DBF
TAZ area that is within walking distance from transit stops	Areawalk.txt**	Text
Factors used to determine the amount of speed degradation, due to congestion, for local bus routes	Bus_Factor_File.dbf	DBF
Station mode-station type-max access dist. Lookup	StaAcc.dbf	DBF
Jurisdiction code- jurisdiction group lookup	Jur.dbf	DBF
List of zones connected to the Pentagon Metrorail station for the purpose of creating long-distance kiss-and-ride (KNR) links, which represent "slugging" or informal, ad-hoc carpooling	Pen.dbf	DBF
TAZ XY co-ordinates	TAZ_xys.dbf	DBF
SOV AM/Off-peak highway time skims file	AM_SOV_MOD.SKM, MD_SOV_MOD.SKM	Binary

\*\* Areawalk.txt contains information needed to calculate zonal percent-walk-to-transit (PWT) values.

## Input File Descriptions and Formats

Table 35 Variables in the transit station file (Station.dbf)

Name	Type	Field Description
SEQNO	N	Sequence Number
MM	C	Mode Code (M=Metrorail, C=Commuter rail, B=Bus, L=Light rail, N=BRT/streetcar)
<b>NCT</b>	<b>N</b>	<b>Access distance code (1, 2, 3, 0, 9, 8)</b> (See Table 36)
STAPARK	C	Does the station have a park-and-ride lot? (Y=yes; blank=no)
STAUSE	C	Is the station in use for the given year? (Y=yes; blank=no)
SNAME	C	Station Name/PNR lot name
STAC	N	Station centroid number (5001-7999), also known as a park-and-ride (PNR) lot centroid or a dummy PNR centroid"
STAZ	N	For the purposes of path building, the TAZ (1-3722) that represents the location of the station PNR lot. Usually the closest TAZ to the PNR lot.
STAT	N	Station Node (8000-8999, 9000-9999, 10000-10999)
STAP	N	Station park-and-ride (PNR) node number (11000-13999)
STAN1	N	Station bus node #1 (used to generate a station-to-bus-node connector)
STAN2	N	Station bus node #2 (used to generate a station-to-bus-node connector)
STAN3	N	Station bus node #3 (used to generate a station-to-bus-node connector)
STAN4	N	Station bus node #4 (used to generate a station-to-bus-node connector)
<b>STAPCAP</b>	<b>N</b>	<b>Parking capacity (number of spaces at the PNR lot)</b>
STAX	N	X coordinate of station/PNR lot (MD State Plane, NAD83, feet)
STAY	N	Y coordinate of station/PNR lot (MD State Plane, NAD83, feet)
<b>STAPKCOST</b>	<b>N</b>	<b>Peak period parking cost (daily cost, cents)</b>
<b>STAOPCOST</b>	<b>N</b>	<b>Off-peak parking cost (hourly cost, cents)</b>
<b>STAPKSHAD</b>	<b>N</b>	<b>Peak-period shadow price (currently not used)</b>
<b>STAOPSHAD</b>	<b>N</b>	<b>Off-peak-period shadow price (currently not used)</b>
FIRSTYR	N	Year of Station/PNR lot Opening (unused by scripts, but used as metadata)
STA_CEND	N	Project ID (Metadata)
	C	Scenario name, or left blank (Metadata)
	C	Comments, if any, regarding the file, since file cannot accept comment lines preceding the data lines

Notes: New variables are shown with bold font. The SEQNO variable does not correspond to the station node (STAT), and, unlike the STAT, cannot be assumed to stay the same over time.

Source: Jain, M. (2010, October). MWCOG network coding guide for Nested Logit Model (First draft: September 20, 2007; Updated February 2008 and October 2010). Memorandum.

The "access distance code," known as NCT in the autoacc5.s script, is a newly added variable in the station file that controls the number, extent, and directionality of PNR/KNR access links generated for each parking lot (in the case of PNR) or each station (in the case of KNR). Table 36 describes the meaning of each of the six access distance codes.

**Table 36 Interpretation of transit access distance codes (NCT): Metrorail, light rail, and bus PNR access distance codes and their meaning for the**

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

The access distance code, along with the transit mode, determines the maximum link distance for the drive-access-to-transit links generated by autoacc5.s for the TPB nested-logit mode choice model. The maximum link distances for PNR are shown in Table 78. Although not shown in the table, the maximum allowed link distance for KNR links is 3 miles. It is also important to note that the KNR links are generated to Metrorail stations, light rail stations, streetcar stops, and bus stops with parking lots, but not commuter rail stations.

**Table 37 HBW zonal parking costs/terminal time file (HBWV2a1.dbf)**

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

For more information about short walk and long walk to transit, see section 21.4 ("Market segmentation") on page 167.

Table 38 Walk Access Links (WalkAcc\_Lnks.dbf)

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

Table 39 Outputs of pre-transit network processing

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

## 15.2 Application Details

It is important to understand the various elements of the Version 2.3 transit network system. The elements are listed in Table 40. The network consists of highway links, transit stops, PNR lots, rail stations, rail links, and transit lines (modes 1-10). The transit network also contains access links relating to zonal access connections including zone-to-transit-stop walking links (mode 16), and zone-to-KNR/PNR auto links (mode 11). The network also includes other walk-related connections such as sidewalk links used in transferring (mode 13), rail station-to-bus stop connections (mode 12), and PNR lot-to-station connections (mode 15). The above scripts are used to develop all of these types of “support” links, with the exception of station-to-bus transfer links which are addressed as part of pre-network development.

The Mode Choice Model chapter of this report addresses the how access links are developed by the WALKACC.S, Parker.S, and the AutoAcc5.S programs.

**Table 40 Overview of Version 2.3 Transit Network Elements**

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





## 16 Transit Skim File Development

### 16.1 Overview

The transit skimming file process involves the development of 22 sets of level-of-service (LOS) skims corresponding to two time-of-day period (peak and off-peak)<sup>81</sup> by four sub-mode groups (Bus only, Metrorail only, Bus-Metrorail combination, and commuter rail) by three access mode (walk, PNR, KNR).<sup>82</sup> As shown on page A-5 of Appendix A, the transit network building and skimming scripts are named TRANSIT\_SKIMS\_CR.S, TRANSIT\_SKIMS\_MR.S, TRANSIT\_SKIMS\_AB.S, TRANSIT\_SKIMS\_BM.S. These four scripts are launched using two batch files:

- Transit\_Skim\_All\_Modes\_Parallel.bat
- Transit\_Skim\_LineHaul\_Parallel.bat

Additionally, transit accessibility summaries are needed to support the vehicle ownership model. The Transit\_Accessibility.S script is used for this purpose. The inputs and outputs to transit skimming are shown in Table 41 and Table 42, respectively.

**Table 41 Inputs to transit skim file development**

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

<sup>81</sup> For the calculation of average headways and run times, the peak period is represented by the AM peak hour and the off-peak period is represented by the five-hour midday period.

<sup>82</sup> This should equal 24 (2x3x4), but KNR access to commuter rail mode is not considered by the mode choice model, and so the total number of required path sets equals 22.

**Table 42** Outputs of transit skim file development

Commuter rail skim files	SUPL_<Prd>_<AA>_CR.ASC SUPN_<Prd>_<AA>_CR.DBF TRNL_<Prd>_<AA>_CR.DBF <ITER>_<Prd>_<AA>_CR.STA <ITER>_<Prd>_<AA>_CR.SKM <ITER>_<Prd>_<AA>_CR.TTT	Text DBF DBF Binary Binary Binary
Metrorail support skim files	SUPL_<Prd>_<AA>_MR.ASC SUPN_<Prd>_<AA>_MR.DBF TRNL_<Prd>_<AA>_MR.DBF <ITER>_<Prd>_<AA>_MR.STA <ITER>_<Prd>_<AA>_MR.SKM <ITER>_<Prd>_<AA>_MR.TTT	Text DBF DBF Binary Binary Binary
All Bus support skim files	SUPL_<Prd>_<AA>_AB.ASC SUPN_<Prd>_<AA>_AB.DBF TRNL_<Prd>_<AA>_AB.DBF <ITER>_<Prd>_<AA>_AB.STA <ITER>_<Prd>_<AA>_AB.SKM <ITER>_<Prd>_<AA>_AB.TTT	Text DBF DBF Binary Binary Binary
Bus/Metrorail support skim files	SUPL_<Prd>_<AA>_BM.ASC SUPN_<Prd>_<AA>_BM.DBF TRNL_<Prd>_<AA>_BM.DBF <ITER>_<Prd>_<AA>_BM.STA <ITER>_<Prd>_<AA>_BM.SKM <ITER>_<Prd>_<AA>_BM.TTT	Text DBF DBF Binary Binary Binary
Job accessibility by transit	<ITER>_<Prd>_<AA>_[BM MR]_JobAcc.dbf	DBF

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

## 16.2 Application Details

### 16.2.1 Skim file names and list of transit skim tables in the skim files

The skim files developed by the transit skimming process in the Ver. 2.3 travel demand model are shown in Table 43. Each filename is preceded by the speed feedback iteration: pp (pump prime), i1, i2, i3, i4. Each skim file contains 16 tables of information, as shown in Table 44.

Table 43 Skim files developed by the transit skimming process

Submode	Time Period	
	AM Peak Skim Files	Off-Peak Skim Files
Commuter Rail	<iter>_AM_WK_CR.SKM <iter>_AM_DR_CR.SKM <iter>_AM_KR_CR.SKM	<iter>_OP_WK_CR.SKM <iter>_OP_DR_CR.SKM <iter>_OP_KR_CR.SKM
Metrorail Only	<iter>_AM_WK_MR.SKM <iter>_AM_DR_MR.SKM <iter>_AM_KR_MR.SKM	<iter>_OP_WK_MR.SKM <iter>_OP_DR_MR.SKM <iter>_OP_KR_MR.SKM
Bus Only	<iter>_AM_WK_AB.SKM <iter>_AM_DR_AB.SKM <iter>_AM_KR_AB.SKM	<iter>_OP_WK_AB.SKM <iter>_OP_DR_AB.SKM <iter>_OP_KR_AB.SKM
Metrorail and Bus	<iter>_AM_WK_BM.SKM <iter>_AM_DR_BM.SKM <iter>_AM_KR_BM.SKM	<iter>_OP_WK_BM.SKM <iter>_OP_DR_BM.SKM <iter>_OP_KR_BM.SKM

Table 44 Skim tables contained in each transit skim file

Table No.	Table Description
1	In-Vehicle Time-Local Bus (0.01 min)
2	In-Vehicle Time-Express Bus (0.01 min)
3	In-Vehicle Time-Metrorail (0.01 min)
4	In-Vehicle Time-commuter rail (0.01 min)
5	In-Vehicle Time-new rail mode (0.01 min)
6	In-Vehicle Time-new bus mode (0.01 min)
7	Initial wait time (0.01 min)
8	Transfer wait time (0.01 min)
9	Walk access time (0.01 min)
10	Other walk time (0.01 min)
11	Added Transfer time (0.01 min)
12	No. of tranfers (0 to N)
13	Drive access time (0.01 min)
14	Drive acc distance (0.01 mile)
15	PNR-to-Station time (0.01 min)
16	PNR Cost (2007 cents)

Ref: Transit\_Skim\_Specs\_2.xlsx

### 16.2.2 Description of local bus, future time degradation factors

Transit service is represented in the transit network using a series of transit routes, which are stored in transit “line” or “mode” files. There is one set of transit routes for the peak period (represented by AM peak period service) and one set of transit routes for the off-peak period (represented by the midday period). For each of the two time-of-day periods, each transit route has the following:

- Name (such as “WM04AI,” or WMATA bus 4A, inbound),
- Flag indicating whether the route is one way or two way,
- Mode code (e.g., 1 = local bus),
- Average headway (FREQ[1]= 30, which means the bus comes every 30 minutes), and
- Average run time (i.e., the number of minutes from the start of the route to the end of the route, e.g., RUNTIME= 42 min.).

When developing the transit networks for a base year (i.e., a year close to the current year, such as 2015), the average headways and average run times come directly from the published schedules from the transit providers. These schedules can be in paper format or electronic format, such as GTFS. For a future-year transit network (such as 2040), however, the average headway and run time are unknown, so we use information from the latest published schedule (e.g., 2015). However, simply using the published schedules would likely result in bus speeds that are too fast, since they don’t account for the added roadway congestion that is likely to occur in the future, i.e., it is likely that worsening road congestion over time would result in slower bus speeds. In particular, **local** bus service, which travels on local roads, might be slowed more than express bus service, which makes use of freeways and expressways for all or part of its routes. Thus, it would be good to have a relationship that relates future-year, congested road/link speeds to bus speeds. Before 2004, the COG/TPB travel model had no such relationship. In 2004, AECOM recommended that COG develop a relationship between link speeds and bus speeds, but cautioned against developing overly sensitive relationships.<sup>83</sup> For example, if one develops a direct relationship between the link speed and the bus travel times over that link, and if one road link becomes hyper congested, due, say, to excessive traffic or a network coding error, then the bus speed will drop to near zero. Consequently, COG/TPB staff developed a proposed solution that followed what was proposed in the Bruggeman/Woodford memo. The solution was what is known as the local bus, future time degradation factors, which are used to represent the fact that, as the highway network becomes more congested, there will be a slight degradation in **local** bus speeds over time. This technique was first used in the Version 2.1/D Travel Model,<sup>84</sup> and has been retained in the Version 2.2 and 2.3 travel models. In 2015, the local bus speed degradation factors were re-estimated,<sup>85</sup> and those re-estimated factors were part of the Ver. 2.3.57a travel demand model.

---

<sup>83</sup> Jeff Bruggeman and Bill Woodford to Ronald Milone, “Comments on MWCOG Modeling Procedures,” Memorandum, (June 30, 2004).

<sup>84</sup> Ronald Milone to Files, “Methodology for Linking Future Bus Speeds to Highway Congestion in the Version 2.1/D Model,” Memorandum, (July 14, 2004).

<sup>85</sup> Meseret Seifu and Ronald Milone, “Update of Local Bus Speed Degradation Model,” Memorandum, (March 19, 2015).

For the Ver. 2.3.66 model, the process has been significantly updated, as was described in Section 1.3.4 and a technical memo.<sup>86</sup> The major changes are as follows:

1. The text file Lbus\_TimFTRS.asc has been replaced with the dBase file Bus\_Factor\_File.dbf.
2. A new script has been added (Adjust\_Runtime.s). This script reads in the transit line files associated with the local bus routes (mode codes 1, 6, and 8), adjusts the RUNTIME values by the factors contained in Bus\_Factor\_File.dbf, and writes out revised mode 1, 6, and 8 files with the revised RUNTIME values. The script Adjust\_Runtime.s is called from Transit\_Skim\_All\_Modes\_Parallel.bat.

An example of a local bus route **before** the adjustment of its RUNTIME variable is shown in Figure 22. An example of a local bus route **after** the adjustment of its RUNTIME variable is shown in Figure 23.

```
LINE NAME="ART43N",
OWNER="ART Bus;Crystal City Bay A, S Bell St, SB @ S Hayes St S;Crystal City Bay A, S Bell St, SB @ S Hayes St S;2014;base",
ONEWAY= Y,MODE= 01,FREQ[1]= 20,RUNTIME= 40,
N= 30247 30666 30279 -30246 -30280 -30243 -30244 -30207 30206,
    30204 -30286 -30316 -30315 -30211 30115 30120 -30115 30116,
    30520 30117 -30122 30123
```

Figure 22 A local bus route before its RUNTIME value is updated

```
LINE NAME="ART43N",
OWNER="ART Bus;Crystal City Bay A, S Bell St, SB @ S Hayes St S;Crystal City Bay A, S Bell St, SB @ S Hayes St S;2014;base",
ONEWAY= Y,MODE= 01,FREQ[1]= 20,RUNTIME=43.9, ; Base RUNTIME= 40.00 Time Factor: 1.098 Year: 2040
N= 30247 30666 30279 -30246 -30280 -30243 -30244 -30207 30206,
    30204 -30286 -30316 -30315 -30211 30115 30120 -30115 30116,
    30520 30117 -30122 30123
```

Figure 23 A local bus route after its RUNTIME value is updated to reflect road congestion predicted to occur in future years

Transit accessibility outputs are listed on Table 45.

Table 45 Job accessibility by transit file format description (<ITER>\_<Prd>\_<AA>\_[BM|MR]\_JobAcc.dbf)

Variable Name	Description
TAZ	TAZ (1-3722)
EMP35	Number of jobs accessible, from a given zone, within 35 minutes' travel time
EMP40	Number of jobs accessible, from a given zone, within 40 minutes' travel time
EMP45**	Number of jobs accessible, from a given zone, within 45 minutes' travel time
EMP50	Number of jobs accessible, from a given zone, within 50 minutes' travel time
EMPTOT	Total number of jobs accessible, from a given zone

\*\* Only EMP45 is used from this file.

<sup>86</sup> Ronald Milone to Feng Xie et al., "Update to the V2.3.57a Model's Treatment of Bus Speed Factors," Memorandum, (March 17, 2016).



## 17 Transit Fare Development

### 17.1 Overview

Zone-to-zone transit fares are developed for the 22 paths sets described in the transit skimming section (section 16). As shown on page A-12 of Appendix A, the fares are developed using the scripts named Prefarv23.s, Metrorail\_Skims.S, Mfare1.s, and Mfare2.s. The inputs to the fare process are shown in Table 46 and the outputs are shown in Table 47. After the fare process is executed, four scripts are used to combine transit skims and fares into consolidated submode files: Assemble\_Skims\_CR.s, Assemble\_Skims\_MR.s, Assemble\_Skims\_AB.s, and Assemble\_Skims\_BM.s.

Table 46 Inputs to transit fare development

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

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

\*\* These two files are originally in the Inputs folder, and then are copied to the Output folder in Transit\_Skim\_All\_Modes\_Parallel.bat

Table 47 Outputs of transit fare development

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



Table 48 TAZ / Bus Fare Zone Equivalency File Format Description (TAZFRZN.ASC)

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

## 17.2 Application Details

The purpose of transit fare process is to develop a zonal matrix containing total transit costs as expressed in 2007 cents. The core components of the transit fare process are two scripts: MFARE1.S which develops Metrorail station-to-station fares and MFARE2.S which develops zone-to-zone transit fares using the MFARE1.S output. Twenty-two fare matrices are developed sub-mode, time period, and access type, specifically:

- Four sub-modes (Bus Only Metrorail only, Metrorail/ Bus, and Commuter Rail) by;
- Two time periods (AM, off-peak), by;
- Three access types (Walk, PNR, and KNR)

Since commuter rail access is distinguished by walk and auto access only, 22 matrices are developed (instead of 24 which is implied above).

The fare process is executed with a batch file named TRANSIT\_FARE.BAT. The batch file calls four scripts that are used to formulate the zone-to-zone transit fares for each market:

- PrefarV23.S: This script reads a zonal transit walk area file (NLWalkPct.txt) which includes walk areas pertaining to Metrorail stations only. It also reads an equivalency file (TAZFRZN.ASC) that equates TAZs to bus fare zones and Metrorail station numbers to bus fare zones. The program essentially merges the Metrorail walk percent information into the zonal equivalency file. The resulting file is named fare\_a2.asc. This file is called by the MFARE2.S script and is needed for the zonal transit fare calculation. This script is also used to develop the zonal parking costs that are input into the mode choice model.
- Metrorail\_Skims.S: This script reads a Metrorail link and node file, and then develops Metrorail station-to-station distance skims. The file is need for the Metrorail station-to-station fare calculation.

- MFARE1.S: This script calculates the Metrorail station-to-station fares for AM and off-peak periods. The script reads in a fare parameter file that is consistent with WMATA's Metrorail fare policy (tariff.txt), station coordinates (MFARE1.A1), and a station discount file (MFARE1\_STA\_DISC.ASC). The script writes two text files containing Metrorail fares: AM\_Metrorail\_Fares.txt and OP\_Metrorail\_Fares.txt.
- MFARE2.S: This script calculates the total transit fare between TAZs for AM and off-peak periods. The script reads in several files:
  - the Metrorail station-to-station fares developed by MFARE1.s,
  - tariff.txt (transit fare policy parameters contain rail-to-bus discounts)
  - TRN\_Deflator.txt (the transit deflation factor)
  - Fare\_a2.asc (file containing zonal walk percentages to Metrorail stations)
  - BUSFAREAM/OP.ASC: AM and off-peak bus and commuter rail fares between bus fare policy zones. TPB currently uses 21 bus fare zones for the region. While most TAZs fall into a single bus policy zone, the fare calculation also accounts for the possibility that a single TAZ may be straddle 2 bus policy zones
  - Zonal skim files containing Metrorail on/off stations (\*.STA) and in-vehicle travel times by transit mode (\*.SKM). A set of transit skims must exist for each of the 22 transit paths.

The transit fare files are written to 22 binary file (\*.FAR) each containing 1 table (total transit fare in 2007 cents). The batch file calls four additional scripts (Assemble\_Skims.?.S) which are used to consolidate the 22 binary fare files into 4 files associated with each sub-mode. The consolidated files are subsequently used as inputs to the mode choice model.

The fare construction process between zonal pairs essentially consists of blending the Metrorail station-to-station fares with the bus zone-to-bus zone fares. The consideration of Metrorail fares is dependent upon individual path characteristics, i.e., whether or not the Metrorail in-vehicle time is greater than zero. If the path is not Metrorail-related then the fare is developed from the bus fare matrix input. If the path is Metrorail-related then the transit fare is based on the Metrorail station-to station fare (from MFARE1), bus access and/or egress fares developed from the bus fare matrix, zonal Metrorail walk potential, and the Rail-to-Bus policy discount. The MFARE2 computation may be explained as a series of discrete conditions.

Condition 1: Non-Metrorail related path / Single bus fare zone origin to Single bus fare destination zone

Transit fare = Bus Fare(bi1/bj1)

Condition 2: Non-Metrorail related path / Single bus fare zone origin to Double bus fare destination zone:

Transit fare = [(Bus Fare( bi1/bj1) + Bus Fare( bi1/bj2)) / 2.0

Condition 3: Non-Metrorail related path / Double bus fare zone origin to Double bus fare destination zone

Transit fare = [(Bus Fare( bi1/bj1) + Bus Fare(b i1/bj2) + (Bus Fare(b i2/bj1) + Bus Fare(b i2/bj2))] / 4.0

Condition 4: Metrorail related paths

Transit Fare = (Bus Access fare \* (1.0-Origin Metrorail walk Pct.)) + Metrorail fare( $s_i/s_j$ ) +  
(Bus Egress fare \* (1.0-Destin. Metrorail walk Pct.))

Bus Access Fare Single bus fare zone to single Metrorail bus fare zone =  
Bus Fare( $bi_1/mi_1$ ) – 0.5 Rail-Bus Discount

Bus Access Fare Single bus fare zone to double Metrorail bus fare zone =  
[Bus Fare( $bi_1/mi_1$ ) + Bus Fare( $bi_1/mi_1$ )]/2.0 – 0.5 Rail-Bus Discount

Bus Access Fare Double bus fare zone to double Metrorail bus fare zone =  
[Bus Fare( $bi_1/mi_1$ ) + Bus Fare( $bi_1/mi_2$ ) + Bus Fare( $bi_2/mi_1$ ) + Bus Fare( $bi_2/mi_2$ )]/4.0 – 0.5 Rail-Bus  
Discount

Bus egress fares are calculated in the same way that bus access fares are calculated. A fare discount is applied to the fare calculation before it is written out to the binary output.



## 18 Demographic Submodels

Demographic submodels are applied within the Trip\_Generation.bat batch file using the Demo\_Models.s Cube Voyager script (see page A-6 of Appendix A). This script applies the three demographic submodels that are run prior to trip generation: household size, household income, and vehicle availability (see Chapter 3 of the calibration report for more details). The inputs to the model are zonal land use data (zone.dbf), data about area types (areaType\_File.dbf), and information about the accessibility to jobs via transit. The zone.dbf file contains zonal households, population, jurisdiction code, and income index, as well as the household size and household income submodels (in the form of lookup tables). The households in each TAZ are then allocated to a household size group (1, 2, 3, or 4+) and an income group (<50K, 50K-100K, 100K-150K, or 150+K).

Next, the Demo\_Models.s reads in the number of jobs accessible by AM Metrorail and Bus/Metrorail service within 45 minutes for each TAZ (see Table 45). This information along with household size, household income, area type, and the DC dummy variable are used to allocate households to the four vehicle ownership categories (0, 1, 2, or 3+).

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

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

Process inputs and outputs are shown in Table 49 and Table 50.

**Table 49 Inputs to the Demographic Models**

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

Note: <ITER> =PP, i1...i4

**Table 50 Outputs of the Demographic Models**

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

## 19 Trip Generation

### 19.1 Control/Support File(s):

TRIP\_GENERATION.S, TRIP\_GENERATION\_Summary.S, Truck\_Com\_Trip\_Generation.s

### 19.2 Application Details:

Trip generation is executed within the Trip\_Generation.bat batch file using three Cube Voyager scripts: Trip\_Generation.s, Trip\_Generation\_Summary.s, and Truck\_Com\_Trip\_Generation.s (as shown on page A-6 of Appendix A). The inputs to the Trip\_Generation.bat batch file are shown in Table 51.

Table 51 Inputs to trip generation

Zonal land use file	zone.dbf	DBF
Zonal Area Type File	AreaType_File.dbf	DBF
Zonal HHs stratified by income level, HH size, & vehs available	<iter>_Demo_Models_HHbyISV.dbf	DBF
Zonal GIS variable file	GIS_variables.dbf	DBF
Trip production rates	weighted_trip_rates.dbf	DBF
External Production and Attraction File	Ext_PsAs.dbf	DBF
Non-motorized trip production share model coefficients	NMPrates.dbf	DBF
Non-motorized trips Attraction share model coefficients	NMArates.dbf	DBF
Trip attraction rates	AttrRates.dbf	DBF
HB income shares	HBINCRAT.dbf	DBF
Consolidated zonal land use file	TripGen_LUFile.dbf	DBF
Truck and commercial vehicles trip rates	support\truck_com_trip_rates.dbf	DBF
Zonal access verification file	Skimtot<ITER>.txt	Text
	JurCore.dbf	DBF

The Trip\_Generation.s script calculates zonal trip productions and attractions. The Trip\_Generation\_Summary.s summarizes the demographic information and the trip ends by jurisdiction. The Truck\_Com\_Trip\_Generation.s produces trip ends for commercial vehicles and trucks.

The Trip\_Generation.s script is very long (almost 1,500 lines). Figure 24 presents an outline or pseudo code of the steps in the trip generation script. The script has three phases, as indicated in the figure. There are a few points to note: **First**, the program is applied to compute zonal initial trip productions and (unscaled) zonal trip attractions. Attraction scaling is performed later, in the “Prepare\_Internal\_Ends.s” script. **Second**, the program makes sparing use of two sets of adjustments: jurisdiction level adjustments (end of phase 1) and area-type level adjustments (phases 1 and 2). The model does not make use of any “special generators” and the model does not make use of zone-level adjustments, which are used in some models and are usually referred to production modification factors (“p-mods”) and attraction modification factors (“a-mods”).<sup>87</sup>

<sup>87</sup> See, for example, William W. Mann, “TRIMS - Four Steps: One Execution,” *ITE Journal* 52, no. 12 (December 1982): 16, <http://www.ite.org/membersonly/itejournal/pdf/JLA82A13.pdf>.

---

**Phase 1: Read in input data and trip rates and establish parameters**

1. Read input files into arrays. The inputs include: zonal land activity, external Ps/As, zonal area types, zonal HHs stratified by Inc./Size/ VA., zonal GIS variables, trip production rates, trip attraction rates, and income attraction shares by HB purpose area type
2. Establish output files:
  - a. Report file (%\_iter\_%\_Trip\_Generation.txt')
  - b. Computed Zonal trip productions ('%\_iter\_%\_Trip\_Gen\_Productions\_Comp.dbf')
  - c. Computed Zonal trip Attractions ( '%\_iter\_%\_Trip\_Gen\_Attractions\_Comp.dbf')
3. Establish Area-Type trip end (motorized, non-motorized) factors by purpose and area type
4. Establish External trip parameters (Share of ext. NHB travel that is NHW and NHO, auto occupancies of external autos, by purpose)
5. Establish Jurisdictional trip end factors by purpose

**Phase 2: Compute Initial Trip Productions and Attractions**

1. Loop through each **internal** zone
  - a. Apply trip production rates to stratified HHs by income, size, vehav. To arrive at total Ps
  - b. Computed non-motorized production shares by purpose and area type
  - c. Apply non-motorized shares and adjustment parameters to total Ps to arrive at final motorized & non-motorized Ps
  - d. Summarize and write out internal computed trip Ps stratified by income
  - e. Apply trip attraction rates to land activity
  - f. Computed non-motorized attraction shares by purpose and area type
  - g. Apply non-motorized shares and adjustment parameters to total As to arrive at final motorized & non-motorized As
  - h. Disaggregate total final attractions to income strata, by purpose and area type

End **internal** zone loop

2. Loop through each internal zone: Summarize and write out internal computed attractions by income
3. Loop through each **external** zone
  - a. Read external auto driver trip Ps and As
  - b. Convert external vehicle Ps and As to auto person trips based on car occ. parameters
  - c. Disaggregate total external NHB auto persons among NHW and NHO based on parameters
  - d. Write out external Ps and As

End **external** zone loop

**Phase 3: Print out regional totals of computed trip productions/attractions**

---

Figure 24 Outline/pseudo code for trip\_generation.s



Trip\_generation.s begins, in phase 1, by reading the zonal land use (Zone.dbf); the area type file (AreaType\_File.dbf); external trip productions and attractions (EXT\_PsAs.dbf, described in Table 52); zonal households stratified by income, size, and vehicles available (<iter>\_Demo\_Models\_HHbyISV.dbf); zonal walkability factors (GIS\_variables.dbf); trip production rates (weighted\_trip\_rates.dbf); non-motorized production model coefficients (NMPrates.dbf); non-motorized attraction model coefficients (NMArates.dbf); trip attraction model coefficients (AttrRates.dbf); and income shares for home-based trips (HBINCRAT.dbf).

The zonal GIS variable file (GIS\_variables.dbf) contains a number of built-environment variables that describe the walkability of an area, such as the number of 3-legged intersections per TAZ, the number of cul-de-sacs per TAZ, the number of street blocks per TAZ, and the number of Census blocks per TAZ. Although the GIS file contains a number of variables, the trip generation process uses only one: **the number of street blocks per TAZ ("BLOCKS")**.<sup>88</sup> Since these built-environment variables are intended to deal with issues of walkability, it is best to use a detailed street network when calculating these metrics (as opposed to simply using the highway network itself, which is quite coarse). In our case, we used NAVTEQ's NAVSTREETS Street Data<sup>89</sup> (for which COG pays a license fee) and the work was performed in 2010 by COG/TPB staff.<sup>90</sup> A block is defined as a 2-dimensional area (polygon) that is completely enclosed by a series of NAVTEQ street segments. Prior to forming blocks, the following segments were removed from the NAVTEQ street network:

- Street segments with no name (ST\_NAME=blank), since these are not actually street segments;
- "Major highways" (NAVTEQ functional class [FUNC\_CLASS] equal to 1 or 2).<sup>91</sup>
- Ramps (RAMP = Y)

Figure 25 shows an example of the seven blocks that are contained within TAZ 283 (Union Station), as defined by NAVTEQ street segments (omitting major highways and ramps, as discussed above).

---

<sup>88</sup> It has been found that areas with a higher density of street blocks are more walkable.

<sup>89</sup> NAVTEQ, "NAVTEQ's NAVSTREETS Street Data, Reference Manual v3.2," Proprietary and Confidential (Chicago, Illinois: NAVTEQ, April 1, 2009).

<sup>90</sup> Mary Martchouk to Mark S. Moran, "Developing GIS Walkability Measures," Memorandum, (June 2, 2010), 6–7.

<sup>91</sup> NAVTEQ, "NAVTEQ's NAVSTREETS Street Data, Reference Manual v3.2", p. 4-5.

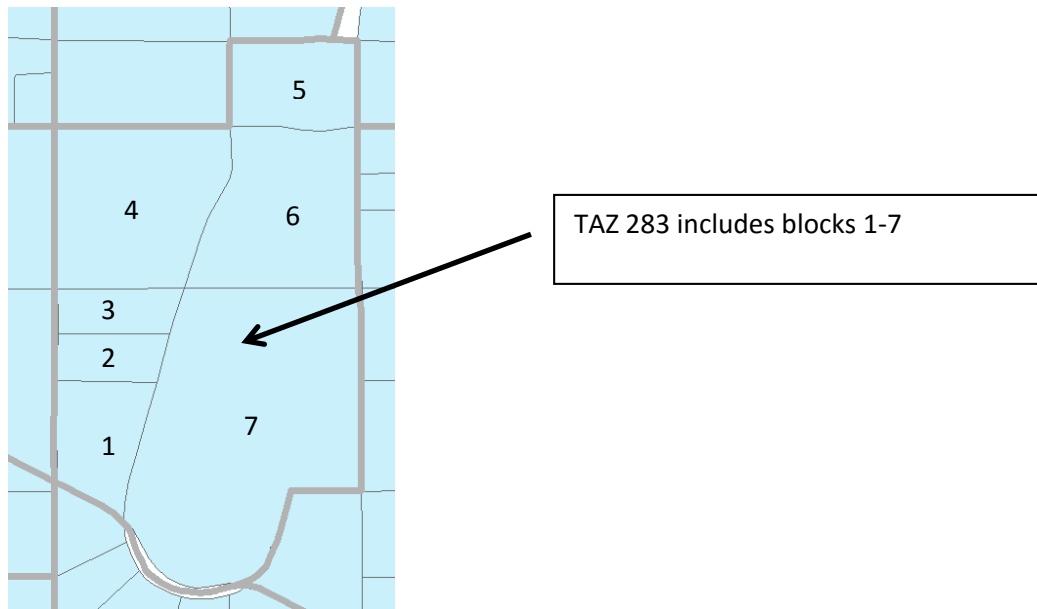


Figure 25 Example of seven NAVTEQ street blocks within TAZ 283 (Union Station)

In the trip generation script (Trip\_Generation.s), the number of blocks per TAZ is then converted to a density measure, specifically the floating 0.5-mile block density for each TAZ (BLOCKS05, lines 180-215 of Trip\_Generation.s). In the travel model, it is assumed that the block density has an effect on non-motorized trip productions and attractions **only for area types 1 and 2** (See, for example, Tables 27-29 of the calibration report for non-motorized productions and Tables 30-32 for non-motorized attractions, where the floating 0.5-mile block density is called BLKDEN05).<sup>92</sup>

It is assumed that the model user will rarely change the value of BLOCKS (or its derivative, BLKDEN05) when running the model (i.e., the modeler will freeze the base-year levels of block density). This is analogous to the way that household income distributions are generally frozen in the model. The exception to this rule is if the modeler believes that the street network in an area will become denser or sparser, then the modeler can make appropriate manual adjustments to the zonal BLOCK variable.

The trip generation process also reads external trip ends from a file (Ext\_PsAs.dbf) that is developed exogenously. The data items are shown in Table 52. The Trip\_Generation.S script writes out an intermediate dBase file containing land activity, one-mile “floating” land use density, one-half mile “floating” block density, and jurisdictional and area type codes. The file is a consolidation of input data from various zone files and derived variables.<sup>93</sup> The specific data elements are shown on Table 53. Note that, in the file Ext\_PsAs.dbf (Table 52), the last two variables are medium truck external-internal (X-I) trip ends (MTK\_XI) and heavy truck external-internal (X-I) trip ends (HTK\_XI), but there are no

<sup>92</sup> Milone et al., “Calibration Report for the TPB Travel Forecasting Model, Version 2.3,” 4-17-20.

<sup>93</sup> Floating densities are calculated using the centroids of the TAZs and street blocks, with a point buffer around the centroid with the given radius (0.5 mile or 1.0 mile).

corresponding variables for the internal-external movements (e.g., MTK\_IX and HTK\_IX). This is because it is assumed that the two movements (XI and IX) are the same.

Table 52 External Production and Attraction File (Ext\_PsAs.dbf)

Variable	Description
TAZ	External station no. (3676-3722)
FACILITY	Facility route no./name
AAWT_CTL	Average annual weekday traffic count (observed or forecasted)
CNTFTR	(unused)
AUTO_XI	Auto driver external-internal (X-I) trip ends
AUTO_IX	Auto driver internal-external (I-X) trip ends
AUTO_XX	Auto driver through (X-X) trip ends
CV_XX	Commercial vehicle through (X-X) trip ends
HBW_XI	HBW external-internal (X-I) trip ends
HBS_XI	HBS external-internal (X-I) trip ends
HBO_XI	HBO external-internal (X-I) trip ends
NHB_XI	NHB external-internal (X-I) trip ends
CV_XI	Commercial vehicle external-internal (X-I) trip ends
HBW_IX	HBW internal-external (I-X) trip ends
HBS_IX	HBS internal-external (I-X) trip ends
HBO_IX	HBO internal-external (I-X) trip ends
NHB_IX	NHB internal-external (I-X) trip ends
CV_IX	Commercial vehicle internal-external (I-X) trip ends
TRCK_XX	Truck through (X-X) tip ends (medium and heavy truck)
TRCK_XI	Truck external-internal (X-I) trip ends (medium and heavy truck)
TRCK_IX	Truck internal-external (I-X) trip ends (medium and heavy truck)
MTK_XI	Medium truck external-internal (X-I) trip ends
HTK_XI	Heavy truck external-internal (X-I) trip ends

Source: Milone, R. (2011, July 1). Version 2.3 Exogenous Trip Files. Memorandum.

The one-mile floating density is then calculated for population and employment and a half-mile floating density is calculated for street blocks. These are saved in an intermediate file named TripGen\_LUFile.dbf (Table 53). Then, the script calculates zonal trip productions based on demographic data and applies the non-motorized production model to the results. Motorized internal trips productions are then obtained by subtracting the estimated non-motorized trips. The output production file data items are shown on Table 54.

Table 53 Consolidated Zonal Land Use File

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

Table 54 Computed zonal trip productions file (<iter>\_Trip\_Gen\_Productions\_Comp.dbf)

Variable Name	Description
TAZ	TAZ Number (1-3,722)
HBW_MTR_PS	Home-Based-Work motorized person trip productions
HBW_NMT_PS	Home-Based-Work non-motorized person trip productions
HBW_ALL_PS	Home-Based-Work motorized and non-motorized person trip productions
HBWMTRP_I1	Home-Based-Work Motorized person trip productions, Income level 1
HBWMTRP_I2	Home-Based-Work Motorized person trip productions, Income level 2
HBWMTRP_I3	Home-Based-Work Motorized person trip productions, Income level 3
HBWMTRP_I4	Home-Based-Work Motorized person trip productions, Income level 4
HBS_MTR_PS	Home-Based-Shop motorized person trip productions
HBS_NMT_PS	Home-Based-Shop non-motorized person trip productions
HBS_ALL_PS	Home-Based-Shop motorized and non-motorized person trip productions
HBSMTRP_I1	Home-Based-Shop Motorized person trip productions, Income level 1
HBSMTRP_I2	Home-Based-Shop Motorized person trip productions, Income level 2
HBSMTRP_I3	Home-Based-Shop Motorized person trip productions, Income level 3

HBSMTRP_I4	Home-Based-Shop Motorized person trip productions, Income level 4
HBO_MTR_PS	Home-Based-Other motorized person trip productions
HBO_NMT_PS	Home-Based-Other non-motorized person trip productions
HBO_ALL_PS	Home-Based-Other motorized and non-motorized person trip productions
HBOMTRP_I1	Home-Based-Other Motorized person trip productions, Income level 1
HBOMTRP_I2	Home-Based-Other Motorized person trip productions, Income level 2
HBOMTRP_I3	Home-Based-Other Motorized person trip productions, Income level 3
HBOMTRP_I4	Home-Based-Other Motorized person trip productions, Income level 4
NHW_MTR_PS	Non-Home-Based Work-Related motorized person trip productions
NHW_NMT_PS	Non-Home-Based Work-Related non-motorized person trip productions
NHW_ALL_PS	Non-Home-Based Work-Related motorized & non-motorized person trip productions
NHO_MTR_PS	Non-Home-Based Non-Work-Related motorized person trip productions
NHO_NMT_PS	Non-Home-Based Non-Work-Related non-motorized person trip productions
NHO_ALL_PS	Non-Home-Based Non-Work-Related motorized & non-motorized person trip productions

Next, the zonal trip attractions are calculated by applying the attraction trip models to the land use file. Non-motorized trip attractions are then determined and subtracted from the total trip attractions. Similar to productions, attractions are multiplied by an adjustment factor (Appendix A of the Calibration Report) and disaggregated by income level. The computed trip attractions are then written out to <ITER>\_Trip\_Gen\_Attractions\_Comp.dbf file. The final trip attractions are saved in the <ITER>\_Trip\_Gen\_Attractions\_Final.dbf described in Table 55.

Table 55 Computed zone trip attractions file (<iter>\_Trip\_Gen\_Attractions\_Comp.dbf)

Variable Name	Description
TAZ	TAZ Number (1-3,722)
HBW_MTR_AS	Home-Based-Work motorized person trip Attractions
HBW_NMT_AS	Home-Based-Work non-motorized person trip Attractions
HBW_ALL_AS	Home-Based-Work motorized and non-motorized person trip Attractions
HBWMTRA_I1	Home-Based-Work motorized person trip Attractions, Income level 1
HBWMTRA_I2	Home-Based-Work motorized person trip Attractions, Income level 2
HBWMTRA_I3	Home-Based-Work motorized person trip Attractions, Income level 3
HBWMTRA_I4	Home-Based-Work motorized person trip Attractions, Income level 4
HBS_MTR_AS	Home-Based-Shop motorized person trip Attractions
HBS_NMT_AS	Home-Based-Shop non-motorized person trip Attractions
HBS_ALL_AS	Home-Based-Shop motorized and non-motorized person trip Attractions
HBSMTRA_I1	Home-Based-Shop motorized person trip Attractions, Income level 1
HBSMTRA_I2	Home-Based-Shop motorized person trip Attractions, Income level 2
HBSMTRA_I3	Home-Based-Shop motorized person trip Attractions, Income level 3
HBSMTRA_I4	Home-Based-Shop motorized person trip Attractions, Income level 4

HBO_MTR_AS	Home-Based-Other motorized person trip Attractions
HBO_NMT_AS	Home-Based-Other non-motorized person trip Attractions
HBO_ALL_AS	Home-Based-Other motorized and non-motorized person trip Attractions
HBOMTRA_I1	Home-Based-Other motorized person trip Attractions, Income level 1
HBOMTRA_I2	Home-Based-Other motorized person trip Attractions, Income level 2
HBOMTRA_I3	Home-Based-Other motorized person trip Attractions, Income level 3
HBOMTRA_I4	Home-Based-Other motorized person trip Attractions, Income level 4
NHW_MTR_AS	Non-Home-Based Work-Related motorized person trip Attractions
NHW_NMT_AS	Non-Home-Based Work-Related non-motorized person trip Attractions
NHW_ALL_AS	Non-Home-Based Work-Related motorized & non-motorized person trip Attractions
NHO_MTR_AS	Non-Home-Based Non-Work-Related motorized person trip Attractions
NHO_NMT_AS	Non-Home-Based Non-Work-Related non-motorized person trip Attractions
NHO_ALL_AS	Non-Home-Based Non-Work-Related motorized & non-motorized person trip Attractions

The Trip\_Generation\_Summary.s creates a summary text file, <ITER>\_Trip\_Generation\_Summary.txt, which includes the following tables:

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

The Truck\_Com\_Trip\_Generation.s script reads in the zonal land use file (Zone.dbf), the area type file (AreaType\_File.dbf), external trip productions and attractions (EXT\_PsAs.dbf), demographic model outputs (%\_iter\_%\_Demo\_Models\_HHbyISV.dbf), truck and commercial trip model coefficients (truck\_com\_trip\_rates.dbf), and the zonal access verification file (Skimtot<ITER>.txt). For the list of inputs, see Table 51. The script then uses the truck and commercial trip model coefficients and the land use data to calculate medium and heavy truck and commercial vehicle zonal trips. After an adjustment factor is applied, these are written out to a ComVeh\_Truck\_Ends\_<ITER>.dbf file described in Table 56.

Table 56 Truck and commercial vehicles trip ends (<iter>\_ComVeh\_Truck\_Ends.dbf)

Variable Name	Description
TAZ	TAZ number (1-3722)
COMM_VEH	Commercial vehicle trip ends
MED_TRUCK	Medium truck trip ends
HVY_TRUCK	Heavy truck trip ends
ICOMM_VEH	Commercial vehicle trip ends (internal only)
IMED_TRUCK	Medium truck trip ends (internal only)
IHVY_TRUCK	Heavy truck trip ends (internal only)

The script also generates a summary text file- <ITER>\_Truck\_Com\_Trip\_Generation.txt, which includes the following tables:

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

The trip generation process is currently applied to produce computed trip productions and computed (un-scaled) attractions by trip purpose. The computed productions and attractions are provided explicitly as motorized and non-motorized. The Home-Based motorized Ps and As are further stratified by income level. In prior trip generation versions, an Internal to External production share model was employed to extract the external travel component of total trip productions (of I-X trips). The extraction was necessary because external trip ends are prepared exogenously based on projected traffic counts. The potential problem with an I-X extraction model is that there is no guarantee that the model would yield I-X productions already developed exogenously at the external station level. It was ultimately decided that the approach for treating external trips in the generation and distribution process, and the approach for trip attraction scaling would be modified to ensure that I-X trips would be better preserved.

The modified process now involves the following Trip Generation and Trip distribution steps:

1. Trip\_Generation.S: Computed trip productions and computed trip attractions are developed by purpose and mode (motorized and non-motorized). Trip attraction scaling is not undertaken.
2. Prepare\_Ext\_Auto\_Ends.S, Prepare\_Ext\_ComTruck\_Ends.S: External trip-ends (Ps and As) are prepared
3. Trip\_Distribution\_External.s: External trip-ends are distributed, resulting in external trip tables, by purpose
4. Prepare\_Internal\_Ends.s: Final internal trip-ends are computed as follows:
  - External trip ends (I-X) trips and (X-I) trips-ends are summarized by purpose from the external trip matrices developed in Step 3
  - The zonal I-X trip ends are subtracted from the motorized trip productions computed in Step 1. This results in final motorized productions. Non-motorized productions are unaffected.

- Scaling factors for internal trip attractions are computed by purpose. The factor is<sup>94</sup>

$$\text{IntAttrScaleFtr} = (\text{"Final" Intl P's} + \text{Extl. P's} - \text{Extl. A's}) / \text{Intl. "Computed" A's}$$

The above factor is applied to both motorized and non-motorized trip attractions

5. Trip\_Distribution\_Internal.s: The final internal P's and balanced A's are run through trip distribution. The resulting internal trips are combined with the external trips developed in step three.

While this process is slightly more complicated than the prior approach it better ensures that external trips developed exogenously are preserved through the trip distribution stage.

---

<sup>94</sup> This equation was developed by Bill Mann.



## 20 Trip Distribution

### 20.1 Overview

The trip distribution process (shown on page A-7 of Appendix A) is invoked by the Trip\_Distribution.bat file. The input and output files are listed in Table 57 and Table 58. As stated in the calibration report, the gravity model is doubly-constrained for all five trip purposes.

The trip distribution process entails five Cube Voyager steps that involve two separate trip distribution procedures: one to distribute external auto person trips by purpose, and another to distribute internal motorized person trips by purpose. As explained in the trip generation chapter, this dual distribution procedure enables external trips (I-X) trips to be more precisely preserved at the station level compared to the prior trip generation/distribution approach.

Table 57 Inputs to trip distribution

Item	Filename	Format
Computed zonal motorized trip productions	<iter>_Trip_Gen_Productions_Comp.dbf	dBase
Computed zonal motorized trip attractions (un-scaled)	<iter>_Trip_Gen_Attractions_Comp.dbf	dBase
Computed zonal commercial, truck trip ends (Ps, As)	<iter>_ComVeh_Truck_Ends.dbf	dBase
AM highway skims	<Prelter>_AM_SOV.SKM	Binary
OP highway skims	<Prelter>_OP_SOV.SKM	Binary
AM Walk Access Metrorail-only total travel time	<iter>_AM_WK_MR.ttt	Binary
AM Drive Access Metrorail-only total travel time	<iter>_AM_DR_MR.ttt	Binary
OP Walk Access Metrorail-only total travel time	<iter>_OP_WK_MR.ttt	Binary
OP Drive Access Metrorail-only total travel time	<iter>_OP_DR_MR.ttt	Binary
Toll-time equiv. file (by Income/purpose)	Equiv_Toll_Min_by_Inc.s	Text
K-Factor matrices	HBW_K.mat, HBS_k.mat, ... ,NHO_k.mat	binary
Friction factors	Ver23_F_Factors.dbf	dBase

Note: <ITER> =PP, i1, ..., i4

Table 58 Outputs of trip distribution

Item	Filename	Format
HBW Motorized Psn. Trips (internal & external)	<iter>_HBW.PTT	Binary
HBS Motorized Psn. Trips (internal & external)	<iter>_HBS.PTT	Binary
HBO Motorized Psn. Trips (internal & external)	<iter>_HBO.PTT	Binary
NHW Motorized Psn. Trips (internal & external)	<iter>_NHW.PTT	Binary
NHO Motorized Psn. Trips (internal & external)	<iter>_NHO.PTT	Binary
Commercial Vehicle Trips (internal & external)	<iter>_Commer.PTT	Binary

Medium Truck Trips (internal & external)	<iter>_MTruck.PTT	Binary
Heavy Truck Trips (internal & external)	<iter>_HTruck.PTT	Binary
HBW Motorized Psn. Trips (internal only)	<iter>_HBW_NL.PTT	Binary
HBS Motorized Psn. Trips (internal only)	<iter>_HBS_NL.PTT	Binary
HBO Motorized Psn. Trips (internal only)	<iter>_HBO_NL.PTT	Binary
NHW Motorized Psn. Trips (internal only)	<iter>_NHW_NL.PTT	Binary
NHO Motorized Psn. Trips (internal only)	<iter>_NHO_NL.PTT	Binary

## 20.2 Application Details

The Trip Distribution process is executed with the batch file named, Trip\_Distribution.bat. Five Cube Voyager scripts are used to carry out the process.

The first two scripts, Prepare\_Ext\_Auto\_Ends.S and Prepare\_Ext\_ComTruck\_Ends.S read the computed zonal Productions and Attraction resulting trip generation and prepares trip ends that are suitable for applying trip distributing for external Ps and As only.

The Trip\_Distribution\_External.s script executes the distribution of external trip-ends, resulting in external trip tables, by purpose. The script also calculates zonal impedances that are used in both the distribution of external and internal trips.

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

The script first prepares zonal highway terminal times, which are based on the zonal area type. The terminal times, which represent the time needed to park and un-park a vehicle, range from 1 minute in the least developed areas to 5 minutes for highly developed areas. The terminal times are then added to the over-the-network highway travel time skims. Next, composite impedance tables are developed combining transit time and highway times, based on the formula shown in Equation 2:

Equation 2 Composite time

$$CT_i = \frac{1}{\frac{1}{HT + TollT_i} + \frac{P_i}{TT}}$$

where

$CT_i$  = Composite time for income level i

$HT$  = Congested highway time (minutes), including terminal time

$TollT_i$  = Time equivalent (minutes) of tolls associated with the minimum-time path for income i

$P_i$  = Regional transit share of income i for the trip purpose

$TT$  = Metrorail-related transit time (min.), including in-vehicle and out-of-veh. time components

The basis of the  $TollT_i$  term calculation is specified in Table 59. The table indicates the average time valuation, in minutes, per year-2007 dollar, that is assigned to a toll value by income level and trip type. The table indicates, for example, that a \$1.00 toll equates to 8.7 minutes of travel time for a traveler in income level 1. More generally, the table indicates that travelers commuting to work are less sensitive to tolls than non-work-bound travelers because the time valuation of commuters is relatively high. The table also reflects the intuitive generalization that lower income travelers are more sensitive to tolls than the higher income travelers.

**Table 59 Time Valuation (Minutes/2007\$) by Purpose and Income Level**

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

Notes:

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

Prepare\_Internal\_Ends.s reads the external trip tables created above, and summarizes the trip-ends from those trip tables. It also reads the internal trip-ends from the trip generation process. The script subtracts I-X trips from the total computed trip productions (by purpose), to arrive at "final" internal trip productions. An internal trip attraction trip scaling factor is next computed. The factor is computed by purpose as:

$$IntAttrScaleFtr = ("Final" Intl Ps + Extl. Ps - Extl. As) / Intl. "Computed" As$$

The internal Ps and As in the above equation includes both motorized and non-motorized trips. A summary of the initial and final/scaled trip-ends is provided in a small text file named <iter>\_Prepare\_Internal\_Ends.txt.

Trip\_Distribution\_Internal.S: The final internal Ps and scaled As are run through trip distribution. The resulting internal trips are combined with the external trips developed above. The trip distribution process produces complete (internal and external) trip tables by purpose and produces internal-to-internal (I-I) trip tables which will be inputs to the mode choice model later in the model stream. A

complete set of jurisdictional trip tables by purpose is reported in a text file named <iter>\_Trip\_Distribution\_Internal.tab.

## 21 Mode Choice

### 21.1 Travel modes represented in the mode choice model

The mode choice model in the Version 2.3 Travel Model includes 15 choices as shown in Figure 26 and described in Chapter 6 of the Calibration Report. The model includes three auto modes (drive alone, shared ride 2 person, and shared ride 3+ person) and four transit modes (commuter rail, all bus, all Metrorail, and combined bus/Metrorail). The four transit modes are stratified by three modes of access to transit (park and ride [PNR]; kiss and ride [KNR or “ride to transit”]; and walk). As discussed in the calibration report, the definition of high-occupancy vehicle (HOV) trips has changed, compared to the definition that was used in the Version 2.2 Travel Model. Previously, HOV trips coming out of the mode choice model referred to *only those that use HOV facilities for a substantial portion of their trip*. Similarly, in previous models, the definition of low-occupancy vehicle (LOV) included both drive-alone and carpools (provided the carpools did not use a preferential HOV facility). By contrast, in the Version 2.3 NLMC model, the term LOV refers to only the drive-alone trips. Similarly, HOV refers to all shared-ride 2 (2-person carpools) and shared-ride 3 (3+ person carpools), irrespective of whether they use an HOV facility or not.

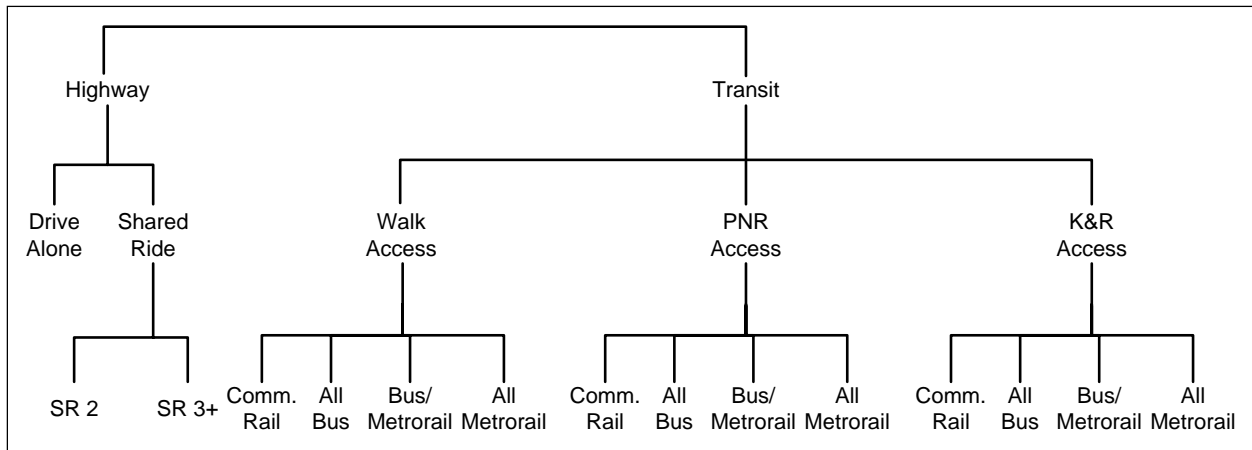


Figure 26 Nesting structure of the nested-logit mode choice model in the Version 2.3 travel model

Ref: "I:\ateam\nest\_log\NestedChoice\_Struct4.vsd"

#### 21.1.1 Treatment of LRT, BRT, and streetcar

Note that the nesting structure of the TPB Version 2.3 NLMC model does not include branches for specialized transit modes, such as light-rail transit (LRT), bus rapid transit (BRT), and streetcar. From this, one might conclude that the mode choice model is not designed to deal with these special transit modes. In fact, the model is designed to deal with these special transit modes. This section of the report discusses how these modes are treated in both the mode choice model and the transit path skimming process that feeds the mode choice model. This is the scheme that was developed by AECOM in 2004-2005 and has been retained by TPB staff. One of the underlying assumptions is that “premium” transit

modes (e.g., Metrorail, commuter rail, LRT, BRT, and streetcar) will typically travel faster than buses, since they have one or more of these characteristics:

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

#### **21.1.1.1 Network representation: LRT, BRT, and streetcar**

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

In cases where a travel demand modeler is coding a new transit line representing a “premium” transit mode,<sup>95</sup> the modeler must add “transit-only” links to the transit network to represent the new service, since the line requires a dedicated ROW which is not part of the highway network. In the past, one would have added these transit-only links to the rail link file (rail\_link.bse). However, with the advent of TPB staff using an ESRI geodatabase to manage the highway and transit networks, the rail\_link.bse file no longer exists. For a modeler working at COG, one should add transit-only links directly into the highway/transit network geodatabase. For a modeler working external to COG (who will not have access to the COGTools ArcGIS add-in for managing the geodatabase), one should modify the text \*.tb files that are output from the create\_support\_files.s Cube Voyager script.<sup>96</sup>

The “station file” (station.dbf) contains information about transit stations in the modeled area. More formally, the station file contains information about Metrorail stations, commuter rail stations, light rail stations, bus rapid transit stations/stops, streetcar stations/stops, express-bus bus stops, and park-and-ride (PNR) lots that serve these stations/stops. One must add Mode 5 and Mode 10 station nodes to the station file using a mode code of “L” for LRT/Mode 5 and “N” for New/BRT/streetcar/Mode 10. Mode 5 and 10 stations do not require a station centroid number,<sup>97</sup> though the designated node number ranges in recent network documentation has designated the following range (7000-7999).<sup>98</sup> Cube Voyager cannot combine headways for routes unless they are part of the same mode code, so, in cases where

---

<sup>95</sup> Such as Metrorail (Mode 3), commuter rail (Mode 4), LRT (Mode 5), and BRT/streetcar (Mode 10).

<sup>96</sup> Meseret Seifu to Files, “Creating a station file,” Memorandum, July 11, 2011.

<sup>97</sup> Jain to Milone and Moran, “MWCOC Network Coding Guide for Nested Logit Model (First Draft: September 20, 2007; Updated February 2008 and October 2010),” 6.

<sup>98</sup> Milone et al., “Highway and Transit Networks for the Version 2.3.57a Travel Model, Based on the 2015 CLRP and FY 2015-2020 TIP,” page 2-12.

Mode 10 routes share a street segment with local bus (Mode 1), these two routes will not be represented with a combined headway.

#### **21.1.1.2 Transit path building and skimming, mode choice, and transit assignment: LRT, BRT, and streetcar**

In transit path building and skimming, mode choice, and transit assignment, the following two rules apply:<sup>99</sup>

- LRT: Mode 5 is treated like Metrorail (Mode 3)
- BRT: Mode 10 is treated like local bus (Modes 1, 6, & 8)

#### **21.1.1.3 Fares: LRT, BRT, and streetcar**

Fares for Mode 5 and Mode 10 are computed like those for local bus (Modes 1, 6, & 8).

#### **21.1.1.4 Inclusion of LRT, BRT, and streetcar trips in trip tables**

Following the mode choice step, the output trip table files (\*.MTT) each contain 14 tables, as shown in Table 64. Any table that lists "MR" (Metrorail) actually includes both Metrorail and LRT, since Mode 5 [LRT] is treated like Metrorail (Mode 3) in transit path building/skimming, mode choice and transit assignment. Similarly, any table that lists "BU" (Bus) actually includes both bus and BRT/streetcar. The only way to get the actual breakout of the estimated level of LRT or BRT/streetcar travel is to look at the transit assignment results (keeping in mind that, although we assign all transit trips, we validated only Metrorail trips, and, at the current time, these are validated only to station groups, not to individual stations). So, after transit assignment, one is able to see how many trips/boardings/alightings occurred on a given LRT line, but, since we do not validate results at the LRT line level, model users are recommended to use caution when using these numbers.

### **21.1.2 Other issues relating to travel modes**

Table 60 list the ten transit modes that are handled by the Version 2.3 mode choice model and lists the mode code used in the station file (station.dbf), which is an input to the parker.s script that is part of the transit\_skim\_all\_modes.bat batch file (see Section 16, Transit Skim File Development). Note that the consolidated station file does not include bus stops, except for bus stops that have their own PNR lot (generally express bus service). Transit routes are represented in Cube Voyager's TRNBUILD module using the LINE command, which is usually placed in a \*.LIN file or, using COG/TPB convention, in a MODE\*.TB file (a "mode" file).

---

<sup>99</sup> Jain to Milone and Moran, "MWCOC Network Coding Guide for Nested Logit Model (First Draft: September 20, 2007; Updated February 2008 and October 2010)," 10.

Table 60 Transit sub-modes represented in the Version 2.3 travel model

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

In addition, there are five non-transit modes that are used to access transit and make transfers to, from, and between transit services. These are detailed in Table 61.

Table 61 Transit Access and Transfer Links

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

All the modes described in Table 60 and Table 61 can be used in the path-building process (see 16). If no prohibitions are imposed, path building assumes that transfers between all modes are possible. For example, a person could theoretically access Metrorail by driving (mode 11) to the station, use Metrorail (mode 3), and egress Metrorail by driving (mode 11) as well. When trips are in production-attraction format, as is the case for transit path-building and mode choice, a person cannot egress from a station and take a car. To prevent the foregoing behavior in the model, some limitations with regard to transfers need to be imposed. These are described in Table 62. The mode interchanges where transfers are prohibited are denoted by "Y".



Table 62 Transfer Prohibitions (No Transfer or NOX)

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

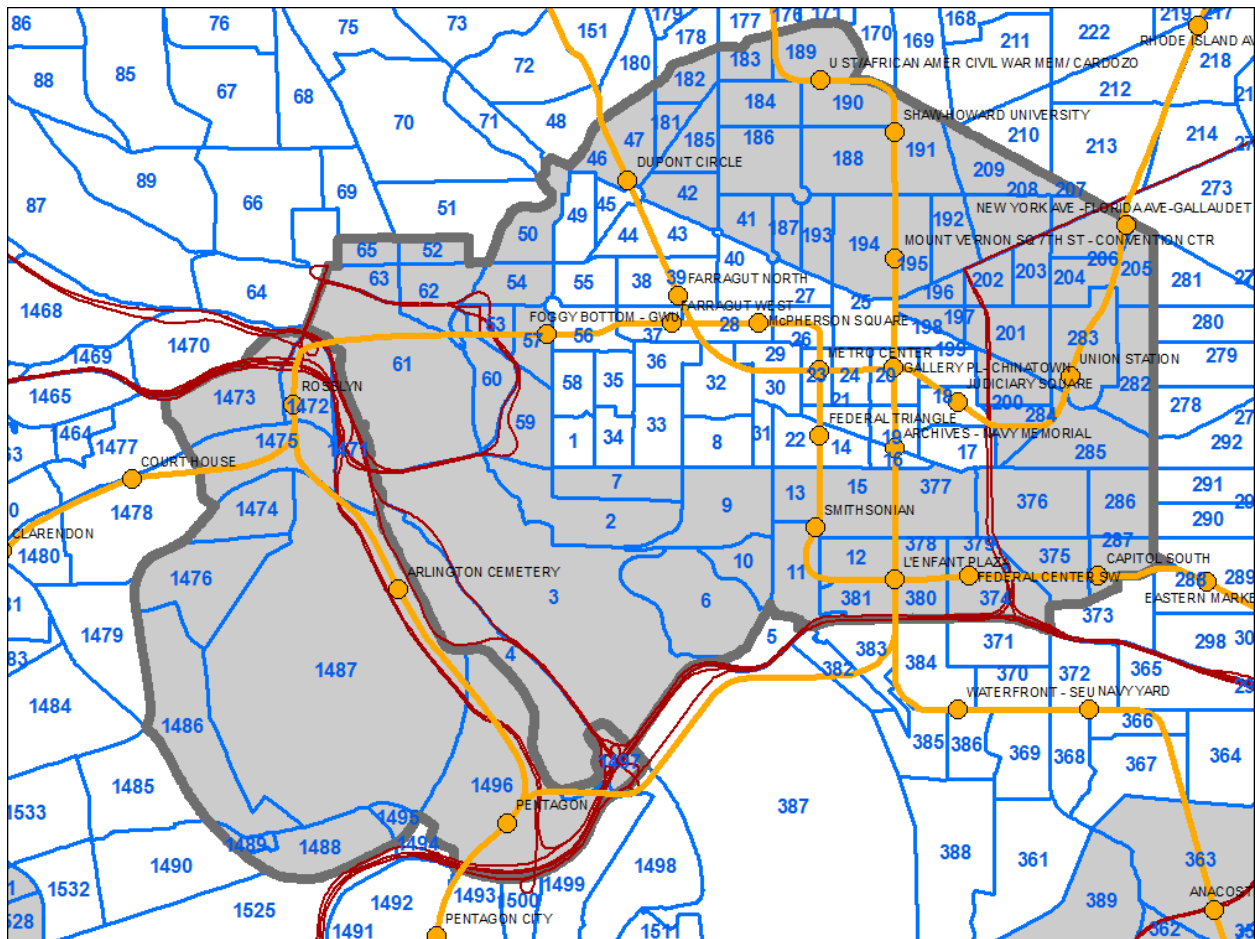
## 21.2 Metrorail constraint through the regional core

The Metrorail constraint through the regional core (sometimes referred to using the less precise term “transit constraint through the regional core”) is a technical adjustment to the trip tables coming out of the mode choice process designed to reflect a WMATA policy assumption that, during peak periods, the Metrorail system may have insufficient capacity to handle all the demand traveling to and through the regional core. Typically, it is assumed that the Metrorail system will be able to handle all of the peak-period demand to and through the regional core in the near term, but, since demand is growing through time, the system might not be able to handle all the peak-period demand at some future time, depending on the amount of growth in demand and the number of rail cars available in a given year. The assumed year at which the Metrorail system will be at its peak capacity during the peak periods to and through the regional core is known as the “binding year.” For years beyond the binding year, it is assumed that any growth in peak-period Metrorail demand to and through the regional core will be forced to switch to other travel modes (specifically, auto person trips). The Metrorail constraint was initiated by WMATA in 2000 to address funding shortfalls restricting the expansion of the rail fleet.<sup>100</sup> WMATA policy sets the binding year, which is currently set at 2020. This means that, for any forecast year past 2020, the Metrorail constraint is applied, i.e., forecasted peak-period Metrorail trips to and

<sup>100</sup> Ronald Milone, “TPB Version 2.3 Travel Model on the 3,722-TAZ area system: Status report” (presented at the September 23, 2011 meeting of the Travel Forecasting Subcommittee of the Technical Committee of the National Capital Region Transportation Planning Board, held at the Metropolitan Washington Council of Governments, Washington, D.C., September 23, 2011).

through the regional core are shifted to other travel modes (specifically, auto person trips). The regional core is defined as the set of Metrorail stations in the central employment area, i.e., the portion of the system bounded by Dupont Circle, U Street, New York Avenue (NOMA), Capital South, L'Enfant Plaza, Pentagon, Arlington Cemetery, and Rosslyn stations. This area is also sometimes referred to by technical audiences as "Ring 0" and "Ring 1." In Figure 27, Ring 0 is shown as the white area shaped like a trapezoid in the center of downtown Washington, D.C. Ring 1 is shown as the gray area surrounding Ring 0. The two areas together comprise the regional core. Note that non-Metrorail-related transit trips and off-peak Metrorail trips are not affected by the Metrorail constraint process.

Figure 27 Ring 0 (white trapezoid) and Ring 1 (gray polygon), which form the "core" area used in the Metrorail constraint through the regional core



Ref: I:\ateam\gis\taz\taz\_2191\_3722.mxd

The Metrorail constraint is applied in the following way (assuming that 2020 is the binding year). Model runs representing the binding year and years prior to the binding year are conducted in the normal fashion, i.e., using the **mode\_choice.bat** batch file (see page A-13 of Appendix A). Model runs representing any year following the binding year, e.g., 2030, are conducted using the **mode\_choice\_tc\_v23.bat** batch file (see page A-13 of Appendix A), as follows:

- Peak 2020 Metrorail trips to and through the core are estimated using a time-of-day model.

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

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

- Mode choice model application program (AEMS.EXE);
- Jurisdictional summary script (MC\_NL\_Summary.s);

By contrast, the mode choice model and Metrorail constraint process are executed using the mode\_choice\_tc\_v23.bat batch file, which invokes the following:

- Mode choice model application program (AEMS.EXE);
- Jurisdictional summary script (MC\_NL\_Summary.s);

Constraint adjustment script (mc\_constraint\_v23.s);

### 21.3 Control/Support Files

The nested-logit mode choice (NLMC) model is applied using a Fortran program called AEMS.<sup>101</sup> AEMS.EXE is the compiled version of the source code AEMS.FOR. In order to run, AEMS.EXE needs to have several DLL files. The model is run one for each of the five trip purposes, as shown on page A-13 of the flowchart in Appendix A. Each run of the mode choice model requires a “control file,” so there are five in total: HBW\_NL\_MC.CTL, HBS\_NL\_MC.CTL, HBS\_NL\_MC.CTL, NHW\_NL\_MC.CTL, and NHO\_NL\_MC.CTL. After the five mode choice models run, there is a Cube Voyager script, MC\_NL\_Summary.s, which is used to create jurisdiction-to-jurisdiction tabulations of the trip tables output from the mode choice model. The inputs to the AEMS mode choice application program are shown in Table 63. The outputs are shown in Table 64.

Table 63 Inputs to the AEMS mode choice application program

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

<sup>101</sup> “AECOM Consult Mode Choice Computation Programs, AEMS, Users Guide,” Draft report (Fairfax, Virginia: AECOM Consult, Inc., April 5, 2005).

Metrorail transit skims (INFILE 5)	trnam_mr.skm, trnop_mr.skm	Binary
Bus/Metrorail transit skims (INFILE 6)	trnam_bm.skm, trnop_bm.skm	Binary
Zonal data (INFILE 8)	zonev2.a2f	Text

Table 64 Outputs from the AEMS mode choice application program

<p>Daily person trips, stratified by travel mode (14 tables):</p> <ol style="list-style-type: none"> <li>1. DR ALONE</li> <li>2. SR2</li> <li>3. SR3+</li> <li>4. WK-CR</li> <li>5. WK-BUS</li> <li>6. WK-BU/MR</li> <li>7. WK-MR</li> <li>8. PNR-CR &amp; KNR-CR</li> <li>9. PNR-BUS</li> <li>10. KNR-BUS</li> <li>11. PNR-BU/MR</li> <li>12. KNR-BU/MR</li> <li>13. PNR-MR</li> <li>14. KNR-MR</li> </ol>	<p>hbw_nl_mc.mtt, hbs_nl_mc.mtt, hbs_nl_mc.mtt, nhw_nl_mc.mtt, nho_nl_mc.mtt</p>	Binary
---	--	--------

## 21.4 Market segmentation

Most mode choice models used in large urban areas in the U.S. have historically been estimated at a disaggregate level, but are applied at an aggregate level. Specifically, these models are typically estimated at the person-trip level, but applied at the zone-to-zone interchange level. Furthermore, in application mode, within each zone-to-zone interchange, many models subdivide the travel market into homogeneous groups, known as market segments. The nested-logit mode choice model (NLMC) that is used in the Version 2.3 Travel Model uses three types of market segmentation:

- Household income
- Geography
- Access to transit

Note that there has been a recent trend away from disaggregate estimation, due, in part to guidance from the FTA.<sup>102</sup>

### 21.4.1 Market segmentation by household income

The income segmentation is the same that is used for the first two steps of the travel model (i.e., trip generation and trip distribution), namely households are segmented by the four household income quartiles, which are shown in Table 65.<sup>103</sup>

Table 65 Household income quartiles computed from the ACS

Quartile	Income range (2007 dollars)
First	Less than \$50,000
Second	\$50,000 to \$99,999
Third	\$100,000 to \$149,999
Fourth	\$150,000 or more

### 21.4.2 Market segmentation by geography

When AECOM Consult, Inc. first developed a mode choice model for the Washington, D.C. metropolitan area in 2004-2005, it divided the modeled area into seven superdistricts:<sup>104</sup>

1. DC core
2. VA core
3. DC urban

---

<sup>102</sup> See, for example, Federal Transit Administration, "Discussion Piece #16: Calibration and Validation of Travel Models for New Starts Forecasting" (Workshop on Travel Forecasting for New Starts Proposals, Minneapolis, Minnesota, 2006), [http://www.fta.dot.gov/planning/newstarts/planning\\_environment\\_5402.html](http://www.fta.dot.gov/planning/newstarts/planning_environment_5402.html).

<sup>103</sup> Hamid Humeida to Files, "Analysis of Data from the American Community Survey (ACS): Households by Household Income, Household Size, and Vehicle Availability," Memorandum, (March 19, 2010).

<sup>104</sup> Bill Woodford, "Development of Revised Transit Components of Washington Regional Demand Forecasting Model" (Transit Modeling Meeting, held at the Metropolitan Washington Council of Governments, Washington, D.C., December 1, 2004), 30.

4. MD urban
5. VA urban
6. MD suburban
7. VA suburban

AECOM's mode choice model was applied as a post process to the COG/TPB travel model (the Version 2.1 Travel Model). COG/TPB staff used the AECOM post-process mode choice model as a starting point for its work on the Version 2.3 Travel Model in work done from 2008 to 2011. TPB staff integrated the mode choice model into the modeling chain (i.e., moved from a post process for the regional model to its normal position in the speed feedback loop, following trip distribution), and re-calibrated the model.

**When COG/TPB staff retained and re-calibrated the NLMC model, it retained the same geographic market segmentation that had been developed by AECOM.**

These seven superdistricts are shown in Figure 28 and in Table 66. Table 66 shows the equivalency between the seven NLMC superdistricts and the new 3,722-TAZ area system.

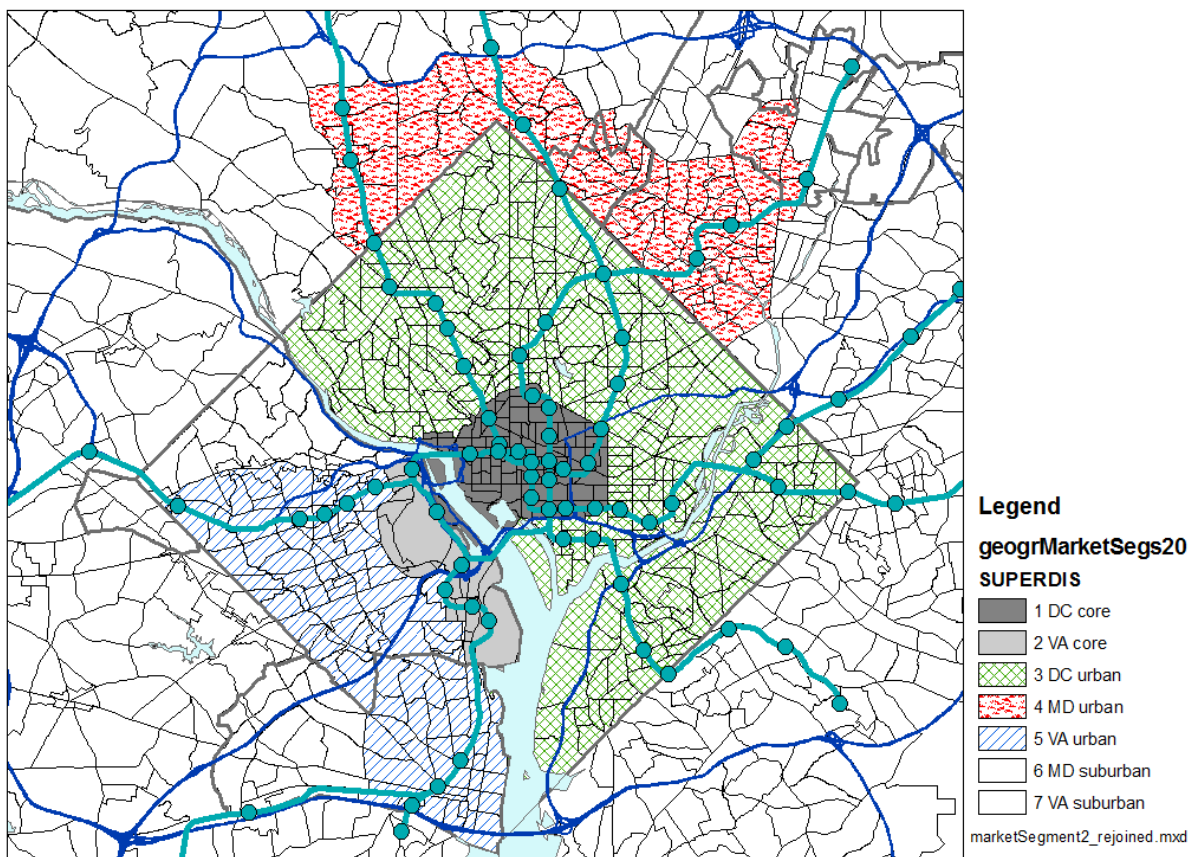


Figure 28 Seven superdistricts used in the Version 2.3 nested-logit mode choice model

Ref: "I:\ateam\nest\_log\marketsegment2\_rejoined.tif"

Table 66 Equivalency between nested-logit mode choice superdistricts and TPB TAZ 3,722

No.	Name	TAZs (TPB TAZ 3,722)
1	DC core	1-4,6-47,49-63,65,181-287,374-381
2	VA core	1471-1476,1486-1489,1493,1495-1504,1507,1508,1510,1511
3	DC urban	5,48,51,64,66-180,210-281,288-373,382-393
4	MD urban	603,606,612-628,630-640,662-664,669,670,913,916,917,939-957,959,961-982,985,
4	MD urban	986
5	VA urban	1405-1422,1427-1435,1448,1452,1454-1464,1477-1485,1490-1492,1494,1505,1506,
5	VA urban	1509,1512-1545,1569-1609
6	MD suburban	394-602,604,605,607-611,629,641-661,665-668,671-912,914,915,918-938,958,960,
6	MD suburban	983,984,987-1404,2820-3102,3104-3409
7	VA suburban	1423-1426,1436-1447,1449-1451,1453,1465-1470,1546-1568,1610-2554,2556-2628,
7	VA suburban	2630-2819,3410-3477,3479-3481,3483-3494,3496-3675

Ref: "I:\ateam\neat\_log\equiv\_tpbTaz3722\_nlmc\_superdistr.txt" and "I:\ateam\neat\_log\Market\_segment\_NewTAZs\_sorted.xlsx"

The TAZs in Table 66 are referred to as “TPB TAZ” to distinguish them from “COG TAZ.” In 2008 and 2009, the COG GIS staff developed a new system of transportation analysis zones (TAZs), which had more zones, but did not increase the size of the modeled area. In other words, the new zones were, on average, smaller than the previous zone system, which is useful for better modeling of transit trips. The old zone system had 2,191 TAZs and the new system has 3,722 TAZs. After the COG GIS staff was finished with their work, the COG models development group reviewed the new zone system and found a few cases where the zone boundaries needed adjustment.<sup>105</sup> The final result was that there were now two sets of zones for the 3,722-TAZ area system:

- COG TAZs: For land activity forecasts (COGTAZ3722\_TPBMOD)
- TPB TAZs: For transportation modeling (TPBTAZ3722\_TPBMOD)<sup>106</sup>

Although seven market areas could lead to 49 (= 7 x 7) geographic interchanges, AECOM Consult, Inc. grouped them into the 20 paired production/attraction areas shown in Table 67 and Table 68. Another way to view the 20 geographic market segments is shown in Table 69.

<sup>105</sup> Meseret Seifu, “Review of New Zone System: 3722 Transportation Analysis Zones (TAZ)” (January 22, 2010 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., January 22, 2010), 4, <http://www.mwcog.org/uploads/committee-documents/ZI5aV1dd20100122152445.pdf>.

<sup>106</sup> Ibid., 23.



**Table 67 Production and attraction market segments used in the TPB Version 2.3 NLMC model**

<b>Production Areas</b>	<b>Attraction Areas</b>
1. DC Core / Urban	1. DC Core
2. MD Urban	2. VA Core
3. VA Core / Urban	3. Urban
4. MD Suburban	4. Suburban
5. VA Suburban	

Ref: "I:\ateam\nest\_log\marketSeg.xls"

**Table 68 20 geographic market segments used in the TPB nested-logit mode choice model**

<b>Market Seg No.</b>	<b>Prod Superdis</b>	<b>Attr Superdis</b>	<b>Production Area</b>	<b>Attraction Area</b>
1	1,3	1	DC	DC core
2	1,3	2	DC	VA core
3	1,3	3,4,5	DC	Urban DC, MD, VA
4	1,3	6,7	DC	Suburban MD, VA
5	4	1	MD urban	DC core
6	4	2	MD urban	VA core
7	4	3,4,5	MD urban	Urban DC, MD, VA
8	4	6,7	MD urban	Suburban MD, VA
9	2,5	1	VA core/urban	DC core
10	2,5	2	VA core/urban	VA core
11	2,5	3,4,5	VA core/urban	Urban DC, MD, VA
12	2,5	6,7	VA core/urban	Suburban MD, VA
13	6	1	MD suburban	DC core
14	6	2	MD suburban	VA core
15	6	3,4,5	MD suburban	Urban DC, MD, VA
16	6	6,7	MD suburban	Suburban MD, VA
17	7	1	VA suburban	DC core
18	7	2	VA suburban	VA core
19	7	3,4,5	VA suburban	Urban DC, MD, VA
20	7	6,7	VA suburban	Suburban MD, VA

Ref: "I:\ateam\nest\_log\marketSeg.xls"

**Table 69 Equivalency between seven super-districts and the 20 geographic market segments**

	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>
	<b>DC core</b>	<b>VA core</b>	<b>DC urban</b>	<b>MD urban</b>	<b>VA urban</b>	<b>MD suburban</b>	<b>VA suburban</b>
<b>1 DC core</b>	1	2	3	3	3	4	4
<b>3 DC urban</b>	1	2	3	3	3	4	4
<b>4 MD urban</b>	5	6	7	7	7	8	8
<b>2 VA core</b>	9	10	11	11	11	12	12
<b>5 VA urban</b>	9	10	11	11	11	12	12
<b>6 MD suburban</b>	13	14	15	15	15	16	16
<b>7 VA suburban</b>	17	18	19	19	19	20	20

Ref: "I:\ateam\nest\_log\superDistr\_marketSeg.xlsx"



### 21.4.3 Market segmentation by access to transit

The section of the report contains two subsections. The first includes a general discussion about how transit-access markets are developed in relatively simple mode choice models. It gives the example of the three transit access markets that are often used by the Federal Transit Administration (FTA): “can walk,” “must drive,” and “no transit.”<sup>107</sup> The second subsection describes the more specific case of the seven transit-access markets used in the mode choice model of the Version 2.3 Travel Model. In both the general discussion and the more specific case, zonal percent-walk-to-transit (PWT) values are used to develop the transit access markets. In the latter case, the mode choice model application program is AEMS.EXE, developed by AECOM.

#### 21.4.3.1 General discussion

The purpose of a mode choice model is to predict the number and or share of trips that will be made by each major travel mode represented in a model. Transit, in one form or another, is usually one of the travel modes represented in most mode choice models. In order to use transit, one must be able to access it, either via non-motorized modes, such as walking and biking, or motorized modes, such as driving an automobile. Many mode choice models segment transit trips by walk access and drive access. A typical zonal metric for how easily one may walk to transit is the “percent walk to transit” (PWT) value, which is defined as the percent of a zone’s area that is within walking distance to transit service. So, for example, a PWT value of 20% means that 20% of the zone’s area lies within walking distance to transit service. If walking distance has been defined to be one mile, then this means that 20% of the zone lies within one mile of transit service. The walking distance threshold is set by the modelers in each urban area and should reflect the typical distance that people are likely to walk to reach transit. Typical values range from 0.5 miles to 1 mile. Some travel models, such as the TPB Version 2.3 Travel Model, make use of two walk-to-transit threshold distances, e.g., a short-walk distance (e.g., 0.5 miles) and a long-walk distance (e.g., 1 mile). The TPB travel model is discussed in the next section of the report. For this section of the report, it is assumed that there is only one walk-to-transit threshold distance (e.g., 1 mile).

A typical method for calculating the percent walk to transit for each zone in the modeled area is the following:

1. Determine a threshold distance for walking to transit (or two threshold distances may be used).
2. Determine point locations where transit service can be accessed (i.e., transit stop nodes and transit stations). In other words, create a geographic data set that includes all the points representing transit stop nodes and transit stations.
3. Determine transit walksheds, which are polygons composed of circular areas around transit stop nodes. In other words, create a geographic data set that represents point buffers (i.e., circles of radius  $X$  = the threshold walking distance) around each transit stop node and transit station.

---

<sup>107</sup> See, for example, Federal Transit Administration, “Discussion Piece #11: Illustrative Mode-Choice and Summit Calculations for Travel by One Market Segment between a Pair of Zones for Base and Build Alternatives” (Workshop on Travel Forecasting for New Starts Proposals, Minneapolis, Minnesota, 2006), [http://www.fta.dot.gov/planning/newstarts/planning\\_environment\\_5402.html](http://www.fta.dot.gov/planning/newstarts/planning_environment_5402.html).

4. Given that there is already a polygon layer of TAZ boundaries, perform a polygon-on-polygon overlay (TAZ boundaries and walkshed boundaries) to create a new geographic data set that can be used to calculate the percent walk to transit value for each zone.
5. Calculate the percent walk to transit values for each zone.<sup>108</sup>

For a number of years, the Federal Transit Administration (FTA) has used a simple transit-access market segmentation system that has three segments known as “can walk,” “must drive,” and “no transit.” These segments are defined at the zone-to-zone interchange level (the level used by most mode choice application programs) and can be determined using the percent-walk-to-transit (PWT) values in the production and attraction zones of the interchange. Before defining these three transit access markets, one must make a few assumptions:

1. Trips are in production/attraction format, not origin/destination format:
  - a. A trip **production** is defined as the **home**-end of a home-based trip, or the **origin** of a non-home-based trip.
  - b. A trip **attraction** is defined as the **non-home**-end of a home-based trip, or the **destination** of a non-home-based trip.
2. Travelers “access” transit at the production end of the trip and “egress from” transit at the attraction end of the trip.
3. At the production end of the trip, one may access the transit system by either walking or driving. Bike access is considered part of “walking.”
4. At the attraction end of the trip, the only egress option is walking, since it is assumed that travelers do not have an automobile available at the non-home end of the trip.
5. The zonal PWT value functions as a probability value. Thus, if the PWT is 20%, this can be interpreted as meaning that, for trips that start (are produced in) or end (are attracted to) this zone, there is a 20% chance of that the trips will access or egress from the transit system via walking.

The “can walk” market is defined as the set of trips, within a given zone-to-zone interchange, where one can walk to transit at the production end of the trip (One can also walk from transit at the attraction end of the trip, but this is not a distinguishing feature, since “must drive” trips also walk from transit at the attraction end of the trip). Even though a trip may be included in the “can walk” segment, it is understood that drive access to transit is also a possibility for this market. In probability theory, if two events, A and B, are independent, the probability of the intersection of A and B equals the product of the probabilities of A and B, i.e.,

$$P(AB) = P(A) * P(B)$$

---

<sup>108</sup> See, for example, Yew Yuan, “Transit Walkshed Generator: A GIS Application to Generate Transit Walksheds, Technical Report,” Draft (Washington, D.C.: Metropolitan Washington Council of Governments, National Capital Region Transportation Planning Board, November 15, 2012).

Since the PWT is considered a probability or likelihood of walking, and since the PWT for two given zones are considered to be independent, then, for a given zone-to-zone interchange, the probability of being in the “can walk” market --  $P(\text{“can walk”})$  or  $P(CW)$  -- is simply the product of the PWT of the production zone and the PWT of the attraction zone:

$$P(\text{“can walk” for interchange } ij) = PWT(i) * PWT(j)$$

The “must drive” market includes trips that must access the transit market via driving since the trip begins outside of the transit walk-access threshold distance. The “no transit” market includes trips for which transit is not an option, since, at the attraction end of the trip, there is no transit available within walking distance. So, for a given interchange, the probability of being in the “must drive” market --  $P(\text{“must drive”})$  or  $P(MD)$  -- is simply the product of the non-walkable share of the production zone and the PWT of the attraction zone:

$$P(\text{“must drive” for interchange } ij) = (1 - PWT(i)) * PWT(j)$$

Similarly, for a given interchange, the probability of being in the “no transit” market --  $P(\text{“no transit”})$  or  $P(NT)$  -- is simply the non-walkable share of the attraction zone:

$$P(\text{“no transit” for interchange } ij) = (1 - PWT(j))$$

So, whereas the  $P(CW)$  and  $P(MD)$  are a function of the PWT in both the production and attraction zones, the  $P(NT)$  is a function of only the PWT in the attraction zone. For a given interchange

$$P(CW) + P(MD) + P(NT) = 100\%$$

Table 70 presents 11 examples, or cases, of how various production and attraction PWT values are combined to get the probabilities of being in the “can walk,” “must drive” and “no transit” zone-to-zone interchange market segments. For example, in the case #1, both the production zone and the attraction zone have percent-walk-to-transit (PWT) values of 0%, which results in the all the trips in the interchange being in the “no transit” market segment. By contrast, in case #2,  $PWT(i) = 0\%$  and  $PWT(j) = 50\%$ , which results in a 50%/50% split of trips in that interchange into the “must drive” and “no transit” markets. When, in case #3,  $PWT(i) = 0\%$  and  $PWT(j) = 100\%$ , this results in all trips being allocated to the “must drive” market.

In any of these cases, the number of trips in each of the three markets is equal to the total number of person trips in the zone-to-zone interchange times each of the three probabilities. **After trips have been assigned to the three markets, then the mode choice model is applied**, as described FTA's Discussion Piece #11 (Discussion\_11\_Summit\_Calcs.doc) and shown in its associated spreadsheet (Discussion\_11\_Summit\_Example\_Calcs.xls).<sup>109</sup>

---

<sup>109</sup> Federal Transit Administration, “Discussion Piece #11: Illustrative Mode-Choice and Summit Calculations for Travel by One Market Segment between a Pair of Zones for Base and Build Alternatives.”

Table 70 Eleven examples showing how zonal percent-walk-to-transit values translate into probabilities of being in three transit-access markets: can walk, must drive, and no transit

	Zonal Attributes		Zone-to-Zone Interchange Attributes			Total Probability
	Percent Walk to Transit Prod. Zone	Percent Walk to Transit Attr. Zone	Probability "Can Walk"	Probability "Must Drive"	Probability "No Transit"	
	PWT(i)	PWT(j)	P(CW,ij)	P(MD,ij)	P(NT,ij)	
Case	A	B	A*B	(1-A)*B	(1-B)	
1	0%	0%	0.0%	0.0%	100.0%	100.0%
2	0%	50%	0.0%	50.0%	50.0%	100.0%
3	0%	100%	0.0%	100.0%	0.0%	100.0%
4	50%	0%	0.0%	0.0%	100.0%	100.0%
5	50%	50%	25.0%	25.0%	50.0%	100.0%
6	50%	100%	50.0%	50.0%	0.0%	100.0%
7	100%	0%	0.0%	0.0%	100.0%	100.0%
8	100%	50%	50.0%	0.0%	50.0%	100.0%
9	100%	100%	100.0%	0.0%	0.0%	100.0%
10	75%	50%	37.5%	12.5%	50.0%	100.0%
11	100%	75%	75.0%	0.0%	25.0%	100.0%

Ref: "percent\_walk\_transit\_can\_walk.xlsx"

Table 71, Table 72, and Table 73 provide a more complete picture of how P(CW), P(MD), and P(NT) each vary with the production and attraction PWT values. For example, Table 71 shows the probability of being in the “can walk” market segment for a zone-to-zone interchange as a function of the production and attraction percent-walk-to-transit values. The probability of “can walk” is zero if either the production PWT or the attraction PWT equal zero. By contrast, the probability of “can walk” is 100% only if the production PWT and the attraction PWT equal 100%.

**Table 71 Probability of being in the “can walk” market segment for a zone-to-zone interchange, based on the production and attraction percent-walk-to-transit values**

		PWT(i)										
		0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
PWT(j)	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	10%	0%	1%	2%	3%	4%	5%	6%	7%	8%	9%	10%
	20%	0%	2%	4%	6%	8%	10%	12%	14%	16%	18%	20%
	30%	0%	3%	6%	9%	12%	15%	18%	21%	24%	27%	30%
	40%	0%	4%	8%	12%	16%	20%	24%	28%	32%	36%	40%
	50%	0%	5%	10%	15%	20%	25%	30%	35%	40%	45%	50%
	60%	0%	6%	12%	18%	24%	30%	36%	42%	48%	54%	60%
	70%	0%	7%	14%	21%	28%	35%	42%	49%	56%	63%	70%
	80%	0%	8%	16%	24%	32%	40%	48%	56%	64%	72%	80%
	90%	0%	9%	18%	27%	36%	45%	54%	63%	72%	81%	90%
100%	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%	

Ref: "percent\_walk\_transit\_can\_walk.xlsx"

Table 72 shows the probability of being in the “must drive” market segment for a zone-to-zone interchange as a function of the production and attraction percent-walk-to-transit values. The probability of “must drive” is zero if either the production PWT equals 100% or the attraction PWT equal zero. By contrast, the probability of “must drive” is 100% only if the production PWT equals zero and the attraction PWT equals 100%.

**Table 72 Probability of being in the “must drive” market segment for a zone-to-zone interchange, based on the production and attraction percent-walk-to-transit values**

		PWT(i)										
		0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
PWT(j)	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	10%	10%	9%	8%	7%	6%	5%	4%	3%	2%	1%	0%
	20%	20%	18%	16%	14%	12%	10%	8%	6%	4%	2%	0%
	30%	30%	27%	24%	21%	18%	15%	12%	9%	6%	3%	0%
	40%	40%	36%	32%	28%	24%	20%	16%	12%	8%	4%	0%
	50%	50%	45%	40%	35%	30%	25%	20%	15%	10%	5%	0%
	60%	60%	54%	48%	42%	36%	30%	24%	18%	12%	6%	0%
	70%	70%	63%	56%	49%	42%	35%	28%	21%	14%	7%	0%
	80%	80%	72%	64%	56%	48%	40%	32%	24%	16%	8%	0%
	90%	90%	81%	72%	63%	54%	45%	36%	27%	18%	9%	0%
100%	100%	90%	80%	70%	60%	50%	40%	30%	20%	10%	0%	

Ref: "percent\_walk\_transit\_can\_walk.xlsx"

Table 73 shows the probability of being in the “no transit” market segment for a zone-to-zone interchange as a function of solely on the attraction percent-walk-to-transit values. The probability of “no transit” is zero only if the attraction PWT equals 100%. By contrast, the probability of “no transit” is 100% only if attraction PWT equals zero.

**Table 73 Probability of being in the “no transit” market segment for a zone-to-zone interchange, based solely on the attraction percent-walk-to-transit values**

		PWT(i)										
		0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
PWT(j)	0%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
	10%	90%	90%	90%	90%	90%	90%	90%	90%	90%	90%	90%
	20%	80%	80%	80%	80%	80%	80%	80%	80%	80%	80%	80%
	30%	70%	70%	70%	70%	70%	70%	70%	70%	70%	70%	70%
	40%	60%	60%	60%	60%	60%	60%	60%	60%	60%	60%	60%
	50%	50%	50%	50%	50%	50%	50%	50%	50%	50%	50%	50%
	60%	40%	40%	40%	40%	40%	40%	40%	40%	40%	40%	40%
	70%	30%	30%	30%	30%	30%	30%	30%	30%	30%	30%	30%
	80%	20%	20%	20%	20%	20%	20%	20%	20%	20%	20%	20%
	90%	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%
100%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	

Ref: "percent\_walk\_transit\_can\_walk.xlsx"

### 21.4.3.2 Version 2.3 Travel Model and AEMS

Regarding the percent-walk-to-transit (PWT) values used by the mode choice model of the TPB Version 2.3 Travel Model, two distance thresholds are used:

- Short walk to transit: ≤ 0.5 mile
- Long walk to transit: > 0.5 mile and ≤ 1 mile

Furthermore, the mode choice model differentiates between peak period transit service and off-peak period transit service. When calculating average headways and run times for transit routes running during the peak and off-peak periods, the historical practice, which is continued to this day, has been to use a subset of the period to represent service during the entire period. Specifically, the one-hour time period from 7:00 AM to 7:59 AM is used to represent peak-period conditions, and the five-hour time period from 10:00 AM to 2:59 PM is used to represent off-peak-period conditions.<sup>110</sup> It is also assumed that home-based-work (HBW) trips occur in the peak periods, and thus make use of the peak-period transit skims and peak-period PWT values. Similarly, it is assumed that the other trip purposes (HBO, HBS, NHW, and NHO) occur in the off-peak periods, and thus make uses of the off-peak transit skims and off-peak PWT values.

The mode choice model in the TPB Version 2.3 Travel Model is a 15-choice, nested-logit mode choice (NLMC) model that includes

- Three auto modes (drive alone [DA], shared ride 2-person [SR2], and shared ride 3+person [SR3]);

<sup>110</sup> Robert Snead et al., “FY-2010 Network Documentation: Highway and Transit Network Development,” Draft (Washington, D.C.: National Capital Region Transportation Planning Board, June 30, 2010), 3.

- Four transit modes (commuter rail [CR], all bus [AB], all Metrorail [MR], and combined bus/Metrorail[BM]); and
- Three modes of access to transit (park and ride [PNR], kiss and ride [KNR], and walk [WK])

These 10 modes are combined in nests, in such a way that there are 15 choices in the mode choice model, as shown in

Figure 29.

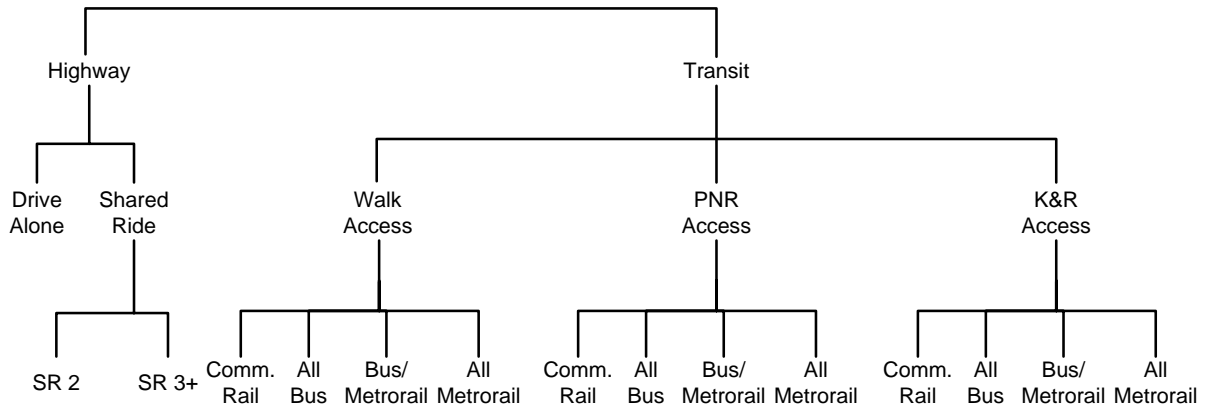


Figure 29 Nesting structure of the nested-logit mode choice model in the Version 2.3 Travel Model

Ref: "I:\ateam\nest\_log\NestedChoice\_Struct4.vsd"

The NLMC model is applied using the AECOM mode choice application program (AEMS).

Although light-rail transit (LRT), bus rapid transit (BRT), and streetcar are not explicit transit modes in the mode choice model, the model has, nonetheless, been designed to deal with these three special transit modes. Mode 5 is reserved for modeling LRT. Mode 10 is reserved for modeling BRT and streetcar. It is assumed that Mode 5 (LRT) will travel mostly on its own, dedicated right of way. By contrast, it is assumed that Mode 10 (BRT and streetcar) will travel mostly in mixed traffic on a shared right of way. Full details of how these three transit modes are modeled can be found in either the calibration report<sup>111</sup> or in section 21.1.1 of this report, but one of the key assumptions is the following:

- For transit path building/skimming, mode choice, and transit assignment
  - Mode 5 (LRT) is treated like Mode 3 (Metrorail)
  - Mode 10 (BRT or streetcar) is treated like Mode 1 (local bus)

When AECOM first developed the nested-logit mode choice model that TPB staff later adopted and recalibrated, AECOM used six percent-walk-to-transit values:<sup>112</sup>

- Percent of the zone within a short walk to Metrorail (Mode 3): PSWMET

<sup>111</sup> Milone et al., “Calibration Report for the TPB Travel Forecasting Model, Version 2.3,” 6-3-5.

<sup>112</sup> AECOM Consult, Inc., “Post MWCOG – AECOM Transit Component of Washington Regional Demand Forecasting Model: User’s Guide” (AECOM Consult, Inc., March 2005), 11.

- Percent of the zone within a long walk to Metrorail (Mode 3): PLWMET
- Percent of the zone within a short walk to any transit in the AM peak period: PSWALLAM
- Percent of the zone within a long walk to any transit in the AM peak period: PLWALLAM
- Percent of the zone within a short walk to any transit in the off-peak period: PSWALLOP
- Percent of the zone within a long walk to any transit in the off-peak period: PLWALLOP

However, TPB staff has recently become aware, thanks to work done by Dusan Vuksan and Feng Xie in September 2012, that **the first two PWT values should include both Metrorail and LRT, not simply Metrorail**. This oversight had not been noticed before, since 1) LRT was not part of the base-year (year-2007) calibration networks, and 2) when LRT was modeled in close-in areas that already had significant transit service, the omission of LRT PWT values from the Metrorail/LRT group was hard to detect. However, in the work conducted by Dusan and Feng, the LRT service was in suburban areas without significant surrounding transit service, and it became apparent that the model was underestimating LRT ridership. The net effect is that the first two zonal PWT values now include both Metrorail and LRT together:

- Percent of the zone within a short walk to Metrorail (Mode 3) **or LRT (Mode 5)**: PSWMET
- Percent of the zone within a long walk to Metrorail (Mode 3) **or LRT (Mode 5)**: PLWMET
- Percent of the zone within a short walk to any transit in the AM peak period: PSWALLAM
- Percent of the zone within a long walk to any transit in the AM peak period: PLWALLAM
- Percent of the zone within a short walk to any transit in the off-peak period: PSWALLOP
- Percent of the zone within a long walk to any transit in the off-peak period: PLWALLOP

“Any transit” includes all transit, including Metrorail and LRT service.

In the past, TPB staff has developed a set of procedures for creating transit walksheds and calculating PWT values.<sup>113</sup> Now, given the recent realization that the term “Metrorail” should actually refer to both “Metrorail and LRT,” COG/TPB staff has updated the documentation to indicate this definitional change.<sup>114</sup> These new definitions have now been used in the automated ArcPy transit walkshed process that is part of the Ver. 2.3.57 model and later models, such as Ver. 2.3.66.

There are two other assumptions governing the use of the six PWT values that need to be kept in mind. The first is definitional and the second relates to differentiating between peak-period and off-peak-

---

<sup>113</sup> See, for example, Yew Yuan, “Walk Shed Generator: A GIS Application To Generate Transit Walk Sheds, Technical Report,” Draft (Washington, D.C.: Metropolitan Washington Council of Governments, National Capital Region Transportation Planning Board, May 2006); Yew Yuan, “Walkshed Generator User Guide” (Washington, D.C.: National Capital Region Transportation Planning Board, August 25, 2010); Mark S. Moran, “Creating Transit Stop Node Shapefiles Needed as Inputs to the TPB Procedure for Generating Transit Walksheds and Calculating the Share of Each Zone within Walking Distance to Transit,” Memorandum, (September 6, 2011).

<sup>114</sup> Yuan, “Transit Walkshed Generator: A GIS Application to Generate Transit Walksheds, Technical Report”; Yew Yuan, “Walkshed Generator User Guide” (Washington, D.C.: Metropolitan Washington Council of Governments, National Capital Region Transportation Planning Board, November 16, 2012); Mark S. Moran to Files, “Creating Transit Stop Node Shapefiles Needed as Inputs to the TPB Procedure for Generating Transit Walksheds and Calculating the Share of Each Zone within Walking Distance to Transit,” Memorandum, (November 16, 2012).



period transit service. Regarding the definitional difference, when AECOM first developed the percent walk values, it defined them based on zonal areas:

$$\text{Percent} = \frac{\text{walkshed area}}{\text{total zonal land area}}$$

As an example, if a zone has half of its land area in the short-walk-to-Metrorail area and half of its land area in the long-walk-to-Metrorail area, one might expect that PSWMET = 50% and the PLWMET = 50%. However, the real PWT values for this scenario would be PSWMET = 50% and the PLWMET = 100%, since the short-walk area is always contained within the long-walk area. Consequently, if one wants the net area that is in the long walk area, one must subtract the two areas:

$$\text{Net Percent Long Walk} = (\text{Percent long walk}) - (\text{Percent short walk})$$

Evidence of this will be seen in later calculations discussed in this report.

The second assumption about PWT values relates to the coverage of transit service in the peak period versus in the off-peak period. It is assumed that transit service is accessed at the transit stop nodes (e.g., bus stops) and transit stations. In the case of Metrorail, there are no examples of stations that operate in the peak period, but do not operate in the off-peak period. Instead, all stations operate in all periods, even though the frequency of service changes (peak versus off-peak), and there are some segments that exist in the off-peak but not in the peak (e.g., in 2006, WMATA began running the Yellow Line from Gallery Place to Fort Totten, but only in the off-peak). However, since the transit walkshed buffers are drawn around points, and not segments, this does not affect Metrorail, meaning that the percent-walk-to-transit values need not be calculated separately for peak and off-peak Metrorail. Furthermore, now that we are including LRT with Metrorail for determining walksheds and calculating PWT values, it is also assumed that there is no difference between LRT stations operating in the peak periods and those in the off-peak.

Finally, transit access markets are determined within the mode choice application program (AEMS) by combining information from the six PWT values already discussed. AEMS is a compiled Fortran program, which requires a control file (\*.CTL) for each mode choice model. The Version 2.3 Travel Model uses five mode choice models (one per trip purpose), and so it requires five control files (e.g., HBW\_NL\_MC.CTL, HBS\_NL\_MC.CTL, HBO\_NL\_MC.CTL, etc.). Percent-walk-to-transit values are stored in a zonal data file (ZONEV2.A2F) that is read into AEMS.

**The remainder of this section of the report draws heavily from a memo from AECOM staff to COG/TPB staff.<sup>115</sup>**

In each of the AEMS control files, the six PWT values are referenced using the following 4-character pattern:

---

<sup>115</sup> David Roden to Mark S. Moran, "Memorandum for Task Order 7 (FY13 Task 1) of COG Contract 12-006, Interpreting AEMS Market Shares," Memorandum, (September 24, 2012).

<production or attraction indicator (1 char)><file number (1 char)><table number (2 char)>

The production or attraction zone status is indicated using the letter “i” (production”) or “j” (attraction). In the current AEMS control files, the file number for the zonal data file is “8.” Using current modeling conventions, the table numbers for the percent-walk-to-transit values go from 7 to 12 (and this information is noted in comment records in the AEMS control files). Thus, the following 3-digit codes refer to the six PWT values:

- 807: Percent of the zone within a short walk to Metrorail (Mode 3) or LRT (Mode 5): PSWMET
- 808: Percent of the zone within a long walk to Metrorail (Mode 3) or LRT (Mode 5): PLWMET
- 809: Percent of the zone within a short walk to any transit in the AM peak period: PSWALLAM
- 810: Percent of the zone within a long walk to any transit in the AM peak period: PLWALLAM
- 811: Percent of the zone within a short walk to any transit in the off-peak period: PSWALLOP
- 812: Percent of the zone within a long walk to any transit in the off-peak period: PLWALLOP

For example, if the control file refers to “i807”, this means the percent of the zone within a short walk to Metrorail or LRT for production zone “i”.

The aforementioned six percent-walk-to-transit values define the percentage of the zonal area that is within walking distance to transit, but they do not indicate the share of productions or attractions are assumed to walk. For example, not all transit trips that begin in a long-walk area will actually end up walking to transit (some will drive access). Consequently, the next step in the process is to calculate six values representing the likely walk-access markets. To do this, two assumptions are made:

1. 100% of the transit trips beginning or ending in the short-walk area will access transit via walking;
2. Only 25% of the transit trips beginning or ending in the long-walk area will access transit via walking (i.e., 75% are assumed to use drive access).

The six assumed walk markets are the following:

PCMI	Percent of trips assumed to access Metrorail/LRT via walking at the production zone
PCMJ	Percent of trips assumed to access Metrorail/LRT via walking at the attraction zone
PCTIAM	Percent of trips assumed to access all transit via walking at the production zone, AM peak per.
PCTJAM	Percent of trips assumed to access all transit via walking at the attraction zone, AM peak per.
PCTIOP	Percent of trips assumed to access all transit via walking at the production zone, off-peak per.
PCTJOP	Percent of trips assumed to access all transit via walking at the attraction zone, off-peak per.

Percent-walk-to-transit values are calculated using point buffers around transit stop nodes (i.e., stations, bus stops, etc.). As is the case with the original percent-walk-to-transit values, it is assumed that Metrorail and LRT service, in terms of stations in service, does not vary by time of day. By contrast, it is assumed that time-of-day variations in other transit modes, such as bus or commuter rail, will mean that the set of AM stop nodes will be different from the off-peak stop nodes. For this reason, there are two

sets of percent-walk-to-transit values for all transit (one for AM and one for off peak), but only one for Metrorail and LRT.

For each of the five mode choice models (HBW, HBS, HBO, NHW, NHO), only four of these values are used at once (HBW gets AM and the other purposes get off-peak):

- PCMI: Percent of trips assumed to access Metrorail/LRT via walking at the production zone
- PCMJ: Percent of trips assumed to egress from Metrorail/LRT via walking at the attraction zone
- PCTI: Percent of trips assumed to access all transit via walking at the production zone
- PCTJ: Percent of trips assumed to egress from all transit via walking at the attraction zone

In all four cases, the following is assumed:

Percent of trips in the interchange assumed to be in one of the four categories =

$$(100\% \text{ of the trips in the short-walk area}) + (25\% \text{ of the trips in the long-walk area})$$

In terms of equations in the mode choice control files, one finds:

- PCMI: Percent of trips assumed to access Metrorail/LRT via walking at the production zone
  - $= (i807 + 0.25 * (i808 - i807)) / 100$
- PCMJ: Percent of trips assumed to egress from Metrorail/LRT via walking at the attraction zone
  - $= (j807 + 0.25 * (j808 - j807)) / 100$
- PCTI: Percent of trips assumed to access “all transit” via walking at the production zone
  - $= (i809 + 0.25 * (i810 - i809)) / 100$  for AM (used for HBW purpose)
  - $= (i811 + 0.25 * (i812 - i811)) / 100$  for off-peak (used for non-work purposes)
- PCTJ: Percent of trips assumed to egress from “all transit” via walking at the attraction zone
  - $= (j809 + 0.25 * (j810 - j809)) / 100$  for AM (used for HBW purpose)
  - $= (j811 + 0.25 * (j812 - j811)) / 100$  for off-peak (used for non-work purposes)

A distinction is drawn between Metrorail/LRT and “other transit” (i.e., transit that is neither Metrorail nor LRT). In the “can walk” market, there are four sub-markets, as shown in Table 74. Similarly, in the “must drive” market, there are two sub-markets, as shown in Table 75. Lastly, there is the “no access to transit” market, which is not part of either table.

Table 74 Four “can walk” sub-markets

Sub-mkt	Transit Service Available		Description
	Production TAZ	Attraction TAZ	
WM	MR/LRT	MR/LRT	Share that can use MR/LRT at both ends of the trip
W1	Other transit	MR/LRT	Share that can use “other transit” at production end and MR/LRT at attraction end of the trip
W2	Other transit	Other transit	Share that can use “other transit” at both ends of the trip
W3	MR/LRT	Other transit	Share that can use MR/LRT at production end and “other transit” at attraction end of the trip

Table 75 Two "must drive" sub-markets

Sub-mkt	Transit Service Available		Description
	Production TAZ	Attraction TAZ	
M1	Any transit	MR/LRT	Share that must drive to any transit at the production end and can use MR/LRT at the attraction end of the trip
M2	Any transit	Other transit	Share that must drive to any transit at the production end and can use "other transit" at the attraction end of the trip

AEMS makes use of WALK SEG commands to allow the model users to specify subzone segmentation such as "can walk," "must drive," and "no transit,"<sup>116</sup> so **the six sub-markets above, along with the "no transit" segment, are represented with seven WALK SEG (WS) variables in the AEMS control files**, and these six variables are calculated as follows from the previously-defined walk percentages:

"Can walk" market

- WSWM – Share of the "walk segment" that can use Metrorail/LRT at both ends of the trip  
= PCMI \* PCMJ
- WSW1 – Share of the "walk segment" that can use "other transit" at production end and MR/LRT at attraction end of the trip  
= (PCTI – PCMI) \* PCMJ
- WSW2 – Share of the "walk segment" that cannot walk to Metrorail/LRT at either end of the trip (i.e., can use "other transit" at both ends of the trip)  
= (PCTI – PCMI) \* (PCTJ – PCMJ)
- WSW3 – Share of the "walk segment" that can use MR/LRT at production end and "other transit" at attraction end of the trip  
= PCMI \* (PCTJ – PCMJ)

"Must drive" market

- WSM1 – Share of the "walk segment" that must drive (to any transit) at the production, but can walk to Metrorail/LRT at the attraction  
= (1 – PCTI) \* PCMJ
- WSM2 – Share of the "walk segment" that must drive (to any transit) at the production, but cannot walk to Metrorail/LRT at the attraction (i.e., must use "other transit" at the attraction)  
= (1 – PCTI) \* (PCTJ – PCMJ)

"No transit" market

- WSNT – Share of the "walk segment" with no access to transit at the attraction end (thus, no access to transit for this zone-to-zone interchange)

<sup>116</sup> "AECOM Consult Mode Choice Computation Programs, AEMS, Users Guide," 29–31.

$$= (1 - WSWM - WSW1 - WSW2 - WSW3 - WSM1 - WSM2)$$

As was the case before with just three transit-access markets, the sum of the seven transit-access shares must equal 100% for any given interchange.

$$WSWM + WSW1 + WSW2 + WSW3 + WSM1 + WSM2 + WSNT = 100\%$$

### Application of seven transit market segments to travel modes in the mode choice model

Given the aforementioned definitions of the transit-access market segments, the mode choice model must be applied to estimate the mode shares for each zone-to-zone interchange. In this case, the seven transit-access markets are applied in AEMS to the travel modes represented in the Ver. 2.3 nested-logit mode choice model as shown in Table 76.

**Table 76 Application of the seven transit-access segments to travel modes represented in the Ver. 2.3 mode choice model**

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
		SOV	SR2	SR3+	WK-CR	WK-BUS	WK-BU/MR	WK-MR	PNR-CR	KNR-CR	PNR-BUS	KNR-BUS	PNR-BU/MR	KNR-BU/MR	PNR-MR	KNR-MR
1	WM Can walk MR/LRT	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
2	W1 Can walk 1	x	x	x	x	x	x		x	x	x	x	x	x	x	x
3	W2 Can walk 2	x	x	x	x	x	x		x	x	x	x	x	x		
4	W3 Can walk 3	x	x	x	x	x	x		x	x	x	x	x	x		
5	M1 Must drive 1	x	x	x					x	x	x	x	x	x	x	x
6	M2 Must drive 2	x	x	x					x	x	x	x	x	x		
7	NT No transit	x	x	x												

Ref: "percent\_walk\_transit\_can\_walk.xlsx"

### Where

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

KNR-MR = kiss-n-ride to Metrorail/Light Rail

## 21.5 Transit access coding

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

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

### 21.5.1 Station file

The station file is a dBase file (station.dbf) that contains information about Metrorail stations, commuter rail stations, light rail stations, bus rapid transit stations/stops, street car stations/stops, express-bus bus stops, and park-and-ride lots that serve these stations/stops. Each station file is associated with one scenario, with the most typical scenarios being the “modeled year” (e.g., 2017, 2020, 2040). This file contains information such as:

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

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

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

The full list of variables in the station file is described in Table 35 on page 126, with the new variables in bold font.

The “access distance code,” known as NCT in the autoacc5.s script, is a newly added variable in the station file that controls the number, extent, and directionality of PNR/KNR access links generated for each parking lot (in the case of PNR) or each station (in the case of KNR). Table 36 describes the meaning of each of the six access distance codes.

The access distance code, along with the transit mode, determines the maximum link distance for the drive-access-to-transit links generated by autoacc5.s for the TPB nested-logit mode choice model. The maximum link distances for PNR are shown in Table 78. Although not shown in the table, the maximum allowed link distance for KNR links is 3 miles. It is also important to note that the KNR links are

generated to Metrorail stations, light rail stations, streetcar stops, and bus stops with parking lots, but not commuter rail stations.

Table 77 shows the mode codes that are used in the station file. "Station centroids" are used to build minimum-impedance paths to all Metrorail and commuter rail stations. In the table below, even though modes 5 and 10 are shown as having a range of numbers designated for station centroids, only Metrorail and commuter rail actually require station centroids.

**Table 77 Mode codes used in the consolidated station file/database (station.dbf)**

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

Notes: \* Station PNR centroids (a.k.a. dummy station centroids) are not required for Mode 5 (LRT) or Mode 10 (BRT/streetcar).<sup>117</sup> For the sake of consistency, the current COG coding practice is to refrain from using station PNR centroids for LRT, BRT, and streetcar. In other words, in the station file, the STAC variable is coded with a value of zero.

### 21.5.2 Sidewalk links and zonal walk links

In the Version 2.2 travel model and earlier models, there was a walk network (sidewalk network), used for transferring from one transit line to another, in downtown DC and downtown Silver Spring, Maryland. In the Version 2.3 travel model, there is a sidewalk network in almost the entire modeled area. The regional sidewalk network is generated automatically using a script walkacc.s (see p. A-5 of the flowchart in Appendix A). Walkacc.s creates a sidewalk network by converting all suitable highway links into sidewalk links (Mode 13). Examples of highway links that are not converted into sidewalk links include freeways, parkways, and ramps (Facility Type = 1, 5, or 6). In order to limit the size of the sidewalk network to links that are likely used for walking, walkacc.s eliminates sidewalk links from zones where the "percent walk to transit" is zero. There is also a way to supply the program with a list of sidewalk links to be manually added or subtracted to the automated list of sidewalk links. For example, one can manually add a sidewalk link for Memorial Bridge, and one can manually remove sidewalk links that should not exist due to a physical barrier. See Jain's 2010 memo for more details.<sup>118</sup>

Walkacc.s also generates zonal walk-access-to-transit links (Mode 16 links). It automatically sweeps each TAZ, generating walk-access links from the zone centroid to all highway network nodes within a maximum walk distance (See Equation 3).

<sup>117</sup> Jain to Milone and Moran, "MWCOG Network Coding Guide for Nested Logit Model (First Draft: September 20, 2007; Updated February 2008 and October 2010)," 6 and 10.

<sup>118</sup> Manish Jain to Ronald Milone and Mark Moran, "MWCOG network coding guide for Nested Logit Model (First draft: September 20, 2007; Updated February 2008 and Oct. 2010)," Memorandum, October 2010, 7.

Equation 3 Maximum walk distance formula, used for generating walk-access-to-transit links

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

So, for a small, downtown zone with an area of 0.1 square miles, the program would calculate a maximum walk distance of 0.237 miles and connect all highway network stop nodes that lie within that distance from the zone centroid. There is an absolute maximum of 1.0 mile, which would be obtained for zones with a size of 1.78 square miles or greater. The actual calculated (straight-line) distance and computed walk time are stored on each link. No walk-access links are generated for zones with a zero percent walk to transit. Figure 30 shows zonal walk access links and sidewalk links in downtown Washington, D.C., near Farragut Square (TAZ 37, which is in the center of the figure). The thickest gray lines are the TAZ boundaries, which are not part of the actual highway or transit network, but are shown for reference. The lines emanating from each TAZ centroid (dark-blue, when the figure is viewed in color) are the zonal walk access links (Mode 16). The rectilinear (green) lines over many, but not all roads, are the sidewalk links (Mode 13). Mode 13 and 16 links can be shown in Cube Base by adding the four files "support link" files associated with walk-access to transit

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

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



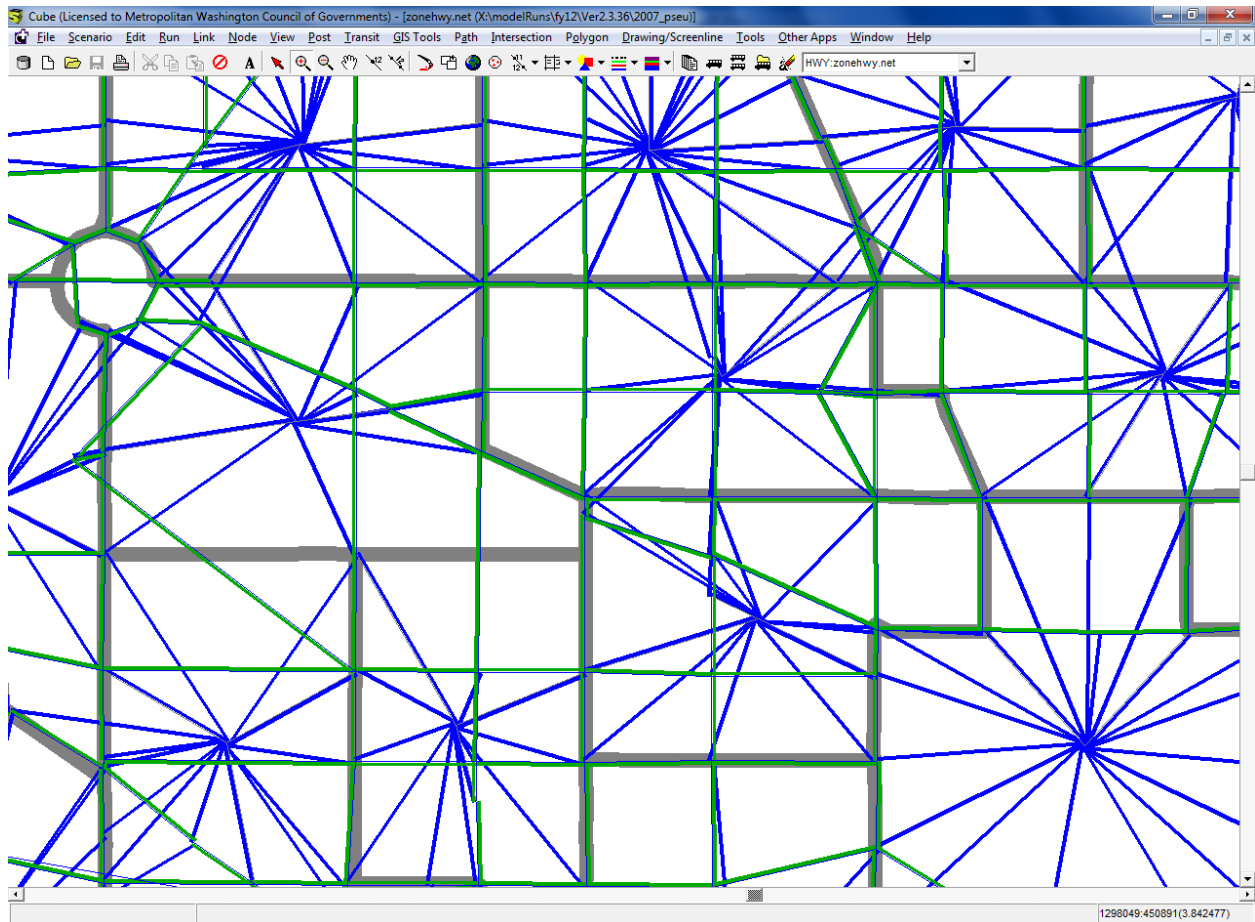


Figure 30 Zonal walk access links and sidewalk links in downtown DC near Farragut Square (Ver. 2.3 NL MC model)

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

### 21.5.3 Zonal auto-access links

The Cube Voyager script `autoacc5.s`, originally created by AECOM Consult as a Fortran program and later transferred to Voyager script by TPB staff, is used to generate auto-access-to-transit links. Zonal auto access links are generated by transit mode (Metrorail, commuter rail, light rail, BRT, streetcar, and bus) for both the peak ("AM") and off-peak ("mid-day") time periods. Auto access links (Mode 11) are a function of multiple criteria:

- Orientation toward downtown (defined as TAZ 8, which corresponds to The Ellipse, just south of The White House)
- A backtracking penalty and a prohibition of crossing the Potomac River (except for trips from Loudoun County to MARC commuter rail);
- A maximum link distance, which is a function of station type (e.g., terminal vs. non-terminal) and transit mode;
- Manually specified overrides; and
- Distances based on the highway skims from the highway network that includes dummy centroids representing Metrorail and commuter rail stations.

Table 78 Maximum link distances for drive-access-to-transit links: Ver. 2.3 NL MC model

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

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

Figure 31 shows kiss-and-ride (KNR) auto-access-to-transit links for the AM period associated with Metrorail stations in Northern Virginia.

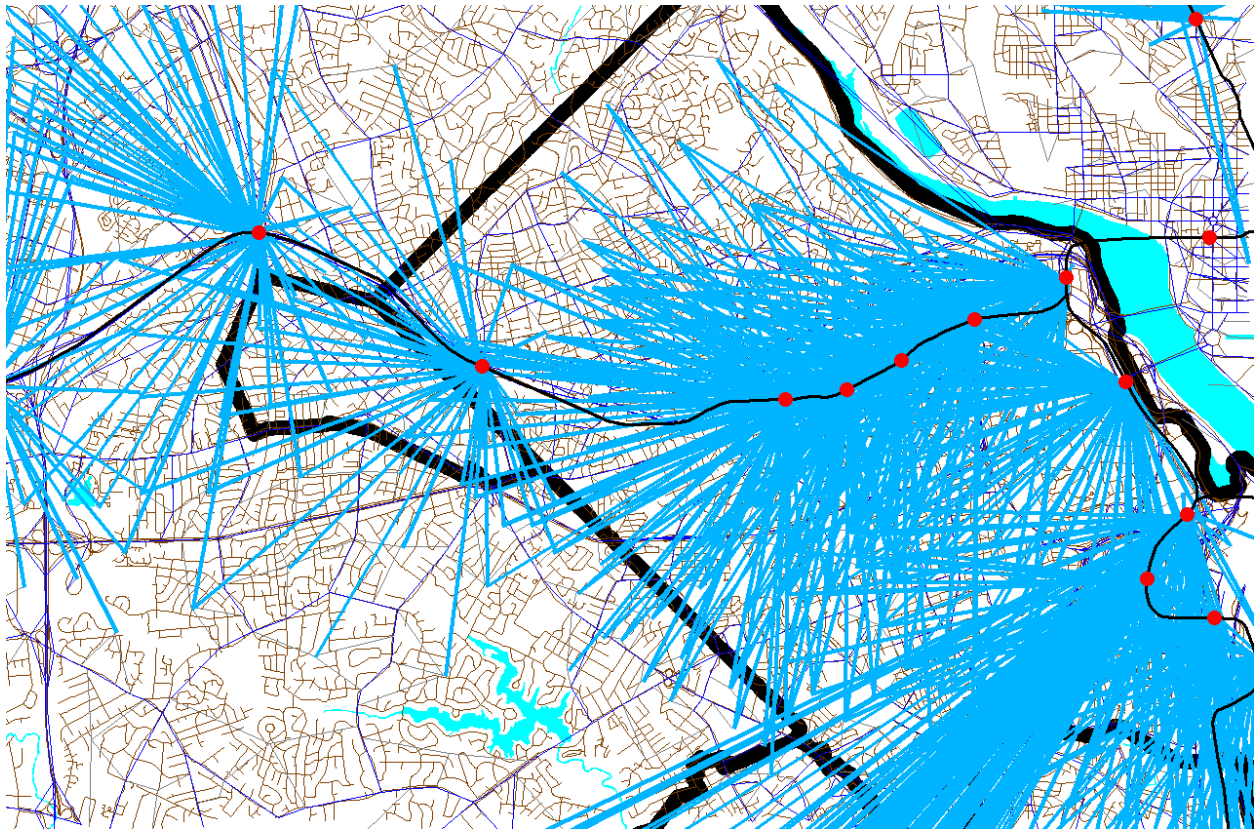


Figure 31 Kiss-and-ride (KNR) auto access links to Metrorail stations in Northern Virginia

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

Figure 32 shows park-and-ride (PNR) auto-access-to-transit links for the AM period associated with Metrorail stations in Northern Virginia. Notice that the Orange Line stations from Clarendon to Rosslyn do not have PNR-access links, since they do not have PNR lots. By contrast, these stations do have KNR-access links, since these stations can have KNR access. The Pentagon Metrorail station is another example of a station where the model does not allow travelers to have PNR access, but they may have KNR access. Notice that the KNR-access links and PNR-access links are not shaped like a circular “starburst,” but are somewhat flattened, due to the backtracking penalty. This was done to mimic the behavior of travelers who tend not to want to backtrack when driving to park at or be dropped off at a Metrorail or commuter rail station.

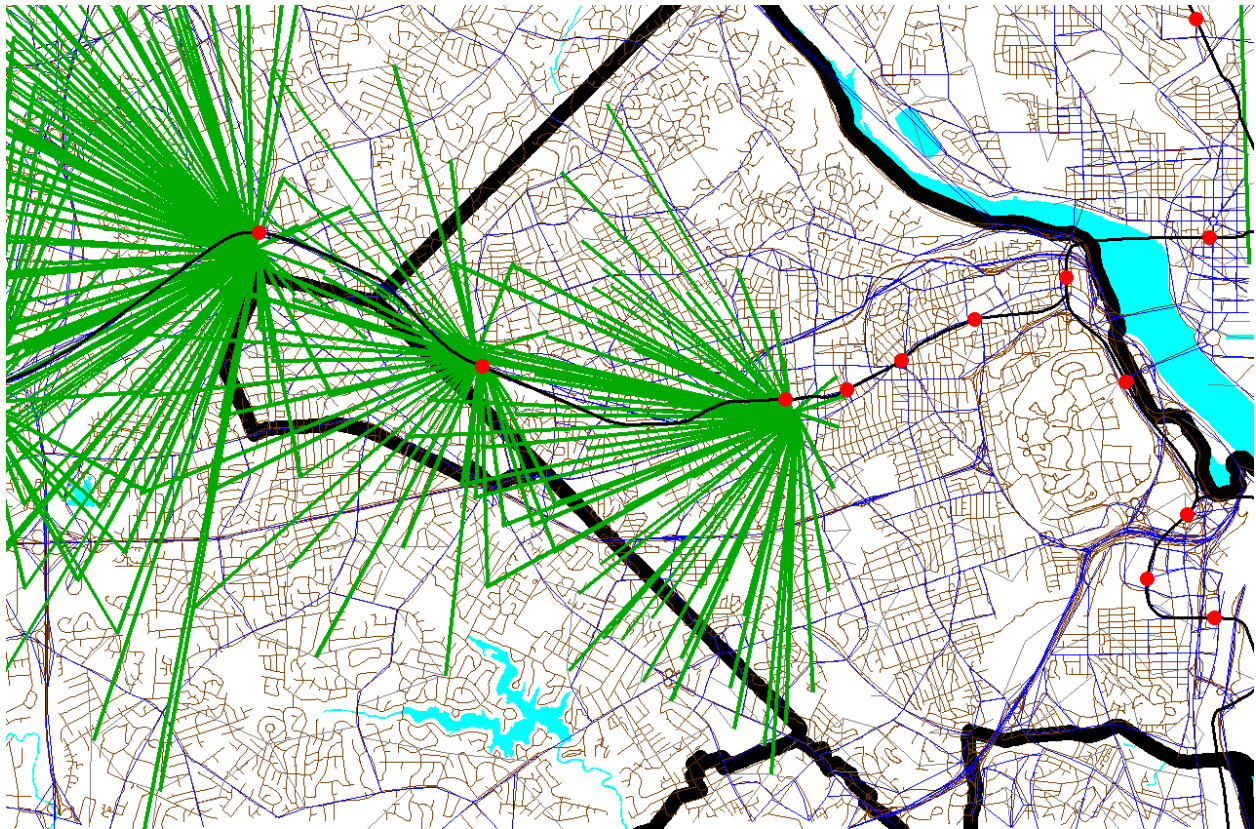


Figure 32 Park-and-ride (PNR) auto access links to Metrorail stations in Northern Virginia

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

#### 21.5.4 Station transfer links

Station transfer links are walk links connecting:

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

These links are generated automatically from data in the station file. **For PNR-station transfer links, the walk time is a function of parking capacity and parking cost, since it is reasoned that bigger parking lots and more expensive parking lots make them more burdensome to use.**<sup>119</sup>

The station file also includes shadow parking price variables (STAPKSHAD & STAOPSHAD, see Table 35) which are not currently used. However, the PNR capacity and parking costs are coded into the station file and are used by PARKER.S to create PNR-to-station links read into TRNBUILD. The PNR-to-station time/impedance is computed as:<sup>120</sup>

<sup>119</sup> Ibid., 6.

<sup>120</sup> Personal communication from Ron Milone, 9/25/13.

**Equation 4 PNR-to-station time/impedance**

$$(PNR\ time) = walk + SHAD + (MinPerDollar * park)$$

where

$(PNR\ time)$  = PNR – to – station link time/impedance

$walk$  = (1 to 5 min)Base time, which is a function of the number of PNR spaces

$SHAD$  = Shadow cost (min.)

$MinPerDollar$  = Equivalent minutes per dollar. Currently set to 6 (each dollar paid = 6 min.)

$Park$  = *Parking cost at station divided by two*

The Mode 15 links are generated by the script parker.s (see page A-5 of the flowchart in Appendix A).

The following files contain the mode 15 links:

busampnr.tb  
busoppnr.tb  
comampnr.tb  
comoppnr.tb  
lrtampnr.tb  
lrtoppnr.tb  
metampnr.tb  
metoppnr.tb  
newampnr.tb  
newoppnr.tb

The mode 12 links are developed manually using the COGTools geodatabase. These links can be found in the following files (see page A-5 of the flowchart in Appendix A):

Com\_Bus.tb  
LRT\_bus.tb  
Met\_Bus.tb  
NEW\_bus.tb

### 21.5.5 Zonal percent walk to transit calculations

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

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



These walk-to-transit areas are sometimes called transit walksheds. Under contract with COG, AECOM has developed a new, automated/integrated Python/ArcPy walkshed process that is describe in Chapter 11 (“Building transit walksheds and calculating zonal walk percentages”) on p. 101.

## 21.6 Transit path-building procedures

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

Three modes of access to transit

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

Four transit modes/combinations

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

Two time-of-day periods

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

However, at present, PNR and KNR to commuter rail are combined as a single path, since, for commuter rail, the PNR- and KNR-access links are identical. Consequently, the number of transit paths built between each production/attraction zone pair is 22. Table 79 summarizes the paths and available transit sub modes in each path. Again, in this figure, “drive to commuter rail” and “KNR to commuter rail” are combined into one category.

Run times for transit routes are controlled by the RUNTIME keyword (TRNBUILD).<sup>121</sup> As stated previously, path weights are consistent with the weights used in the mode choice model:

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

Headway combination between two or more transit routes is allowed to occur provided 1) the routes share the same transit mode code and 2) the difference between the run time and the minimum run time is less than a designated number of minutes (5 minutes for AM and 10 minutes for off peak). A maximum path time is set at 360 weighted minutes. There is no weighting of in-vehicle time by transit

---

<sup>121</sup> In Ver. 2.3.57a and earlier, bus IVT skims were adjusted to reflect the general level of road congestion using the factor table Lbus\_TimFTRS.asc. In Ver. 2.3.66, this adjustment is now done directly to the mode 1, 6, and 8 local bus line files (\*.TB) using the script Adjust\_Runtime.s.

sub-modes (i.e., all transit modes have an IVT weight of 1.0). The maximum initial wait time for all ten transit modes is set at 60 perceived minutes. The minimum transfer wait time is 4.0 minutes for bus (Modes 1, 2, 6, 7, 8), 0 minutes for Metrorail (Mode 3), 4.0 minutes for commuter rail (Mode 4), 0 minutes for LRT (Mode 5), 10.0 minutes for express bus (Mode 9), and 4.0 minutes for Mode 10 (streetcar and/or BRT).

Table 79 Path-specific parameters used in transit path building

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

Source: AECOM Consult, Inc.<sup>122</sup>

<sup>122</sup> AECOM Consult, Inc., *Post MWCOG – AECOM Transit Component of Washington Regional Demand Forecasting Model: User's Guide* (AECOM Consult, Inc., March 2005).



## 21.7 Treatment of parking costs and terminal times for non-transit-related trips

Parking costs can be associated with either a transit trip (in the case of a drive-access transit trip) or a non-transit trip (an auto person trip, where no transit is involved). For drive-access transit trips, the cost of parking is stored in the station file. For park-and-ride (PNR)-to-station transfer links, the walk time is a function of parking capacity and parking cost,<sup>123</sup> but parking cost is not used as part of the transit path-building. For driving trips not involving transit, a parking cost model is used, where parking cost is a function of employment density. The next section of the report concerns parking costs that are not associated with a transit trip.

### 21.7.1 Non-transit-related parking costs

In applying the Version 2.3 model, prior to the execution of the mode choice model, a Voyager script (prefarv23.s) is used to generate zonal files containing zonal parking costs and highway terminal times (the time to park and “un-park” a vehicle). The files are, in turn, read into the mode choice model upon execution. The Version 2.3 model includes a new parking cost model estimated based on the 2007/2008 HTS.<sup>124</sup> HBW trip purpose utilizes the daily parking rate, while all other purposes use the hourly parking rate. Thus, two separate parking cost models were estimated, one for daily rates and one of hourly rates. For the daily rates model, the observed data indicated that it is rare for a traveler to incur parking costs in area types 4 and above, thus the model was estimated only for area types 1-3. A daily parking cost was estimated to be:

#### Equation 5 Daily non-transit-related parking cost for area types 1-3

$$\text{Non-transit-related parking cost} = 2.1724 * \ln(\text{floating employment density}) - 15.533$$

The resulting non-transit-related parking costs are also shown in Figure 33.

---

<sup>123</sup> Jain to Milone and Moran, “MWCOG Network Coding Guide for Nested Logit Model (First Draft: September 20, 2007; Updated February 2008 and October 2010),” 6.

<sup>124</sup> Mary Martchouk to Mark S. Moran, “Developing a Parking Cost Model for Automobile Modes in the Version 2.3 Travel Model,” Memorandum, June 14, 2010.

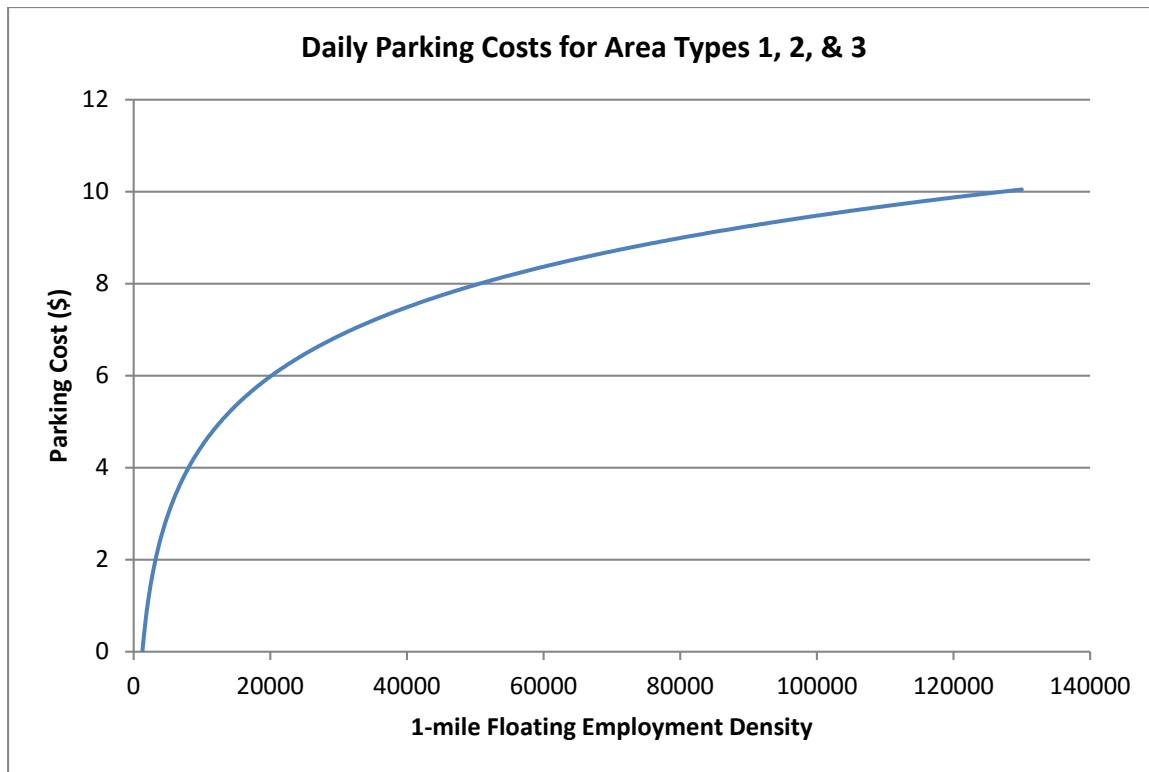


Figure 33 Non-transit-related, daily parking cost model used in the Version 2.3 Model

Ref: I:\ateam\docum\FY11\Ver2.3\modelDoc\_v3\02\_userGuide\parking\_scatterplots.xlsx

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

### 21.7.2 Non-transit-related highway terminal time assumptions

Non-transit-related highway terminal time is typically associated with the average time spent parking or “un-parking” an automobile. The current mode choice model application program considers highway terminal time only at the attraction end. Highway time is calculated as a function of employment density, as shown in Table 80.

**Table 80 Non-transit-related highway terminal time as a function of employment density**

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

### 21.8 Auto Operating Costs

The auto operating cost in the mode choice model relate to out-of-pocket expenditures directly associated with the requirements of an automobile trip, including fuel, oil, maintenance, tire wear, etc. (auto ownership costs including insurance, registration fees are not included). The mode choice model expresses operating costs as a per-mile rate (year-2007 cents) that is specified as a parameter in the nested-logit mode choice model control files. We are currently using 10 cents per mile and this rate is not varied over time (i.e., the auto operating cost for 2016 and 2030 are both assumed to be 10 cents per mile, in year-2007 cents).



## 22 Time-of-Day Processing

### 22.1 Overview

The time of day process is applied to convert daily vehicles among the four modeled time periods, prior to being assigned to the network. The process is applied with the Time-of-Day.s and Misc\_Time-of-Day.s scripts. The Prepare\_Trip\_Tables\_for\_Assignment.s script is used to combine the various trips by time period into combined trip tables for the traffic assignment process. The input and output files are listed in Table 81 and Table 82.

Table 81 Inputs to time-of-day process

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

Note: <ITER> =PP, i1...i4

Table 82 Outputs of time-of-day process

Trip Tables by Time Period	AM<ITER>.ADR, MD<ITER>.ADR, PM<ITER>.ADR, NT<ITER>.ADR,	Binary
Miscellaneous Time-of-Day Files	MISCAM<ITER>.TT, MISCMD<ITER>.TT, MISCPM<ITER>.TT, MISCNT<ITER>.TT	Binary
Total Vehicle Trips by Six Markets T1 – SOVs T2 – 2 occ. vehicles T3 – 3+ occ. vehicles T4 – Commercial vehicles T5 – Medium + Heavy Trucks Combined T6 – Airport passenger auto-driver trips/vehs.	<ITER>AM.VTT, <ITER>MD.VTT, <ITER>PM.VTT, <ITER>NT.VTT	Binary



## 23 Traffic Assignment

### 23.1 Overview

As mentioned in section 2.3 (“Modeling steps and the speed feedback loop”), the Version 2.3 Travel Model uses a user-equilibrium (UE) traffic assignment, which is the generally accepted method for static traffic assignments. The user equilibrium condition was defined by Wardrop in 1952.<sup>125</sup> According to Wardrop’s first principle, in the case where all trip makers perceive costs the same way (i.e., no stochastic effects):

*Under equilibrium conditions, traffic arranges itself in congested networks such that all used routes between an O-D pair have equal and minimum costs, while all unused routes have greater or equal costs.*<sup>126</sup>

Furthermore, the assignment process is a multi-class UE assignment, meaning that separate user classes can be assigned at the same time. The Version 2.3 model includes six user classes:

1. Single-occupant vehicle (SOV)
2. High-occupant vehicle with two persons (HOV2)
3. High-occupant vehicle with three+ persons (HOV3+)
4. Medium and heavy trucks
5. Commercial vehicles
6. Airport passengers traveling to/from the three commercial airports

In Version 2.2, there were only five user classes, since the commercial vehicles category was grouped with medium/heavy truck. The primary reason for distinguishing truck markets is to allow for the option of using passenger car equivalents (PCEs) in the traffic assignment process. The use of PCE’s has not yet been implemented, but they will be considered in future developmental work.

Additionally, the Version 2.3 model includes four time-of-day periods for traffic assignment:

- AM peak period (3 hours: 6:00 AM to 9:00 AM)
- Midday period (6 hours: 9:00 AM to 3:00 PM)
- PM peak period (4 hours: 3:00 PM to 7:00 PM)
- Night/early morning period (11 hours: 7:00 PM to 6:00 AM)

Most MPOs use a UE traffic assignment that relies on an optimization algorithm known as the Frank-Wolfe (FW) algorithm.<sup>127</sup> The FW algorithm is essentially a series of all-or-nothing traffic assignments where flows are combined using weights from an optimization process whose goal is to minimize an

---

<sup>125</sup> John Glen Wardrop, “Some Theoretical Aspects of Road Traffic Research,” *Proceedings of the Institution of Civil Engineers* 1, no. 3 (January 1952): 325–62, doi:10.1680/ipeds.1952.11259.

<sup>126</sup> Juan de Dios Ortúzar and Luis G. Willumsen, *Modelling Transport*, 2nd ed. (John Wiley & Sons, 1994), 304.

<sup>127</sup> Frank and Wolfe, “An Algorithm for Quadratic Programming.”

objective function. The process stops when a stopping criterion is met. Previously, the Version 2.3 Travel Model used the following UE stopping criterion: When the relative gap  $\leq 10^{-3}$  OR the number of UE iterations  $\geq 300$ . The relative gap threshold was always intended to be the primary stopping criterion, with the number of UE iterations functioning as a backup criterion. Now, however, we have moved to what we call a “progressive” relative gap stopping criterion. The idea is that, in the early SFB iterations, the UE closure criterion will be relatively loose, but, in the later SFB iterations, the UE closure criterion will tighten, as shown in Table 3.

**Table 83 User equilibrium closure criterion (relative gap) varies by speed feedback iteration**

Speed feedback iteration	Primary closure criterion for UE traffic assignment	Secondary closure criteria for UE traffic assignment
Pump prime	Relative gap $\leq 10^{-2}$ (i.e., 0.01)	Number of UE iterations $\geq 1000$
1	Relative gap $\leq 10^{-2}$ (i.e., 0.01)	Number of UE iterations $\geq 1000$
2	Relative gap $\leq 10^{-2}$ (i.e., 0.01)	Number of UE iterations $\geq 1000$
3	Relative gap $\leq 10^{-3}$ (i.e., 0.001)	Number of UE iterations $\geq 1000$
4	Relative gap $\leq 10^{-4}$ (i.e., 0.0001)	Number of UE iterations $\geq 1000$

By using the higher value for UE iterations (1000 vs. 300), we were able to ensure that this secondary criterion is unlikely to be used as the stopping criterion. Based on a series of sensitivity tests,<sup>128</sup> we found that the new progressive relative gap scheme results in a relatively converged traffic assignment, without the extremely lengthy model run times that would be needed if one were to use a high threshold (e.g.,  $10^{-4}$  relative gap) for each of the five SFB iterations. The Version 2.3 Travel Model uses a slight variation of the FW algorithm, called the *bi-conjugate* Frank-Wolfe algorithm, which converges marginally faster than the classic FW algorithm.

### 23.2 Two-step assignment

The Version 2.2 traffic assignment process prior to the fall of 2008 consisted of three separate assignment executions for each speed feedback (SFB) loop: AM peak period, PM peak period, and the off-peak period (see Table 84). To respect the various highway path options and prohibitions in the Washington region, five separate markets or “user classes” (trip tables) were loaded during each assignment execution:

1. Single-occupant vehicles, including commercial vehicles (SOV),
2. 2-occupant vehicles (HOV2),
3. 3+occupant vehicles (HOV3+),
4. Trucks (medium and heavy), and
5. Airport passenger vehicles.

<sup>128</sup> Moran and Milone, “Status Report on the Version 2.3 Travel Model: Updates to the Model and Year-2010 Validation,” 7–11.



**Table 84 Traffic assignment in the Version 2.2 Travel Model prior to fall 2008: Three multiclass assignments**

For each SFB loop	Assignment period	Trip markets assigned
Assignment 1	AM peak	1. SOV 2. HOV2 3. HOV3+ 4. Trucks 5. Airport passengers
Assignment 2	PM peak	1. SOV 2. HOV2 3. HOV3+ 4. Trucks 5. Airport passengers
Assignment 3	Off-peak	1. SOV 2. HOV2 3. HOV3+ 4. Trucks 5. Airport passengers

In the fall of 2008, as part of air quality conformity work, the traffic assignment process was modified to improve the assignment of HOV/HOT traffic on the Capital Beltway in Virginia and the I-395 Shirley Highway.<sup>129</sup> In the revised process, shown in Table 85, the AM traffic assignment was split into two parts: non-HOV 3+ (i.e., SOV, HOV2, trucks, and airport passengers) and HOV 3+. Similarly, the PM traffic assignment was also split into two parts: non-HOV 3+ and HOV3+. This new traffic assignment process is sometimes referred to as the “two-step assignment,” since it splits the AM and PM assignment each into two parts.<sup>130</sup>

**Table 85 Traffic assignment in the Version 2.2 Travel Model prior to fall 2008: Five multiclass assignments**

For each SFB loop	Assignment period	Trip markets assigned
Assignment 1	AM peak (non-HOV3+)	1. SOV 2. HOV2 3. Trucks 4. Airport passengers
Assignment 2	AM peak (HOV3+)	1. HOV3+
Assignment 3	PM peak (non-HOV3+)	1. SOV 2. HOV2 3. Trucks 4. Airport passengers
Assignment 4	PM peak (HOV3+)	1. HOV3+
Assignment 5	Off-peak	1. SOV 2. HOV2 3. HOV3+ 4. Trucks 5. Airport passengers

<sup>129</sup> Ronald Milone and Mark S. Moran, “TPB Models Development Status Report” (November 21, 2008 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., November 21, 2008).

<sup>130</sup> Jinchul Park to Files, “Two Step Traffic Assignment for HOT Lane Modeling in 2008 CLRP,” Memorandum, (December 2, 2008).

The result was five (not three) traffic assignments, with either four, one, or five user classes, depending on which assignment was being conducted. The fifth traffic assignment, representing the off-peak period, included all five trip markets (it was only the two peak-period assignments where the non-HOV 3+ and HOV 3+ were split out).

In the first step of the two-step assignment (assignments #1 and #3), non-HOV 3+ traffic (i.e., SOV, HOV 2, truck, and airport passenger trips) is assigned to all facilities (HOV and general purpose). In the second step, HOV 3+ traffic is assigned to HOT lanes and other facilities on the partially loaded network. The pre-assignment of non-HOV 3+ traffic results in congested link speeds for the general purpose lanes. This means that HOV 3+ traffic has a greater incentive to use HOV facilities, which results in improved HOV 3+ loadings on priority-use and general-use facilities.

**Recent versions of the regional travel demand model (e.g., 2.3.52, 2.3.57, 2.3.57a, and 2.3.66) continue to use the same two-step assignment**, but there are now six assignments in each speed feedback loop, since the off-peak period has been split into midday and nighttime. Also, commercial vehicles are split out from trucks, as shown in Table 86. Both of these changes occurred in earlier builds of the Version 2.3 Travel Model.

**Table 86 Traffic assignment in the Version 2.3.66 Travel Model: Six multiclass assignments**

For each SFB loop	Assignment period	Trip markets assigned
Assignment 1	AM peak (non-HOV3+)	1. SOV 2. HOV2 3. Trucks 4. Commercial vehicles 5. Airport passengers
Assignment 2	AM peak (HOV3+)	1. HOV3+
Assignment 3	PM peak (non-HOV3+)	1. SOV 2. HOV2 3. Trucks 4. Commercial vehicles 5. Airport passengers
Assignment 4	PM peak (HOV3+)	1. HOV3+
Assignment 5	Off-peak, midday	1. SOV 2. HOV2 3. HOV3+ 4. Trucks 5. Commercial vehicles 6. Airport passengers
Assignment 6	Off-peak, nighttime	1. SOV 2. HOV2 3. HOV3+ 4. Trucks 5. Commercial vehicles 6. Airport passengers

### 23.3 Application details

The traffic assignment process is shown on page A-10 of the flowchart in Appendix A. The Highway\_assignment\_parallel.bat batch file calls the highway\_assignment\_parallel.s script. As described

in Chapter 8 (“Use of parallel processing to reduce model run times”), the highway assignment process has been “parallelized” by using Cube Cluster (both IDP and MDP), which is Cube’s implementation of distributed processing. See section 8.2.1 for terminology related to distributed processing, and see section 8.2.4 for details about how Cube Cluster has been implanted in the Version 2.3.52 model (and later versions, such as 2.3.66), including the traffic assignment step.

The highway assignment process uses a generalized cost or impedance, which is function of both travel time and cost. Cost is converted to travel time based on the vehicle class and time of day, as described in Table 87. These minutes/per-dollar factors are used for both variably-priced facilities, such as the I-495 HOT lanes in Virginia, and for fixed-price facilities, such as the Governor Nice Bridge.

**Table 87 Time Valuation by Vehicle Type and Time Period (minutes/dollar, in year-2007 prices)**

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

(Time\_Valuation\_V2.3.xls)

The inputs and outputs of the highway\_assignment\_parallel.s script are shown in Table 88 and Table 89, respectively.

**Table 88 Inputs to traffic assignment process**

Volume delay parameters and free-flow speed assumptions	support\hwy_assign_Conical_VDF.s support\hwy_assign_capSpeedLookup.s	Script block
Total vehicle trips by 4 time-of-day periods and 6 user classes	<ITER>_AM.VTT, <ITER>_MD.VTT, <ITER>_PM.VTT, <ITER>_NT.VTT	Binary
Toll minutes equivalence file	support\toll_minutes.txt	Text
AM Toll Factors by Vehicle Type	Inputs\AM_Tfac.dbf	DBF
Midday Toll Factors by Vehicle Type	Inputs\MD_Tfac.dbf	DBF
PM Toll Factors by Vehicle Type	Inputs\PM_Tfac.dbf	DBF
Night Toll Factors by Vehicle Type	Inputs\NT_Tfac.dbf	DBF
Network files	ZONEHWY.NET, <ITER>_HWY.NET	Binary

Note: <ITER> =PP, i1...i4

The Version 2.3 uses conical volume-delay functions (VDFs). More information about these VDFs can be found on pp. 8-13 to 8-17 of the calibration report dated 1/20/12.<sup>131</sup>

<sup>131</sup> Milone et al., “Calibration Report for the TPB Travel Forecasting Model, Version 2.3.”

When the traffic assignment process is run, the script creates a series of user equilibrium convergence report files, as shown in Table 89. Each file contains the relative gap by user equilibrium iteration. By using these files with a spreadsheet, one can make plots of the rate of convergence of the traffic assignment.

Table 89 Outputs of traffic assignment process

Loaded-link files by time period	<ITER>_am_load_link.asc, <ITER>_md_load_link.asc, <ITER>_pm_load_link.asc, <ITER>_nt_load_link.asc,	Text
Loaded Highway Network	<ITER>_Assign_output.net	Binary
UE convergence report files	<iter>_ue_iteration_report_AM_nonHov.txt <iter>_ue_iteration_report_AM_hov.txt <iter>_ue_iteration_report_PM_nonHov.txt <iter>_ue_iteration_report_PM_hov.txt <iter>_ue_iteration_report_MD.txt <iter>_ue_iteration_report_NT.txt	Text

Note: <ITER> =PP, i1...i4

Table 90 provides further details regarding the attributes of the final loaded highway network.

Table 90 Variables included in the final iteration, loaded highway network (i4\_Assign\_output.net)

Variable Name	Description
A	A-Node
B	B-Node
DISTANCE	Link Distance in miles (x.xx)
SPDC	(Not used)
CAPC	(Not used)
JUR	Jurisdiction Code (0-23) 0/dc, 1/mtg, 2/pg, 3/alr/, 4/alx,5, ffx, 6/ldn, 7/ pw, 8/(unused), 9/frd, 10/how, 11/aa, 12/chs, 13/(unused), 14/car, 15/cal, 16/stm, 17/ kg, 18/fbg, 19/stf, 20/spts, 21/fau, 22/clk, 23/jef
SCREEN	Screenline Code (1-38)
FTYPE	Link Facility Type Code (0-6) 0/Centroids, 1/Freeways, 2/Major Art., 3/Minor Art, 4/Collector, 5/Expressway, 6/Ramp
TOLL	Toll Value in current year dollars
TOLLGRP	Toll Group Code (1-9999)
<Period> LANE	<Period> No. of Lanes
<Period>LIMIT	<Period> Limit Code (0-9)
EDGEID	Geometry network link identifier
LINKID	Logical network link identifier
NETWORKYEA	Planning year of network link

SHAPE_LEN	Geometry length of network link (in feet)
PROJECTID	Project identifier
TAZ	Nearest TAZ centroid to midpoint of link (1-3,722)
ATYPE	Area Type (1-6)
SPDCCLASS	Speed Class
CAPCLASS	Capacity Class
DEFLATIONFTR	Factor for deflating current year tolls to constant year tolls
<Period>TOLL	<Period> Toll Value in current year dollars - all tolled facilities
<Period>TOLL_VP	<Period> Toll Value in current year dollars - Variably priced tolled facilities only
<Period> HTIME	<Period> Highway Time - based on initial highway lookup speeds
I4<Period>SOV	Iteration 4 <Period> assigned SOV Volume
I4<Period>HV2	Iteration 4 <Period> assigned HOV2 Volume
I4<Period>HV3	Iteration 4 <Period> assigned HOV3 Volume
I4<Period>CV	Iteration 4 <Period> assigned Commercial Vehicle Volume
I4<Period>TRK	Iteration 4 <Period> assigned Truck Volume
I4<Period>APX	Iteration 4 <Period> assigned Airport Passenger Volume
I4<Period> VOL	Iteration 4 <Period> assigned Volume
I4<Period>VMT	Iteration 4 <Period> Vehicle Miles Travelled (VMT)
I4<Period>FFSPD	Iteration 4 <Period> free flow speed (mph)
<Period>HRLKCAP	<Period> hourly link capacity
<Period>HRLNCA P	<Period> hourly lane capacity
I4<Period>VC	Iteration 4 <Period> Volume Capacity ratio
I4<Period>VDF	Iteration 4 <Period> Volume Delay function
I4<Period>SPD	Iteration 4 <Period> Speed (mph)
I424VOL	Iteration 4 Daily (24 hour) Volume
<b>KEY</b>	
<Period>= AM	AM Peak Period (6:00-9:00 AM)
MD	Mid Day (9:00 AM - 3:00 PM)
PM	PM Peak Period (3:00 - 7:00 PM)
NT	All remaining hours

Since the travel model includes speed feedback, in order to ensure that highway volumes and hence speeds are stabilizing with each successive speed feedback iteration, it is necessary to apply a link-level “method of successive averaging” (MSA) process. The MSA averaging is performed on the basis of total (non-segmented) link volumes, and is performed individually for each time period. This process is performed after each successive highway assignment process using the Average\_Link\_Speeds.bat file that includes the Average\_Link\_Speeds.s script. This script uses the current iteration and previous iteration loaded networks to develop a network with volume averaging named <ITER>\_HWY.net.



## 24 Transit Assignment, Including Summary Process (LineSum)

### 24.1 Transit assignment process

Transit assignment is a new feature of the Version 2.3 Travel Model that was not part of the Version 2.2 Travel Model. Transit assignment is where transit trips are loaded on to the transit network. Although highway and transit assignment have some similarities, it is useful to point out some of the differences between these two assignment procedures. First, whereas highway assignment is done with trip tables in origin/destination (O/D) format, transit assignment is done with trip tables in production/attraction (P/A) format. Second, whereas highway assignment is capacity constrained, transit assignment is not. Lastly, whereas highway assignment is done in each of the five speed feedback loops (i.e., pump prime, i1, i2, i3, and i4), transit assignment is conducted only at the conclusion of the i4 speed feedback loop (See Figure 2 and Figure 3).

Procedures for transit assignment are shown on pages A-15 through A-17 in the flowchart in Appendix A. The transit assignment is run using the `Transit_Assignment_Parallel.bat` and `Transit_Assignment_LineHaul_Parallel.bat` batch files, the first of which is called from the "run model steps" batch file. Prior to transit assignment, the five mode choice trip tables (HBW, HBS, HBO, NHW, and NHO) are combined into two trip tables (AM = HBW; and OP = HBS + HBO + NHW + NHO), using the `Combine_Tables_For_TrAssign_Parallel.s` script. After the transit assignment has been run, the transit assignment output files are summarized using the LineSum program. This is also called from the `Transit_Assignment_Parallel.bat` batch file via the `TranSum.bat` batch file. The transit assignment process is run in the standard scenario/output folder (e.g., `2016_Final`), but the transit assignment summary files are stored in a sub-folder called "transum." The inputs to the `Transit_Assignment_Parallel.bat` batch file are shown in Table 91 and the outputs are shown in Table 92.

Table 91 Inputs to transit assignment process

Trip tables segmented by mode (coming from the mode choice model)	i4_HBW_NL_MC.MTT i4_HBS_NL_MC.MTT i4_HBO_NL_MC.MTT i4_NHW_NL_MC.MTT i4_NHO_NL_MC.MTT	Binary
Highway network	Zonehwy.net	Binary
AM peak transit lines	Inputs\MODE1AM...MODE10AM.TB	Text
Off peak transit lines	Inputs\MODE1OP...MODE10OP.TB	Text
Transit network files	met_node.tb, met_bus.tb, met_link.tb, com_node.tb, com_bus.tb, com_link.tb, lrt_node.tb, lrt_bus.tb, lrt_link.tb new_node.tb, new_bus.tb, new_link.tb  met_pnrn.tb, com_pnrn.tb, bus_pnrn.tb, lrt_pnrn.tb, new_pnrn.tb  met_[AM OP]_pnr.tb, com_[AM OP]_pnr.tb, bus_[AM OP] _pnr.tb, lrt_[AM OP]_pnr.tb, new_[AM OP]_pnr.tb  met_[AM OP]_pnr.asc, com_[AM OP]_pnr.asc, bus_[AM OP] _pnr.asc, lrt_[AM OP]_pnr.asc, new_[AM OP]_pnr.asc  met_[AM OP]_knr.asc, bus_[AM OP]_knr.asc, lrt_[AM OP] _knr.asc, new_[AM OP]_knr.asc  met_bus.tb, com_bus.tb, lrt_bus.tb, new_bus.tb	Text
Transit network walk links	walkacc.asc, sidewalk.asc	Text

Note: <ITER> =PP, i1...i4

Table 92 Outputs of transit assignment process

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

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

The transit assignment is done for two time-of-day periods: the peak period and the off peak period. The peak period is represented by the three-hour AM peak period. The off-peak period is represented by the five-hour midday period. Thus, when calculating peak-period travel times on transit (“skims”) the AM peak period is used to represent the level of service in both the AM and PM peak period. Similarly, when calculating the average headway and average run time for each transit route, these calculations



are done for the peak period (represented by the AM peak) and the off-peak period (represented by the midday period). It is assumed that the majority of HBW trips will occur in the peak periods and that the majority of non-work trips will occur in the off-peak periods. Consequently, prior to the actual transit assignment, the five trip tables coming out of mode choice are combined into two tables: one for the peak period and one for the off-peak period. The peak-period trip table ("AM") contains only one trip table (HBW). By contrast, the off-peak period trip table ("OP") contains the trip tables from the other four trip purposes (HBS, HBO, NHW, NHO) as shown in Table 93.

**Table 93 Mapping/concatenation of trip tables by trip purposes into peak and off-peak period trip tables prior to transit assignment**

Before combining trip tables	After combining trip tables
i4_HBW_NL_MC.MTT	i4_AMMS.TRP
i4_HBS_NL_MC.MTT i4_HBO_NL_MC.MTT i4_NHW_NL_MC.MTT i4_NHO_NL_MC.MTT	i4_OPMS.TRP

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

WK\_CR, WK\_BUS, WK\_BUS\_MR, WK\_MR,

PNR\_KNR\_CR, PNR\_BUS, KNR\_BUS, PNR\_BUS\_MR, KNR\_BUS\_MR, PNR\_MR, KNR\_MR

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

transit\_assignment\_CR.s  
transit\_assignment\_MR.s  
transit\_assignment\_AB.s  
transit\_assignment\_BM.s

### 24.1.1 Inputs to the transit assignment

As can be seen on page A-15 of Appendix A, the specific list of inputs for transit assignment varies for each of the four transit submodes.

### 24.1.2 Outputs of the transit assignment

The output of the four transit assignment scripts are a series of transit link files and transit node files in dBase (DBF) format. These files are generated in Cube Voyager's TRNBUILD module using the LINKO and NODEO keywords. The transit node files (NODEO) simply contain the node number and its X and Y coordinates, as shown in Figure 34.

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

Figure 34 Excerpt from one of the transit node DBF files output from transit assignment (i4\_WKMRAMnode.dbf)

Ref: "X:\modelRuns\fy12\Ver2.3.36\2007\_pseu\i4\_WKMRAMnode.dbf"

Transit link files (LINKO) files include the following attributes.<sup>132</sup>

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

The following additional attributes are included due to transit assignment:

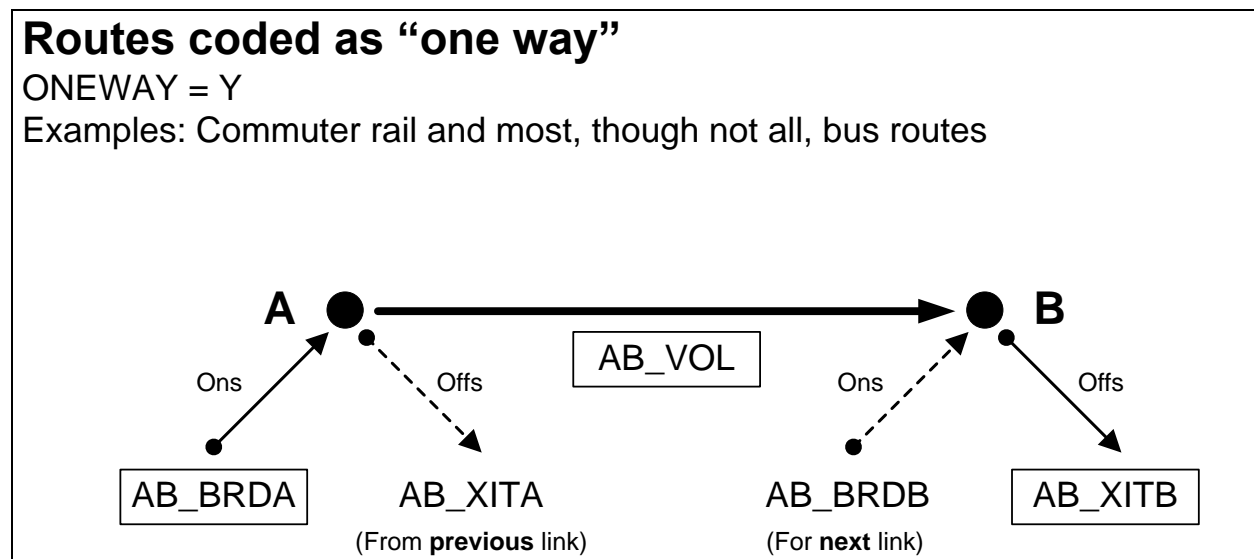
- SEQ: Link sequence in the line
- OWNER: Line owner (first ten characters)
- AB\_VOL: Volume
- AB\_BRDA: Number of trip boardings at A
- AB\_XITA: Number of exits at A

<sup>132</sup> Citilabs, Inc., "Cube Voyager Reference Guide, Version 6.0.2," 958.

- AB\_BRDB: Number of boardings at B
- AB\_XITB: Number of exits at B
- (last 5 variables are also repeated for B-A direction)

Figure 35 and Figure 36 show the naming conventions used for transit volumes from a TRNBUILD-based transit assignment. Both figures show the associated volumes (“ons,” “throughs,” and “offs”) for a hypothetical transit link AB. Figure 35 is for the case of a one-way transit route, and Figure 36 is for the case of a two-way transit route. These figures can also be useful when interpreting reports from the LineSum transit assignment summary program (covered in the next section of the report).

Figure 35 Transit volumes from transit assignment using TRNBUILD: One-way route



Ref: transit\_volumes\_ab\_ba\_trnbuild\_v2.vsd

Note: For a description of AB\_VOL, AB\_BRDA, AB\_XITA, etc., see page 958, Cube Voyager Reference Guide, Version 6.0.2. Citilabs, Inc., July 26, 2012.

The simplest case is the one-way route (Figure 35). In this case, the three important values for the link AB are:

- AB\_VOL: Transit person trips on link AB (“throughs”)
- AB\_BRDA: Transit person boardings (“ons”) at the “from” node (node A in the figure)
- AB\_XITB: Transit person alightings (“offs”) from the “to” node (node B in the figure)

All three of these variables are shown in rectangular boxes in Figure 35. The other two values shown in Figure 35 (AB\_XITA and AB\_BRDB) are associated with the **link prior to link AB** (AB\_XITA) and the **link after link AB** (AB\_BRDB).

For routes coded as two-way (Figure 36), the situation is similar, but a bit more complex. When traveling in the A-to-B direction, the three important variables for transit volumes are the same as before:

- AB\_VOL: Transit person trips on link AB (“throughs”)
- AB\_BRDA: Transit person boardings (“ons”) at the “from” node (node A in the figure)

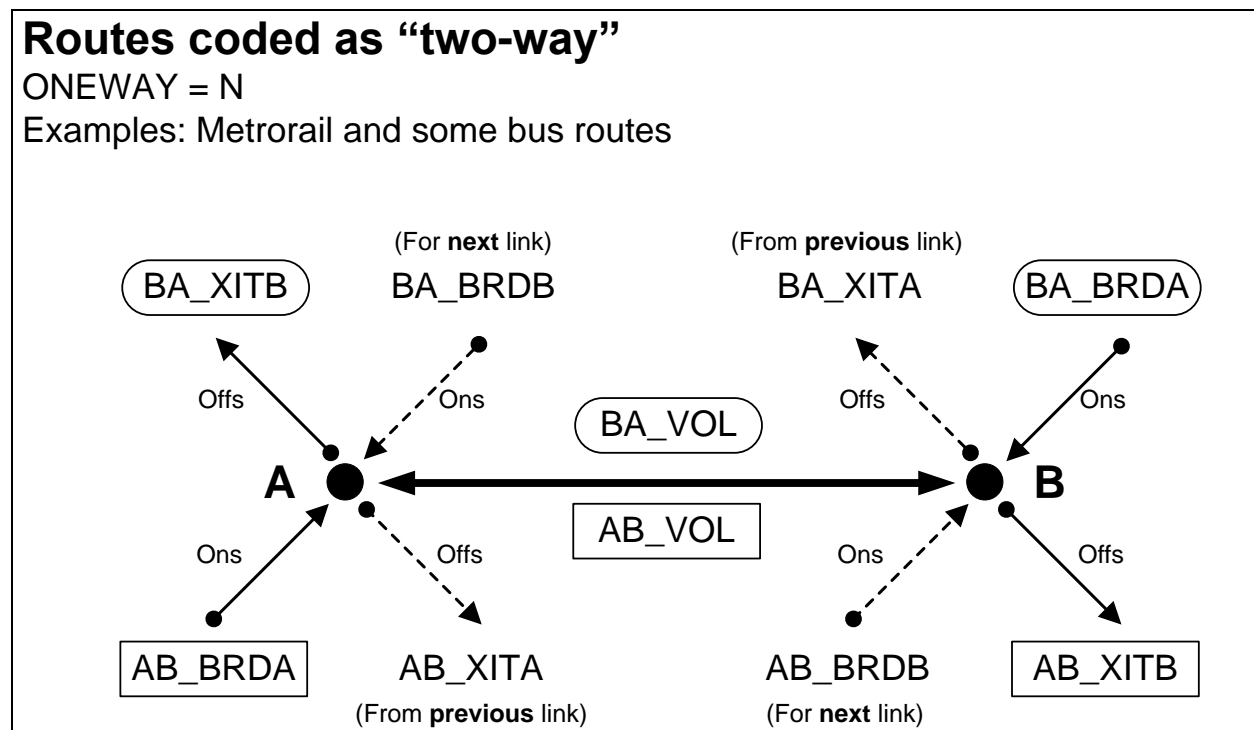
- AB\_XITB: Transit person alightings (“offs”) from the “to” node (node B in the figure)

However, when travelling in the B-to-A direction, the three relevant variables are:

- BA\_VOL: Transit person trips on link AB in the B-to-A direction (“throughs”)
- BA\_BRDA: Transit person boardings (“ons”) in the B-to-A direction at the “from” node (node B in the figure)<sup>133</sup>
- BA\_XITB: Transit person alightings (“offs”) in the B-to-A direction from the “to” node (node A in the figure)

These are indicated in Figure 36 with rectangular boxes that have rounded corners.

Figure 36 Transit volumes from transit assignment using TRNBUILD: Two-way route



Ref: transit\_volumes\_ab\_ba\_trnbuild\_v2.vsd

Note: For a description of AB\_VOL, AB\_BRDA, AB\_XITA, etc., see page 958, Cube Voyager Reference Guide, Version 6.0.2. Citilabs, Inc., July 26, 2012.

Keep in mind that, since transit path-building and assignment are conducted in production/attraction (P/A) format, **all of the values on these tables are also in P/A format**. Conducting transit assignment in production/attraction format is state of the practice for transit assignments and has the benefit of

<sup>133</sup> Typically, the convention is that the “from” node is the A node and the “to” node is the B node. However, in Figure 36, for movement in the B-to-A direction, the “from” node is labeled B and the “to” node is labeled A, since those were the labels used for movement in the A-to-B direction.

showing the peak orientation of the transit line. **To estimate the boardings at a given station in origin/destination format, you need to add the “ons” and “offs” together and divide by two.**<sup>134</sup>

Some examples of the LINKO attribute values can be found in Figure 37 through Figure 39. For example, Figure 37 shows a portion of the AM walk-access to Metrorail LINKO file (i4\_WKMRAMlink.dbf) that has mode-16 links (walk access to transit). Similarly, Figure 38 shows a portion of the AM walk-access to Metrorail LINKO file (i4\_WKMRAMlink.dbf) that has mode-3 links (Metrorail line segments). Lastly, Figure 39 shows a portion of the AM walk-access to Metrorail LINKO file (i4\_WKMRAMlink.dbf) that has mode-12 links (walk transfer links).

---

<sup>134</sup> AECOM, “LineSum (Version 5.0.17)” (Arlington, Virginia: AECOM, June 13, 2012), 14.

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

Figure 37 Excerpt from one of the transit link DBF files output from transit assignment (i4\_WKMRAMlink.dbf) showing mode-16 links

Ref: "X:\modelRuns\fy12\Ver2.3.36\2007\_pseu\i4\_WKMRAMlink.dbf"

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

Figure 38 Excerpt from one of the transit link DBF files output from transit assignment (i4\_WKMRAMlink.dbf) showing mode-3 links

Ref: "X:\modelRuns\fy12\Ver2.3.36\2007\_pseu\i4\_WKMRAMlink.dbf"

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

Figure 39 Excerpt from one of the transit link DBF files output from transit assignment (i4\_WKMRAMlink.dbf) showing mode-12 links

Ref: "X:\modelRuns\fy12\Ver2.3.36\2007\_pseu\i4\_WKMRAMlink.dbf"

## 24.2 Transit assignment summary process

The purpose of the transit assignment *summary* process is to summarize the output from the transit assignment process. The process is run with the `transum.bat` batch file, which, like the `Transit_Assignment_Parallel.bat` batch file, is called from the “run model steps” batch file (see page A-16 of Appendix A). Whereas the transit assignment process is run in the scenario/output folder (e.g., `2016_Final`), the transit assignment *summary* process is run in the **transum** folder (e.g., `2016_Final\transum`), which is a subfolder of the scenario/output folder. In the past, such as the Version 2.3.36 model, there were two transit assignment summary programs (`LineVol` and `LineSum`). `LineVol` was used to merge transit assignment output files into peak and off-peak files. Now, there is only one of these (`LineSum`, ver. 6.0.2),<sup>135</sup> since `LineSum` performs all the functionality needed, including the merging of output files.

An excerpt from the `transum.bat` batch file is shown in Figure 40 and the complete batch file can be found in Appendix B. When the model run is begun, the “transum” folder must exist under the scenario folder (e.g., `2010_Final\transum`), **but the folder is completely empty**. The control files needed to run `LineSum` are stored in the “controls” folder. Although the station names file (`station_names.dbf`) used to be stored in the “controls” folder, this file is now generated by `set_factors.s` and is stored in the “inputs” folder (this change occurred in the Ver. 2.3.57a). The station names file includes Metrorail stations, commuter rail station, and other “named” nodes, such as the following:

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

Once the transit assignment summary process is finished, the folder will include both a copy of the control files that were used and the report files generated by `LineSum`.

The following control files, associated with `LineSum`, are stored in the “controls” folder and are called by the `transum.bat` batch file:

```
LineSum_Volume.ctf
lineSum_MR_access.ctf
lineSum_MR_line.ctf
```

These files are described below and the model user can always develop more control files to generate more reports.

<sup>135</sup> AECOM, *LineSum*, version 6.0.2 (Arlington, Virginia: AECOM, 2014).



At the beginning of the transum.bat batch file (line 8 in Figure 40), the change directory command is used to change the working directory to the “transum” folder. In line 11, a local copy of the LineSum control files is made in the transum folder. In line 14, we create a peak-period and off-peak period file containing the transit assignment. In line 18 of Figure 40, we generate a Metrorail station access report. This station access report does not include transfers from one Metrorail line to another, just the number of boardings at each station. Lastly, in line 22, we create line summaries for the Metrorail system.

Figure 40 An excerpt of tranSum.bat transit summary batch file

```
1  :: TranSum.bat
2  :: To be run from the root directory (e.g., E:\modelRuns\fy13\Ver2.3.46)
3
4
5  REM Change to the Transum folder, under the scenario-specific folder
6  REM Output report files will be stored in the Transum folder
7  REM The Transum folder starts out empty, since station_names.dbf is stored in Controls
8  CD %1\Transum
9
10 REM Copy the lineSum control files from the Controls folder to the Transum folder
11 copy ..\..\Controls\LineSum*.ctl
12
13 REM Consolidate peak and off-peak volumes from transit assignment
14 ..\..\software\LineSum.exe LineSum_Volume.ctl
15 if %ERRORLEVEL% == 1 goto error
16
17 REM Metrorail station access (does not include transfers)
18 ..\..\software\LineSum.exe lineSum_MR_access.ctl
19 if %ERRORLEVEL% == 1 goto error
20
21 REM Metrorail line summaries
22 ..\..\software\LineSum.exe lineSum_MR_line.ctl
23 if %ERRORLEVEL% == 1 goto error
24
25 (etc.)
```

Ref: M:\fy17\CGV2\_3\_66\_Conformity2016CLRP\_Xmittal\TranSum.bat

### 24.2.1 Consolidating transit assignment output and displaying results

As shown on page A-16 of Appendix A, the LineSum\_Volume.ctl (Figure 41) control file is used to consolidate the transit assignment volume DBF files into two summary volume files, one for the peak period (PK\_VOL.DBF, equal to the HBW transit volumes) and one for the off-peak period (OP\_VOL.DBF, equal to the sum of the HBS, HBO, NHW, and NHO transit volume files).

Figure 41 Consolidating peak and off-peak transit assignment volumes (LineSum\_Volume.ctl)

1	TITLE	Merge the Transit Volumes	
2			
3	DEFAULT_FILE_FORMAT	DBASE	
4			
5	PEAK_RIDERSHIP_FILE_1	..\i4_DRABAMlink.dbf	//DRIVE ACCESS
6	PEAK_RIDERSHIP_FILE_2	..\i4_DRBMAMlink.dbf	
7	PEAK_RIDERSHIP_FILE_3	..\i4_DRCRAMlink.dbf	
8	PEAK_RIDERSHIP_FILE_4	..\i4_DRMRAMlink.dbf	
9	PEAK_RIDERSHIP_FILE_5	..\i4_KRABAMlink.dbf	//KISS AND RIDE ACCESS
10	PEAK_RIDERSHIP_FILE_6	..\i4_KRBAMlink.dbf	
11	PEAK_RIDERSHIP_FILE_7	..\i4_KMRAMlink.dbf	
12	PEAK_RIDERSHIP_FILE_8	..\i4_WKABAMlink.dbf	//WALK ACCESS
13	PEAK_RIDERSHIP_FILE_9	..\i4_WKBAMlink.dbf	
14	PEAK_RIDERSHIP_FILE_10	..\i4_WKCRAMlink.dbf	
15	PEAK_RIDERSHIP_FILE_11	..\i4_WKMRAMlink.dbf	
16			
17	OFFPEAK_RIDERSHIP_FILE_1	..\i4_DRABOPlink.dbf	//DRIVE ACCESS
18	OFFPEAK_RIDERSHIP_FILE_2	..\i4_DRBMOPlink.dbf	
19	OFFPEAK_RIDERSHIP_FILE_3	..\i4_DRCROPlink.dbf	
20	OFFPEAK_RIDERSHIP_FILE_4	..\i4_DRMROPlink.dbf	
21	OFFPEAK_RIDERSHIP_FILE_5	..\i4_KRABOPlink.dbf	//KISS AND RIDE ACCESS
22	OFFPEAK_RIDERSHIP_FILE_6	..\i4_KRBMOPlink.dbf	
23	OFFPEAK_RIDERSHIP_FILE_7	..\i4_KRMROPlink.dbf	
24	OFFPEAK_RIDERSHIP_FILE_8	..\i4_WKABOPlink.dbf	//WALK ACCESS
25	OFFPEAK_RIDERSHIP_FILE_9	..\i4_WKBMOPlink.dbf	
26	OFFPEAK_RIDERSHIP_FILE_10	..\i4_WKCROPlink.dbf	
27	OFFPEAK_RIDERSHIP_FILE_11	..\i4_WKMROPlink.dbf	
28			
29	NEW_PEAK_RIDERSHIP_FILE	PK_VOL.dbf	
30	NEW_PEAK_RIDERSHIP_FORMAT	DBASE	
31	NEW_OFFPEAK_RIDERSHIP_FILE	OP_VOL.dbf	
32	NEW_OFFPEAK_RIDERSHIP_FORMAT	DBASE	

The output from the LineSum\_Volume.ctl process is pk\_vol.dbf and op\_vol.dbf. Either of these transit loaded link files can be brought into Cube Base as the transit layer, as is shown in Figure 42 through Figure 46.

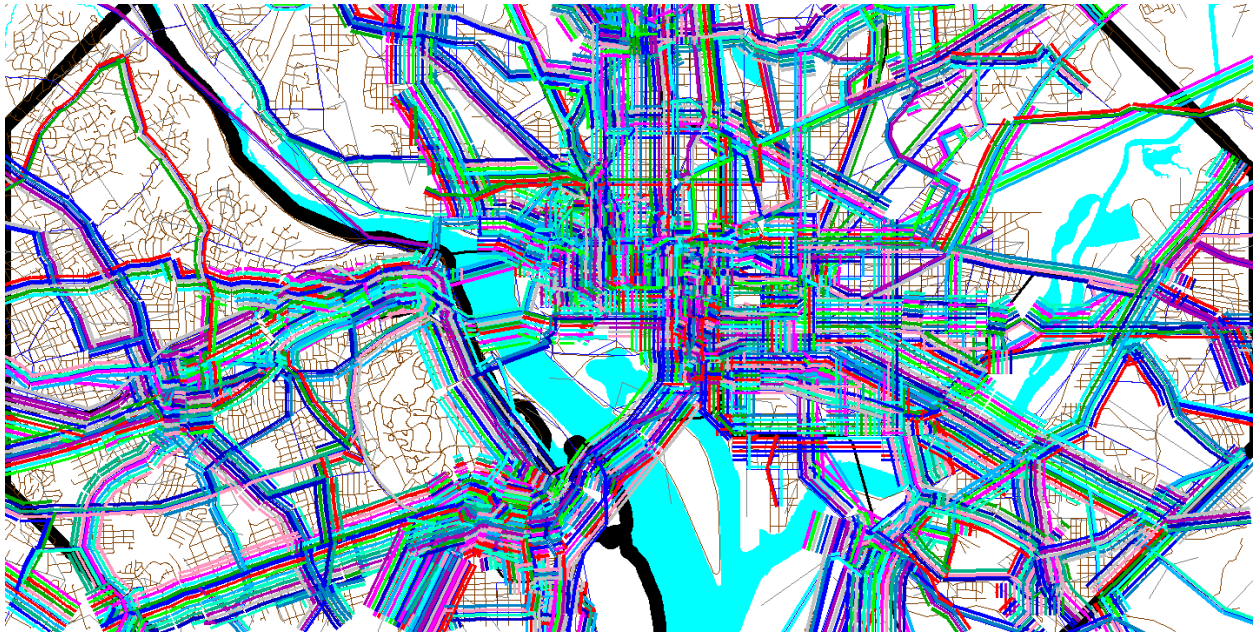


Figure 42 Using the pk\_vol.dbf file in Cube Base as the transit layer: All transit routes turned on, but non-transit links (modes 11-16) turned off

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

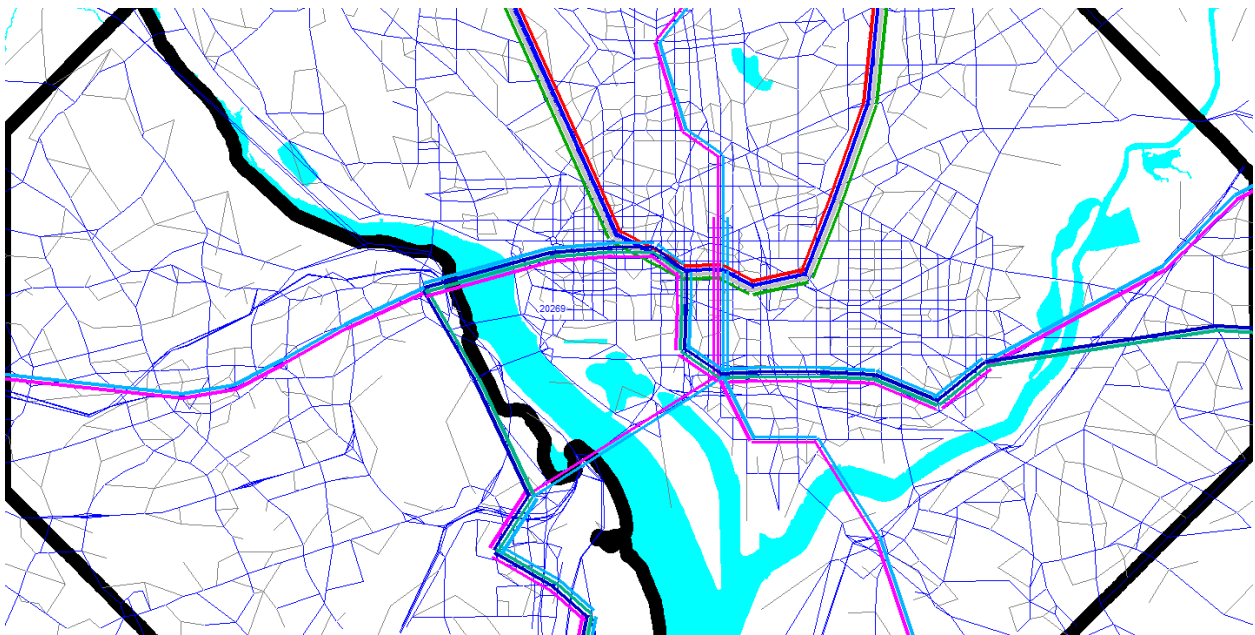


Figure 43 Using the pk\_vol.dbf file in Cube Base as the transit layer: Only mode-3 (Metrorail) links turned on

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

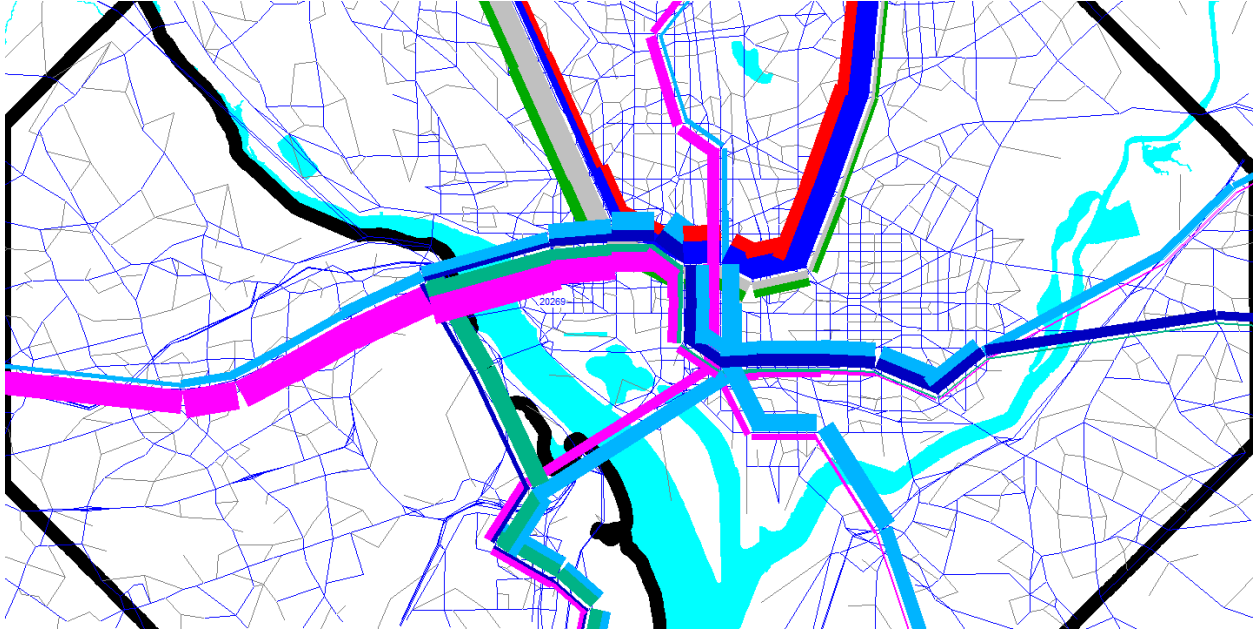


Figure 44 Using the pk\_vol.dbf file in Cube Base as the transit layer: Only mode-3 (Metrorail) links turned on; using multi-bandwidth to represent transit loads (ab\_vol): Arlington and DC

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

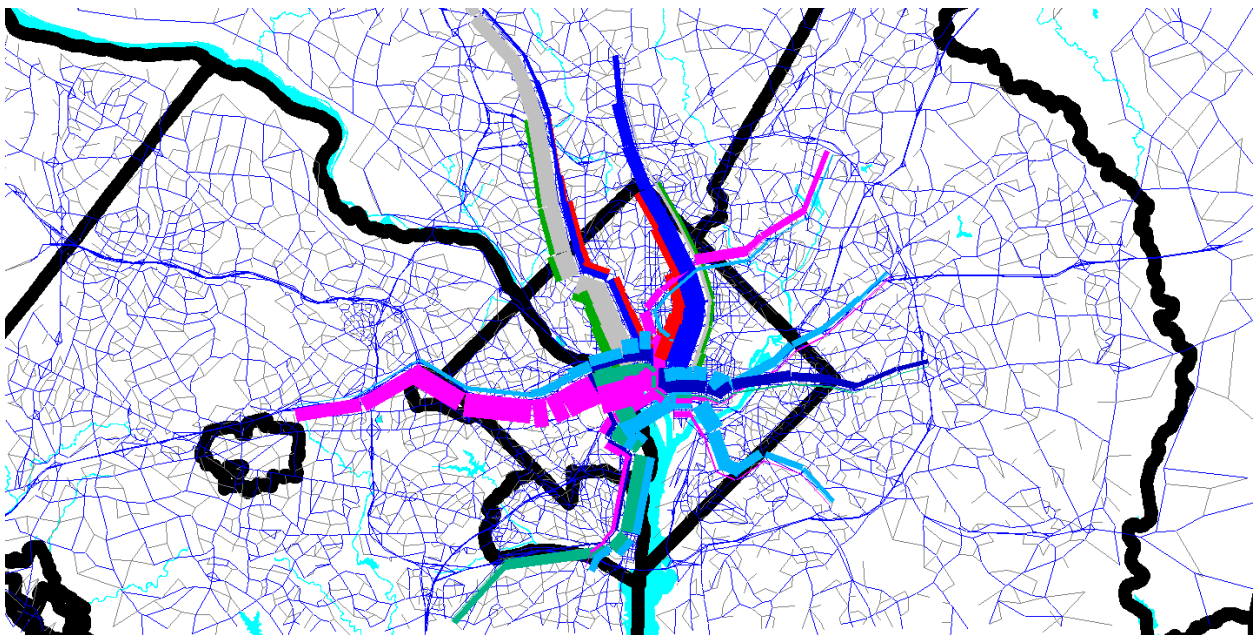


Figure 45 Using the pk\_vol.dbf file in Cube Base as the transit layer: Only mode-3 (Metrorail) links turned on; using multi-bandwidth to represent transit loads (ab\_vol): Metrorail system

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

**Transit Layer Link Band Width Settings**

Set: 1:  Name:

Attributes	Color Settings	value/pixel	Value Range
AB_VDL	<input checked="" type="radio"/> Link Color <input type="radio"/> Fix Color <input type="radio"/> Dynamic Color	2000	0-66856
<input type="text"/>	<input checked="" type="radio"/> Link Color <input type="radio"/> Fix Color <input type="radio"/> Dynamic Color	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input checked="" type="radio"/> Link Color <input type="radio"/> Fix Color <input type="radio"/> Dynamic Color	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input checked="" type="radio"/> Link Color <input type="radio"/> Fix Color <input type="radio"/> Dynamic Color	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input checked="" type="radio"/> Link Color <input type="radio"/> Fix Color <input type="radio"/> Dynamic Color	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input checked="" type="radio"/> Link Color <input type="radio"/> Fix Color <input type="radio"/> Dynamic Color	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input checked="" type="radio"/> Link Color <input type="radio"/> Fix Color <input type="radio"/> Dynamic Color	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input checked="" type="radio"/> Link Color <input type="radio"/> Fix Color <input type="radio"/> Dynamic Color	<input type="text"/>	<input type="text"/>

Selection Criteria:

Scale Range to Show Posting  to

Key Value Key1  Key2  Key3  Key4

Key Min. Width Key1  Key2  Key3  Key4

Figure 46 Using the multi-bandwidth option in Cube Base to show transit volumes on the Metrorail system.

## 24.2.2 Generating transit assignment summaries using LineSum

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

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

An example of a control file used to generate an access report showing riders who arrive at and depart from Metrorail stations (via transit access links) can be seen in Figure 47. The station\_names.dbf file is now stored in the “inputs” folder (not the “controls”) folder.

Figure 47 Generating a Metrorail station access report (lineSum\_MR\_access.ctf)

```

1  ## Access reports focus on riders who arrive or depart using transit access links
2  ## i.e., the summary does not include transfers
3  TITLE Metrorail Station Access Summary
4  DEFAULT_FILE_FORMAT DBASE
5
6  PEAK_RIDERSHIP_FILE_1 PK_VOL.DBF
7  PEAK_RIDERSHIP_FORMAT_1 DBASE
8  OFFPEAK_RIDERSHIP_FILE_1 OP_VOL.DBF
9  OFFPEAK_RIDERSHIP_FORMAT_1 DBASE
10
11 STOP_NAME_FILE ..\inputs\station_names.dbf
12 STOP_NAME_FORMAT DBASE
13
14 ACCESS_REPORT_TITLE_1 All
15 ACCESS_REPORT_STOPS_1 8001..8100, 8119..8140, 8145..8148, 8150..8154,
16 8160..8166, 8169..8182
17 ##ACCESS_REPORT_MODES_1 11,12,14,15,16
18 ACCESS_REPORT_MODES_1 ALL
19 ##ACCESS_REPORT_DETAILS_1 MODE
20 NEW_ACCESS_REPORT_FILE_1 MR_access.txt
21 NEW_ACCESS_REPORT_FORMAT_1 TAB_DELIMITED
    
```

Similarly, an example of a control file used to generate a Metrorail line summary can be seen in Figure 48. Once again, the station\_names.dbf file is now stored in the “inputs” folder (not the “controls”) folder.

Figure 48 Generating a Metrorail line summary (lineSum\_MR\_line.ctf)

```

1  ## Line reports summarize boardings, alightings, and ridership for one or more line
2  TITLE Metrorail Line Summary
3  DEFAULT_FILE_FORMAT DBASE
4
5  PEAK_RIDERSHIP_FILE_1 PK_VOL.DBF
6  PEAK_RIDERSHIP_FORMAT_1 DBASE
7  OFFPEAK_RIDERSHIP_FILE_1 OP_VOL.DBF
8  OFFPEAK_RIDERSHIP_FORMAT_1 DBASE
9
10 STOP_NAME_FILE ..\inputs\station_names.dbf
11 STOP_NAME_FORMAT DBASE
12
13 LINE_REPORT_TITLE_1 All
14 LINE_REPORT_LINES_1 All
15 LINE_REPORT_MODES_1 3
    
```

16	NEW_TOTAL_RIDERSHIP_FILE_1	MR_line.txt
17	NEW_TOTAL_RIDERSHIP_FORMAT_1	TAB_DELIMITED

An example of the report generated by the lineSum\_MR\_**access**.ctl control file can be found in Figure 49. Similarly, an example of the report generated by the lineSum\_MR\_**line**.ctl control file can be found in Figure 50.

More information about using LineSum can be found in its documentation:

- AECOM. (2013). LineSum, Quick Reference, Version 5.0.17. Arlington, Virginia: AECOM.
- AECOM. (2014). LineSum (Version 6.0.2). Arlington, Virginia: AECOM.



Figure 49 An excerpt from the report file generated by lineSum\_MR\_access.ct

```

*****
|                                     |
|      LineSum - Version 5.0.17      |
|      Copyright 2012 by TRANSIMS Open-Source |
|      Fri Jul 26 08:42:13 2013      |
|                                     |
*****

Control File = lineSum_MR_access.ct1
Report File = lineSum_MR_access.prn (Create)

Metrorail Station Access Summary

Default File Format = DBASE

LineSum Control Keys:

Peak Ridership File #1 = PK_VOL.DBF

Offpeak Ridership File #1 = OP_VOL.DBF

Stop Name File = ..\..\controls\station_names.dbf

Access Report Title = All
Access Report Stops = 8001..8100, 8119..8140, 8145..8148, 8150..8154, 8160..8166, 8169..8182
Access Report Modes = ALL
New Access Report File #1 = MR_access.txt
New Access Report Format #1 = TAB_DELIMITED

Number of Stop Names = 267 Metrorail Station Access Summary
Fri Jul 26 08:42:13 2013 LineSum page 2

Title: All
Modes: All

      ---- Peak ----  -- Offpeak ---  ---- Daily ---
Stop  Arrive Depart Arrive Depart Arrive Depart
8001  25762  1323  2216  773  27978  2096  Shady Grove
8002  12652  2238  1363  889  14015  3127  Rockville
8003  4537  2836  1376  1500  5913  4336  Twinbrook
8004  3589  2952  1009  1437  4598  4389  White Flint
8005  8768  213  1775  845  10543  1058  Grosvenor
8006  3979  8967  1016  2060  4995  11027  Medical Center
    
```



8007	10802	14528	3884	5686	14686	20214	Bethesda
8008	9886	6177	3634	2954	13520	9131	Friendship Heights
8009	7559	3581	2565	1654	10124	5235	Tenleytown
8010	5066	2765	1945	1510	7011	4275	Van Ness-UDC
8011	4066	885	1653	840	5719	1725	Cleveland Park
8012	6790	2388	2934	1636	9724	4024	Woodley Park-Zoo
8013	9104	27194	3486	7314	12590	34508	Dupont Circle
8014	2588	38091	764	7571	3352	45662	Farragut North
8015	2071	21835	446	3509	2517	25344	Metro Center
8016	814	13274	1662	3035	2476	16309	Gallery Place
8017	304	17877	1303	3823	1607	21700	Judiciary Square
8018	19955	22256	3818	6521	23773	28777	Union Station

Figure 50 The report file generated by lineSum\_MR\_line.ct1

```

*****
|                                     |
|      LineSum - Version 5.0.17       |
|      Copyright 2012 by TRANSIMS Open-Source |
|      Fri Jul 26 08:42:13 2013      |
|                                     |
*****

Control File = lineSum_MR_line.ct1
Report File = lineSum_MR_line.prn (Create)

Metrorail Line Summary

Default File Format = DBASE

LineSum Control Keys:

Peak Ridership File #1 = PK_VOL.DBF

Offpeak Ridership File #1 = OP_VOL.DBF

Stop Name File = ..\..\controls\station_names.dbf

Line Report Title = All
Line Report Lines = All
Line Report Modes = 3

Number of Stop Names = 267      Metrorail Line Summary
Fri Jul 26 08:42:14 2013 LineSum page 2

Title: All
Lines: All
Modes: 3

----- A->B Direction (Read Down) ----- B->A Direction (Read Up) -----Total-----
Dist  Time  -----Peak-----  -----Off-Peak-----  -----Daily-----  -----Peak-----  -----Off-Peak-----  -----Daily-----  -----Daily-----
Stop   (miles) (min)  On  Off  Ride  On  Off  Ride  On  Off  Ride  On  Off  Ride  On  Off  Ride  On  Off  Ride
Franconia-  3.49  6.14  12595  0  12595  1332  0  1332  13927  0  13927  0  718  718  0  288  288  0  1006  1006  13927  1006  14933
Van Dorn S  3.86  4.95  7805  98  20300  1453  23  2760  9258  121  23060  119  432  1027  19  543  812  138  975  1839  9396  1096  24899
King Stree  0.68  2.02  4245  3189  37000  950  445  5882  5195  3634  42882  341  2076  5614  191  967  2937  532  3043  8551  5727  6677  51433
Braddock R  2.80  4.46  5735  1215  41522  1637  857  6661  7372  2072  48183  393  2863  8083  460  1567  4042  853  4430  12125  8225  6502  60308
National A  0.49  2.59  1  971  40554  422  131  6954  423  1102  47508  7  2720  10796  85  895  4851  92  3615  15647  515  4717  63155
Crystal Ci  0.76  2.02  3898  5097  39352  2678  1197  8434  6576  6294  47786  340  18366  28824  608  6286  10532  948  24652  39356  7524  30946  87142
Pentagon C  0.61  0.99  6870  1249  44973  2106  796  9743  8976  2045  54716  1391  4273  31705  1251  2124  11401  2642  6397  43106  11618  8442  97822
Pentagon A  1.24  2.91  12252  5065  32508  1345  1073  6066  13597  6138  38574  3152  2920  11440  1250  1053  5292  4402  3973  16732  17999  10111  55306
Arlington  0.99  2.08  48  0  32556  42  0  6107  90  0  38663  69  0  11370  35  0  5257  104  0  16627  194  0  55290
Rosslyn  1.35  3.12  6219  12112  81284  1574  3597  13273  7793  15709  94557  9984  11122  29578  3809  3735  10601  13793  14857  40179  21586  30566  134736
Foggy Bott  0.57  2.08  1498  12600  70182  1533  2568  12242  3031  15168  82424  813  19224  47990  629  5194  15171  1442  24418  63161  4473  39586  145585
Farragut W  0.38  0.97  642  17242  53580  313  1808  10748  955  19050  64328  1196  12655  59447  358  1750  16567  1554  14405  76014  2509  33455  140342
McPherson  0.46  1.09  3139  14641  42079  643  2224  9170  3782  16865  51249  2359  23637  80724  1134  3172  18603  3493  26809  99327  7275  43674  150576
    
```



Shady Grov	2.61	4.06	25767	0	25767	2221	0	2221	27988	0	27988	0	1321	1321	0	775	775	0	2096	2096	27988	2096	30084
Rockville	2.13	3.29	12366	544	37590	1209	200	3230	13575	744	40820	282	1694	2734	157	689	1307	439	2383	4041	14014	3127	44861
Twinbrook	1.09	3.08	4076	1021	40645	1120	359	3992	5196	1380	44637	460	1816	4089	258	1142	2191	718	2958	6280	5914	4338	50917
White Flin	1.35	2.03	3266	951	42965	794	325	4460	4060	1276	47425	322	2005	5773	218	1112	3084	540	3117	8857	4600	4393	56282
Grosvenor	2.19	3.05	8329	56	51234	1532	245	5748	9861	301	56982	438	156	5488	239	600	3445	677	756	8933	10538	1057	65915
Medical Ce	1.02	2.01	3812	2635	52409	830	629	5951	4642	3264	58360	164	6334	11655	180	1432	4694	344	7766	16349	4986	11030	74709
Bethesda	1.70	3.08	9559	4214	57756	2972	1957	6966	12531	6171	64722	1240	10313	20724	911	3730	7513	2151	14043	28237	14682	20214	92959
Friendship	0.91	2.41	7458	1654	63560	2132	876	8224	9590	2530	71784	2429	4523	22820	1502	2076	8088	3931	6599	30908	13521	9129	102692
Tenleytown	1.09	2.03	5604	1101	68064	1684	394	9509	7288	1495	77573	1949	2480	23351	884	1257	8460	2833	3737	31811	10121	5232	109384
Van Ness-U	0.55	1.82	4258	762	71558	1368	330	10546	5626	1092	82184	803	2003	24548	573	1180	9066	1376	3183	33614	7002	4275	115718
Cleveland	0.80	2.33	3126	209	74474	1184	217	11519	4310	426	85993	941	676	24285	472	625	9222	1413	1301	33507	5723	1727	119500
Woodley Pa	1.15	1.99	5926	589	79815	2321	406	13433	8247	995	93248	866	1801	25216	613	1230	9836	1479	3031	35052	9726	4026	128300
Dupont Cir	0.56	2.28	7870	9817	77865	2744	2025	14150	10614	11842	92015	1235	17374	41357	739	5287	14382	1974	22661	55739	12588	34503	147754
Farragut N	0.79	2.22	1309	19637	59536	487	2457	12183	1796	22094	71719	1280	18453	58531	276	5113	19219	1556	23566	77750	3352	45660	149469
Metro Cent	0.33	1.18	20227	29041	50724	5828	4273	13738	26055	33314	64462	12940	49862	95453	3194	8528	24551	16134	58390	120004	42189	91704	184466
Gallery Pl	0.33	0.92	11227	20619	41334	4283	5359	12658	15510	25978	53992	23308	17434	89580	9071	4309	19792	32379	21743	109372	47889	47721	163364
Judiciary	0.67	2.14	40	13077	28296	201	3000	9858	241	16077	38154	265	4801	94117	1102	823	19510	1367	5624	113627	1608	21701	151781
Union Stat	0.73	1.06	2033	13743	16591	753	4285	6327	2786	18028	22918	17921	8514	84710	3063	2240	18688	20984	10754	103398	23770	28782	126316
New York A	0.96	1.86	245	5411	11422	393	2205	4515	638	7616	15937	3061	2536	84182	2714	998	16972	5775	3534	101154	6413	11150	117091
Rhode Isla	0.93	2.23	710	2733	9401	748	1058	4205	1458	3791	13606	9161	1445	76467	4792	515	12696	13953	1960	89163	15411	5751	102769
Brookland-	1.30	2.98	254	1592	8068	437	894	3747	691	2486	11815	2530	1142	75078	1477	566	11781	4007	1708	86859	4698	4194	98674
Fort Totte	1.89	3.10	2605	1559	9114	1334	1233	3848	3939	2792	12962	23481	2281	53882	5236	871	7420	28717	3152	61302	32656	5944	74264
Takoma	1.47	3.22	526	2248	7393	382	1161	3070	908	3409	10463	6525	574	47931	2311	344	5451	8836	918	53382	9744	4327	63845
Silver Spr	1.75	3.25	293	6150	1537	432	2041	1462	725	8191	2999	20366	1595	29160	3496	752	2705	23862	2347	31865	24587	10538	34864
Forest Gle	1.58	4.07	51	816	771	33	692	801	84	1508	1572	4929	130	24361	994	70	1782	5923	200	26143	6007	1708	27715
Wheaton	1.75	3.31	24	580	215	22	583	241	46	1163	456	12782	125	11705	626	37	1193	13408	162	12898	13454	1325	13354
Glenmont			215			241			456		11705			1193			12898			12898		456	
L'Enfant P	2.35	4.84	10676	0	26918	4116	0	8451	14792	0	35369	0	12704	31123	0	2375	5084	0	15079	36207	14792	15079	71576
Pentagon			6887			2540			9427		11472			1136			12608			12608		9427	
King Stree	0.64	1.78	657	0	3507	179	0	1532	836	0	5039	0	464	16108	0	367	2984	0	831	19092	836	831	24131
Eisenhower	0.55	1.01	26	2821	709	31	1007	556	57	3828	1265	2277	456	14288	757	312	2541	3034	768	16829	3091	4596	18094
Huntington			709			556			1265		14288			2541			16829			16829		1265	
Greenbelt	2.44	2.88	0	0	0	1268	0	1268	1268	0	1268	0	0	0	0	231	231	0	231	231	1268	231	1499
College Pa			0			59			59		0			25			25			25		59	
Total	107.28	219.61		392452		110158		502610		396543		125097		521640			1024250						
Max	3.86	6.14	28760	29041	81284	6139	5673	14150	31422	33314	94557	47485	49862	95453	9071	8528	24551	54745	58390	120004	75566	91704	184466
Passenger Miles			2603041		478079		3081120		2110201		565601		2675802		5756922								
Passenger Hours			95058		18514		113572		82904		22839		105743		219315								
Average Trip Length (miles)			6.6		4.3		11.0		5.3		4.5		9.8		20.8								
Average Trip Length (minutes)			14.5		10.1		24.6		12.5		10.9		23.5		48.1								

Fri Jul 26 08:42:14 2013 -- Process Complete (0:00:01)

## Appendix A. Flowcharts

Ref: Ver2.3.66\_flowchart\_v2.vsd

### Flowchart Numbers associated with Flowchart Steps

This Appendix contains detailed data processing flow charts showing the relationship of input and output files to the processing steps comprising the Version 2.3 travel model. The flowcharts are arranged on the basis of the 17 batch files used in the model application. Many of the batch files are reused during the application of the model. The table below describes the sequence of each batch file used by iteration. The flowcharts are numbered in accordance with the numbering system (1-17), shown in the table below.

Batch File	Initial (Pump Prime) Iteration				
	PP	1	2	3	4
ArcPy_Walkshed_Process.bat	1				
Set_CPI.bat	2				
PP_Highway_Build.bat	3				
PP_Highway_Skims.bat	4				
Transit_Skim_All_Modes_Parallel.bat					
Transit_Skim_LineHaul_Parallel.bat			5		
Trip_Generation.bat			6		
Trip_Distribution.bat			7		
PP_Auto_Drivers.bat	8				
Time-of-Day.bat			9		
Highway_Assignment_Parallel.bat			10		
Highway_Skims.bat / HSR_Highway_Skims.bat			11		
Transit_Fare.bat				12	
Mode_Choice_Parallel.bat					
/Mode_Choice_TC_V23_Parallel.bat				13	
Auto_Driver.bat				14	
Transit_Assignment_Parallel.bat					
Transit_Assignment_LineHaul_Parallel.bat					15
Transum.bat					16-17

Ref: Append\_A\_cover\_Flowchart\_Table\_V2.3.66.xlsx





**TITLE: Process flowchart: Application of the TPB Version 2.3 Travel Model (Build 66)**

**COMPANY: COG/TPB**

**CREATOR: RM, MS, MSM**

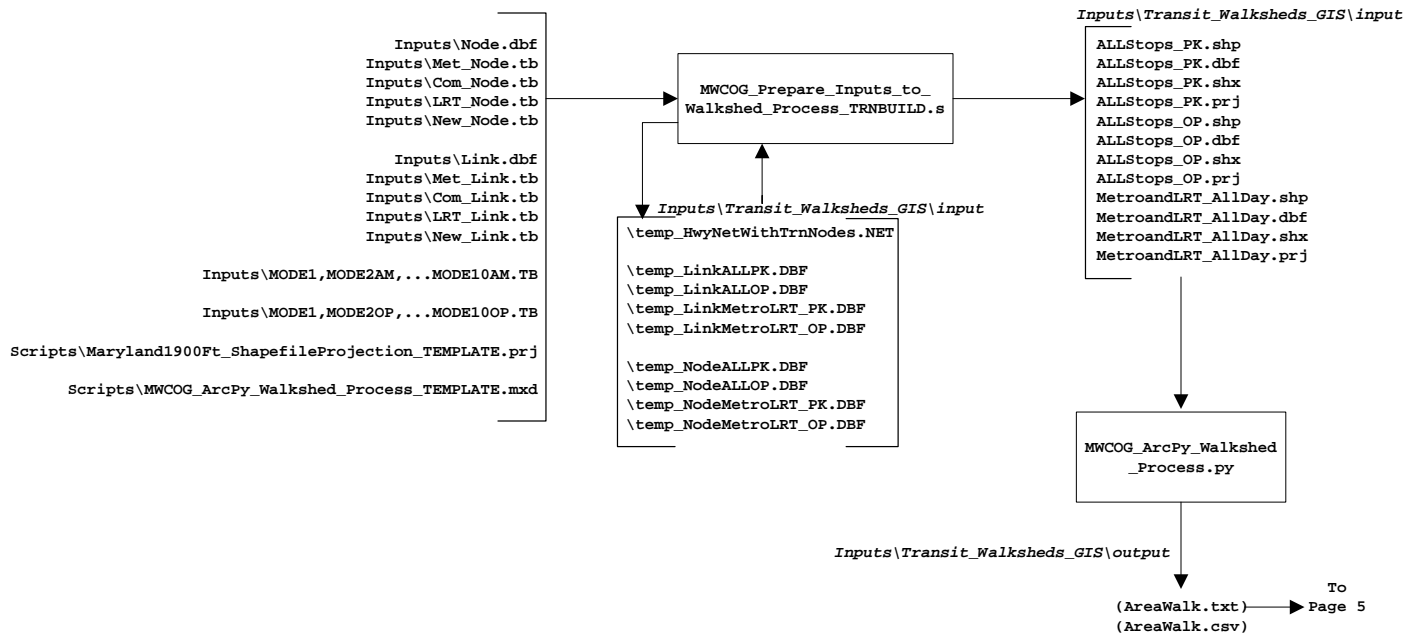
**DATE: 2/10/17**

**PG: 1**

**OF 17**

**FILENAME: Ver2.3.66\_flowchart\_v2.vsd**

## ArcPy\_Walkshed\_Process.bat



Report Files Generated by ArcPy\_Walkshed\_Process.bat:  
AreaWalk\_REPORT.prn



**TITLE: Process flowchart: Application of the TPB Version 2.3 Travel Model (Build 66)**

**COMPANY: COG/TPB**

**CREATOR: RM, MS, MSM**

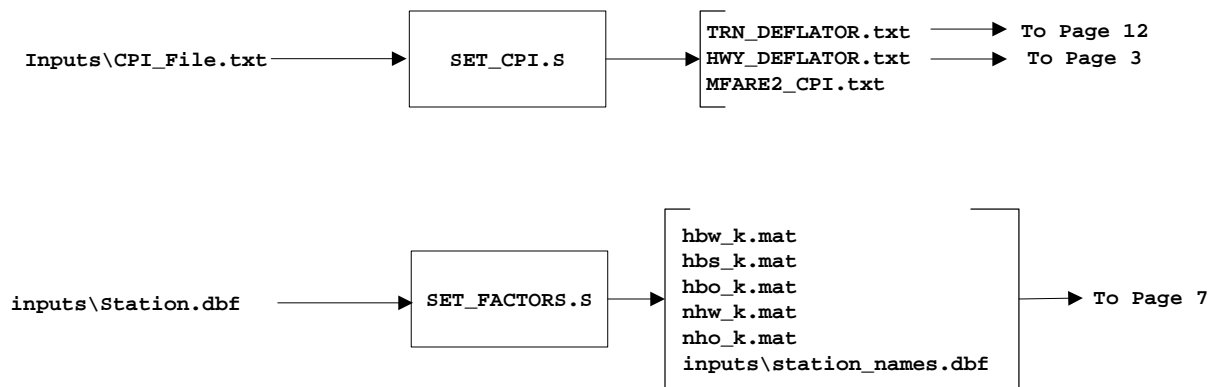
**DATE: 2/10/17**

**PG: 2**

**OF 17**

**FILENAME: Ver2.3.66\_flowchart\_v2.vsd**

### Set\_CPI.bat: Develop CPI and K-factors



Report Files Generated by Set\_CPI.bat:

Set\_CPI.rpt  
Set\_Factors.rpt





**TITLE: Process flowchart: Application of the TPB Version 2.3 Travel Model (Build 66)**

**COMPANY: COG/TPB**

**CREATOR: RM, MS, MSM**

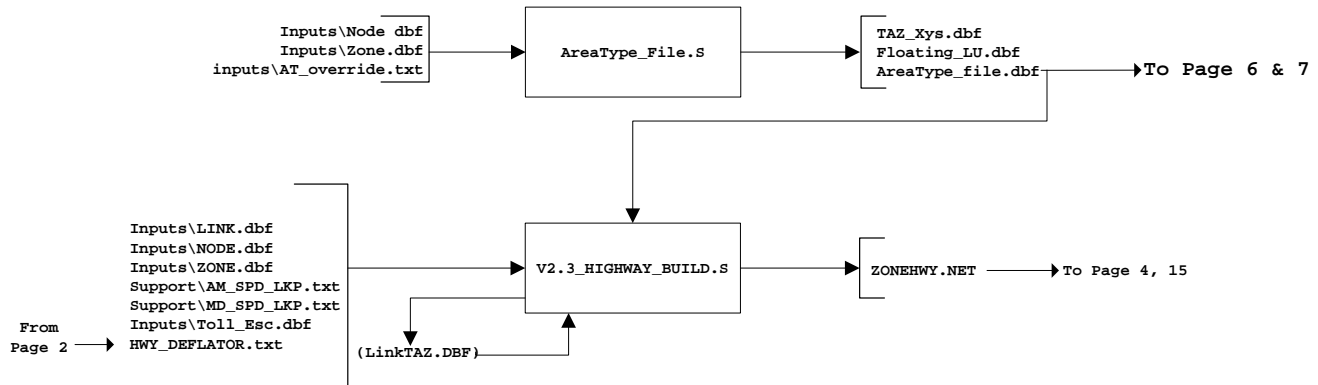
**DATE: 2/10/17**

**PG: 3**

**OF 17**

**FILENAME: Ver2.3.66\_flowchart\_v2.vsd**

## PP\_Highway\_Build.bat



## Optional

### True shape display

In Cube Open:

ZONEHWY.NET  
i1..i4\_HWY.NET  
i4\_Assign\_Output.net

Open the "Layer Control"  
- Double Click on "Polyline"  
- browse to ..support\\"True\_Shape\_2040\_Nov20.shp"

From the "GIS Tools" pull down menu select "True shape display" and click the "ON" tab.

Save the project with the same file name as the network file (ZONEHWY.VPR,i1HWY.VPR..i4HWY.VPR) in the same subdirectory.

Report Files Generated by PP\_Highway\_Build.bat:

AreaType\_File.rpt  
V2.3\_highway\_build.rpt



TITLE: Process flowchart: Application of the TPB Version 2.3 Travel Model (Build 66)

COMPANY: COG/TPB

CREATOR: RM, MS, MSM

DATE: 2/10/17

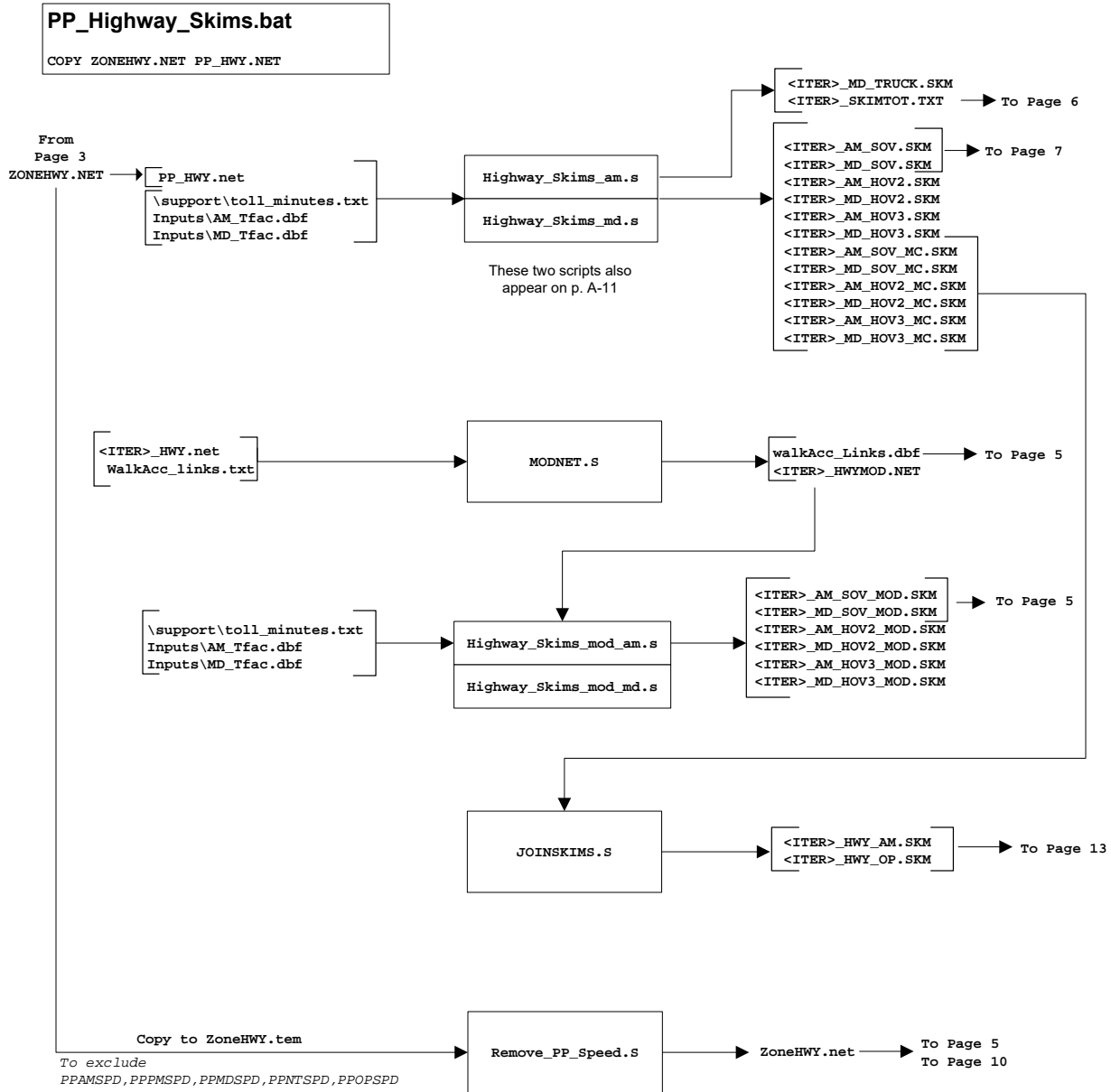
PG: 4

OF 17

FILENAME: Ver2.3.66\_flowchart\_v2.vsd

## PP\_Highway\_Skims.bat

(See also page A-11 for highway skimming process used in speed feedback iterations 1-4)



Report Files Generated by PP\_Highway\_Skims.bat:

pp\_Highway\_skims.rpt  
pp\_Joinskims.rpt  
pp\_Modnet.rpt  
pp\_Highway\_skims\_mod.rpt  
pp\_RemovePPSpeed.rpt



TITLE: Process flowchart: Application of the TPB Version 2.3 Travel Model (Build 66)

COMPANY: COG/TPB

CREATOR: RM, MS, MSM

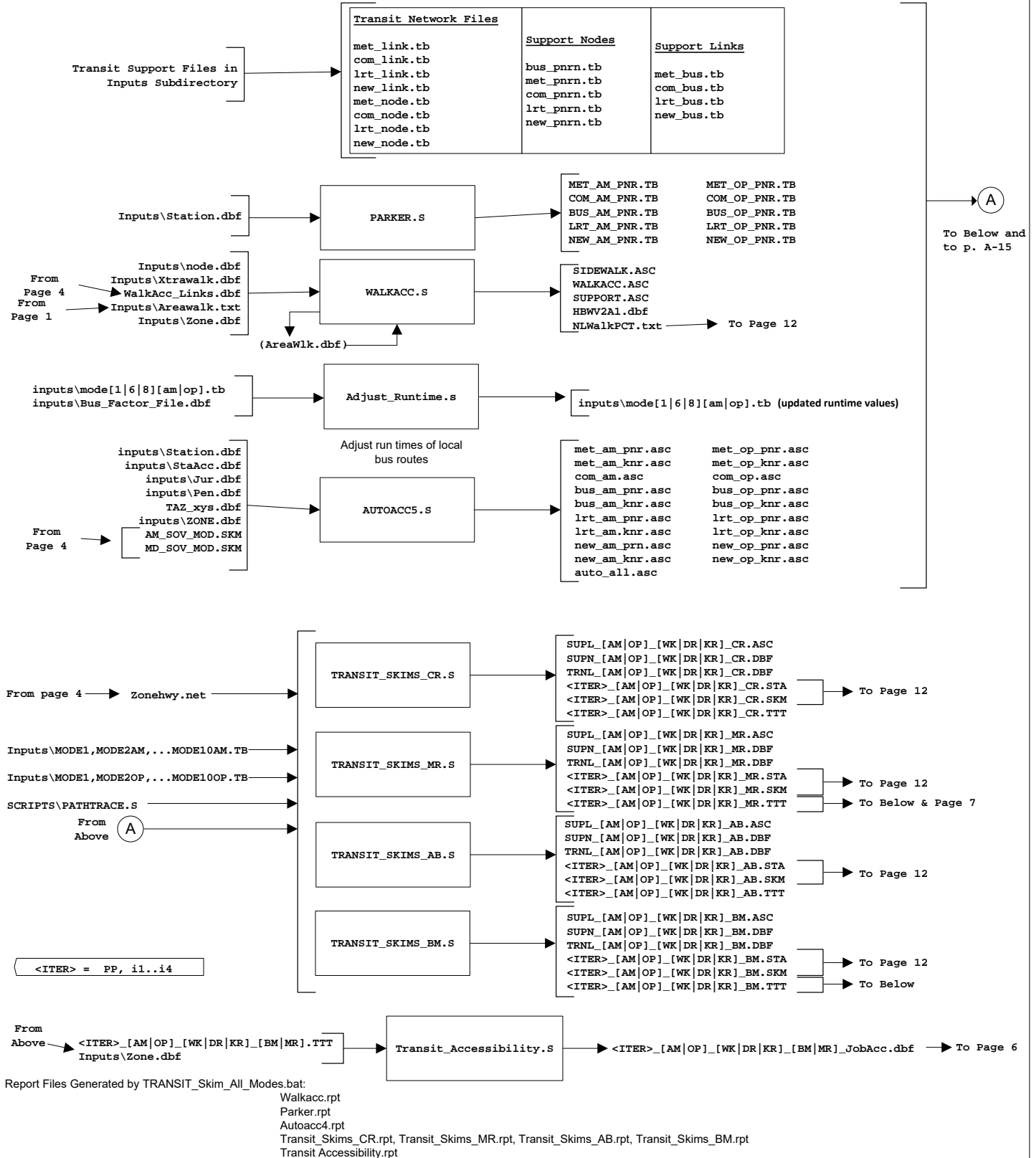
DATE: 2/10/17

PG: 5

OF 17

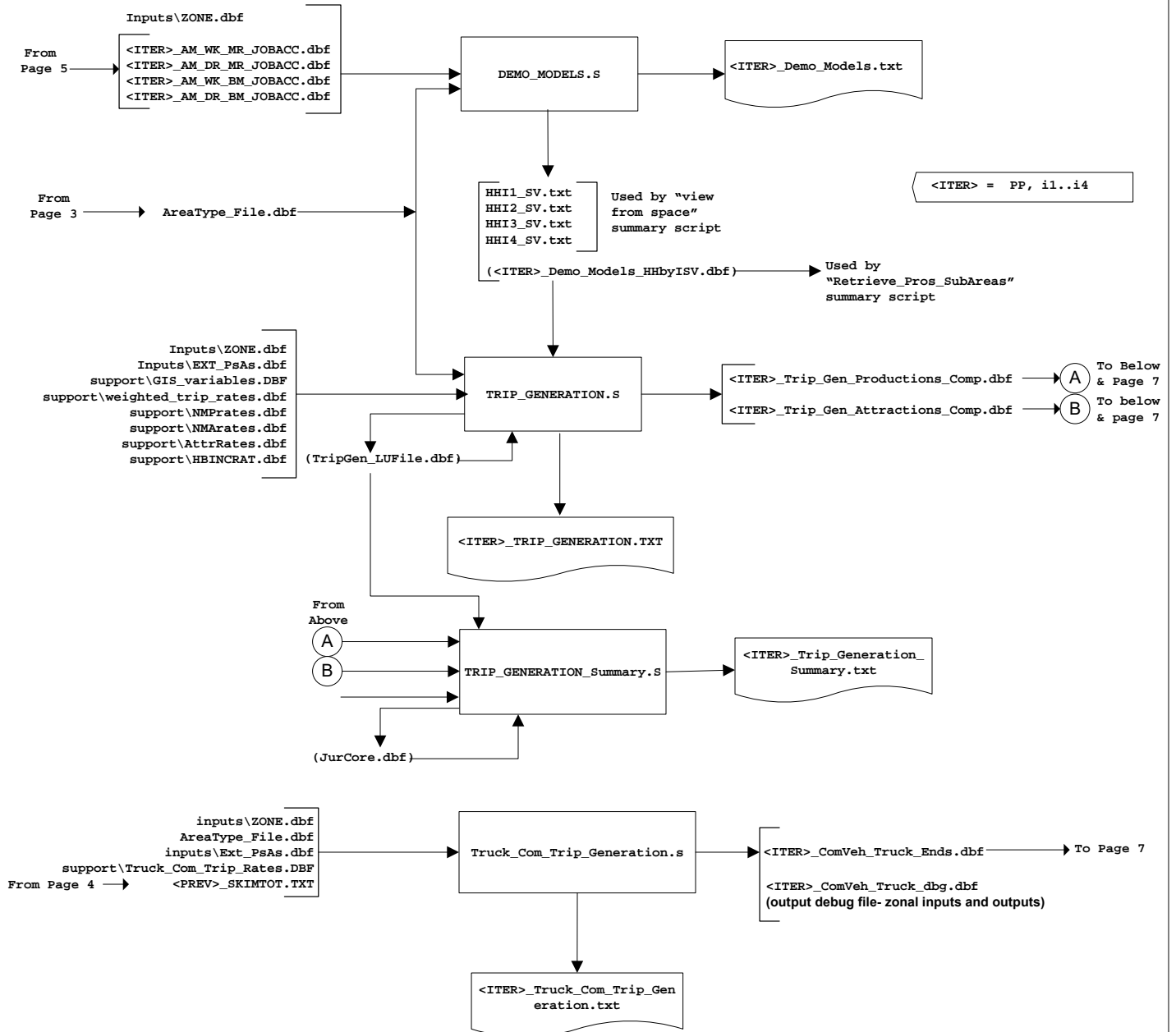
FILENAME: Ver2.3.66\_flowchart\_v2.vsd

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





### Trip\_Generation.bat



Report Files Generated by Trip\_Generation.bat:

- <ITER>\_Demo\_Models.rpt
- <ITER>\_Trip\_Generation.rpt
- <ITER>\_Trip\_Generation\_Summary.rpt
- <ITER>\_Truck\_Com\_Trip\_Generation.rpt



TITLE: Process flowchart: Application of the TPB Version 2.3 Travel Model (Build 66)

COMPANY: COG/TPB

CREATOR: RM, MS, MSM

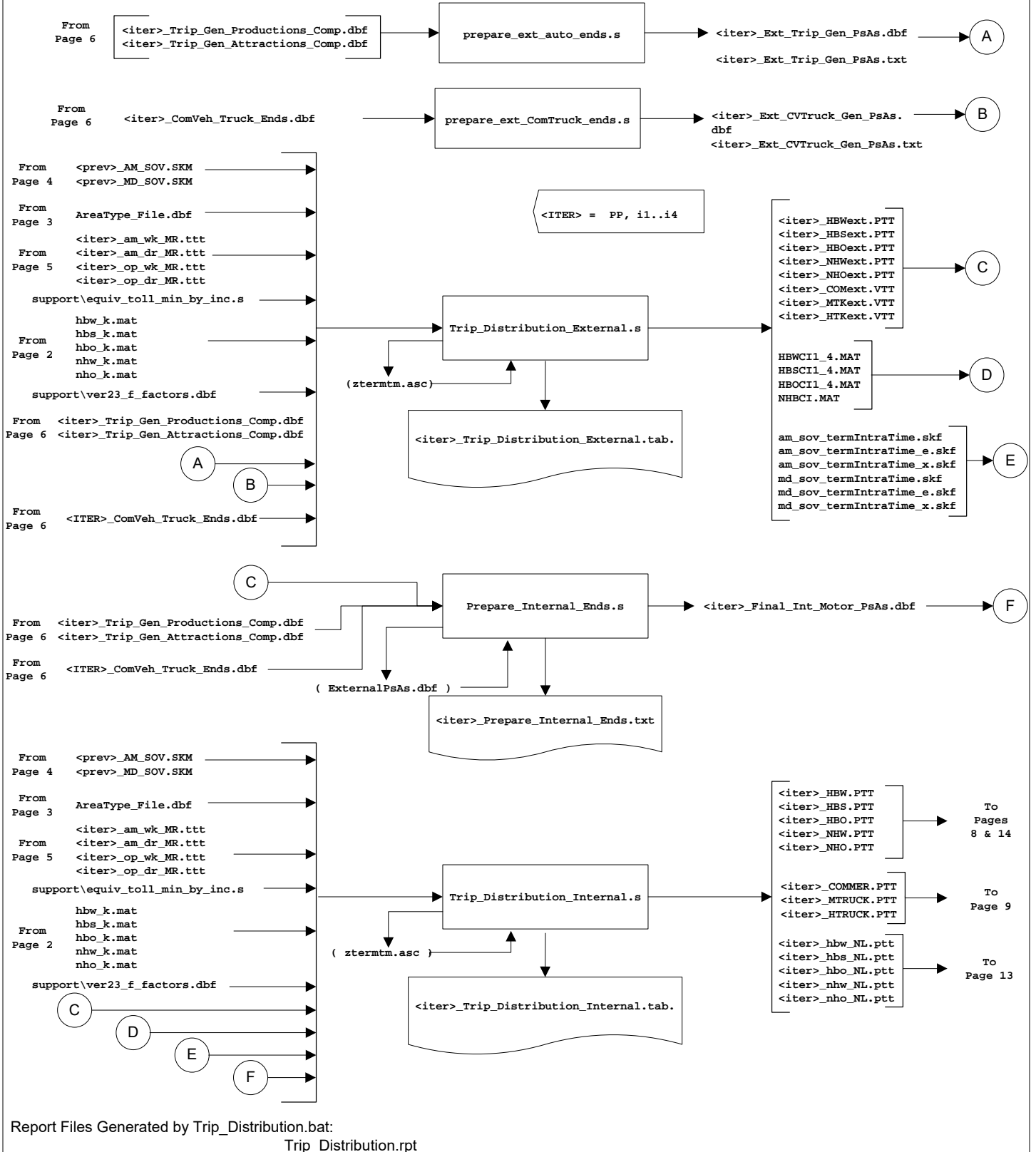
DATE: 2/10/17

PG: 7

OF 17

FILENAME: Ver2.3.66\_flowchart\_v2.vsd

### Trip\_Distribution.bat





**TITLE: Process flowchart: Application of the TPB Version 2.3 Travel Model (Build 66)**

**COMPANY: COG/TPB**

**CREATOR: RM, MS, MSM**

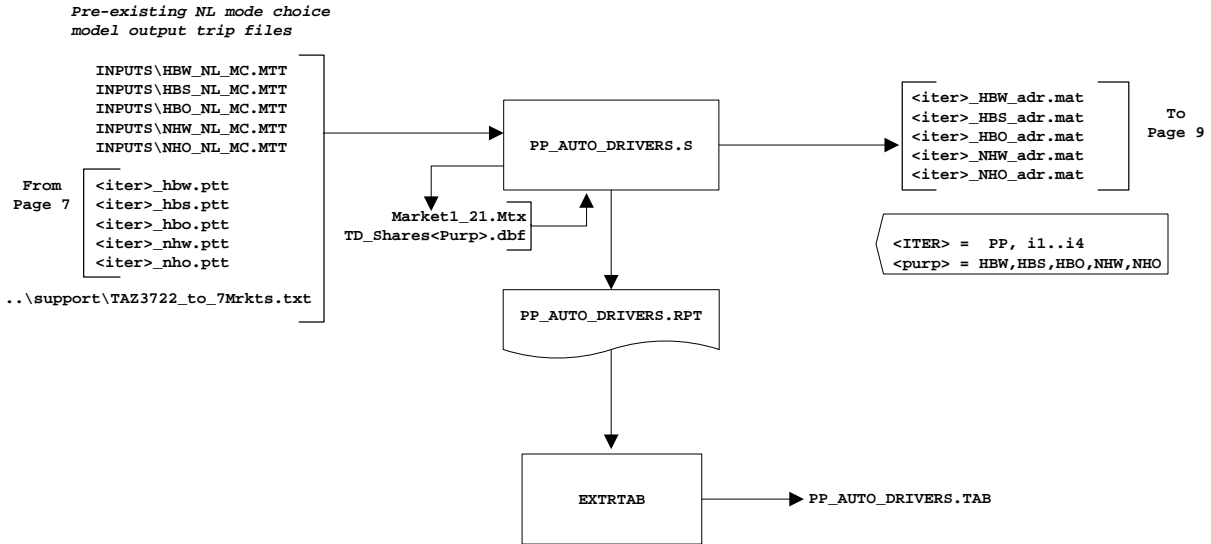
**DATE: 2/10/17**

**PG: 8**

**OF 17**

**FILENAME: Ver2.3.66\_flowchart\_v2.vsd**

## PP\_Auto\_Drivers.bat



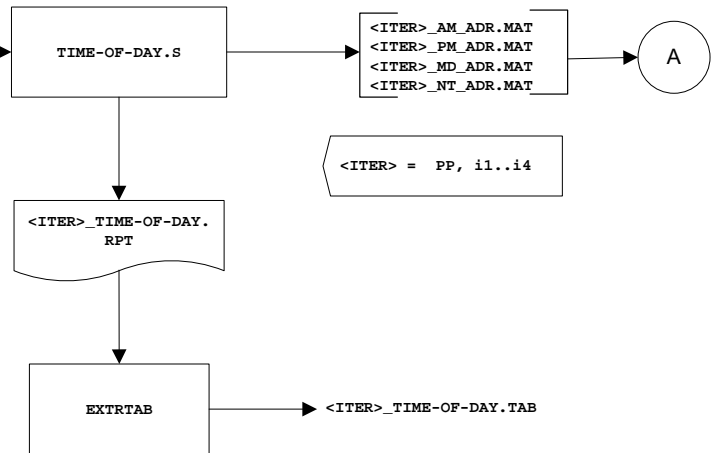
Report Files Generated by PP\_Auto\_Drivers.bat:  
PP\_Auto\_Drivers.rpt



### Time-of-Day.bat

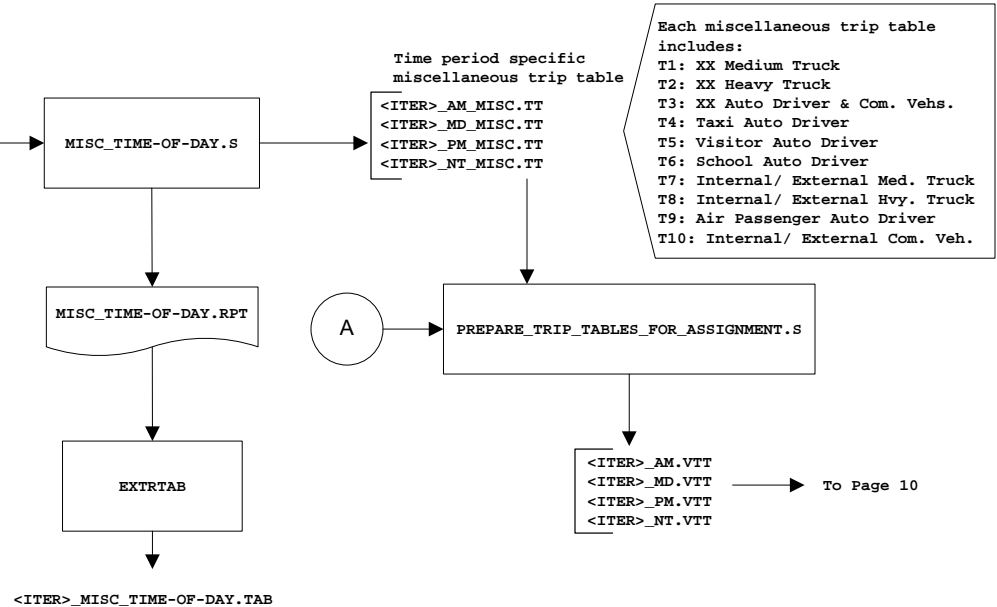
From Page 8 (pp Iter)  
From Page 14  
(i1 to i4 iter)

```
<iter>_HBW_adr.mat  
<iter>_HBS_adr.mat  
<iter>_HBO_adr.mat  
<iter>_NHW_adr.mat  
<iter>_NHO_adr.mat  
..\support\todcomp_2008HTS.dbf
```



From Page 7

```
Inputs\XXCVT.VTT  
Inputs\XXAUT.VTT  
Inputs\TAXI.ADR  
Inputs\SCHL.ADR  
Inputs\VISI.ADR  
Inputs\AIRPAX.ADR  
<iter>_COMMER.PTT  
<iter>_MTRUCK.PTT  
<iter>_HTRUCK.PTT  
..\Support\cvdelta_3722.trp  
..\Support\tkdelta_3722.trp
```



Report Files Generated by Time-of-Day.bat:

```
<ITER>_Time-of-Day.rpt  
<ITER>_Misc_Time-of-Day.rpt
```



**TITLE: Process flowchart: Application of the TPB Version 2.3 Travel Model (Build 66)**

**COMPANY: COG/TPB**

**CREATOR: RM, MS, MSM**

**DATE: 2/10/17**

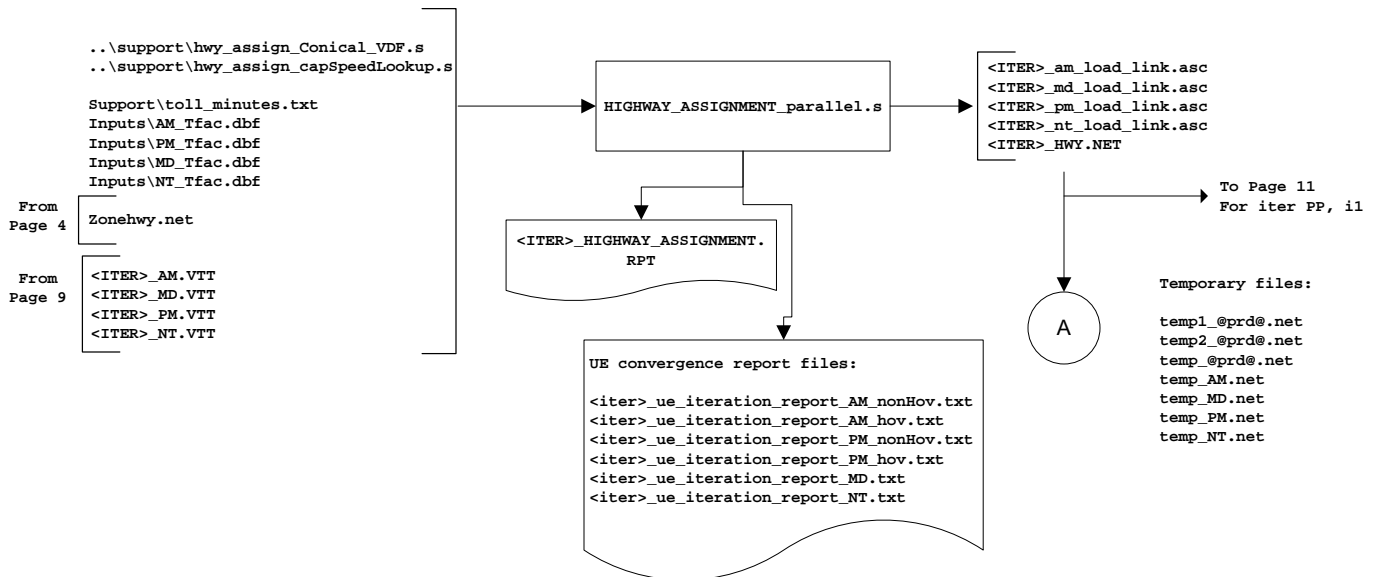
**PG: 10**

**OF 17**

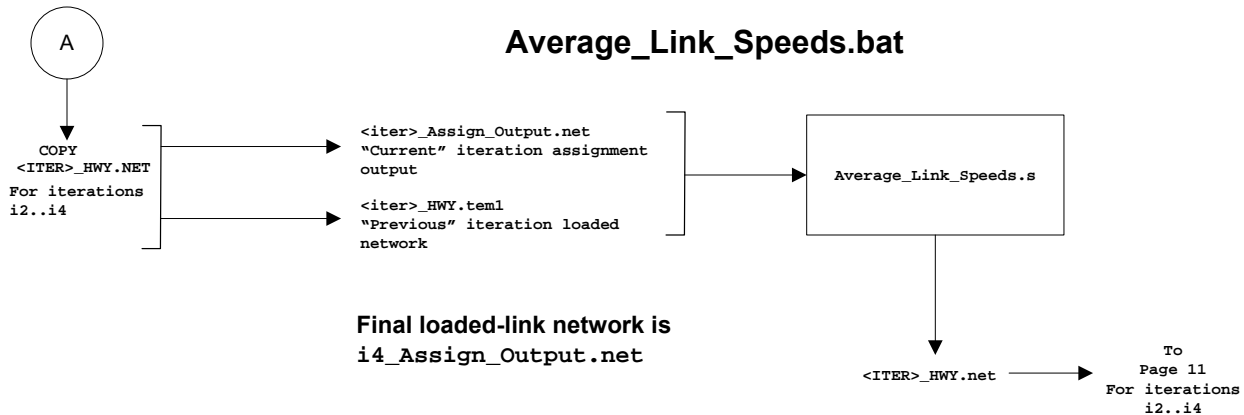
**FILENAME: Ver2.3.66\_flowchart\_v2.vsd**

## Highway\_Assignment\_parallel.bat

<ITER> = PP, i1..i4



## Average\_Link\_Speeds.bat



Report Files Generated by Highway\_Assignment.bat:

<ITER>\_Highway\_assignment.rpt  
Average\_Link\_Speeds.rpt





TITLE: Process flowchart: Application of the TPB Version 2.3 Travel Model (Build 66)

COMPANY: COG/TPB

CREATOR: RM, MS, MSM

DATE: 2/10/17

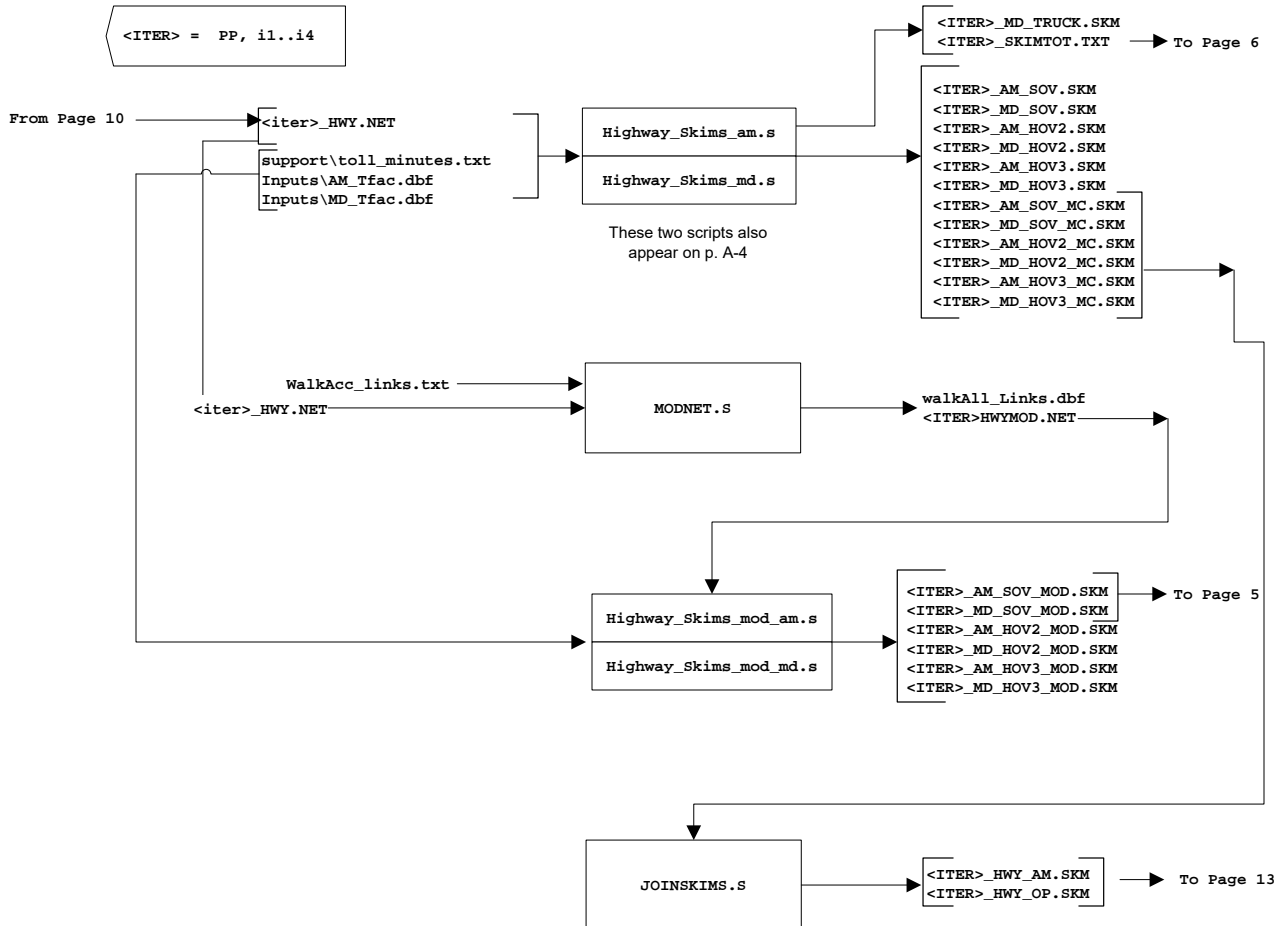
PG: 11

OF 17

FILENAME: Ver2.3.66\_flowchart\_v2.vsd

## Highway\_Skims.bat (or HSR\_Highway\_Skims.bat for analysis years with HOT lanes)

(See also page A-4 for highway skimming process used in the "pump prime" speed feedback iteration)



Report Files Generated by Highway\_Skims.bat:

<ITER>\_Highway\_skims.rpt  
<ITER>\_Joinskims.rpt  
<ITER>\_Modnet.rpt  
<ITER>\_Highway\_skims\_mod.rpt



TITLE: Process flowchart: Application of the TPB Version 2.3 Travel Model (Build 66)

COMPANY: COG/TPB

CREATOR: RM, MS, MSM

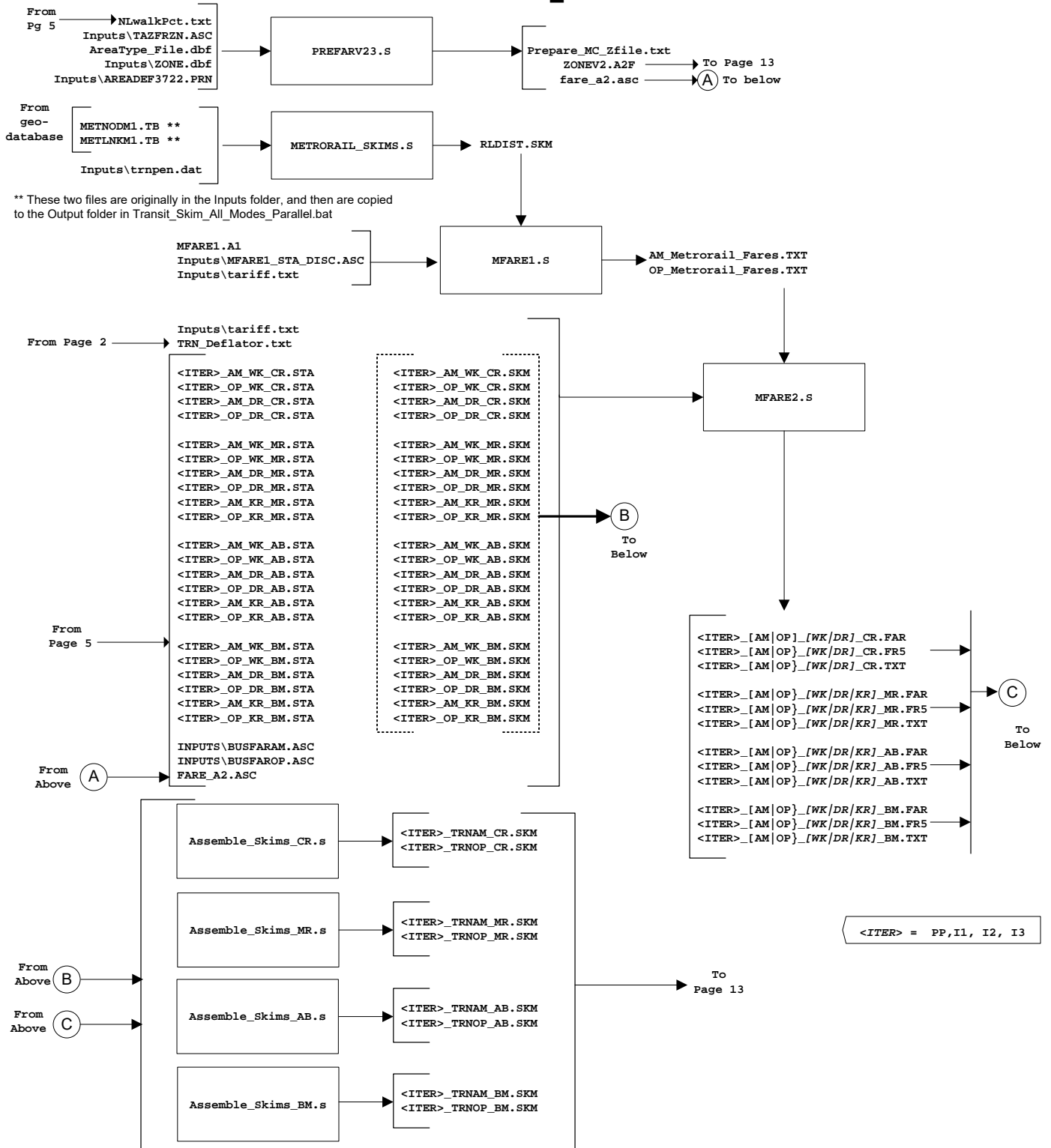
DATE: 2/10/17

PG: 12

OF 17

FILENAME: Ver2.3.66\_flowchart\_v2.vsd

### Transit\_Fare.bat



Report Files Generated by Transit\_Fare.bat:

- Prepare\_MC\_ZFile.rpt
- <ITER>\_prefarV23.rpt
- <ITER>\_Metrorail\_Skims.rpt
- <ITER>\_Mfare1.rpt
- <ITER>\_Mfare2.rpt
- <ITER>\_Assemble\_Skims\_CR.rpt, Assemble\_Skims\_MR.rpt, Assemble\_Skims\_AB.rpt, Assemble\_Skims\_BM.rpt



TITLE: Process flowchart: Application of the TPB Version 2.3 Travel Model (Build 66)

COMPANY: COG/TPB

CREATOR: RM, MS, MSM

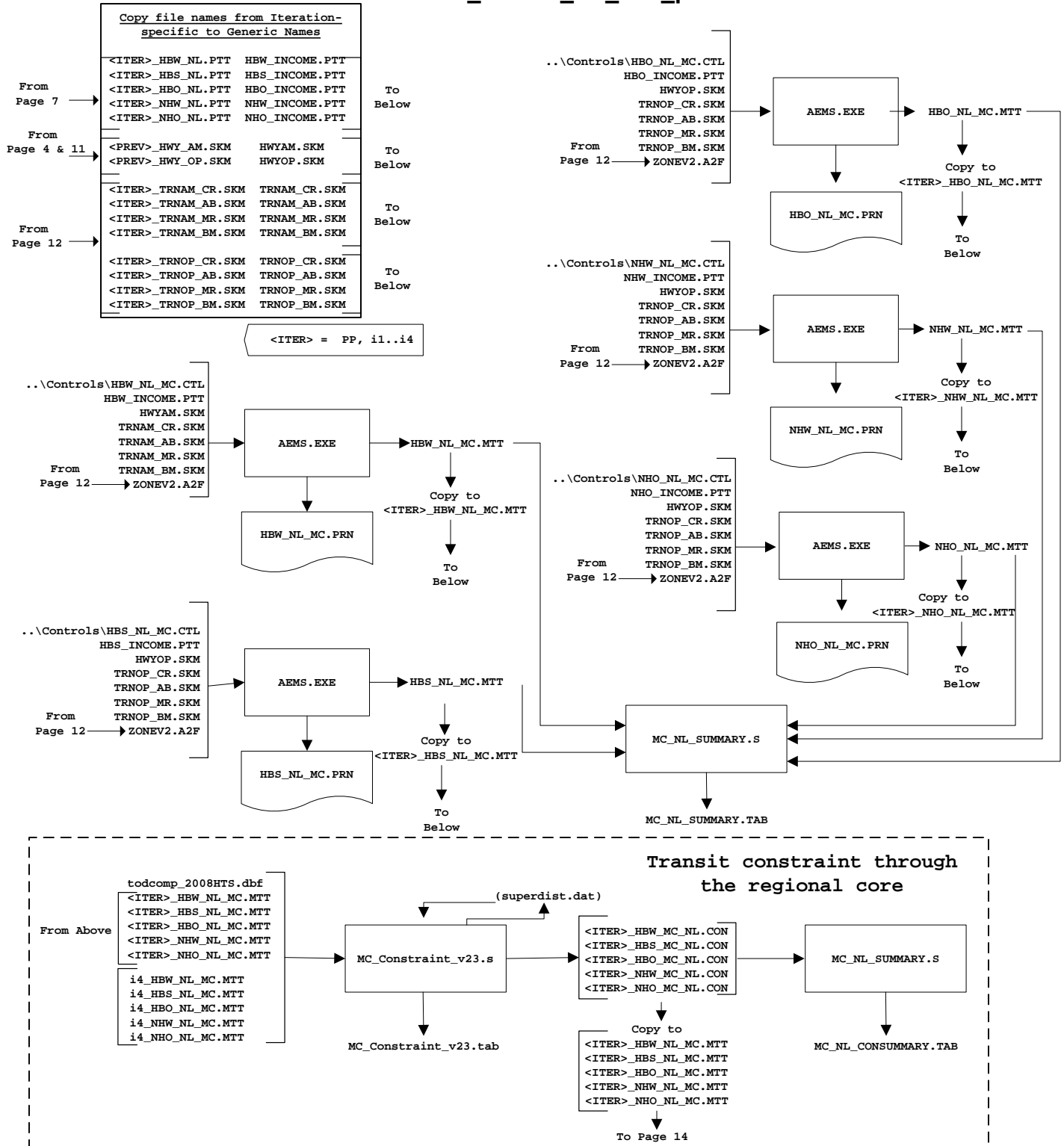
DATE: 2/10/17

PG: 13

OF 17

FILENAME: Ver2.3.66\_flowchart\_v2.vsd

### Mode\_Choice\_parallel.bat/ Mode\_Choice\_TC\_V23\_parallel.bat



Report Files Generated by Mode\_Choice.bat/Mode\_Choice\_TC\_V23.bat:

- <ITER>\_MC\_NL\_Summary.rpt
- <ITER>\_MC\_Constraint\_v23.rpt
- <ITER>\_mc\_NL\_CONSUMMARY.rpt



**TITLE: Process flowchart: Application of the TPB Version 2.3 Travel Model (Build 66)**

**COMPANY: COG/TPB**

**CREATOR: RM, MS, MSM**

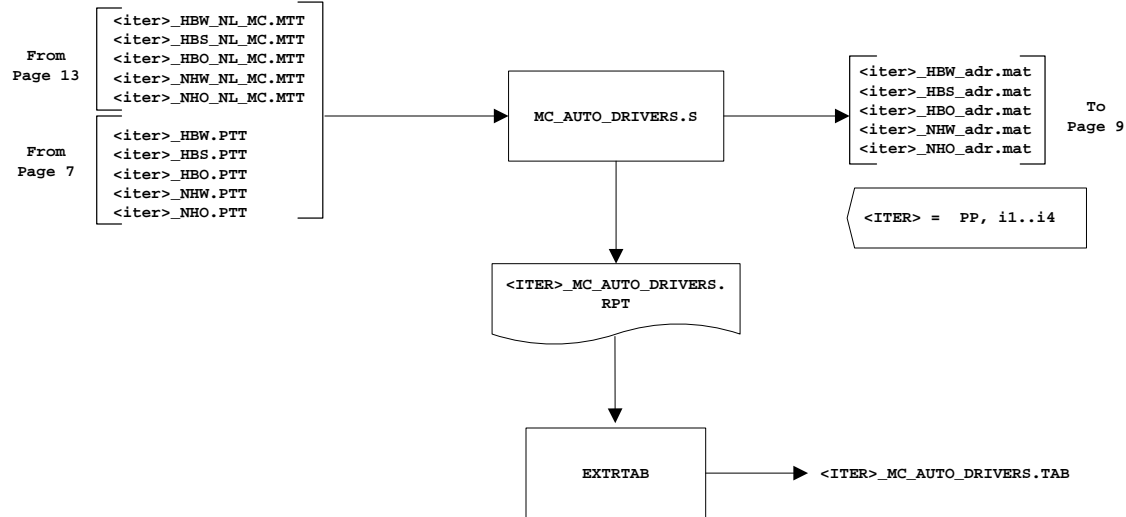
**DATE: 2/10/17**

**PG: 14**

**OF 17**

**FILENAME: Ver2.3.66\_flowchart\_v2.vsd**

### Auto\_Driver.bat



Report Files Generated by Auto\_driver.bat:  
<ITER>\_MC\_Auto\_Drivers.rpt



TITLE: Process flowchart: Application of the TPB Version 2.3 Travel Model (Build 66)

COMPANY: COG/TPB

CREATOR: RM, MS, MSM

DATE: 2/10/17

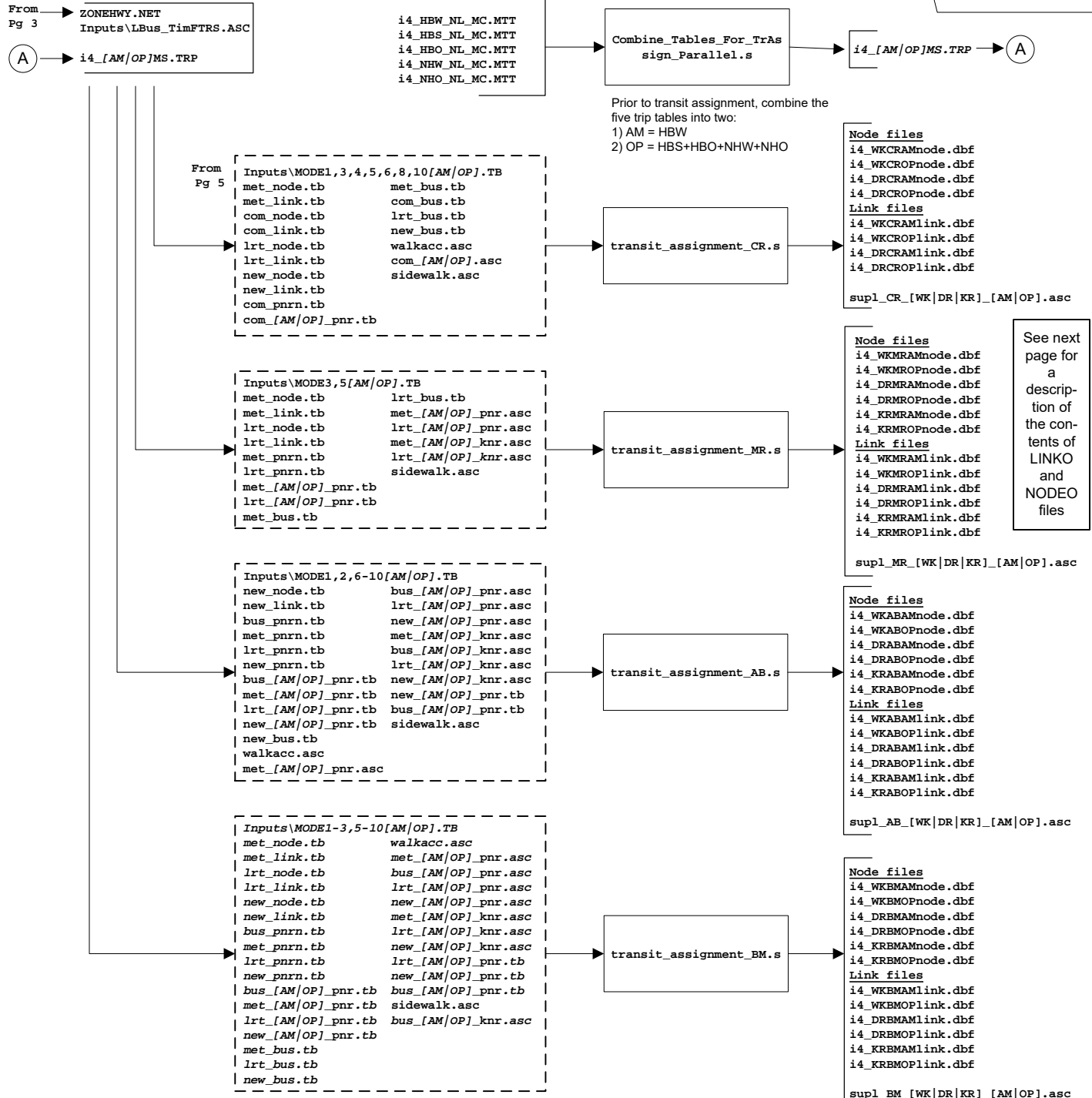
PG: 15

OF 17

FILENAME: Ver2.3.66\_flowchart\_v2.vsd

### Transit\_Assignment\_Parallel.bat Transit\_Assignment\_LineHaul\_Parallel.bat

AA = WK, DR, KR  
(Note: No KR for CR)



Report Files Generated by Transit\_Assignment.bat:  
Combine\_Tables\_For\_TrAssign\_Parallel.rpt  
transit\_assignment\_CR.rpt, transit\_assignment\_MR.rpt, transit\_assignment\_AB.rpt, transit\_assignment\_BM.rpt



**TITLE: Process flowchart: Application of the TPB Version 2.3 Travel Model (Build 66)**

**COMPANY: COG/TPB**

**CREATOR: RM, MS, MSM**

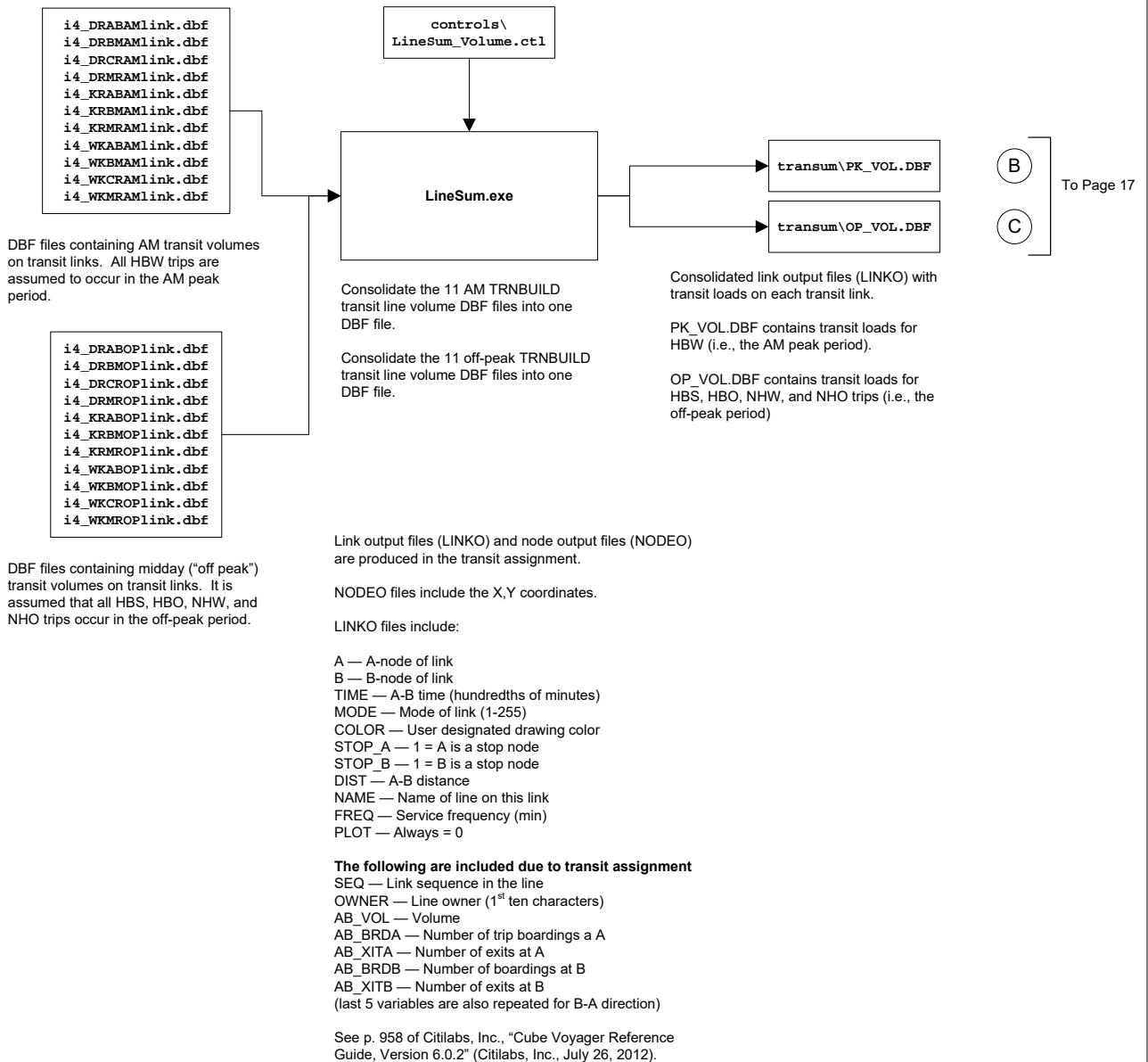
**DATE: 2/10/17**

**PG: 16**

**OF 17**

**FILENAME: Ver2.3.66\_flowchart\_v2.vsd**

## Transum.bat (page 1 of 2)





**TITLE: Process flowchart: Application of the TPB Version 2.3 Travel Model (Build 66)**

**COMPANY: COG/TPB**

**CREATOR: RM, MS, MSM**

**DATE: 2/10/17**

**PG: 17**

**OF 17**

**FILENAME: Ver2.3.66\_flowchart\_v2.vsd**

### Transum.bat (page 2 of 2)

