

# USER'S GUIDE

For the COG/TPB Gen2/Version 2.4.6 Travel Demand Forecasting Model

July 13, 2023



National Capital Region  
**Transportation Planning Board**



## **USER'S GUIDE FOR THE COG/TPB GEN2/VERSION 2.4.6 TRAVEL DEMAND FORECASTING MODEL**

July 13, 2023

### **ABOUT THE TPB**

The National Capital Region Transportation Planning Board (NC RTPB or 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, 24 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.

### **ABOUT COG**

The Metropolitan Washington Council of Governments (M WCOG or 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 24 local governments, the Maryland and Virginia state legislatures, and U.S. Congress.

### **CREDITS**

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

Director, Travel Forecasting and Emissions Analysis Program, DTP: Mark Moran

Editors: Meseret Seifu, Ray Ngo, Feng Xie, and Mark Moran

Contributing Editors: Ron Milone (retired Oct. 2018) and Jane Posey

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 © 2023 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 regional travel demand forecasting model by the TPB</b> .....	<b>2</b>
1.2	<b>History of the production-use Version 2.3 and Version 2.4 Travel Models</b> .....	<b>3</b>
1.3	<b>Recent changes to the Ver. 2.4.6 Travel</b> .....	<b>5</b>
1.3.1	Update 1: Migration of the Transit Walkshed Process from ArcPy to GeoPandas .....	9
1.3.2	Update 2: Migration from AEMS to TRANSIMS Mode Choice .....	12
1.3.3	Update 3: Updated process to compute new bus speed degradation factors .....	20
1.3.4	Update 4: Developed a process that would ensure that no row in the transit line files exceeds 144 columns .....	22
1.3.5	Update 5: Clean up transit skimming/assignment report files .....	24
1.3.6	Update 6: Added an automatic shutdown option at the end of a model run .....	28
<b>2</b>	<b><i>Overview of the model</i></b> .....	<b>30</b>
2.1	<b>Model inputs</b> .....	<b>31</b>
2.2	<b>Model outputs</b> .....	<b>34</b>
2.3	<b>Modeling steps and the speed feedback loop</b> .....	<b>35</b>
2.4	<b>Special modeling procedures used in earlier versions of the travel model</b> .....	<b>41</b>
2.4.1	Toll estimation for high occupancy/toll (HOT) lanes .....	42
2.4.2	Select-link analyses.....	44
<b>3</b>	<b><i>Hardware and software requirements</i></b> .....	<b>45</b>
3.1	<b>Hardware</b> .....	<b>45</b>
3.2	<b>Software</b> .....	<b>46</b>
3.3	<b>Examples of computer hardware used at COG for modeling</b> .....	<b>47</b>
<b>4</b>	<b><i>Mechanics of the model application process</i></b> .....	<b>51</b>
<b>5</b>	<b><i>Preparing to run the model</i></b> .....	<b>63</b>
5.1	<b>Installing software and setting the Windows PATH environment variable</b> .....	<b>63</b>
5.2	<b>Preparing input files and calculating TAZ walk areas</b> .....	<b>73</b>
<b>6</b>	<b><i>Running the model</i></b> .....	<b>75</b>
6.1	<b>Parent batch files</b> .....	<b>75</b>
6.1.1	Description of the “run model”/wrapper batch file .....	80
6.1.2	Description of the “run model steps” batch files.....	81
6.2	<b>Running the model: An example</b> .....	<b>86</b>
<b>7</b>	<b><i>Summarizing model output and other utilities</i></b> .....	<b>89</b>
7.1	<b>Summary Scripts</b> .....	<b>89</b>

7.2	Utilities .....	90
<b>8</b>	<b>Use of parallel processing to reduce model run times .....</b>	<b>93</b>
8.1	Model run times.....	93
8.2	Use of parallel processing to reduce model run times.....	94
8.2.1	Background and terminology .....	95
8.2.2	Effect of Cube Cluster on modeled results .....	96
8.2.3	History of adding parallelization to the Version 2.3 Travel Models.....	96
8.2.4	Implementation of parallelization in the Version 2.3.52 through 2.4.6 Travel Models.....	97
<b>9</b>	<b>Debugging cases where the model run stops prematurely or crashes.....</b>	<b>109</b>
<b>10</b>	<b>Known issues related to running the model.....</b>	<b>113</b>
10.1	Cube Cluster differences.....	113
10.2	Model run stops before finishing .....	113
10.3	Issues with traffic assignment convergence.....	113
10.4	Running multiple concurrent model runs on one computer/server .....	113
<b>11</b>	<b>Building transit walksheds using Geopandas.....</b>	<b>115</b>
11.1	Overview .....	115
11.2	Application Details .....	115
<b>12</b>	<b>Set-Up Programs and Highway Network Building.....</b>	<b>119</b>
12.1	Overview .....	119
12.2	Application Details .....	125
<b>13</b>	<b>Highway Skim File Development .....</b>	<b>133</b>
13.1	Overview .....	133
13.2	Application Details .....	134
<b>14</b>	<b>Auto Driver Trip Development.....</b>	<b>137</b>
14.1	Overview .....	137
14.2	Application Details .....	137
<b>15</b>	<b>Pre-Transit Network Processing .....</b>	<b>139</b>
15.1	Overview .....	139
15.2	Application Details .....	143
<b>16</b>	<b>Transit Skim File Development.....</b>	<b>145</b>
16.1	Overview .....	145

<b>16.2</b>	<b>Application Details</b> .....	<b>146</b>
16.2.1	Skim file names and list of transit skim tables in the skim files .....	146
16.2.2	Description of local bus, future time degradation factors .....	148
<b>17</b>	<b>Transit Fare Development</b> .....	<b>151</b>
17.1	Overview .....	151
17.2	Application Details .....	153
<b>18</b>	<b>Demographic Submodels</b> .....	<b>157</b>
<b>19</b>	<b>Trip Generation</b> .....	<b>159</b>
19.1	Control/Support File(s):.....	159
19.2	Application Details: .....	159
<b>20</b>	<b>Trip Distribution</b> .....	<b>171</b>
20.1	Overview .....	171
20.2	Application Details .....	172
<b>21</b>	<b>Mode Choice</b> .....	<b>174</b>
21.1	Travel modes represented in the mode choice model.....	174
21.1.1	Other issues relating to travel modes.....	175
21.2	Elimination of Metrorail constraint to and through the regional core.....	176
21.2.1	Treatment of LRT, BRT, and streetcar.....	179
21.3	Control/Support Files .....	181
21.4	Market segmentation.....	183
21.4.1	Market segmentation by household income .....	183
21.4.2	Market segmentation by geography .....	183
21.4.3	Market segmentation by access to transit.....	187
21.5	Transit access coding .....	199
21.5.1	Station file.....	199
21.5.2	Sidewalk links and zonal walk links .....	200
21.5.3	Zonal auto-access links .....	202
21.5.4	Station transfer links.....	205
21.5.5	Zonal percent walk to transit calculations .....	206
21.6	Transit path-building procedures .....	207
21.7	Treatment of parking costs and terminal times for non-transit-related trips .....	209
21.7.1	Non-transit-related parking costs .....	209
21.7.2	Non-transit-related highway terminal time assumptions.....	210
21.8	Auto Operating Costs .....	211
<b>22</b>	<b>Time-of-Day Processing</b> .....	<b>213</b>

22.1	Overview .....	213
<b>23</b>	<b>Traffic Assignment.....</b>	<b>215</b>
23.1	Overview .....	215
23.2	Two-step assignment .....	216
23.2.1	Prior to 2008: 5 user classes .....	216
23.2.2	After 2008: 6 user classes .....	218
23.3	Application details .....	219
23.3.1	Generalized cost .....	219
23.3.2	Inputs and outputs .....	220
23.3.3	Multi-class assignment .....	221
23.3.4	Volume-delay functions.....	222
23.3.5	Convergence of user equilibrium traffic assignment.....	222
23.3.6	Loaded link highway network.....	223
23.3.7	Averaging of link volumes.....	224
23.3.8	Treatment of airport passenger auto driver trips on HOV and HOT lane facilities.....	225
<b>24</b>	<b>Transit Assignment, Including Summary Process (LineSum) .....</b>	<b>231</b>
24.1	Transit assignment process.....	231
24.1.1	Inputs to the transit assignment.....	233
24.1.2	Outputs of the transit assignment.....	233
24.2	Transit assignment summary process.....	240
24.2.1	Consolidating transit assignment output and displaying results .....	242
24.2.2	Generating transit assignment summaries using LineSum .....	246



## List of Figures

Figure 1 The New Transit Walkshed Process, <i>Walkshed_Process.py</i> in the Ver. 2.4.6 model .....	11
Figure 2 Excerpt from the new Transit Walkshed Process, <i>Walkshed_Process.bat</i> .....	12
Figure 3 AEMS Control Files Converted to the ModeChoice Setup .....	14
Figure 4 Changes made to the MC_Purp.bat file .....	16
Figure 5 Beginning and end sections of changes made to <i>Mode_Choice_Parallel.bat</i> .....	17
Figure 6 Example HBW AM ModeChoice Control File .....	18
Figure 7 Example HBW AM ModeChoice Constants File .....	18
Figure 8 Example of beginning and end section of <i>ModeChoice_CommonKeys.ctf</i> file .....	19
Figure 9 Example ModeChoice Script File .....	20
Figure 10 beginning and end sections of changes made to <i>Adjust_Runtime.s</i> in Ver. 2.4.6 .....	22
Figure 11 Changes made to <i>Transit_Skim_All_Modes_Parallel.bat</i> in Ver. 2.4.6 .....	22
Figure 12 Beginning section of <i>Adjust_Runtime.s</i> in Ver. 2.4.6 .....	23
Figure 13 Beginning and end sections of <i>Transit_Skims_MR</i> script .....	25
Figure 14 Beginning and end sections of <i>Transit_Assignment_MR.S</i> .....	26
Figure 15 Changes made to <i>Transit_Assignment_LineHaul_Parallel.bat</i> .....	27
Figure 16 Changes made to <i>Transit_Skim_LineHaul_Parallel.bat</i> .....	27
Figure 17 An excerpt from the wrapper batch file .....	29
Figure 18 Batch file to call the automatic shutdown application program .....	29
Figure 19 Major inputs and outputs of the TPB Version 2.4.6 Travel Model .....	31
Figure 20 Major steps of the Version 2.4.6 Travel Model .....	36
Figure 21 Application process of the Version 2.4.6 Travel Model .....	40
Figure 22 Subdirectory structure for executing the Version 2.4.6 Travel Model .....	52
Figure 23 Example for the correct paths of Mambaforge and the gen2_model environment .....	64
Figure 24 Number of cores and logical processors .....	65
Figure 25 Bentley Systems Cube 6.4, "About" message window .....	66
Figure 26 Bentley Systems Cube 6.5, "About" message window .....	66
Figure 27 "About Voyager": Presence of Cube Cluster license .....	67
Figure 28 Windows PowerShell version .....	68
Figure 29 "Run model" batch file for 2021 .....	77
Figure 30 "Run model steps" batch file for 2021 .....	78
Figure 31 "Run model steps" batch file for 2040_final .....	84
Figure 32 First five lines of the LOV/HOV link equivalency <i>HOV_Links.txt</i> .....	90
Figure 33 Convert DBFs to CSVs dialog box .....	92
Figure 34 Schematic of IDP and MDP in the highway assignment process of the Ver. 2.4.6 Travel Model ( <i>Highway_Assignment_Parallel.s</i> ) .....	104
Figure 35 Excerpts from the <i>Highway_Assignment_Parallel.s</i> script (triple ampersand => code removed) .....	104
Figure 36 An excerpt from an example of the "full output" text file that is created during a Ver.2.4.6 model run .....	110
Figure 37 An excerpt from the "search for errors" file that is created during a model run .....	111
Figure 38 Folder structure for the automated Geopandas walkshed process .....	117
<b>Figure 39 Walkshed buffers for a typical base-year scenario</b> .....	118
Figure 40 Consumer price index file ( <i>CPI_File.txt</i> ) .....	125
Figure 41 A local bus route before its RUNTIME value is updated (excerpt from the Ver2.3.78 travel model Year 2045) .....	149
Figure 42 A local bus route after its RUNTIME value is updated to reflect road congestion predicted to occur in future years ..	150
Figure 43 Outline/pseudo code for <i>trip_generation.s</i> .....	161
Figure 44 Example of seven NAVTEQ street blocks within TAZ 283 (Union Station) .....	163
Figure 45 Designed nesting structure of the nested-logit mode choice model in the Version 2.4.6 Model .....	174
Figure 46 Ring 0 (white trapezoid) and Ring 1 (gray polygon), which form the "core" area used in the Metrorail constraint through the regional core .....	178
Figure 47 Seven superdistricts used in the Versions 2.3 and 2.4 nested-logit mode choice model .....	184

Figure 48 Zonal walk access links and sidewalk links in downtown DC near Farragut Square (Ver. 2.4.6 NL MC model) .....	202
Figure 49 Kiss-and-ride (KNR) auto access links to Metrorail stations in Northern Virginia .....	204
Figure 50 Park-and-ride (PNR) auto access links to Metrorail stations in Northern Virginia .....	205
Figure 51 Non-transit-related, daily parking cost model used in the Version 2.4.6 Model .....	210
Figure 52 Excerpt from one of the transit node DBF files output from transit assignment (i4_WKMRAMnode.dbf) .....	234
Figure 53 Transit volumes from transit assignment using TRNBUILD: One-way route.....	235
Figure 54 Transit volumes from transit assignment using TRNBUILD: Two-way route.....	236
Figure 55 Excerpt from one of the transit link DBF files output from transit assignment (i4_WKMRAMlink.dbf) showing mode-16 links .....	238
Figure 56 Excerpt from one of the transit link DBF files output from transit assignment (i4_WKMRAMlink.dbf) showing mode-3 links .....	238
Figure 57 Excerpt from one of the transit link DBF files output from transit assignment (i4_WKMRAMlink.dbf) showing mode-12 links .....	239
Figure 58 An excerpt of <i>tranSum.bat</i> transit summary batch file.....	241
Figure 59 Consolidating peak and off-peak transit assignment volumes (LineSum_Volume.ctl) .....	242
Figure 60 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.....	243
Figure 61 Using the pk_vol.dbf file in Cube Base as the transit layer: Only mode-3 (Metrorail) links turned on .....	243
Figure 62 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 .....	244
Figure 63 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 .....	244
Figure 64 Using the multi-bandwidth option in Cube Base to show transit volumes on the Metrorail system. ....	245
Figure 65 Generating a Metrorail station access report (lineSum_MR_access.ctl) .....	246
Figure 66 Generating a Metrorail line summary (lineSum_MR_line.ctl) .....	246
Figure 67 An excerpt from the report file generated by lineSum_MR_access.ctl .....	248
Figure 68 The report file generated by lineSum_MR_line.ctl .....	250

## List of Tables

Table 1 Updates made to the Ver. 2.4.6 Model, compared to Ver. 2.4	6
Table 2 Affected model components due to Update 1	10
Table 3 Affected model components due to Update 2	14
Table 4 Affected model components due to Update 3	21
Table 5 Affected model components due to Update 4	23
Table 6 Affected model components due to Update 5	24
Table 7 Affected model components due to Update 6	28
Table 8 Outputs of the travel model: Disaggregate-level output that is not validated versus aggregate-level output that is validated	35
Table 9 User equilibrium closure criterion (relative gap) varies by speed feedback iteration	38
Table 10 Previously used Computer storage drives for travel demand modeling	49
Table 11 Current drive mappings for travel demand modeling	49
Table 12 Input files needed to run the Version 2.4.6 Travel Model, stored in the CONTROLS, INPUTS, and SUPPORT folders	55
Table 13 Fortran and C++ executable files and dynamic-link library files required for running the Version 2.4.6 Travel Model	58
Table 14 Child batch files used to run the Version 2.4.6 Travel Model	58
Table 15 Sequence of the batch files used to run the Version 2.4.6 Travel Model	61
Table 16 Summary of differences to the "run model steps" batch files for the years 2019, 2020, and 2045 in Ver 2.3.70 and Ver 2.4.6 (or Ver 2.3.75, Ver 2.3.78)	83
Table 17 Travel Model Summary Scripts	89
Table 18 Five traffic assignments in the Version 2.2 Travel Model became six in the Version 2.3 and Version 2.4 Travel Models	94
Table 19 Modeling steps where parallelization is used, including the maximum number of threads/cores used	99
Table 20 Running the Version 2.4.6 Travel Model on computers with fewer than 8 cores: Changes that need to be made to the "run model"/wrapper batch file	101
Table 21 Inputs to the set-up and highway network building process	119
Table 22 Land Use File Format Description (zone.dbf)	119
Table 23 Node Coordinate File Format Description (node.dbf)	120
Table 24 Base Highway Link File Format Description (link.dbf)	120
Table 25 Link limit codes	121
Table 26 Toll Parameter File (Toll_esc.dbf)	122
Table 27 Outputs of the set-up and highway network building process	122
Table 28 Zonal Area Type File (AreaType_File.dbf)	123
Table 29 Unloaded binary highway network file (Zonehwy.net)	124
Table 30 Elements of the highway network	127
Table 31 Area type codes, based on population and employment density	128
Table 32 zonehwy.net file Variables description	130
Table 33 Inputs to the highway skim file development	133
Table 34 Outputs of the highway skim file development	134
Table 35 Inputs to auto driver trip development	137
Table 36 Outputs of auto driver trip development	137
Table 37 Inputs to pre-transit network processing	139
Table 38 Variables in the transit station file (Station.dbf)	140
Table 39 Interpretation of transit access distance codes (NCT): Metrorail, light rail, and bus PNR access distance codes and their meaning for the	141
Table 40 HBW zonal parking costs/terminal time file (HBWV2a1.dbf)	142
Table 41 Walk Access Links (WalkAcc_Lnks.dbf)	142
Table 42 Outputs of pre-transit network processing	142
Table 43 Overview of Version 2.4.6 Transit Network Elements	143
Table 44 Inputs to transit skim file development	145
Table 45 Outputs of transit skim file development	146

Table 46 Skim files developed by the transit skimming process.....	147
Table 47 Skim tables contained in each transit skim file .....	147
Table 48 Job accessibility by transit file format description (<ITER>_<Prd>_<AA>_[BM MR]_JobAcc.dbf).....	150
Table 49 Inputs to transit fare development .....	151
Table 50 Outputs of transit fare development .....	152
Table 51 TAZ/Bus Fare Zone Equivalency File Format Description (TAZFRZN.ASC) .....	153
Table 52 Inputs to the Demographic Models.....	157
Table 53 Outputs of the Demographic Models.....	158
Table 54 Inputs to trip generation .....	159
Table 55 External Production and Attraction File (Ext_PsAs.dbf) .....	164
Table 56 Consolidated Zonal Land Use File.....	165
Table 57 Computed zonal trip productions file (<iter>_Trip_Gen_Productions_Comp.dbf) .....	165
Table 58 Computed zone trip attractions file (<iter>_Trip_Gen_Attractions_Comp.dbf) .....	166
Table 59 Truck and commercial vehicles trip ends (<iter>_ComVeh_Truck_Ends.dbf) .....	168
Table 60 Inputs to trip distribution.....	171
Table 61 Outputs of trip distribution .....	171
Table 62 Time Valuation (Minutes/2007\$) by Purpose and Income Level .....	173
Table 63 Transit sub-modes represented in the Version 2.4 Model.....	175
Table 64 Transit Access and Transfer Links.....	175
Table 65 Transfer Prohibitions (No Transfer or NOX) .....	176
Table 66 Inputs to the ModeChoice mode choice application program.....	181
Table 67 Outputs from the ModeChoice mode choice application program .....	182
Table 68 Household income quartiles computed from the ACS .....	183
Table 69 Equivalency between nested-logit mode choice superdistricts and TPB TAZ 3,722 .....	185
Table 70 Production and attraction market segments used in the TPB Versions 2.3 and 2.4 NLMC model.....	186
Table 71 20 geographic market segments used in the TPB nested-logit mode choice model.....	186
Table 72 Equivalency between seven super-districts and the 20 geographic market segments.....	187
Table 73 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.....	190
Table 74 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 .....	191
Table 75 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 .....	191
Table 76 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 .....	192
Table 77 Four “can walk” sub-markets .....	197
Table 78 Two “must drive” sub-markets .....	197
Table 79 Application of the six transit-access segments to travel modes represented in the Ver. 2.4.6 mode choice model.....	198
Table 80 Mode codes used in the consolidated station file/database (station.dbf).....	200
Table 81 Maximum link distances for drive-access-to-transit links: Ver. 2.4 NL MC model .....	203
Table 82 Path-specific parameters used in transit path building.....	208
Table 83 Non-transit-related highway terminal time as a function of employment density.....	211
Table 84 Inputs to time-of-day process .....	213
Table 85 Outputs of time-of-day process .....	213
Table 86 User equilibrium closure criterion (relative gap) varies by speed feedback iteration.....	216
Table 87 Traffic assignment in the Version 2.2 Travel Model prior to fall 2008: Three multiclass assignments .....	217
Table 88 Traffic assignment in the Version 2.2 Travel Model prior to fall 2008: Five multiclass assignments .....	218
Table 89 Traffic assignment in the Version 2.3.52 and later travel model: Six traffic assignments per speed feedback loop .....	219
Table 90 Time Valuation by Vehicle Type and Time Period (minutes/dollar, in year-2007 prices).....	220
Table 91 Inputs to traffic assignment process .....	220
Table 92 Lookup table: Highway link capacities in free-flow conditions (vehicles per hour per lane).....	220

Table 93 Lookup table: Highway link speeds in free-flow conditions (mph).....	221
Table 94 Link limit code, traffic assignment add group, and its meaning.....	221
Table 95 Outputs of traffic assignment process .....	223
Table 96 Variables included in the final iteration, loaded highway network (i4_Assign_output.net) .....	223
Table 97 HOV and HOT-lane facilities in the Washington, D.C. area .....	225
Table 98 Use of HOV and HOT-lane facilities by autos serving airport passengers: Real world, Ver. 2.3.66 Model (and later, including Ver. 2.4.6 Model) .....	228
Table 99 Inputs to transit assignment process .....	232
Table 100 Outputs of transit assignment process .....	232
Table 101 Mapping/concatenation of trip tables by trip purposes into peak and off-peak period trip tables prior to transit assignment .....	233

## **List of Equations**

Equation 1 Converting tolls into time-equivalent minutes of impedance .....	134
Equation 2 Composite time .....	172
Equation 3 Maximum walk distance formula, used for generating walk-access-to-transit links .....	201
Equation 4 PNR-to-station time/impedance.....	206
Equation 5 Daily non-transit-related parking cost for area types 1-3.....	209

## **Appendices**

- A. Flowcharts

## **Colophon**

This report was created using Microsoft Word and Visio from Office 365 and was converted to a PDF file using Adobe Acrobat DC. In the past, this report was divided into two sections, but the current version of this report is contained in one report. The files for this report can be found in a folder on COG's internal file server (I:\ateam\docum\fy23\Version2.4.6\_Development\travel\_model\_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 (MWC OG or simply COG). The TPB is staffed by COG's Department of Transportation Planning (DTP). The COG/TPB staff develops, maintains, and improves, with consultant assistance, a series of regional travel demand forecasting 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: one or more adopted, production-use models and one or more developmental models. The production-use model is the one that is used in planning studies conducted by COG/TPB and is made available to outside parties.<sup>1</sup> The developmental model(s) are the ones that are currently under development by COG/TPB staff, and are generally not made available to outside parties, since they are not yet considered a finished product. Currently, TPB staff provides two production-use travel models (i.e., Ver. 2.4 and Ver. 2.4.6). The Data Request webpage provides details on the model transmittal packages for each production-use model.

This report explains how to set up and run the TPB's **Generation 2, or Gen2, Version 2.4.6 Travel Demand Forecasting Model, which is the latest in a series of regional, trip-based, production-use travel demand models.** The Gen2/Ver. 2.4.6 Model was derived from the Gen2/Ver. 2.4 Model. The Ver. 2.4 Model is considered an adopted, production-use model, since it was used for the air quality conformity analysis of the 2022 update of the Long-Range Transportation Plan (LRTP), known as Visualize 2045. By contrast, the Ver. 2.4.6 Model is deemed ready for production-use, but it has not yet been used in an official capacity for COG/TPB work, so it is not yet deemed adopted. See the next section for more information about adopted and production-use models. All the models in the Generation 2 family of models are aggregate, trip-based, four-step travel demand models estimated, calibrated, and validated to conditions in the metropolitan Washington region.

Between 2008 and 2011, the TPB Gen2/Ver. 2.3 Travel Model was calibrated and validated to year-2007 conditions.<sup>2</sup> In 2013, the Ver. 2.3 Travel Model was also 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 the Ver. 2.3.52 Travel Model (a.k.a., Build 52). In 2019, the **Ver. 2.3.75**

---

<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.: Metropolitan Washington Council of Governments, National Capital Region Transportation Planning Board, January 20, 2012), <https://www.mwcog.org/transportation/data-and-tools/modeling/model-documentation/>.

<sup>3</sup> Ronald Milone to Files, "2010 Validation of the Version 2.3 Travel Demand Model," Memorandum, June 30, 2013, <https://www.mwcog.org/transportation/data-and-tools/modeling/model-documentation/>.

**Model was validated to year-2014 conditions**, mainly to fulfill a federal requirement associated with the air quality conformity (AQC) determination.<sup>4</sup> In November 2019, **staff re-calibrated** the nested-logit mode choice (NLMC) model of the TPB Version 2.3 Travel Demand Model to year-2007 conditions, following updates to person-trip calibration targets for commuter rail.<sup>5</sup> In 2020, the TPB Ver. 2.3 Travel Demand Model (Ver. 2.3.85) was re-validated to year-2014 conditions and evaluated for its performance, based on a series of sensitivity tests. Following a series of important updates and recalibration/revalidation efforts, the resulting Ver. 2.3.87 Model was rebranded as Version 2.4. In 2022, the Ver. 2.4 Model was validated to 2018 conditions to fulfil the same federal requirement described above.<sup>6</sup>

COG is currently developing, with consultant assistance, a simplified, disaggregate, activity-based model (ABM) to be known as the Generation 3, or Gen3, Travel Model. The Gen3 Model is developed based on the 2017/2018 Regional Travel Survey (RTS) data and implemented in the open-source ActivitySim software platform. The model is expected to be released for production use in 2024. In the meantime, however, COG plans to maintain and improve the Gen2 Model for the foreseeable future. For example, the Gen2/Ver. 2.4.6 Model or its successor will be used for the upcoming 2025 Long-Range Transportation Plan (LRTP) update, to be known as Visualize 2050.

The user's guide for the Ver. 2.4.6 Model is contained in one volume. As of the Ver. 2.4 Model,<sup>7</sup> the model user's guide no longer includes Volume 2, which had contained appendices of batch files, Cube Voyager scripts, and control files (Instead, these files are available via the standard model/network transmittal package).<sup>8</sup>

## 1.1 Adoption of the regional travel demand forecasting model by the TPB

The TPB does not *explicitly* adopt a specific version of the regional travel demand model. Instead, the adoption is typically made *implicitly* when the TPB adopts both 1) a given version of its Long-Range Transportation Plan (LRTP) and Transportation Improvement Program (TIP); and 2) the findings from an

---

<sup>4</sup> Specifically, according to federal regulations 40 CFR 93.122(b)(1)(i), “network-based travel models must be validated against observed counts (peak and off-peak, if possible) for a base year that is not more than 10 years prior to the date of the conformity determination.” The year-2014 validation enabled the Ver. 2.3 Model to be used for Air Quality Conformity (AQC) determinations through 2024. Refer to the following document for more details: Feng Xie to Dusan Vuksan and Mark Moran, “Year-2014 Validation of TPB’s Version 2.3 Travel Demand Model”, Memorandum, March 12, 2019.

<sup>5</sup> Feng Xie, “TPB Version 2.3 Travel Demand Model: Re-Calibrating the Nested-Logit Mode Choice Model Following the Updates to Commuter Rail Person Trip Targets,” Memorandum, November 15, 2019.

<sup>6</sup> Seifu, Meseret and Ko, Sanghyeon to Feng Xie, “Year-2018 Validation of TPB Version 2.4 Travel Model,” Memorandum, August 17, 2021.

<sup>7</sup> Ray Ngo, Feng Xie, and Mark Moran, “User’s Guide for the COG/TPB Travel Demand Forecasting Model, Version 2.4” (Washington, D.C.: Metropolitan Washington Council of Governments, National Capital Region Transportation Planning Board, March 15, 2021),

[https://www.mwcog.org/assets/1/6/mwcog\\_tpb\\_travel\\_model\\_v2.4\\_user\\_guide\\_final.pdf](https://www.mwcog.org/assets/1/6/mwcog_tpb_travel_model_v2.4_user_guide_final.pdf)

<sup>8</sup> See, for example, Meseret Seifu to Feng Xie and Mark S. Moran, “Transmittal Package: TPB Gen2/Ver. 2.4 Travel Demand Forecasting Model, Transportation Networks, and Land Use Data Associated with the Air Quality Conformity Analysis of the 2022 Update to Visualize 2045 with Revised Transit Networks,” Memorandum, November 14, 2022,

[https://www.mwcog.org/assets/1/6/Rev\\_Xmittal\\_Ver2.4\\_2022Upd\\_Viz2045\\_model\\_and\\_inputs\\_v3.pdf](https://www.mwcog.org/assets/1/6/Rev_Xmittal_Ver2.4_2022Upd_Viz2045_model_and_inputs_v3.pdf)



AQC analysis of the financially constrained element of the LRTP and the associated TIP. The LRTP undergoes a major update every four years. The last of these updates was done in 2022 and was known as Visualize 2045.<sup>9</sup> The latest version of the LRTP is known as the 2022 Update to Visualize 2045, which was finalized in 2022 and has an out year of 2045. The 2022 Update to Visualize 2045 has both a financially constrained element and an aspirational element. The constrained element is what used to be called the Constrained Long-Range Plan (CLRP). The purpose of the AQC analysis is to determine whether the air pollution created by motor vehicles (“mobile emissions”) traveling on the transportation network represented in the constrained element of the LRTP (in this case, the 2022 Update to Visualize 2045) is consistent with (conforms to) the state air quality implementation plans (SIPs). The *implicit adoption* of a specific model version generally occurs when the TPB adopts an AQC analysis that made use of that model version. For instance, the Gen2/Ver. 2.4 TPB Travel Demand Forecasting Model (TDFM) became the adopted, production-use model on June 15, 2022. Since the Ver. 2.4 Model was used for that analysis, it has now considered the adopted, production-use travel demand model. The TPB staff currently offers two production-use versions of the TPB regional travel demand forecasting model: Gen2/Ver. 2.4 and Gen2/Ver. 2.4.6. The Ver. 2.4.6 Model has not been used for an AQC analysis and thus has not been implicitly adopted by the TPB yet. However, the Ver. 2.4.6 Model is being released because TPB staff deem that it is ready for production use based on a comprehensive internal evaluation. Compared to the Ver. 2.4 Model, the Ver. 2.4.6 Model incorporates some major updates which will be discussed in Section 1.3.

## 1.2 History of the production-use Version 2.3 and Version 2.4 Travel Models

The adopted, production-use Generation 2, or Gen2, Version 2.3 Travel Model was a series of model versions ending with Version 2.3.78. The first iteration of the Version 2.3 Model became the adopted regional travel model for the metropolitan Washington region on November 16, 2011. In 2012, a newer version of the model (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, the Ver. 2.3.52 Model was used for the air quality conformity analysis of the 2013 CLRP and FY 2013-2018 TIP. In 2015, the Ver. 2.3.57a Model became the production-use model. In 2016, 2017, 2018, and 2020 the Ver. 2.3.66, Ver. 2.3.70, Ver. 2.3.75, and Ver. 2.3.78 models became the production-use models, respectively.

The Gen2/Version 2.4 Travel Model is the latest series of the Generation 2 family of model versions. COG staff implemented some major updates to Version 2.3.78 to create Version 2.4. In early 2021, the Version 2.4 Model became the production-use regional travel demand model.

Below is a list of milestones in the development of the TPB regional travel demand model from 2008 to the present:

---

<sup>9</sup> “Visualize 2045, A Long-Range Transportation Plan for the National Capital Region,” Draft (Washington, D.C.: National Capital Region Transportation Planning Board, Metropolitan Washington Council of Governments, September 2018), <https://www.mwcog.org/visualize2045/document-library/>.

- March 1, 2008: The TPB Version 2.2 Travel Model was released.<sup>10</sup>
  - 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>11</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 the 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: The TPB Version 2.3 Travel Model, Build 9, was released (i.e., Ver. 2.3.9).<sup>12</sup>
  - From 2008 to 2012, TPB staff conducted the following activities:
    - Compiled and cleaned new survey data.
    - Prepared calibration files based on the new 3,722 TAZ system.
    - Estimated and calibrated various sub-models in the regional travel model.
- November 11, 2011: The TPB Version 2.3.36 Model was released.<sup>13</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.

---

<sup>10</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>.

<sup>11</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>12</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>13</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>.

- December 21, 2011: The TPB Version 2.3.38 Model was released. This model was documented in January 2012.<sup>14</sup>
- July 17, 2013: The TPB Version 2.3.52 Model became the production-use travel model.
- October 15, 2014: The TPB Version 2.3.57 Model became the production-use travel model.
- October 21, 2015: The TPB Version 2.3.57a Model became the production-use travel model.
- November 16, 2016: The TPB Version 2.3.66 Model became the production-use travel model.
- October 18, 2017: The TPB Version 2.3.70 Model became the production-use travel model.
- October 17, 2018: The TPB Version 2.3.75 Model became the production-use travel model.
- March 18, 2020: The TPB Version 2.3.78 Model became the production-use travel model.
- March 26, 2021: The TPB Version 2.4 Model became the production-use travel model and was released without a TPB adoption.
- June 15, 2022: The TPB Version 2.4 Model became the adopted production-use travel model following the TPB's approval of 2022 Amendment of Visualize 2045.
- July 21, 2023: The TPB Version 2.4.6 Model became the production-use travel model.

### 1.3 Recent changes to the Ver. 2.4.6 Travel

Relative to the TPB Gen2/Version 2.4 Model, there have been six major updates to the Gen2/Version 2.4.6 Travel Model. The updates are listed in Table 1 and are described in more detail later in this document.

The TPB Model can have four types of updates: a bug fix, a new feature, a feature enhancement, and updated documentation. Bug fixes are the most important type of update and have the highest priority. Software bugs can be found by either internal or external model users. New or enhanced features bring improvements to the model or make it easier to use. Documentation relates to comments or annotations within scripts or batch files that explain what is occurring in the software code. As shown in Table 1, four of the six model updates associated with the Ver. 2.4.6 Model were feature enhancements while two of them were bugfixes. As will be explained later, the feature enhancements in Updates 1, 2, and the bugfix in Update 3 impact the model outputs (though each having a very marginal effect at the aggregate level), whereas Updates 4, 5, 6 do not affect the model results. The first 3 updates (Updates 1-3) have been documented in three separated memoranda.<sup>15 16 17</sup>

---

<sup>14</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>.

<sup>15</sup> Ray Ngo, "Migration of the Transit Walkshed Process from ArcPy to GeoPandas," Memorandum, March 6, 2023

<sup>16</sup> Ray Ngo, "Migration from AEMS to TRANSIMS ModeChoice for the Mode Choice Modeling of the Gen2/Ver. 2.4.6 Model," Memorandum, April 18, 2023.

<sup>17</sup> Feng Xie, "Implement a Bugfix to the Bus Speed Degradation Process in Gen2/Ver. 2.4 Model," Memorandum, May 20, 2022

Table 1 Updates made to the Ver. 2.4.6 Model, compared to Ver. 2.4

#	Description	Type of update	Further details and benefit(s)	Changes model results?
1	Migration of the Transit Walkshed Process from ArcPy to GeoPandas	Feature enhancement	<p>To resolve compatibility and stability issues associated with the ArcPy process, the process was migrated to GeoPandas, a Python library that combines the capabilities of Pandas and Shapely to work with geospatial data. The migration to GeoPandas was driven by several key advantages of the GeoPandas library:</p> <ul style="list-style-type: none"> <li>• First, GeoPandas is open-source and independent from Cube's ArcGIS Runtime Engine, making it more accessible to users and less prone to issues caused by ArcGIS Engine Runtime or ArcGIS Desktop.</li> <li>• Second, the process that utilizes the GeoPandas library is compatible with any Cube 6 released version and potentially with future Cube releases as well.</li> </ul>	Yes
2	Migration of the mode choice process from AEMS to TRANSIMS ModeChoice	Feature enhancement	<p>Due to issues such as instability of running the AEMS mode choice step with newer Cube versions, unavailability of source code, and a lack of technical support, staff decided to migrate to a software program known as TRANSIMS ModeChoice, an open-source C++ program, as a replacement for AEMS. ModeChoice has several advantages over AEMS, including active maintenance and support, better support for Cube, and enhanced functionalities such as:</p> <ul style="list-style-type: none"> <li>• Ease of compilation for various operating systems.</li> <li>• Well-organized and structured code that can be easily understood and updated by other programmers.</li> <li>• Improved accuracy, easy integration into the model workflow, more flexibility and computational clarity.</li> <li>• More efficient processing procedures, more report and output file options, and built-in calibration capabilities for both mode constants and demographic subdivision constants.<sup>18</sup></li> </ul>	Yes

<sup>18</sup> AECOM, "FY 2014 Final Report: Assistance with Development and Application of the National Capital Region Transportation Planning Board Travel Demand Model," Final Report (National Capital Region Transportation Planning Board, Metropolitan Washington Council of Governments, August 18, 2014).

#	Description	Type of update	Further details and benefit(s)	Changes model results?
3	Compute new bus speed degradation factor	Bugfix	The latest local bus speed/runtime degradation process was developed in 2016 <sup>19</sup> based on the factors estimated using modeling data from 2015. <sup>20</sup> This was not a problem when the degradation factors were initially developed. However, as time went by, staff noticed a discrepancy in the current local bus speed/runtime degradation process, for the bus degradation factors were not reflecting the degradation of local bus runtimes relative to the baseline when the base year changes. A fix was instituted without needing to update the original bus degradation factors. The updated process looks up the bus factors for both the current CPI year and modeled year (relative to 2015) and calculates the new bus degradation factor based on the original factors for both years. <sup>21</sup>	Yes
4	Check rows in transit mode files exceeding 144 columns	Feature enhancement	To prevent transit line ("mode") files from being truncated when exceeding 144 columns, a process was developed that would check each row in the transit line files.	No
5	Clean up transit skimming/assignment report files	Feature enhancement/bugfix	Produce mode-specific transit skim/assignment report files.	No
6	Added automated shutdown feature	Feature enhancement	An optional function to shut down the computer automatically after completing a model run. This feature is useful when running the model on an on-demand cloud server where the cost is incurred based on time used.	No

<sup>19</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.

<sup>20</sup> Meseret Seifu and Ronald Milone, "Update of Local Bus Speed Degradation Model," Memorandum, March 19, 2015.

<sup>21</sup> Feng Xie, "Implement a Bugfix to the Bus Speed Degradation Process in Gen2/Ver. 2.4 Model," Memorandum, May 20, 2022.



### 1.3.1 Update 1: Migration of the Transit Walkshed Process from ArcPy to GeoPandas

This section is largely extracted from a memo documenting the migration of the transit walkshed process from ArcPy to GeoPandas.<sup>22</sup>

#### 1.3.1.1 Motivation

The need to revisit this component of the model was identified due to the compatibility and stability issues associated with the ArcPy process. The ArcPy process requires either Cube's built-in ArcGIS Engine Runtime or ArcGIS Desktop, but each Cube version is compatible with only certain ArcGIS Desktop versions. For instance, Cube 6.4.1 is compatible with ArcGIS versions 9.3 through 10.3.1, while Cube 6.5 is compatible with versions 9.3 through 10.8.1. In cases where two incompatible Cube and ArcGIS Desktop versions are installed, the ArcPy-based process will crash. This means that every time when upgrading to a new Cube version, which typically comes with a new ArcGIS Engine Runtime and a different range of compatible ArcGIS versions, there is a need to make some minor code changes to make the ArcPy process work. But the code change itself does not guarantee a stable performance. Users have reported crashes of the COG/TPB Gen2 Travel Model when running the ArcPy process using compatible Cube and ArcGIS versions or using a computer with only Cube and Cube ArcGIS Engine Runtime installed. These crashes were associated with both Cube 6.4.5 and Cube 6.5.<sup>23</sup>

#### 1.3.1.2 Solution

The migration to GeoPandas<sup>24</sup> was driven by several key advantages of the GeoPandas library. First, GeoPandas is open-source and independent from Cube's ArcGIS Runtime Engine, making it more accessible to users and less prone to issues caused by ArcGIS Engine Runtime or ArcGIS Desktop. Second, the process that utilizes the GeoPandas library is compatible with any Cube 6 released version, and also potentially future Cube releases. As such, GeoPandas was chosen as the replacement for the ArcPy package.

The proposed GeoPandas-based process retains the Cube/Voyager scripts that prepare the inputs from the transit line files coded in Cube TRNBUILD or Public Transport (PT) format, and all the input files to the process remain the same. Like the ArcPy-based process, a working folder called "Transit\_Walksheds\_GIS" is also created with two sub-folders, "input" and "output", in it. The final output file, AreaWalk.txt, is copied to the main scenario "Inputs" folder and replaces any existing copy. By default, the transit walkshed process is turned off in the chain of model execution steps and the AreaWalk.txt file included in the model transmittal package will be used as the model input. When the process is turned on by setting "SET RUN\_WALKACCESS=True" on Line 27 of the "run\_Model\_[year].bat" batch file, however, it will be executed as part of a model run and will update the AreaWalk.txt file during the model run. Users do not need to run the transit walkshed process if their model run does not involve a change to the transit inputs, especially the location of transit stops.

---

<sup>22</sup> Ray Ngo, "Migration of the Transit Walkshed Process from ArcPy to GeoPandas," Memorandum, March 6, 2023.

<sup>23</sup> Ray Ngo, Feng Xie, and Mark Moran, "User's Guide For the COG/TPB Gen2/Version 2.4 Travel Demand Forecasting Model" (Washington, D.C.: Metropolitan Washington Council of Governments, National Capital Region Transportation Planning Board, March 15, 2021), [https://www.mwcog.org/assets/1/6/mwcog\\_tpb\\_travel\\_model\\_v2.4\\_user\\_guide\\_final.pdf](https://www.mwcog.org/assets/1/6/mwcog_tpb_travel_model_v2.4_user_guide_final.pdf).

<sup>24</sup> GeoPandas, <https://geopandas.org>, 2023.

The migration from ArcPy-based transit walkshed process to GeoPandas-based process with a buffering resolution of 128 significantly improves the runtime performance, reducing the runtime by a factor of 3, while maintaining the accuracy level of walkshed areas in the AreaWalk.txt file. The resolution value determines the number of segments used to approximate a quarter circle around a point. Moreover, this migration enhanced the process stability, with both old and new Cube versions.

Update 1 changed one script and three batch files, as listed in Table 2. The modifications made to these files resulted in marginal differences in the AreaWalk.txt file generated by the GeoPandas-based and ArcPy-based transit walkshed processes.

Table 2 Affected model components due to Update 1

Purpose	File name	Changes
Revised treatment in the Transit Walkshed Process	<i>Scripts/Walkshed_Process.py</i>	New
	<i>ArcPy_Walkshed_Process.bat</i>	New
	<i>run_Model_[year].bat</i>	Revised
	<i>run_ModelSteps_[year].bat</i>	Revised

### Changes made to *MWCOG\_ArcPy\_Walkshed\_Process.py*

Figure 1 below shows the new transit walkshed process migrated from the ArcPy to the open source GeoPandas library. The new process, *Walkshed\_Process.py*, is based on a Python library that combines the capabilities of Pandas and Shapely to work with geospatial data. Since the original process is replaced with a new process, only the new process is presented in this user's guide. An interested user can view the changes by comparing the ArcPy process of the Ver. 2.4 Model to that of the GeoPandas process of the Ver. 2.4.6 Model.



```

Scripts/Walkshed_Process.py -0 +41
Hunk History
1 import geopandas as gp
2 import numpy as np
3
4 ALLPK = gp.read_file("./input/AllStops_PK.shp")
5 ALLOP = gp.read_file("./input/AllStops_OP.shp")
6 MTLRT = gp.read_file("./input/MetroandLRT_AllDay.shp")
7
8 taz = gp.read_file("../TPBTAZ3722_TPBMod/TAZLand3722.shp")
9
10 taz['taz_area'] = taz.area
11
12 SHR = 2640
13 LNG = 5280
14 fnames = {'mtlrt': MTLRT, 'allpk': ALLPK, 'allop':ALLOP }
15 dists = {'shr':SHR, 'lng':LNG}
16
17 df = taz[['tazID', 'taz_area']]
18
19 for f_key in fnames:
20     for d_key in dists:
21         buff = fnames[f_key].buffer(dists[d_key], resolution=32)
22         buff.to_file(f"buff_{f_key}{d_key}.shp")
23         buff = buff.unary_union
24         buff_inter = taz['geometry'].intersection(buff)
25         buff_inter.to_file(f"buff_intersect_{f_key}{d_key}.shp")
26         intersect_area = buff_inter.area.rename(f"{f_key}{d_key}")
27         df = df.merge(intersect_area, left_index=True, right_index=True)
28
29 out = df.copy()
30 out.columns = out.columns.str.upper()
31 cols = out.columns[1:]
32 out[cols] = out[cols].apply(Lambda x:(x/(5280*5280)).round(4))
33
34 out.to_csv("output/AreaWalk.csv", index=None)
35
36 formats='%+6s %+9s %+9s %+9s %+9s %+9s %+9s %+9s'
37 headers = [format(str(x),y.replace('%+', '>'))
38             for x, y in zip(out.columns,formats.split())]
39
40 np.savetxt('output/AreaWalk.txt', out.values, fmt=formats,
41            header=' '.join(headers), comments='')
42

```

Figure 1 The New Transit Walkshed Process, *Walkshed\_Process.py* in the Ver. 2.4.6 model

### Changes made to *ArcPy\_Walkshed\_Process.bat*

There were 100 lines of new codes in *ArcPy\_Walkshed\_Process.bat*, but Figure 2 captures only a subset of the code. Interested users may view the batch file using a text editor.

```

1 @ECHO OFF
2 IF [%1] == [] goto usage
3 CD %1
4
5 SET orig_dir=%CD%
6 SET WalkshedDirName=Transit_Walksheds_GIS
7
8 rem Set the Python path by changing the USERNAME in as your log in account. For example,
9 rem if USERNAME is rngo, change to C:\Users\rngo\.conda\envs\gen2_model\python.exe
10 rem If Mambaforge is installed for only one user on the computer (not for all users),
11 rem the path should be: C:\Users\USERNAME\AppData\Local\mambaforge\envs\gen2_model
12 rem SET python_bindir=C:\Users\USERNAME\.conda\envs\gen2_model\python.exe
13
14 ECHO. Using Python from Directory = %python_bindir%
15 CD /D %orig_dir%
16
17 :: Create directories
18 ECHO.
19 ECHO.1) Creating Subdirectories ...
20 ECHO.
21 IF NOT EXIST "inputs\%WalkshedDirName%\input" MKDIR "inputs\%WalkshedDirName%\input"
22 IF NOT EXIST "inputs\%WalkshedDirName%\output" MKDIR "inputs\%WalkshedDirName%\output"
23
24 :: Change Working Directory
25
26 DEL /F /Q /S inputs\%WalkshedDirName%\*
27 CD /D inputs\%WalkshedDirName%

```

Figure 2 Excerpt from the new Transit Walkshed Process, *Walkshed\_Process.bat*

The changes made to the “run model”/wrapper batch files and the “run model steps” batch files due to the updates made to the transit walkshed process are described in detail in sections 6.1.1 and 6.1.2.

In application, the model user does not always need to run the transit walkshed process. Most of the time, one can run the model with “RUN\_WALKACCESS” turned off. The user needs to run the transit walkshed process only if the transit network inputs are altered and the walksheds surrounding transit stations/stops need to be updated. Note that, running the GeoPandas transit walkshed process requires additional software installation and preparation steps, which are detailed in Chapter 5.

### 1.3.2 Update 2: Migration from AEMS to TRANSIMS Mode Choice

This section is largely extracted from a memo documenting migration from AEMS to TRANSIMS ModeChoice for the Mode Choice Modeling of the Gen2/Ver. 2.4.6 Model.<sup>25</sup>

<sup>25</sup> Ray Ngo, “Migration from AEMS to TRANSIMS ModeChoice for the Mode Choice Modeling of the Gen2/Ver. 2.4.6 Model,” Memorandum, April 18, 2023.

### 1.3.2.1 Motivation

The Gen2 Model uses a nested-logit mode choice (NLMC) modeling process to estimate the mode of transportation that travelers will choose. Up until the Ver. 2.4 Model, the mode choice process in the Gen2 Model has been implemented using a software program called AEMS, a proprietary FORTRAN program developed in the early 1990s by Bill Woodford at KPMG. After three decades, however, AEMS has become outdated, and consultants have stopped providing technical support for it. Due to issues such as instability of running the AEMS mode choice step with newer Cube versions, unavailability of source code, and a lack of technical support, staff decided to migrate to a software program known as TRANSIMS ModeChoice,<sup>26</sup> an open-source, C++ program written by David Roden of AECOM in 2010 as a replacement for AEMS.

### 1.3.2.2 Solution

In 2014, AECOM converted the mode choice process in the TPB's Gen2/Ver. 2.3.52 Model from AEMS to ModeChoice 6.0.5. However, staff tested and found that the integration resulted in undesirable changes in model outputs. Thus, staff did not move forward with the migration in the subsequent Gen2/Ver. 2.3 Model versions. Over time, AECOM made a series of updates to the TRANSIMS ModeChoice software, including rounding and software structural changes. In 2022, staff updated the ModeChoice setup and tested the TRANSIMS ModeChoice, Version 7.5.8,<sup>27</sup> which was the latest version that staff retrieved, in the Gen2/Ver 2.3.75 Travel Model and found marginal changes in modeling results after the migration. Because of this success, staff decided to implement the migration on the Gen2/Ver. 2.4 Model, which became the TPB's adopted, production-use regional travel demand model for the metropolitan Washington region on June 15, 2022. The Ver. 2.4 Model includes a NLMC model that was re-calibrated to year-2007 conditions following updates to person trip calibration targets for commuter rail, resulting in a different set of nesting constants from those in the previous model versions. Therefore, all the updated constants in the AEMS formats of the Ver. 2.4 Model were subsequently converted into the ModeChoice formats.

The migration of the mode choice process from AEMS to TRANSIMS ModeChoice was a success. The ModeChoice process improved the model stability/compatibility with newer Cube versions, such as Cube 6.5, and runtime performance, while having little impact on model outputs. With the transition from AEMS to ModeChoice, staff will benefit from the potential software updates and technical support provided by AECOM staff, particularly when newer Cube versions are released.<sup>28</sup>

Overall, the setup of ModeChoice 7.5.8 is very similar to that in ModeChoice 6.0.5. The main differences are the rounding precision and ModeChoice control keys. Staff also changed the location of some setup

---

<sup>26</sup> <https://sourceforge.net/projects/transims/>, 2023.

<sup>27</sup> AECOM, "ModeChoice, Version 7.5.7" (Arlington, Virginia, September 15, 2021), <https://sourceforge.net/projects/transims/files/documentation/TRANSIMS%20Version%207.5%20User%20Guides.7z/download>.

<sup>28</sup> Ray Ngo, "Migration from AEMS to TRANSIMS ModeChoice for the Mode Choice Modeling of the Gen2/Ver. 2.4.6 Model," Memorandum, April 18, 2023.

components to be consistent with the Ver. 2.4 Model setup. For example, the ModeChoice setup files were placed under the controls folder of the Ver. 2.4.6 Model. Section 7.2.1 of AECOM's FY 14 full report<sup>29</sup> detailed the changes needed when migrating from AEMS to ModeChoice. For example, in the Ver. 2.4 Model, the mode choice step utilizes 5 AEMS control files, one for each trip purpose; As shown in Figure 1, however, these 5 files were converted to 11 key files (5 control files, 5 constant files, and 1 script file) and other auxiliary files in the TRANSIMS ModeChoice setup of the Ver. 2.4.6 Model.

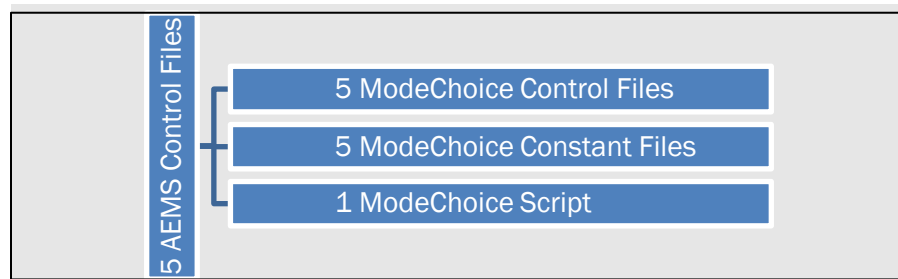


Figure 3 AEMS Control Files Converted to the ModeChoice Setup<sup>30</sup>

Table 3 Affected model components due to Update 2

Purpose	File name	Changes
Migrate the mode choice process from AEMS to TRANSIMS ModeChoice	<i>MC_Purp.bat</i>	Revised
	<i>Mode_Choice_Parallel.bat</i>	Revised
New ModeChoice control files which define the input and output file names, mode choice model nesting structure, market segments, utility function parameter values and selected reports	<i>controls/HBW_AM_ModeChoice.ctl</i>	Revised
	<i>controls/HBS_OP_ModeChoice.ctl</i>	Revised
	<i>controls/HBO_OP_ModeChoice.ctl</i>	Revised
	<i>controls/NHW_OP_ModeChoice.ctl</i>	Revised
	<i>controls/NHO_OP_ModeChoice.ctl</i>	Revised
New ModeChoice constants file which stores mode-specific constants and income-level adjustment constants for each trip purpose	<i>Controls/ ModeChoice_Constants/ HBW_AM_ModeChoice.csv</i> <i>HBW_AM_ModeConstants.csv.def</i> <i>HBW_AM_BiasConstants.csv</i> <i>HBW_AM_BiasConstants.csv.def</i>	new
	<i>Controls/ ModeChoice_Constants/ HBS_OP_ModeChoice.csv</i> <i>HBS_OP_ModeConstants.csv.def</i> <i>HBS_OP_BiasConstants.csv</i>	new

<sup>29</sup> AECOM, "FY 2014 Final Report: Assistance with Development and Application of the National Capital Region Transportation Planning Board Travel Demand Model."

<sup>30</sup> Page 3, Ray Ngo, "Migration from AEMS to TRANSIMS ModeChoice for the Mode Choice Modeling of the Gen2/Ver. 2.4.6 Model," Memorandum, April 18, 2023

Purpose	File name	Changes
	<i>HBS_OP_BiasConstants.csv.def</i>	
	<i>Controls/ ModeChoice_Constants/ HBO_OP_ModeChoice.csv</i> <i>HBO_OP_ModeConstants.csv.def</i> <i>HBO_OP_BiasConstants.csv</i> <i>HBO_OP_BiasConstants.csv.def</i>	new
	<i>Controls/ ModeChoice_Constants/ NHW_OP_ModeChoice.ctl</i> <i>NHW_OP_ModeConstants.csv.def</i> <i>NHW_OP_BiasConstants.csv</i> <i>NHW_OP_BiasConstants.csv.def</i>	new
	<i>Controls/ ModeChoice_Constants/ NHO_OP_ModeChoice.csv</i> <i>NHO_OP_ModeConstants.csv.def</i> <i>NHO_OP_BiasConstants.csv</i> <i>NHO_OP_BiasConstants.csv.def</i>	new
Contains all ModeChoice Control Keys	<i>Controls/ ModeChoice_CommonKeys.ctl</i>	new
Contains utility functions and conditional overrides to customize the mode choice model	<i>Controls/ModeChoice_Script.txt</i>	new
Geographic market segmentation	<i>Controls/Segment_Map.csv</i> <i>Controls/Segment_Map.csv.def</i>	new
Zonal data file to store percent-walk-to-transit values	<i>Controls/ZONEV2.A2F.def</i>	new

### Changes made to *MC\_Purp.bat*

Changes made to the *MC\_Purp.bat* file are shown in Figure 4.

Figure 1<sup>31</sup> below shows the first section of changes made to the *MC\_Purp.bat*. The 8 red and 10 green lines indicate the lines before and after changes. The far-left column shows the line numbers of the scripts before the change and the next column shows the line numbers of the scripts after the change. If a green line is blank, the line is deleted from the batch file. If a green line is added without a prior red line, the green line is simply added to the file.

For example, the original lines 6-7:

```

6  if exist %2_NL_MC.* del %2_NL_MC.*
7  ..\timethis ..\software\AEMS ..\controls\%2_NL_MC.ctl | ..\tee %2.txt

```

<sup>31</sup> We have used the Sublime Merge text editor to create these figures showing the script differences.

are removed and replaced by new lines 6-9 in the updated batch file:

```
6 call ..\software\ModeChoice.exe -K ..\controls\%2_ModeChoice.ct1 @arg_purp@=%2 @arg_threads@=%3 @arg_calibtarget@=
7 set exitcode=%errorLevel%
8 if %exitcode% equ 1 goto error
9 if %exitcode% gtr 2 goto error
```

```
5 @date /t & time /t
6 if exist %2_NL_MC.* del %2_NL_MC.*
7 ..\timethis ..\software\AEMS ..\controls\%2_NL_MC.ct1 | ..\tee %2.txt
6 call ..\software\ModeChoice.exe -K ..\controls\%2_ModeChoice.ct1 @arg_purp@=%2 @arg_threads@=%3 @arg_calibtarget@=
7 set exitcode=%errorLevel%
8 if %exitcode% equ 1 goto error
9 if %exitcode% gtr 2 goto error
8 10 @echo %2 Finished
9 11 @date /t & time /t
10 12
11 13 goto end
12 14
13 15 :error
14 16 REM Processing Error...
15 @echo ERROR: %2 Mode Choice failed >> %2.txt
16 @echo %2 Error > %2.err
17 @echo Error > %2.done
17 @echo ERROR: %2 Mode Choice failed with exitcode %exitcode% > %2.err
18 @echo ERROR: %2 Mode Choice failed with exitcode %exitcode% > %2.done
18 19 @pause
20 set exitcode=
19 21 exit
20 22 :end
21 @echo %2 Mode Choice Done >> %2.txt
22 @echo Done > %2.done
23 exit
23 @echo %2 Mode Choice Done with exitcode %exitcode% > %2.done
24 set exitcode=
```

Figure 4 Changes made to the MC\_Purp.bat file

### Changes made to *Mode\_Choice\_Parallel.bat*

There were 75 modified lines of code in *Mode\_Choice\_Parallel.bat*, but Figure 5 captures only a subset of the changes. Interested users may view the rest of the changes by comparing the updated batch file in Ver. 2.4.6 to its counterpart in Ver. 2.4.

```

Mode_Choice_Parallel.bat -75 +44

2 2 @echo off
3 3 CD %1
4 4
5 @rem Copy iteration-specific inputs to generic names
6
7 if exist %_iter_%_hbw_NL.ptt copy %_iter_%_hbw_NL.ptt HBW_INCOME.PTT /y
8 if exist %_iter_%_hbs_NL.ptt copy %_iter_%_hbs_NL.ptt HBS_INCOME.PTT /y
9 if exist %_iter_%_hbo_NL.ptt copy %_iter_%_hbo_NL.ptt HBO_INCOME.PTT /y
10 if exist %_iter_%_nhw_NL.ptt copy %_iter_%_nhw_NL.ptt NHW_INCOME.PTT /y
11 if exist %_iter_%_nho_NL.ptt copy %_iter_%_nho_NL.ptt NHO_INCOME.PTT /y
12
13 if exist %_prev_%_hwy_AM.SKM copy %_prev_%_hwy_AM.SKM HWYAM.SKM /y
14 if exist %_prev_%_hwy_OP.SKM copy %_prev_%_hwy_OP.SKM HWYOP.SKM /y
15
16 if exist %_iter_%_TRNAM_CR.SKM copy %_iter_%_TRNAM_CR.SKM TRNAM_CR.SKM /y
17 if exist %_iter_%_TRNAM_AB.SKM copy %_iter_%_TRNAM_AB.SKM TRNAM_AB.SKM /y
18 if exist %_iter_%_TRNAM_MR.SKM copy %_iter_%_TRNAM_MR.SKM TRNAM_MR.SKM /y
19 if exist %_iter_%_TRNAM_BM.SKM copy %_iter_%_TRNAM_BM.SKM TRNAM_BM.SKM /y
20
21 if exist %_iter_%_TRNOP_CR.SKM copy %_iter_%_TRNOP_CR.SKM TRNOP_CR.SKM /y
22 if exist %_iter_%_TRNOP_AB.SKM copy %_iter_%_TRNOP_AB.SKM TRNOP_AB.SKM /y
23 if exist %_iter_%_TRNOP_MR.SKM copy %_iter_%_TRNOP_MR.SKM TRNOP_MR.SKM /y
24 if exist %_iter_%_TRNOP_BM.SKM copy %_iter_%_TRNOP_BM.SKM TRNOP_BM.SKM /y
25
5 @echo Copy zonal definition file
6 copy ..\controls\ZONEV2.A2F.def ZONEV2.A2F.def /y

79 57 :Mode_Choice_is_Done
80 58 @rem - This step is to collect all the output from the MC to the log file.
81 @type HBW.txt
82 @type HBS.txt
83 @type HBO.txt
84 @type NHW.txt
85 @type NHO.txt
59 @type %_iter_%_HBW_AM_ModeChoice.prn
60 @type %_iter_%_HBS_OP_ModeChoice.prn
61 @type %_iter_%_HBO_OP_ModeChoice.prn
62 @type %_iter_%_NHW_OP_ModeChoice.prn
63 @type %_iter_%_NHO_OP_ModeChoice.prn
86 64
87 65
88 @echo Finished Mode Choice
89 67 @date /t & time/t
90 68
91 @rem COPY GENERIC MODE CHOICE OUTPUT FILES
92 @rem TO INTERATION-SPECIFIC NAMES
93
94 if exist HBW_NL_MC.MTT copy HBW_NL_MC.MTT %_iter_%_HBW_NL_MC.MTT /y
95 if exist HBS_NL_MC.MTT copy HBS_NL_MC.MTT %_iter_%_HBS_NL_MC.MTT /y
96 if exist HBO_NL_MC.MTT copy HBO_NL_MC.MTT %_iter_%_HBO_NL_MC.MTT /y
97 if exist NHW_NL_MC.MTT copy NHW_NL_MC.MTT %_iter_%_NHW_NL_MC.MTT /y
98 if exist NHO_NL_MC.MTT copy NHO_NL_MC.MTT %_iter_%_NHO_NL_MC.MTT /y
99
100 @ping -n 11 127.0.0.1
69 CHOICE /M:"Pausing for 10 seconds" /C:C /D:C /T:10
    
```

Figure 5 Beginning and end sections of changes made to *Mode\_Choice\_Parallel.bat*

### ModeChoice control files: *HBW\_AM\_ModeChoice.ctl*

ModeChoice control files define the input and output file names, mode choice model nesting structure, market segments, utility function parameter values and selected reports. The ModeChoice constants file stores mode-specific constants and income-level adjustment constants for each trip purpose.

Due to the large number of changes made to the ModeChoice control and constant files, the user's guide presents only excerpts from the ModeChoice control and constants files for the home-based work (HBW) trip purpose, as shown below in Figure 6 and Figure 7, respectively.

```

controls/HBW_AM_ModeChoice.ctf -0 +50
Hunk History
1 TITLE MWMCOG Version 2.4 AM Peak HBW Mode Choice (Peak Home-Based Work)
2 REPORT_FILE %_iter_%_@arg_purp@_ModeChoice.prn
3
4 CONTROL_KEY_FILE ..\Controls\ModeChoice_CommonKeys.ctf
5
6 #---- ModeChoice Control Keys ----
7
8 TRIP_TABLE_FILE %_iter_%_hbw_NL.ptt
9 SKIM_FILE_2 %_prev_%_hwy_AM.SKM
10 SKIM_FILE_3 %_iter_%_TRNAM_CR.SKM
11 SKIM_FILE_4 %_iter_%_TRNAM_AB.SKM
12 SKIM_FILE_5 %_iter_%_TRNAM_MR.SKM
13 SKIM_FILE_6 %_iter_%_TRNAM_BM.SKM
14
15 MODEL_NAMES HBWI1Psn, HBWI2Psn, HBWI3Psn, HBWI4Psn
16 MODEL_PURPOSE_CODES 1 //---- HBW=1, HBS=2, HBO=3, NHW=4, NHO=5 ----
17 MODEL_PERIOD_CODES 1 //---- PEAK=1, OFFPEAK=2 ----
18
19 VEHICLE_TIME_VALUES -0.02128
20 WALK_TIME_VALUES -0.04256
21 DRIVE_ACCESS_VALUES -0.03192
22 WAIT_TIME_VALUES -0.05320
23 TRANSFER_COUNT_VALUES -0.00000
24 PENALTY_TIME_VALUES -0.05320
25 TERMINAL_TIME_VALUES -0.05320
26 COST_VALUES -0.00185, -0.00093, -0.00062, -0.00046
27
28 MODE_CONSTANT_FILE ..\Controls\ModeChoice_Constants\@arg_purp@_ModeConstants.csv
29 MODE_BIAS_FILE ..\Controls\ModeChoice_Constants\@arg_purp@_BiasConstants.csv
    
```

Figure 6 Example HBW AM ModeChoice Control File

SEGMENT	MODE	HBWI1Psn	HBWI2Psn	HBWI3Psn	HBWI4Psn
1	AUTO	0	0	0	0
1	TRANSIT	2.85474	2.85474	2.85474	2.85474
1	SOV	0	0	0	0
1	HOV	-1.28317	-1.28317	-1.28317	-1.28317
1	SR2	0	0	0	0
1	SR3	-1.5417	-1.5417	-1.5417	-1.5417
1	WALK	0	0	0	0
1	PNR	-3.32099	-3.32099	-3.32099	-3.32099
1	KNR	-7.02088	-7.02088	-7.02088	-7.02088
1	WK_CR	-0.74745	-0.74745	-0.74745	-0.74745
1	WK_BUS	-1.37011	-1.37011	-1.37011	-1.37011
1	WK_BUS_M	-1.27764	-1.27764	-1.27764	-1.27764
1	WK_MR	0	0	0	0
1	PNR_CR	-3.39847	-3.39847	-3.39847	-3.39847
1	PNR_BUS	-2.90566	-2.90566	-2.90566	-2.90566
1	PNR_BUS_I	-0.09631	-0.09631	-0.09631	-0.09631

Figure 7 Example HBW AM ModeChoice Constants File

### Example ModeChoice\_CommonKeys.ctf

Figure 8 shows only the beginning and end section of the ModeChoice\_CommonKeys.ctf file, for the file contains 252 new lines of the ModeChoice control keys.



```

controls/ModeChoice_CommonKeys.ctl -0 +252
1 REPORT_DIRECTORY      ..\%scenar%
2
3 REPORT_FLAG           FALSE                //---- TRUE/FALSE, YES/NO, 1/0, T/F, Y/N
4 PAGE_LENGTH           9999                //---- >= 0
5 PROJECT_DIRECTORY     NULL
6 DEFAULT_FILE_FORMAT   COMMA_DELIMITED    //---- TEXT, BINARY, FIXED_COLUMN, COMMA_DELIMITED,
SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, ARCVIEW, SQLITE3, VERSION3, TPPLUS, CUBE, TRASCAD,
STRUCTURE
7 NUMBER_OF_THREADS     @arg_threads@      //---- 1..128
8 RANDOM_NUMBER_SEED    2029623231        //---- 0 = computer clock, > 0 = fixed
9
10 #---- System File Keys ----
11
12 ZONE_FILE             ZONEV2.A2F
13 ZONE_FORMAT           FIXED_COLUMN
14
15 #---- ModeChoice Control Keys ----
16
17 TRIP_TABLE_FORMAT     CUBE:MATRIX
18 SKIM_FORMAT           CUBE:MATRIX
19 NEW_TRIP_TABLE_FORMAT CUBE:MATRIX
20 NEW_TRIP_TABLE_RESOLUTION 2
21
22 MODE_CHOICE_SCRIPT    ..\Controls\ModeChoice_Script.txt
23
24 SEGMENT_MAP_FILE      ..\Controls\Segment_Map.csv
25 ORIGIN_MAP_FIELD      AREA_TYPE
26 DESTINATION_MAP_FIELD AREA_TYPE
--
228 // CALIBRATION_SCALING_FACTOR      1.0                //---- 1.0..5.0
229 // MAX_CALIBRATION_ITERATIONS      20                //---- 1..1000
230 // CALIBRATION_EXIT_RMSE           5.0                //---- 0.0..50.0
231 // ADJUST_FIRST_MODE_CONSTANTS     FALSE              //---- TRUE/FALSE, YES/NO, 1/0, T/F, Y/N
232 // GROUP_MODEL_CONSTANTS           FALSE              //---- TRUE/FALSE, YES/NO, 1/0, T/F, Y/N
233 // PLAN_SKIMS_IN_MEMORY            FALSE              //---- TRUE/FALSE, YES/NO, 1/0, T/F, Y/N
234 // REPORT_AFTER_ITERATIONS         NONE               //---- e.g., 1, 2, 4..10, 100..200, 300
235 // NEW_MODE_CONSTANT_FILE          //---- [project_directory]filename
236 // NEW_MODE_CONSTANT_FORMAT        TAB_DELIMITED      //---- TEXT, BINARY, FIXED_COLUMN,
COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, ARCVIEW, SQLITE3, VERSION3, TPPLUS,
CUBE, TRASCAD, STRUCTURE
237 // NEW_CALIBRATION_DATA_FILE       //---- [project_directory]filename
238 //
239 // #---- Program Report Keys ----
240 //
241 // MODECHOICE_REPORT_1              //---- program report name
242 //                                  //---- MODE_CHOICE_SCRIPT
243 //                                  //---- MODE_CHOICE_STACK
244 //                                  //---- MODE_SUMMARY_REPORT
245 //                                  //---- MARKET_SEGMENT_REPORT
246 //                                  //---- CALIBRATION_REPORT
247 //                                  //---- TARGET_DATA_REPORT
248 //                                  //---- MODE_VALUE_SUMMARY
249 //                                  //---- SEGMENT_VALUE_SUMMARY
250 //                                  //---- ACCESS_MARKET_SUMMARY
251 //                                  //---- LOST_TRIPS_REPORT
252 //

```

Figure 8 Example of beginning and end section of *ModeChoice\_CommonKeys.ctl* file

### Example *ModeChoice\_Script.txt*

The ModeChoice script file, which is common for all trip purposes, contains utility functions and conditional overrides that help customize the mode choice model. Figure 9 shows a section of an example ModeChoice script file.

```

controls/ModeChoice_Script.txt -0 +358
1 //V2
2 // This script is modified from the one in S:\DC\MWCOG Model Assistance\Task 13 - PT & Mode Choice
  II\ModeChoice_Conversion\New CTLs
3 //The walk access, walk to Metrorail, drive access, PNR and KNR to Metrorail have been changed
4
5
6 //--- ASSUMED MATRIX ORGANIZATION
7 //--- TRIP_FILE = TRIP TABLE (SEPARATE FOR EACH PURPOSE)
8 //--- HBWI1Psn INCOME 1 (HOME-BASED)/ALL NHB TRIPS
9 //--- HBWI2Psn INCOME 2 (HOME-BASED)
10 //--- HBWI3Psn INCOME 3 (HOME-BASED)
11 //--- HBWI4Psn INCOME 4 (HOME-BASED)
12
13 //--- SKIM_FILE_2 = HIGHWAY SKIMS (SEPARATE FOR PEAK AND OFFPEAK)
14 //--- SovTime 1 SOV TIME (MIN)
15 //--- SovDst10 2 SOV DIST (0.1 MILES)
16 //--- SovTo11t 3 SOV TOLL (2007 CENTS)
17 //--- Hv2Time 4 HOV2 TIME (MIN)
18 //--- Hv2Dst10 5 HOV2 DIST (0.1 MILES)
19 //--- Hv2To11 6 HOV2 TOLL (2007 CENTS)
20 //--- Hv3Time 7 HOV3+ TIME (MIN)
21 //--- Hv3Dst10 8 HOV3+ DIST (0.1 MILES)
22 //--- Hv3To11 9 HOV3+ TOLL (2007 CENTS)
23
24 //--- SKIM_FILE_3 = COM. RAIL SKIMS (SEPARATE FOR PEAK AND OFFPEAK)
25 //--- SKIM_FILE_4 = BUS SKIMS (SEPARATE FOR PEAK AND OFFPEAK)
26 //--- SKIM_FILE_5 = METRORAIL SKIMS (SEPARATE FOR PEAK AND OFFPEAK)
27 //--- SKIM_FILE_6 = BUS+METRORAIL SKIMS (SEPARATE FOR PEAK AND OFFPEAK)
28 //--- 1 WLK ACC/EGR (.01 MIN) 15 PNR ACC/EGR (.01 MIN) 33 KNR ACC/EGR (.01 MIN)

```

Figure 9 Example ModeChoice Script File

### 1.3.3 Update 3: Updated process to compute new bus speed degradation factors

This section is largely extracted from a memo documenting implementation of a Bugfix to the Bus Speed Degradation Process in the Gen2/Ver. 2.4 Model.<sup>32</sup>

#### 1.3.3.1 Motivation

In the COG/TPB regional travel models, the runtime information of individual bus routes is coded in transit line files. The baseline bus runtimes are extracted from the most recent bus schedules published by transit operators. For future analysis years, the model uses bus speed/runtime degradation factors to account for an expected slight degradation of local bus speeds/runtimes over time due to increasing roadway congestion. The latest local bus speed/runtime degradation process was developed in 2016 based on the factors estimated using the modeling data in 2015. This process has been adopted both in the Gen2/Ver. 2.4.6 Model for production use and in the Gen3 Model that is currently under development.

However, staff noticed a discrepancy in the current local bus speed/runtime degradation process. Specifically, the current degradation factors assume that the baseline bus runtimes coded in the transit mode files are always from 2015, and thus degrades the runtimes of local buses from 2015 to a specific

<sup>32</sup> Feng Xie, "Implement a Bugfix to the Bus Speed Degradation Process in Gen2/Ver. 2.4 Model," Memorandum, May 20, 2022.

modeled year. When the degradation factors were initially developed in 2015, this assumption was not unreasonable, as the runtimes coded in the transit mode files were indeed extracted from the 2015 bus schedules. As time goes by, however, this assumption could lead to a slight over-estimation of local bus speed/runtime degradation. For example, for the 2022 Update of the Visualize 2045 Long-Range Transportation Plan (LRTP), bus runtimes were extracted from the bus schedules published in 2020 or early 2021 (mainly in 2020). Thus, the degradation factors should have reflected the degradation of local bus runtimes relative to the 2020 baseline, but instead, they still reflect the runtime degradation relative to 2015.

### 1.3.3.2 Solution

Staff proposed a fix to this discrepancy without needing to update the bus degradation factors every year when the baseline schedules change. Staff proposed to revise the "Adjust\_Runtime.s" script so that it reads the "CurrCPIYear" variable (i.e., Current CPI Year) from the "CPI\_File.txt" file, located in the scenario "inputs" folder, which indicates the year when the baseline bus runtime information in transit mode files is extracted, looks up the degradation factors both for the "CurrCPIYear" and for the modeled year, and calculates new bus degradation factors between the current year and the modeled year using the current factors for the modeled year (relative to 2015) divided by the corresponding factors for the Current CPI Year (also relative to 2015). The process is described in detail in section 16.2.2.

Staff implemented this fix in the Ver. 2.4.6 Model and conducted a year-2045 test run. Based on the new degradation factors recorded in the updated transit line files (i.e., Mode[1|6|8][AM|OP|.tb), the fix worked as expected. The testing results also suggested that the proposed update to the bus speed/runtime degradation process had a very marginal effect on the aggregate modeling results. As such, a similar fix was instituted in the developmental Gen3 Model.

Table 4 Affected model components due to Update 3

File name	Changes
Scripts\ Adjust_Runtime.s	Revised
Transit_Skim_All_Modes_Parallel.bat	Revised

### Changes made to Adjust\_Runtime.s and Transit\_Skim\_All\_Modes\_Parallel.bat

Figure 10 captures a portion of the script where changes were made in the Adjust\_Runtime.s script. Figure 11 shows changes made to Transit\_Skim\_All\_Modes\_Parallel.bat file, where error levels were changed from 1 to 2.

```

Scripts/Adjust_Runtime_PT.s -5 +8
Hunk History ...
95 95 ; OK, If there are no ABORTS above then all is well- now read through the
96 96 ; AM/OP and Mode 1,6,8 line files and factor the RUNTIME values as appropriate
97 97 ; with the DBF Lookup
98 ;
98 ; rngo, 5/16/22: apply Feng's TRNBUILD update so that it looks up the bus factors for
99 ; both current CPI year and modeled year, and calculates the bus factors as
100 ; bus factors for modeled year divided by bus factors for the current CPI year
99 101 ;=====
100 102 ;
101 103

121 123
122 124 RUN PGM=MATRIX
123 125 ZONES = 1
126 READ FILE=inputs\CPI_File.txt
124 127
125 128 ;; Read Bus Factors as a lookup, indexed to Model Year
126 129 FileI LOOKUPI[1]= "inputs\Bus_Factor_File.dbf"

132 135 INTERPOLATE=N, FAIL= 1,1,1, LIST=Y ; if "ModeledYear" is not in the range of modeled years listed in the
133 136 ; lookup table, ; "FAIL=1,1,1" ensures bus run time will remain the same.
134 137
135 AMIB_FTR = BFTR(1,@ModeledYear@)
136 AMOB_FTR = BFTR(2,@ModeledYear@)
137 OPIB_FTR = BFTR(3,@ModeledYear@)
138 OPOB_FTR = BFTR(4,@ModeledYear@)
138 AMIB_FTR = BFTR(1,@ModeledYear@)/BFTR(1,CurrCPIYear)
139 AMOB_FTR = BFTR(2,@ModeledYear@)/BFTR(2,CurrCPIYear)
140 OPIB_FTR = BFTR(3,@ModeledYear@)/BFTR(3,CurrCPIYear)
141 OPOB_FTR = BFTR(4,@ModeledYear@)/BFTR(4,CurrCPIYear)
139 142
140 143
141 144 ;;inputs\MODE@MOD@PER@.TB",Fields=1-3
    
```

Figure 10 beginning and end sections of changes made to Adjust\_Runtime.s in Ver. 2.4.6

```

Transit_Skim_All_Modes_Parallel.bat -1 +1
...
42 42 if exist voya*.* del voya*.*
43 43 if exist Adjust_Runtime.rpt del Adjust_Runtime.rpt
44 44 start /w Voyager.exe ..\scripts\Adjust_Runtime.s /start -Pvoya -S..\%1
45 if errorLevel 1 goto error
45 if errorLevel 2 goto error
46 46 if exist voya*.prn copy voya*.prn Adjust_Runtime.rpt /y
47 47
48 48
    
```

Figure 11 Changes made to Transit\_Skim\_All\_Modes\_Parallel.bat in Ver. 2.4.6

### 1.3.4 Update 4: Developed a process that would ensure that no row in the transit line files exceeds 144 columns

#### 1.3.4.1 Motivation

In the transit network, transit line files represent transit services by a series of transit routes, which are stored in transit “line” or “mode” files by time-of-day period AM peak and off-peak period (represented

by the midday period). When a row in the transit mode files goes beyond column 144, the specifications for the corresponding transit route could be truncated, which may lead to a fatal error in the subsequent step. Even worse, if the truncation occurs on a row for the last transit route in the mode file, that route may be skipped altogether without being noticed.

### 1.3.4.2 Solution

A bug fix was instituted in the model script to ensure that no row in the transit mode files exceeds 144 columns.

Table 5 Affected model components due to Update 4

File name	Changes
Scripts\Adjust_Runtime.s	Revised

### Changes made to Adjust\_Runtime.s

As shown in Figure 12 shows beginning section of the Adjust\_Runtime.s script where changes were instituted to make sure that each row in the transit mode files does not exceed 144 columns.

```

10 ; First loop through AM/OP and Mode 1,6,8 Line files
11 ; First loop through AM/OP and all Mode files
12 ; To make sure each row in the Mode files does not exceed 144 columns
13 ; Network staff pointed out that the transit skimming script will report
14 ; an error when the route specifications (such as nodes) go beyond column 144, and that
15 ; If the error had been on the last row of the mode file, the route would have been
16 ; truncated and staff would not have known it
17 ;=====
18 Loop PRD = 1,2 ;Loop through 1>AM and 2>OP periods
19 Loop MOD = 1,10 ; Loop through Modes 1-10
20
21 IF (PRD = 1) PER = 'AM'
22 IF (PRD = 2) PER = 'OP'
23
24 RUN PGM=MATRIX
25 ZONES = 1
26
27 FILEI RECI="inputs\MODE@MOD@@PER@.TB"
28
29 lineno = reci.RECNO
30 IF (strlen(reci)>144)
31 List= 'I quit! Number of columns in Mode@MOD@@PER@.TB file exceeds the limit of 144. Please check Line ',
32 lineno(5.0),':',reci
33 ABORT
34 ENDF
35 ENDRUN
36 ENDLOOP
    
```

Figure 12 Beginning section of Adjust\_Runtime.s in Ver. 2.4.6

### 1.3.5 Update 5: Clean up transit skimming/assignment report files

#### 1.3.5.1 Motivation

Currently in the transit skimming and transit assignment steps of both the Gen2 and Gen3 models, when the report file (\*.rpt) for a specific sub-mode (e.g., “Bus-Metrorail” abbreviated as “BM”) is generated, it copies from all existing \*.prn files, including those generated by the parallel processes for the other sub-modes (e.g., “All Bus” abbreviated as “AB”). As a result, the information contained in the transit skimming/assignment report files could be very confusing. Ideally, a transit skim/transit assignment report file for a specific sub mode should contain printout information for that sub mode only.

#### 1.3.5.2 Solution

Staff modified mode specific transit skims and transit assignment script files (total eight scripts) and two batch files as shown in Table 6. The batch files were modified to create temporary sub-mode specific folders, as shown in Figure 15 line #6, to write out the reports and remove them at the end of the transit assignment parallel process as shown on line #25.

Table 6 Affected model components due to Update 5

File name	Changes
<i>Scripts\Transit_Skims_MR AB CR BM.s</i>	Revised
<i>Scripts\Transit_Assignment_MR AB CR BM.s</i>	Revised
<i>Transit_Assignment_LineHaul_Parallel.bat</i>	Revised
<i>Transit_Skim_LineHaul_Parallel.bat.bat</i>	Revised

#### Changes made to *Transit\_Skims\_MR.S* and *Transit\_Assignment\_MR.s*

Due to the large number of changes made to the transit skimming and transit assignment script files, the user's guide presents only example scripts for the Metrorail Only (MR) sub mode (*Transit\_skims\_MR.s* and *Transit\_Assignment\_MR.s*) as shown in Figure 13 and Figure 14, respectively. Interested users may view the rest of the changes by comparing the updated transit skimming and transit assignment script files in Ver. 2.4.6 to those in Ver. 2.4.

```

Scripts/Transit_Skims_MR.s -46 +46

105 105 ;-----
106 106
107 107 RUN PGM=TRNBUILD
108     NETI = ZONEHWY.NET
108     NETI = ..\ZONEHWY.NET
109 109 MATO = transit.temp.mr.skm
110 110 maxnode = 160000
111 111

261 261
262 262 ;---- Rail Stations & Links (modes 3 & 4) ----
263 263
264     READ FILE = met_node.tb ;---- Metrorail stations
265     READ FILE = met_link.tb ;---- Metrorail links
266     ;READ FILE = com_node.tb ;---- Commuter Rail stations
267     ;READ FILE = com_link.tb ;---- Commuter Rail links
268     READ FILE = lrt_node.tb ;---- LRT stations
269     READ FILE = lrt_link.tb ;---- LRT links
270     ;READ FILE = new_node.tb ;---- Mode10 Stations
271     ;READ FILE = new_link.tb ;---- Mode10 links
264     READ FILE = ..\met_node.tb ;---- Metrorail stations
265     READ FILE = ..\met_link.tb ;---- Metrorail links
266     ;READ FILE = ..\com_node.tb ;---- Commuter Rail stations
267     ;READ FILE = ..\com_link.tb ;---- Commuter Rail links
268     READ FILE = ..\lrt_node.tb ;---- LRT stations
269     READ FILE = ..\lrt_link.tb ;---- LRT links
270     ;READ FILE = ..\new_node.tb ;---- Mode10 Stations
271     ;READ FILE = ..\new_link.tb ;---- Mode10 links
314 314 ;---- Transit Line Cards (modes 1-10) ----
315 315
316     ;READ FILE = MODE1@TIME_PERIOD@.TB ;---- M1- metrobus local
317     ;READ FILE = MODE2@TIME_PERIOD@.TB ;---- M2- metrobus express
318     READ FILE = MODE3@TIME_PERIOD@.TB ;---- M3- metrorail
319     ;READ FILE = MODE4@TIME_PERIOD@.TB ;---- M4- commuter rail
320     READ FILE = MODE5@TIME_PERIOD@.TB ;---- M5- other rail (future)
321     ;READ FILE = MODE6@TIME_PERIOD@.TB ;---- M6- other local bus
322     ;READ FILE = MODE7@TIME_PERIOD@.TB ;---- M7- other express bus
323     ;READ FILE = MODE8@TIME_PERIOD@.TB ;---- M8- other local bus
324     ;READ FILE = MODE9@TIME_PERIOD@.TB ;---- M9- other express bus
325     ;READ FILE = MODE10@TIME_PERIOD@.TB ;---- M10- other bus (future)
316     ;READ FILE = ..\MODE1@TIME_PERIOD@.TB ;---- M1- metrobus local
317     ;READ FILE = ..\MODE2@TIME_PERIOD@.TB ;---- M2- metrobus express
318     READ FILE = ..\MODE3@TIME_PERIOD@.TB ;---- M3- metrorail
319     ;READ FILE = ..\MODE4@TIME_PERIOD@.TB ;---- M4- commuter rail
320     READ FILE = ..\MODE5@TIME_PERIOD@.TB ;---- M5- other rail (future)
321     ;READ FILE = ..\MODE6@TIME_PERIOD@.TB ;---- M6- other local bus
322     ;READ FILE = ..\MODE7@TIME_PERIOD@.TB ;---- M7- other express bus
323     ;READ FILE = ..\MODE8@TIME_PERIOD@.TB ;---- M8- other local bus
324     ;READ FILE = ..\MODE9@TIME_PERIOD@.TB ;---- M9- other express bus
325     ;READ FILE = ..\MODE10@TIME_PERIOD@.TB ;---- M10- other bus (future)
326 326
327 327 /* Transit path traces for select i/j pairs */
328     read file = ..\scripts\pathTrace.s
328     READ FILE = ..\..\scripts\pathTrace.s
329 329
330 330 ENDRUN

```

Figure 13 Beginning and end sections of Transit\_Skims\_MR script

```

Scripts/Transit_Assignment_MR.s -46 +46

90 90 ;-----
91 91
92 92 RUN PGM=TRNBUILD
93     NETI = ZONEHWY.NET
94     MATI = @MATIN@
93     NETI = ..\ZONEHWY.NET
94     MATI = ..\@MATIN@
95     maxnode = 160000
96 96
97 97 HWYTIME = @TIME_PERIOD@HTIME

218 218
219 219 ;---- Rail Stations & Links (modes 3 & 4) ----
220 220
221     READ FILE = met_node.tb ;---- Metrorail stations
222     READ FILE = met_link.tb ;---- Metrorail links
223     ;READ FILE = com_node.tb ;---- Commuter Rail stations
224     ;READ FILE = com_link.tb ;---- Commuter Rail links
225     READ FILE = lrt_node.tb ;---- LRT stations
226     READ FILE = lrt_link.tb ;---- LRT links
227     ;READ FILE = new_node.tb ;---- Mode10 Stations
228     ;READ FILE = new_link.tb ;---- Mode10 links
221     READ FILE = ..\met_node.tb ;---- Metrorail stations
222     READ FILE = ..\met_link.tb ;---- Metrorail links
223     ;READ FILE = ..\com_node.tb ;---- Commuter Rail stations
224     ;READ FILE = ..\com_link.tb ;---- Commuter Rail links
225     READ FILE = ..\lrt_node.tb ;---- LRT stations
226     READ FILE = ..\lrt_link.tb ;---- LRT links

269 READ FILE = ..\sidewalk.asc;--- walk network for transfers
270 270
271 271 ;---- Transit Line Cards (modes 1-10) ----
272 272
273     ;READ FILE = MODE1@TIME_PERIOD@.TB ;---- M1- metrobus local
274     ;READ FILE = MODE2@TIME_PERIOD@.TB ;---- M2- metrobus express
275     READ FILE = MODE3@TIME_PERIOD@.TB ;---- M3- metrorail
276     ;READ FILE = MODE4@TIME_PERIOD@.TB ;---- M4- commuter rail
277     READ FILE = MODE5@TIME_PERIOD@.TB ;---- M5- other rail (future)
278     ;READ FILE = MODE6@TIME_PERIOD@.TB ;---- M6- other local bus
279     ;READ FILE = MODE7@TIME_PERIOD@.TB ;---- M7- other express bus
280     ;READ FILE = MODE8@TIME_PERIOD@.TB ;---- M8- other local bus
281     ;READ FILE = MODE9@TIME_PERIOD@.TB ;---- M9- other express bus
282     ;READ FILE = MODE10@TIME_PERIOD@.TB ;---- M10- other bus (future)
273     ;READ FILE = ..\MODE1@TIME_PERIOD@.TB ;---- M1- metrobus local
274     ;READ FILE = ..\MODE2@TIME_PERIOD@.TB ;---- M2- metrobus express
275     READ FILE = ..\MODE3@TIME_PERIOD@.TB ;---- M3- metrorail
276     ;READ FILE = ..\MODE4@TIME_PERIOD@.TB ;---- M4- commuter rail
277     READ FILE = ..\MODE5@TIME_PERIOD@.TB ;---- M5- other rail (future)
278     ;READ FILE = ..\MODE6@TIME_PERIOD@.TB ;---- M6- other local bus
279     ;READ FILE = ..\MODE7@TIME_PERIOD@.TB ;---- M7- other express bus
280     ;READ FILE = ..\MODE8@TIME_PERIOD@.TB ;---- M8- other local bus
281     ;READ FILE = ..\MODE9@TIME_PERIOD@.TB ;---- M9- other express bus
282     ;READ FILE = ..\MODE10@TIME_PERIOD@.TB ;---- M10- other bus (future)
283 283
284 284 ; output files
285 285 @WALK_MODEL@AM@FILEO NODEO = %_iter_%_WKMRAMnode.dbf ; output node file
    
```

Figure 14 Beginning and end sections of Transit\_Assignment\_MR.S



```

Transit_Assignment_LineHaul_Parallel_PT.bat -3 +8
3 3 if exist Transit_Assignment_%2.err del Transit_Assignment_%2.err
4 4 if exist Transit_Assignment_%2.done del Transit_Assignment_%2.done
5 5 @echo Transit_Assignment_%2
6 6 if exist voya*.* del voya*.*
6 6 if not exist %2 mkdir %2
7 7 del .\%2\*. * /f/q
7 8 if exist %_iter_%_Transit_Assgn_%2.RPT del %_iter_%_Transit_Assgn_%2.RPT
8 9
9 10 REM Cluster.exe MWCOG %subnode% start exit
10 start /w voyager.exe ..\Scripts\PT_asgn_%2.s /start -Pvoya -S..\%1
11 start /w voyager.exe ..\Scripts\PT_asgn_%2.s /start -Pvoya -S..\%2
11 12 REM Cluster.exe MWCOG %subnode% close exit
12 13
13 14 if errorLevel 2 goto error
14 14 if exist voya*.prn copy voya*.prn %_iter_%_Transit_Assgn_%2.RPT /y
15 15 if exist .\%2\voya*.prn copy .\%2\voya*.prn %_iter_%_Transit_Assgn_%2.RPT /
16 16 y
16 16 del .\%2\voya*.prn /f/q
17 17 copy .\%2\*. * .\ /y
18 18 del .\%2\*. * /f/q
15 19 goto end
16 20 :error
17 21 echo Error in Transit Assignment %2 > Transit_Assignment_%2.err
18 22 PAUSE
19 23 :end
20 24 echo Finished Transit Assignment %2 > Transit_Assignment_%2.done
25 25 rmdir .\%2 /s/q
    
```

Figure 15 Changes made to *Transit\_Assignment\_LineHaul\_Parallel.bat*

```

Transit_Skim_LineHaul_Parallel_PT.bat -3 +8
5 5 if exist Transit_Skims_%2.done del Transit_Skims_%2.done
6 6 @echo Transit_Skims_%2
7 7
8 8 if exist voya*.* del voya*.*
8 8 if not exist %2 mkdir %2
9 9 del .\%2\*. * /f/q
9 10 if exist %_iter_%_TRANSIT_SKIMS_%2.RPT del %_iter_%_TRANSIT_SKIMS_%2.RPT
10 start /w Voyager.exe ..\scripts\Transit_Skims_PT_%2.s /start /high -Pvoya -S..\%1
11 start /w Voyager.exe ..\scripts\Transit_Skims_PT_%2.s /start /high -Pvoya -S..\%2
11 12 if errorLevel 2 goto error
12 12 if exist voya*.prn copy voya*.prn %_iter_%_TRANSIT_SKIMS_%2.RPT /y
13 13 if exist .\%2\voya*.prn copy .\%2\voya*.prn %_iter_%_TRANSIT_SKIMS_%2.RPT /y
14 14
15 15 del .\%2\voya*.prn /f/q
16 16 copy .\%2\*. * .\ /y
17 17 del .\%2\*. * /f/q
13 18 goto end
14 19 :error
15 20 echo Error in Transit Skim %2 > Transit_Skims_%2.err
    
```

Figure 16 Changes made to *Transit\_Skim\_LineHaul\_Parallel.bat*

### 1.3.6 Update 6: Added an automatic shutdown option at the end of a model run

#### 1.3.6.1 Motivation

Working with COG's Information Technology (IT) staff, TPB staff implemented the migration of workstations and servers to on-demand cloud-based servers, making use of Amazon Web Services (AWS). The migration of travel demand model servers to AWS began in December 2022, and was completed in January 2023. Running the model on an on-demand cloud server is charged based on the time used. On an on-demand cloud server, even if a model run has been completed, the server will continue to incur charges until the server is shut down. It is a hassle for the model user to monitor the model run and shut down the server manually when the model run is complete, especially since model runs often end outside of work hours.

#### 1.3.6.2 Solution

The added shutdown feature is an optional function to shut down the computer automatically after a model run is complete. By default, this command is turned off by including the word REM at the start of line #48, as shown in Figure 17. By removing the word REM in the wrapper batch file, the automatic shutdown batch script (shutdown\_60s.bat), shown in Figure 18, can be called. As explained above, this feature is particularly useful when running the model on an on-demand cloud server that is charged based on the time used.

Table 7 Affected model components due to Update 6<sup>33</sup>

File name	Changes
<i>run_Model_[year].bat</i>	Revised
<i>shutdown_60s.bat</i>	Revised

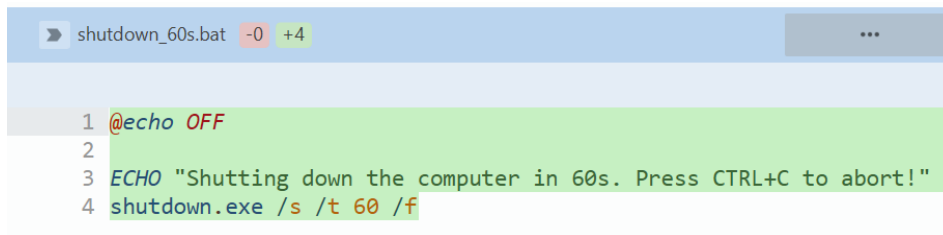
#### Changes made to *run\_Model\_[year].bat*

```

run_Model_2021_Final.bat -0 +2
45 45 cd ..
46 46 move_temp_files_v6.bat %scenar%
47 47
48 REM call shutdown_60s.bat
49
48 50 :: Cleanup
49 51 set root=
50 52 set scenar=
    
```

<sup>33</sup> These batch files are discussed in detail in section 6.1.1 (page 79).

Figure 17 An excerpt from the wrapper batch file



```
shutdown_60s.bat -0 +4 ...  
1 @echo OFF  
2  
3 ECHO "Shutting down the computer in 60s. Press CTRL+C to abort!"  
4 shutdown.exe /s /t 60 /f
```

Figure 18 Batch file to call the automatic shutdown application program

## 2 Overview of the model

The TPB Version 2.4 family of travel models, derived from the Version 2.3 family, 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
- Trip assignment<sup>34</sup>

The first three steps deal with estimating current-year or future-year demand for travel. The last step, trip assignment, where the demand for travel is assigned to a transportation network, represents an equilibration between the transportation demand and the transportation supply. In many models, trip assignment includes only a highway assignment step (“traffic assignment”), where private-use motor vehicles are assigned to a roadway network. In larger urban areas with extensive transit systems, on the other hand, there is often an additional transit assignment component. The TPB Version 2.4.6 Travel Model includes both a highway assignment and a transit assignment step. 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 2.4.6 Travel Model highway assignment is a **macroscopic**, static traffic assignment. This is the standard practice for most 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 through 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 rarely 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 2012 TRB report, “While there is much ongoing research into the use of

---

<sup>34</sup> The Version 2.4 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.

dynamic assignment and traffic simulation procedures, the state of the practice for regional travel models remains static equilibrium assignment.”<sup>35</sup>

## 2.1 Model inputs

The major inputs and outputs of the regional travel demand model are shown in Figure 19. 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.

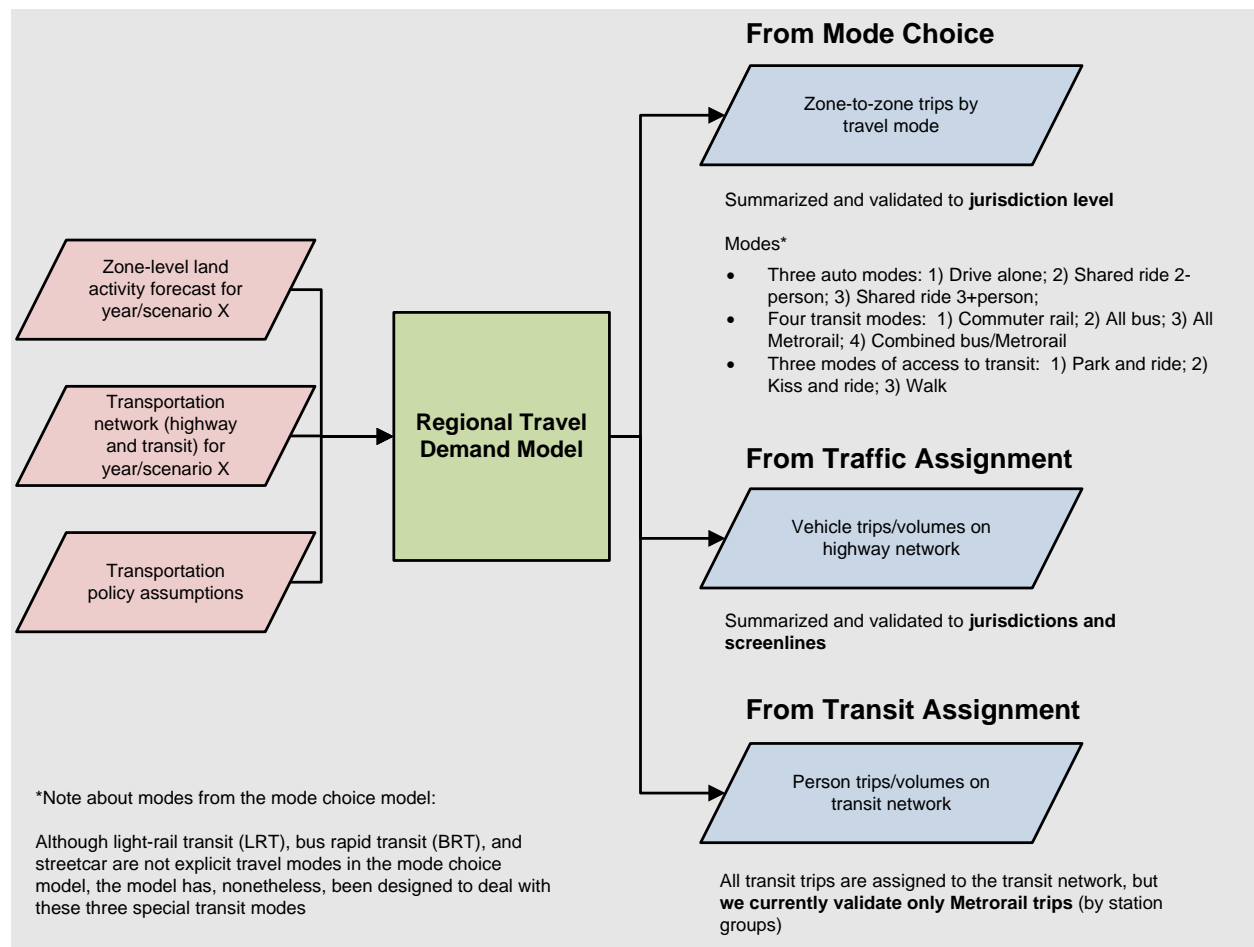


Figure 19 Major inputs and outputs of the TPB Version 2.4.6 Travel Model

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

<sup>35</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>36</sup> COG does not use a formal land use model. In the early 1970s, COG tried using a land use model called EMPIRIC,<sup>37</sup> but COG staff was not satisfied with its performance, and later abandoned its use.<sup>38</sup> According to a recent screening survey of 68 agencies, 37% do not operate a land use model and do not plan to do so in the near future.<sup>39</sup> This is the current case with COG. Instead of a formal land use model, COG uses a process, often known as an "expert panel"<sup>40</sup> or a "modified Delphi process," which involves reconciling top-down and bottom-up land activity forecasts.<sup>41</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." The latest adopted update is Round 9.2, but efforts are underway to develop Round 10.0.

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>42</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

---

<sup>36</sup> "Cooperative Forecasting and Data Subcommittee," Metropolitan Washington Council of Governments, 2021, <https://www.mwcog.org/committees/cooperative-forecasting-and-data-subcommittee/>.

<sup>37</sup> Peat, Marwick, Mitchell and Company, "EMPIRIC Activity Allocation Model: Application to the Washington Metropolitan Region" (Metropolitan Washington Council of Governments, 1972).

<sup>38</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>39</sup> Rolf Moeckel, "NCHRP Synthesis 520: Integrated Transportation and Land Use Models," National Cooperative Highway Research Program, A Synthesis of Highway Practice (Washington, D.C.: Transportation Research Board of the National Academies, 2018), 2, [http://onlinepubs.trb.org/onlinepubs/nchrp/nchrp\\_syn\\_520.pdf](http://onlinepubs.trb.org/onlinepubs/nchrp/nchrp_syn_520.pdf).

<sup>40</sup> Samuel N. Seskin et al., "The Use of Expert Panels in Analyzing Transportation and Land Use Alternatives," Final Report (Portland, Oregon: American Association of State Highway and Transportation Officials (AASHTO), Standing Committee on Planning, April 2002), [http://onlinepubs.trb.org/onlinepubs/nchrp/docs/NCHRP08-36%2804%29\\_FR.pdf](http://onlinepubs.trb.org/onlinepubs/nchrp/docs/NCHRP08-36%2804%29_FR.pdf).

<sup>41</sup> Metropolitan Washington Council of Governments, "Round 9.1 Growth Trends to 2045," Cooperative Forecasting in the Washington Region (Washington, D.C.: Metropolitan Washington Council of Governments, October 2018), <https://www.mwcog.org/documents/2018/10/17/growth-trends-cooperative-forecasting-in-metropolitan-washington-cooperative-forecast-growth--development/>.

<sup>42</sup> Robert E. Griffiths to Ronald Milone, "Travel Model Employment Data Adjustment Factors for Round 7.0," Memorandum, August 10, 2005, 1.

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 is no longer used by BLS.”<sup>43</sup> This year, the most recently developed employment adjustment factors are no longer based on a comparison with CTPP data, so the factors are called “employment definition adjustment factors.” The employment definition adjustment factors were used in Round 9.2, the most recent land activity data.<sup>44</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 2020.<sup>45</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;

---

<sup>43</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>44</sup> Sanghyeon Ko to Mark S. Moran et al., “Developing Land Use Input Files for the Version 2.3 Travel Model Using Round 9.1a Cooperative Forecasts and the Employment Definition Adjustment Factors,” Memorandum, September 17, 2019.

<sup>45</sup> “Highway and Transit Networks Used in the Air Quality Conformity Analysis of the 2020 Amendment to Visualize 2045 and the FY 2021-2024 TIP (Ver. 2.3.78 Travel Model)” (Washington, D.C.: Metropolitan Washington Council of Governments, National Capital Region Transportation Planning Board, April 10, 2020), <https://www.mwcog.org/transportation/data-and-tools/modeling/model-documentation/>.

- 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>46</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”).
- Amount of commuting to and from areas outside the modeled cordon.

## 2.2 Model outputs

The travel model produces many 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 each 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>47</sup> or its update, NCHRP 765<sup>48</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,

---

<sup>46</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.

<sup>47</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>48</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).



- The metropolitan statistical area (MSA), or
- One of the air quality non-attainment areas, which can vary by pollutant.<sup>49</sup>
- Jurisdiction level
- Jurisdiction-to-jurisdiction level
- For highway assignments: Regional screenlines
- For transit assignments: Metrorail station groups

Figure 19 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 8 adds some detail to the information found in Figure 19.

**Table 8 Outputs of the travel model: Disaggregate-level output that is not validated versus aggregate-level output that is validated**

Model producing the output	Disaggregate-level output (Produced by the model, but not recommended for use)	Aggregate-level output (recommended for use)
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 a regional 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 2.4.6 Travel Model, including major inputs and outputs, can be found in Figure 20. 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 20 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.

<sup>49</sup> The modeled area is the largest of these regional areas.

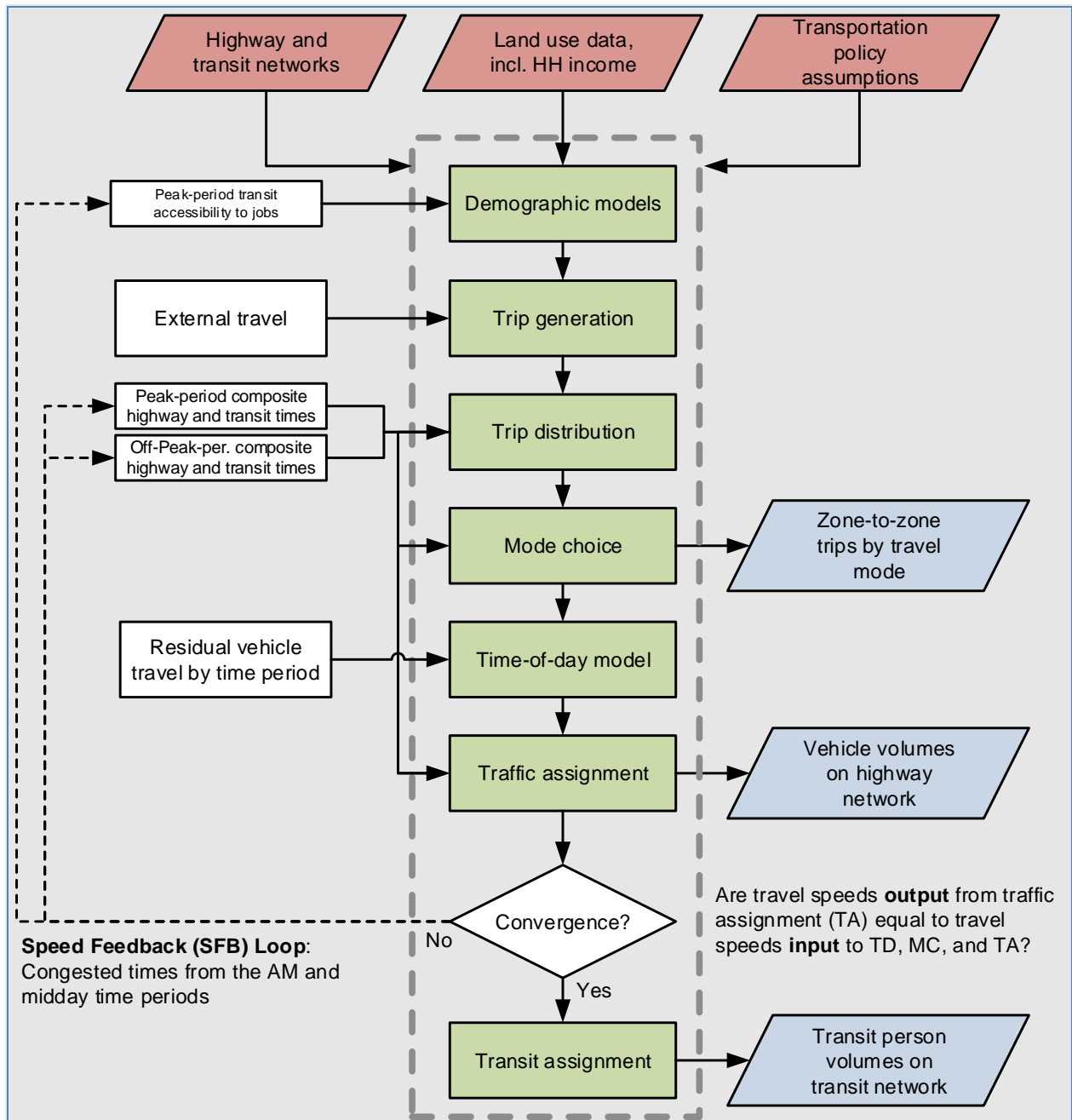


Figure 20 Major steps of the Version 2.4.6 Travel Model

Ref: six\_step\_model\_ver2.3\_v3.vsd

As can be seen in Figure 20, the Version 2.4.6 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>50</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>51</sup> Nonetheless, in 2011, Cambridge Systematics was unable to find any MPOs that used a formal stopping criterion for the SFB loop.<sup>52</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 2.4.6 Travel Model uses a user-equilibrium (UE), static traffic assignment (STA), traffic assignment process, 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 2.4.6 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 2.4.6 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>53</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

---

<sup>50</sup> Ron Milone, “TPB Models Development Status Report,”

<https://www.mwcog.org/file.aspx?&A=%2fTnLbhiKP7J4dc5BCvLqxHQzO%2bq9WHN4K%2bDGCm64j8s%3d>.

<sup>51</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>52</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 to 1–21, <https://www.mwcog.org/transportation/data-and-tools/modeling/review-of-travel-modeling-procedures/>.

<sup>53</sup> Marguerite Frank and Philip Wolfe, “An Algorithm for Quadratic Programming,” *Naval Research Logistics Quarterly* 3, no. 1–2 (1956): 95–110, <https://doi.org/10.1002/nav.3800030109>.

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, in later versions of the Ver. 2.3/Ver. 2.4 Models and also in the latest Ver. 2.4.6 Model, we use 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 9.

Table 9 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>54</sup> we found that the new progressive relative gap scheme resulted 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 2.4.6 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 20) 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 are composed of the following:

- Through trips (auto and truck), also known as external-to-external, or X-X, trips;
- 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).

<sup>54</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.

Figure 21 is another view of the Version 2.4.6 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 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 based on default zone-level mode choice and car occupancy percentages, and these are assigned to the highway network.

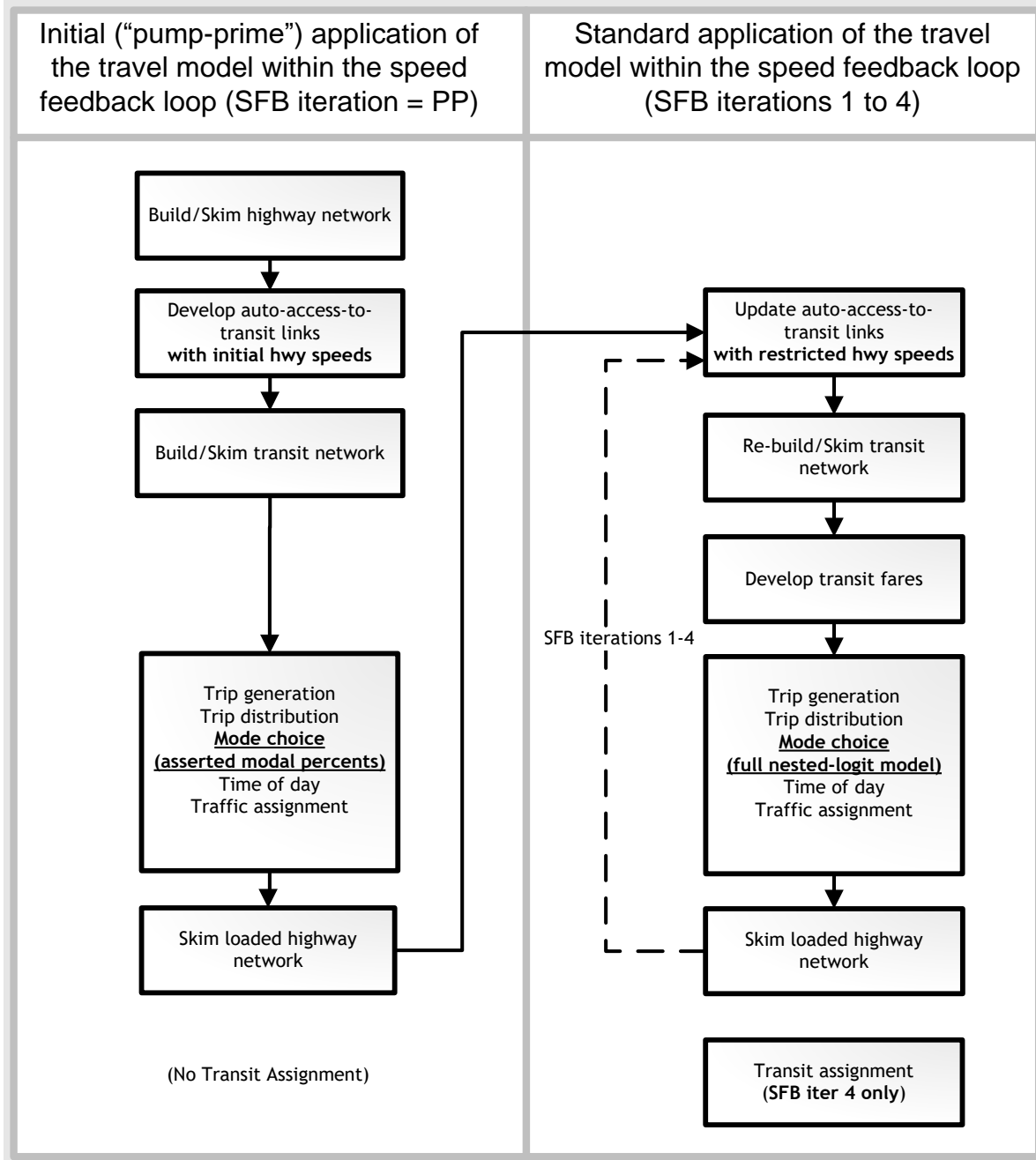


Figure 21 Application process of the Version 2.4.6 Travel Model

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 all the Version 2.2, Version 2.3, and Version 2.4 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 and Version 2.4 models use four speed feedback iterations (in addition to the pump prime iteration). The Version 2.4.6 Model, like the Version 2.3 Model, produces **two final loaded network files** called **i4\_Assign\_Output.net** and **i4\_HWY.NET**. 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 work for the state air quality implementation plans (SIPs).

As shown in both Figure 20 and Figure 21, 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>55</sup>

## 2.4 Special modeling procedures used in earlier versions of the travel model

Historically, there have been two transportation phenomena that have required specialized modeling procedures. The two phenomena were 1) Limited capacity on the Metrorail system to handle the demand for travel to and through the regional core and 2) High Occupancy/Toll (HOT) lanes.

The first phenomenon, a limit on Metrorail's peak-period capacity, was modeled using a procedure called the Metrorail constraint to and through the regional core. This modeling technique was used from about 2001 to 2018. In 2018, however, WMATA received new dedicated funding from the District, Northern Virginia, and suburban Maryland, which meant that the transit authority would likely have the

---

<sup>55</sup> See, for example, Cambridge Systematics, Inc. et al., *NCHRP 716*, 77.

funds to handle its peak volumes to/through the regional core. Thus, in 2018, WMATA requested that this procedure stop being used. The last model to use this procedure was the Ver. 2.3.70 Model.

The second phenomenon, HOT lanes, is still in effect and will be for the foreseeable future, but the technique for modeling it was changed in the Ver. 2.3.75 Model. In the Ver. 2.3.70 Model, and older model versions, a special procedure was used, known as the HOV3+ highway skim replacement (HSR) procedure or the multi-run traffic assignment procedure.

As discussed in Section 1.3 in the previous user's guide,<sup>56</sup> as of the Version 2.3.75 Travel Model, both the Metrorail constraint and the HSR procedure have been eliminated. Nonetheless, the Ver. 2.3.75, Ver.2.3.78, and Ver. 2.4.6 models still include two special modeling procedures, which are not used for general application of the model but can be used when the need arises. One is estimating toll values on HOT lane facilities. The other is performing select-link analyses. Both special procedures are described below.

#### **2.4.1 Toll estimation for 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>57</sup> Most HOT lane facilities exist on freeways which include one or more adjacent general purpose (GP) lanes that are not managed.

##### **2.4.1.1 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, every six minutes or so, 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+.

Given the advantageous treatments in favor of HOV3+ traffic, such as the fact that HOV3+ vehicles are not charged tolls like non-HOV3+ (in both mode choice and traffic assignment), staff decided to eliminate the HSR procedure.<sup>58</sup> 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

---

<sup>56</sup> Ngo et al., "User's Guide for the COG/ TPB Travel Demand Forecasting Model, Version 2.3.75. Volume 1 of 2: Main Report and Appendix A (Flowcharts)."

<sup>57</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).

<sup>58</sup> Feng Xie and Dusan Vuksan to Files, "Evaluating the Modeling Effects of Eliminating the 'HOV Skim Replacement' Process," Memorandum, March 7, 2018.



the HOT facility. To achieve this objective, the following two steps (reduced from four steps in previous model versions when HSR procedure was used) were implemented as of the Ver. 2.3.75 Model (including Ver. 2.4.6) on a year-by-year basis to perform toll setting (i.e., estimate toll values) on HOT lanes.

First, the travel model is fully executed in a Pump Prime (PP) model run. 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. At the end of the Pump Prime model run, a toll searching process will be run to identify HOT lane toll rates (cents/mile) which yield desired Level of Service (LOS) on the facility. The toll rates are developed on a freeway segment by segment basis. At the end of each assignment execution, the segment level volume and its LOS are evaluated, and the toll rate is adjusted incrementally. The toll level is increased when the segment LOS is lower than the desired level. The result of this process is a file containing 'final' toll rates for each HOT lane freeway segment, by time period.

Second, the model is executed again with the final HOT lane toll rates developed in the first step. The result of this execution produces the final loaded links (this step is referred to as the 'final' run).

Note that the toll-setting procedure is conducted by TPB staff to prepare the toll inputs to the Gen2 Model. Thus, many external users of the TPB model never perform this step (since the estimated toll values for future-year networks are provided as part of the model/network transmittal package). For those with an interest in the current toll-setting procedure, please consult the 2018 memo cited here and earlier.<sup>59</sup>

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

From 2017 to 2020, the Virginia Department of Transportation (VDOT) chose to operate the I-66 HOT lanes inside the Beltway as a HOT2+ facility. Before 2017, the only HOT lane facilities in the region were HOT3+. The Air Quality Conformity Analysis of the 2020 Amendment to Visualize 2045 includes six analysis years: 2019, 2021, 2025, 2030, 2040, and 2045. Among those years, only 2019 contained the HOT2+ facility of I-66 HOT lanes inside the Beltway. Since the HSR procedure has been removed, the similar three steps discussed in section 2.4.1.1 were implemented in Ver 2.3.75 for year-2019 to simulate HOT lanes. The only difference is that HOV2+ traffic is free to operate on HOT2+ facility, thus HOV2+ skims developed in Step 2 were used in Step 3.

Please also see the discussion of the treatment of airport passenger auto driver trips on HOV- and HOT-lane facilities in section 23.3.8 (p. 225).

---

<sup>59</sup> Feng Xie and Dusan Vuksan to Files, "Evaluating the Modeling Effects of Eliminating the 'HOV Skim Replacement' Process," Memorandum, March 7, 2018.

### 2.4.1.3 Recent updates to the toll setting process

The core component of the toll setting process is a toll searching algorithm that incrementally raises the tolls on HOT lane facilities until target Volume-to-Capacity (V/C) ratios that would ensure both free-flow conditions and optimal vehicle throughput are achieved. This toll setting procedure was initially developed in the late 2000s by Jinchul Park and has been updated over time. The entire toll-setting process is now integrated with the pump-prime modeling run process. A memorandum dated January 23, 2019 documents this process in detail.<sup>60</sup>

The stopping criteria used in the Gen2/Ver. 2.3 toll-setting process were a V/C threshold range of (0.95-1.01). In 2020, staff recommended new V/C stopping criteria of (0.90-0.95) based on extensive testing, which were then incorporated into the toll setting processes for the Gen2/Ver. 2.4 Model and Ver. 2.4.6 Model. A memorandum dated October 16, 2020 documents the motivation and justification for this update.<sup>61</sup>

### 2.4.2 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>62</sup> but COG/TPB staff often get requests for help with running SLAs. COG/TPB staff has developed some SLA procedures for the regional travel demand forecasting model. The most recent update of these procedures, for the Gen2/Ver. 2.4 Model,<sup>63</sup> can also be used for the Ver. 2.4.6 Model. The SLA procedures may be requested by outside parties by making a request on the Data Request web page (<https://www.mwcog.org/transportation/data-and-tools/modeling/data-requests/>).

---

<sup>60</sup> Feng Xie, Dusan Vuksan, "An Integrated Toll Setting Procedure for COG/TPB's Version 2.3 Travel Demand Model", Memorandum, January 23, 2019.

<sup>61</sup> Anant Choudhary, Dusan Vuksan, "Toll Setting Process Recommendation for the Version 2.4 Travel Demand Model," Memorandum, October 16, 2020.

<sup>62</sup> See, for example, Hillel Bar-Gera and Amos Luzon, "Non-Unique Solutions of User-Equilibrium Assignments and Their Practical Implications (Paper # 07-1335)," in *Compendium of Papers CD-ROM* (Transportation Research Board 86th Annual Meeting, held January 21-25, 2007, Washington, D.C., 2007).

<sup>63</sup> Feng Xie to Files, "Select Link Analysis for TPB's Version 2.4 Travel Demand Model," Memorandum, December 7, 2021.

### 3 Hardware and software requirements

This section of the report describes the hardware and software requirements for running the Version 2.4 family of travel models. 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 2.4.6 Travel Model has been designed to run some steps in parallel using Cube Cluster and/or multiple concurrent command windows.
    - 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 20 on p. 101), 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 Section 8.2.1 for more details. COG/TPB staff has executed the Version 2.4 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 accommodate a large amount of memory (e.g., from 128 GB on Windows 10 Home to 2 TB on Windows 10 Pro). However, based on experience, running the Ver. 2.4 Model is not memory intensive, so 3 to 4 GB of RAM should suffice. Some of our current travel mode servers have 32 GB of RAM, but, again, this does not seem to be required for the current trip-based model.
- 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>64</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 SSDs.

### 3.2 Software

- Operating system: Microsoft Windows (64-bit version), such as Windows 10, Windows Server 2008, Windows Server 2012, or Windows Server 2019. To our knowledge, the Ver. 2.4.6 Model has not been tested at COG using Windows 11, but it should work.
- The Version 2.4.6 Travel Model: The procedures for requesting the travel model and its input files can be found on the “Data Requests” webpage (<https://www.mwcog.org/transportation/data-and-tools/modeling/data-requests/>).
- Bentley Systems Cube software (Bentley Systems acquired Citilabs in 2020): The TPB Version 2.4.6 Travel Model is implemented using Cube software, a proprietary software package, which is produced, licensed, and marketed by Bentley Systems. Thus, to run the regional travel model, you will need to purchase the Cube software from Bentley Systems. COG/TPB staff cannot provide copies of Cube software.
  - 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/Version 2.4 family of travel models have been implemented. Instead, the Version 2.4.6 Model is implemented using a command-line interface (CLI), as described later in this report. **Note that Cube Base is 32-bit software (Cube Voyager is 64 bit).**
  - Cube Voyager: Cube Voyager is the computational engine that powers the Cube suite of software and includes its own proprietary scripting language. The Version 2.4.6 Travel Model has been developed and applied by COG/TPB staff on-premises servers or cloud AWS servers using **Cube version 6.4.1 and 6.5 CE., so it is generally recommended you use 6.5 CE (preferred) OR Cube 6.4.1 with the Ver. 2.4.6 Model.**
  - Cube TRNBUILD: Currently, Cube has two transit modeling software modules: TRNBUILD and Public Transport (PT). The Ver. 2.4.6 Model, like the Ver. 2.3 Model, uses Cube’s TRNBUILD software module for transit modeling (e.g., transit network building, path development, skimming and assignment). For the TPB's next-generation travel model, the Generation-3, or Gen3, Model,<sup>65</sup> on the other hand, the transit modeling module is switched to PT.

---

<sup>64</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, 8.

<sup>65</sup> RSG and Baseline Mobility Group, “Gen3 Model Design Plan” (Metropolitan Washington Council of Governments, July 2, 2020).  
<https://www.mwcog.org/file.aspx?&A=UvEp6mZMXIIwbpwr0%2Bd8dmoshHSSA4wFCs7s8AoszaM%3D>.

- Cube Cluster: Cube Cluster is Bentley Systems' 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 2.4.6 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 both Notepad++ (free and open source) and PSPad (free, but not open source).
- 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.
- Cygwin (optional, <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 no longer part of the model stream, so users no longer need to install this**. Alternatively, Windows now allows one to install Linux on Windows 10.<sup>66</sup>

### 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. Working with COG's Office of Information Technology (IT), TPB staff continued to maintain the software and hardware used to conduct travel demand forecasting and mobile emissions modeling. Hardware includes on-premises travel model servers (TMS) and mobile emissions model workstations (MEMWs) maintained at COG. Software includes Bentley Citilabs Cube, Python, Git/GitHub, R, MOVES, MySQL, MariaDB, and a travel speed post processor, implemented in Cube, for estimation of mobile emissions.

Working with COG's IT, TPB staff implemented the migration of workstations and servers to on-demand AWS. The migration began in December 2022, and was completed in **January 2023**. The on-demand cloud-based AWS servers are referred to as Travel Model Web Servers (TMWS). TPB staff examined the software installed on the TMWS servers and conducted various modeling tests to confirm that the processes that used to be executed using the on-premises computers can now be executed in the cloud.

COG/TPB staff typically access a travel model server (TMS) using a Remote Desktop Connection. We currently have four on-premises travel model servers, named tms5, tms6, tms7, and tms8. Tms8 is the

---

<sup>66</sup> “[Install Windows Subsystem for Linux \(WSL\) on Windows 10](https://docs.microsoft.com/en-us/windows/wsl/install-win10),” September 15, 2020.

most recent travel model server at COG. Tms6 and tms7 are used mainly by COG's Model Applications Group, and tms5 and tms8 are used mainly by COG's Model Development Group.

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. 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 tms7 and tms8 have two processors, tms7 has 10 cores per processor, resulting in a total of 20 physical cores. By contrast, tms8 has 12 cores per processor, resulting in a total of 24 physical cores. Because of Intel's Hyper Threading Technology, each server appears (to the operating system) to have double number of cores. Thus, tms7 appears to the operating system with 40 virtual cores (which can handle 40 threads of instruction) and tms8 appears to the operating system with 48 virtual cores (which can handle 48 threads of instruction). The processor clock speeds for tms7 and tms8 are 3.1 GHz and 3.2 GHz, respectively. Regarding total physical memory, both tms8 and tms7 are equipped with 256 GB of RAM each. Prior to the RAM upgrade, tms8 and tms7 had 64 GB and 32 GB of RAM respectively.

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 2.4.6 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. 2.4.6 Model as well). However, In the past, using Cube 6.1 SP1, we had found that, if two or more users tried to launch concurrent model runs, even if there were only two users, each with one model run, then one of the two model runs would often crash.<sup>67</sup> However, **under Cube 6.4.1 or Cube 6.5, we found that two or three users can submit concurrent model runs.**<sup>68</sup> This is one improvement of newer Cube versions.

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 logical disk drives where the model runs are usually conducted. Each one of these logical disks could be one or more disks, storage arrays, or, conceivably, solid state drives. The previous data drives associated with the on-premises travel model servers at COG prior to the migration to the cloud are shown in Table 10.

---

<sup>67</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.

<sup>68</sup> Dzung Ngo to Mark S. Moran et al., "Testing the COG/TPB Travel Model Servers: 1) Need for Admin Privileges; 2) Ability to Run Two or More Concurrent Model Runs by Two or More Users; 3) Experience with Malware," Memorandum, June 6, 2017, 5.

Table 10 Previously used Computer storage drives for travel demand modeling

Server	UNC Path	Mapped		
		Drive Letter	Size	Drive 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.
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)
tms8	\\tms8\F	Z:	4 TB	RAID 10

Ref: "I:\ateam\docum\fy19\tpb\_tdfm\_gen2\ver2.3\travel\_model\_user\_guide\mapped\_drives\_cog\_2018.xlsx"

For example, tms5 had one data drive with a capacity of 4.5 TB. This drive was mapped as the E drive when logged on to tms5 and was mapped to the X drive when not logged on to the server. By contrast, tms6 had four data drives. The UNC path for each data drive indicated 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 redundancies, 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.

Currently, COG/TPB staff have begun conducting travel model runs in the cloud (i.e., using off-premises servers, such as AWS's m6a.4xlarge instance (AMD 3.6 GHz, 8 cores, 64 GB RAM)). Table 11 shows the current equivalency of DTP drive mappings after migrating travel demand files from on-premises servers to the cloud AWS servers.

Table 11 Current drive mappings for travel demand modeling

Before	After
<a href="#">L: (\\tms6\ateam)</a>	<a href="#">L: (\\mwkog.org\dfs\DTPA\mod_dev\tms6\ateam)</a>
<a href="#">M: (\\tms7\Data)</a>	<a href="#">M: (\\mwkog.org\dfs\DTPA\mod_app\tms7)</a>
<a href="#">N: (\\nas\TMSARCHIVE\MODELAPP)</a>	<a href="#">N: (\\nas\TMSARCHIVE\MODELAPP)</a>
<a href="#">O: (\\nas\TMSARCHIVE\MODELDEV)</a>	<a href="#">O: (\\nas\TMSARCHIVE\MODELDEV)</a>
<a href="#">P: (\\tms6\bteam)</a>	<a href="#">P: (\\mwkog.org\dfs\DTPA\mod_app\tms6\bteam)</a>
<a href="#">T: (\\tms6\ateamarray)</a>	<a href="#">T: (\\mwkog.org\dfs\DTPA\mod_dev\tms6\ateamarray)</a>
<a href="#">V: (\\tms6\bteamarray)</a>	<a href="#">V: (\\mwkog.org\dfs\DTPA\mod_app\tms6\bteamarray)</a>
<a href="#">X: (\\tms5\E)</a>	<a href="#">X: (\\mwkog.org\dfs\DTPA\mod_dev\tms5)</a>
<a href="#">Z: (\\tms8\F)</a>	<a href="#">Z: (\\mwkog.org\dfs\DTPA\mod_dev\tms8)</a>





## 4 Mechanics of the model application process

The Version 2.4 family of travel models, like Ver. 2.3 family, 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 2.4.6 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 2.4.6 Model is shown in Figure 22. 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. Gen2/Ver2.4.6) and the modeling project being undertaken.** For example, an analyst performing model runs to support the Air Quality Conformity (AQC) analysis of the 2022 Update to Visualize 2045, the TPB’s Long-Range Transportation Plan (LRTP) might name the root subdirectory as follows:

C:\mode1Runs\fy23\Ver2.4.6\_aqc\_Update\_Vis2045

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 “fy23” subdirectory. On the left side of Figure 22, there are five specially designated subdirectories under the root that are established:

- SOFTWARE: C++, Fortran executable file and dynamic-link library (DLL) file
- CONTROLS: Control files that are required by the C++ programs
- SCRIPTS: Cube Voyager scripts
- SUPPORT: General parameter files used by the scripts or other programs, such as ModeChoice (C++) 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.

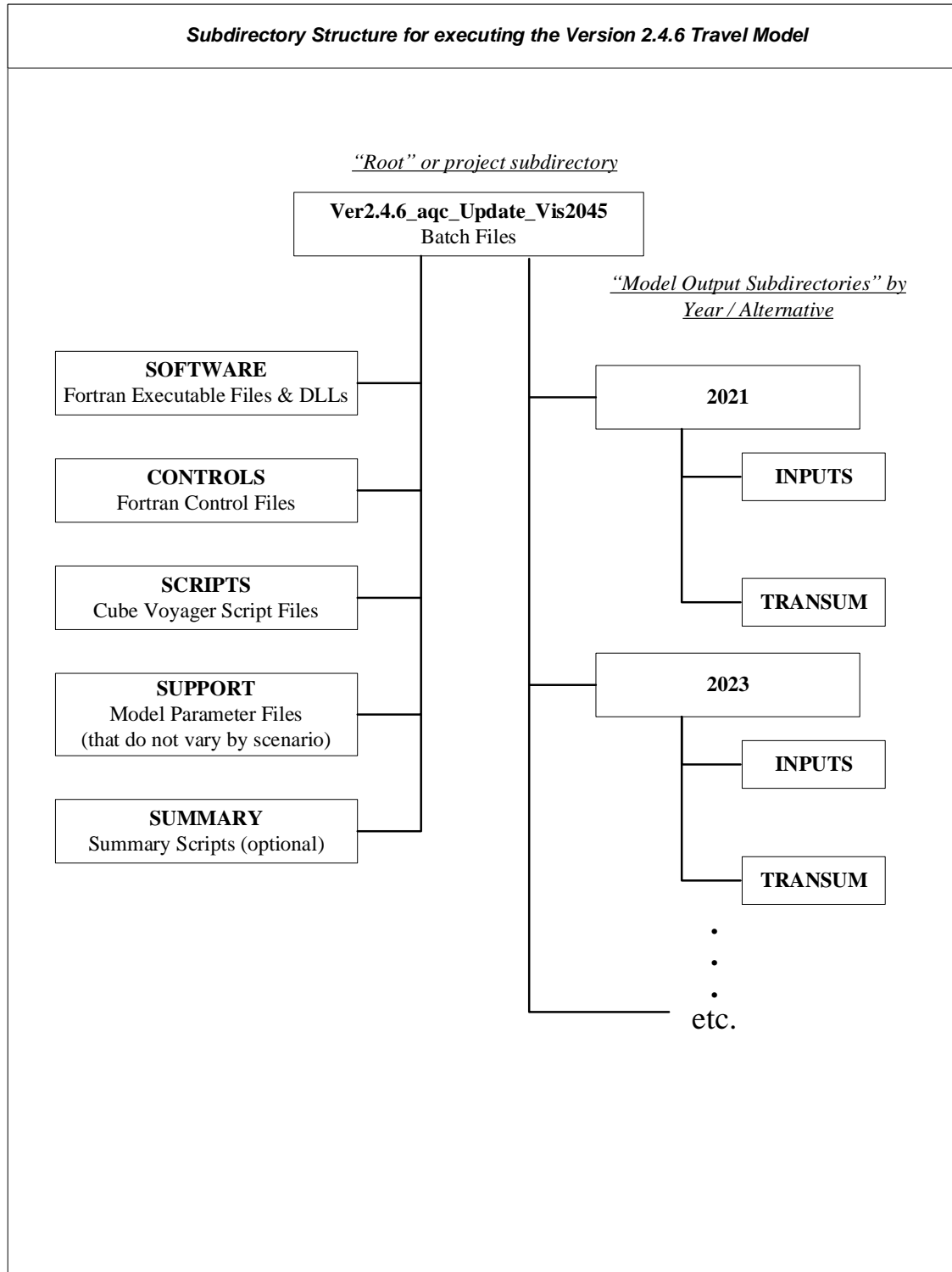


Figure 22 Subdirectory structure for executing the Version 2.4.6 Travel Model

Ref: " I:\ateam\docum\fy23\Ver.2.4.6\_User'sGuide\directoryStruct\_v2.4.6.vsd"

The right side of Figure 22 shows two subdirectories, named “2021” and “2023.”<sup>69</sup> 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, and, 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 summarize the transit assignment results. The “inputs” subdirectory is where one stores all necessary model inputs that are specific to a modeled scenario (see Table 12). Note that some “inputs” that are common to all modeled scenarios are stored in the “support” subdirectory (see Table 12, 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, then the “inputs” folder will contain a new subfolder called “Transit\_Walksheds\_GIS.” In the default setup found with the model transmittal package, the automated transit walkshed process is turned off (**RUN\_WALKACCESS=False**) in the wrapper batch file. Note that the primary output file from the automated transit walkshed process (areawalk.txt) is supplied in the inputs folder as part of the model transmittal package, so users need not turn on and re-run the transit walkshed process unless users make changes to the transit network inputs, in which case it is recommended that the transit walkshed process be run.

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, but the list below is representative:

```
+---2021
|   +---Inputs
|   \---transum
+---2023
|   +---Inputs
|   \---transum
+---Controls
+---Docs
+---Scripts
```

---

<sup>69</sup> As of the Ver. 2.3.75 Model, we no longer require two runs of the travel model (“base” and “final”) for each scenario modeled. However, some modelers may continue to use the name “final” (e.g., 2021\_final), even though there is only one run per scenario.

+---Software  
+---Summary  
+---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 9.1a), WMATA tariff number, etc.
- Key modeling output parameters, such as model run time, regional VMT, total transit, etc.

Table 12 Input files needed to run the Version 2.4.6 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_OP_ModeChoice.ctl	HBO nested-logit mode choice mode control file	Text	Model	Mode choice (MC_Purp.bat & MODECHOICE.EXE)	
2	controls	HBS_OP_ModeChoice.ctl	HBS nested-logit mode choice mode control file	Text	Model	Mode choice (MC_Purp.bat & MODECHOICE.EXE)	
3	controls	HBW_OP_ModeChoice.ctl	HBW nested-logit mode choice mode control file	Text	Model	Mode choice (MC_Purp.bat & MODECHOICE.EXE)	
4	controls	lineSum_MR_access.ctl	Summary of Metrorail riders by access mode	Text	Summary	LineSum.exe	
5	controls	lineSum_MR_line.ctl	Summary of Metrorail boardings, alightings, and ridership	Text	Summary	LineSum.exe	
6	controls	LineSum_Volume.ctl	Consolidate peak & off-peak vols from transit assignment	Text	Summary	LineSum.exe	
7	controls	NHO_OP_ModeChoice.ctl	NHO nested-logit mode choice mode control file	Text	Model	Mode choice (MC_Purp.bat & MODECHOICE.EXE)	
8	controls	NHW_OP_ModeChoice.ctl	NHW nested-logit mode choice mode control file	Text	Model	Mode choice (MC_Purp.bat & MODECHOICE.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	controls	ModeChoice_CommonKeys.ctl	ModeChoice Default Control Keys	Text	Model	Mode choice (MC_Purp.bat & MODECHOICE.EXE)	
11	controls	ZONEV2.A2F.def	Zonal definition file	Text	Model	Mode choice (MC_Purp.bat & MODECHOICE.EXE)	
12	controls	ModeChoice_Script.txt	Customize the mode choice model	Text	Model	Mode choice (MC_Purp.bat & MODECHOICE.EXE)	
13	controls	Segment_Map.csv	contains origin, Destination, and segment fields	CSV	Model	Mode choice (MC_Purp.bat & MODECHOICE.EXE)	
10	controls	ModeChoice_CommonKeys.ctl	ModeChoice Default Control Keys	Text	Assumptions		
14	Controls\Mode	HBW_AM_ModeConstants.csv					
15	Choice	HBS_OP_ModeConstants.csv	Mode specific nesting constants and income	CSV	Assumptions	..\Controls\HBW_AM_ModeChoice.CTL	
16	Choice	HBO_OP_ModeConstants.csv	level adjustment constants for each trip purpose			..\Controls\HBS\HBO\NHW\NHO_OP_ModeChoice.CTL	
17	Constants	NHO_OP_ModeConstants.csv				..\Controls\ModeChoice_Script.txt	
18		NHW_OP_ModeConstants.csv				..\Controls\Segment_Map.csv	
19		HBW_AM_BiasConstants.csv					
20	Controls\Mode	HBS_op_BiasConstants.csv					
21	Choice	HBO_op_BiasConstants.csv	Mode Bias Constants	CSV	Assumptions		
22	Constatnts	NHO_op_BiasConstants.csv					
23		NHW_op_BiasConstants.csv					
1	inputs	airpax.adr	Air Passenger Auto Driver Trips	Binary	Assumptions	Miscellaneous time of day	
2	inputs	AM_Tfac.dbf	AM Toll Factors by Vehicle Type	DBF	Assumptions	Highway skimming and assignment	
3	inputs	areadeF3722.prn	Input TAZ-Mode choice district equivalence	Text	Assumptions	Transit fare development (prefarv23.s)	
4	inputs	AreaWalk.txt	Optional. This file is now generated/re-generated by model	Text	Transit network	Generated by automated ArcPy process	
5	inputs	AT_override.TXT	Cases where zones have area-type override values	Text	Assumptions	AreaType_File.s	
6	inputs	Bus_Factor_File.dbf	Local Bus Time Degradation Factors	DBF	Assumptions	transit_skims_?.s	
7	inputs	bus_pnrn.tb	Bus PNR lots	Text	Transit network	transit_skims_?.s	1,2,6-9
8	inputs	BUSFARAM.ASC	AM Bus Fare matrix (Bus fares zones '1' to '21')	Text	Assumptions	mfare2.s	
9	inputs	BUSFAROP.ASC	OP Bus Fare matrix (Bus fares zones '1' to '21')	Text	Assumptions	mfare2.s	
10	inputs	com_bus.tb	Transfer link (walk) between commuter rail station and bus & LRT stop	Text	Transit network	transit_skims_?.s	12
11	inputs	com_link.tb	Commuter rail links	Text	Transit network	transit_skims_?.s	4
12	inputs	com_node.tb	Commuter rail stations	Text	Transit network	transit_skims_?.s	4
13	inputs	com_pnrn.tb	Commuter rail PNR lots	Text	Transit network	transit_skims_?.s	4
14	inputs	CPI_File.txt	Assumed rate of inflation, based on historical CPI	Text	Assumptions	Set_CPI.s	
15	inputs	Ext_PsAs.dbf	External Productions and Attractions	DBF	Observed data	trip_generation.s	
16	inputs	GIS_Variables.dbf	GIS variables used to calculate non-motorized trips	DBF	Observed data	trip_generation.s	
17	inputs	HBO_NL_MC.MTT	Pre-existing mode choice model output	Binary	Assumptions	pp_auto_drivers.s	
18	inputs	HBS_NL_MC.MTT	Pre-existing mode choice model output	Binary	Assumptions	pp_auto_drivers.s	
19	inputs	HBW_NL_MC.MTT	Pre-existing mode choice model output	Binary	Assumptions	pp_auto_drivers.s	
20	inputs	Jur.dbf	Equiv. between juris and river superdistricts: Disallows river crossings for PNR	DBF	Land use	Autoacc5.s	
21	inputs	Link.dbf	Highway network links	DBF	Highway network	V2.3_Highway_Build.s	
22	inputs	lrt_bus.tb	Transfer link (walk) between LRT station and bus stop	Text	Transit network	transit_skims_?.s	12
23	inputs	lrt_link.tb	LRT links	Text	Transit network	transit_skims_?.s	5
24	inputs	lrt_node.tb	LRT stations/stops	Text	Transit network	transit_skims_?.s	5
25	inputs	lrt_pnrn.tb	LRT PNR lots	Text	Transit network	transit_skims_?.s	5
26	inputs	MD_Tfac.dbf	MD Toll Factors by Vehicle Type	DBF	Assumptions	Highway skimming and assignment	
27	inputs	met_bus.tb	Transfer link (walk) between Metrorail station and bus stop	Text	Transit network	transit_skims_?.s	12
28	inputs	met_link.tb	Metrorail links	Text	Transit network	transit_skims_?.s	3
29	inputs	met_node.tb	Metrorail stations	Text	Transit network	transit_skims_?.s	3
30	inputs	met_pnrn.tb	Metrorail PNR lots	Text	Transit network	transit_skims_?.s	3
31	inputs	metlnkm1.tb	Metrorail Links	Text	Transit network	metrorail_skims.s	
32	inputs	metnodm1.tb	Metrorail Nodes	Text	Transit network	metrorail_skims.s	

User's Guide for the COG/TPB Gen2/Version 2.4.6 Travel Forecasting Model

No.	Folder	Filename	Description	File Type	Category	Modeling step where file is used	Transit/ Non-Transit Mode(s)
33	inputs	mFare1.al	Metrorail Sta XYs scaled to 1/100ths of miles	Text	Transit network	mfare1.s	
34	inputs	mfare1_Sta_Disc.ASC	Metrorail Sta fare discount array in cents	Text	Assumptions	mfare1.s	
35	inputs	MODEL1AM, ... MODEL10AM.tb	AM Transit Line Files	Text	Transit network	transit_skims_??s	
36	inputs	MODEL1OP, ... MODEL10OP.tb	OP Transit Line Files	Text	Transit network	transit_skims_??s	
37	inputs	new_bus.tb	Transfer link (walk) between BRT/streetcar stop and bus stop	Text	Transit network	transit_skims_??s	12
38	inputs	new_link.tb	BRT/streetcar links	Text	Transit network	transit_skims_??s	10
39	inputs	new_node.tb	BRT/streetcar stations/stops	Text	Transit network	transit_skims_??s	10
40	inputs	new_pnrn.tb	BRT/streetcar PNR lots	Text	Transit network	transit_skims_??s	10
41	inputs	NHO_NL_MC.MTT	Pre-existing mode choice model output	Binary	Assumptions	pp_auto_drivers.s	
42	inputs	NHW_NL_MC.MTT	Pre-existing mode choice model output	Binary	Assumptions	pp_auto_drivers.s	
43	inputs	Node.dbf	XY coordinates of nodes in highway network	DBF	Highway network	AreaType_File.s	
44	inputs	NT_Tfac.dbf	NT Toll Factors by Vehicle Type	DBF	Assumptions	Highway skimming and assignment	
45	inputs	Pen.dbf	List of TAZs considered to be in the "slugging" shed of the Pentagon	DBF	Assumptions	Autoacc5.s	
46	inputs	PM_Tfac.dbf	PM Toll Factors by Vehicle Type	DBF	Assumptions	highway_assignment.s	
47	inputs	schl.adr	School Auto Driver Trips	Binary	Assumptions	misc_time-of-day.s	
48	inputs	StaAcc.dbf	Lookup table: Maximum drive-access-to-transit distances	DBF	Assumptions	Autoacc5.s	
49	inputs	station.dbf	Station file: Metrorail, commuter rail, LRT stations/PNR lots and bus PNR lots	DBF	Transit network	parker.s	
50	inputs	tariff.txt	WMATA tariff policy	Text	Assumptions	mfare1.s	
51	inputs	taxi.adr	Taxi Auto Driver Trips	Binary	Assumptions	misc_time-of-day.s	
52	inputs	tazfrzn.asc	Fare Zone File	Text	Assumptions	prefarv23.s	
53	inputs	Toll_Esc.dbf	Toll escalation assumptions: Highway tolls & deflators	DBF	Assumptions	V2.3_Highway_Build.s	
54	inputs	trnpen.dat	Turn Penalty file to ensure correct Metrorail fares	Text	Assumptions	metrorail_skims.s	
55	inputs	visi.adr	Visitor Auto Driver Trips	Binary	Assumptions	misc_time-of-day.s	
56	inputs	xtrawalk.dbf	Extra walk links that the analyst wishes to include	DBF	Transit network	walkacc.s	13
57	inputs	xxaut.vtt	Auto Driver Through Trips	Binary	Assumptions	misc_time-of-day.s	
58	inputs	XXCVT.vtt	Com/Mtk/Htk through Trips	Binary	Calculated data	misc_time-of-day.s	
59	inputs	Zone.dbf	Land use/land activity data at zonal level, 3722 TAZ	DBF	Land use	AreaType_File.s	
1	support	AM_SPD_LKP.txt	Initial lookup speeds used for highway links, AM period	Text	Highway network	V2.3_Highway_Build.s	
2	support	AttrRates.dbf	Trip Attractions	DBF	Calculated data	trip_generation.s	
3	support	cvdelta_3722.trp	Calibration matrix, or "delta table" for commercial vehicles	Binary	Assumptions	misc_time-of-day.s	
4	support	equiv_toll_min_by_inc.s	Equivalent minutes (min/'07\$) by period & income level	Text	Assumptions	trip_distribution.s	
5	support	HBincRat.dbf	HB Income Shares	DBF	Calculated data	trip_generation.s	
6	support	hwy_assign_capSpeedLookup.s	FT x AT Speed & Capacity lookup	Text	Highway network	highway_assignment.s	
7	support	hwy_assign_Conical_VDF.s	Volume Delay Functions file	Text	Highway network	highway_assignment.s	
8	support	MD_SPD_LKP.txt	Initial lookup speeds used for highway links, midday	Text	Highway network	V2.3_Highway_Build.s	
9	support	NMArates.dbf	Non-motorized Trip Attractions	DBF	Calculated data	trip_generation.s	
10	support	NMPrates.dbf	Non-motorized Trip Productions	DBF	Calculated data	trip_generation.s	
11	support	TAZ3722_to_7Mrkts.txt	Equivalency between TAZs and mode choice superdistricts	Text	Assumptions	PP_Auto_Drivers.s	
12	support	tkdelta_3722.trp	Calibration matrix, or "delta table" for med and hvy truck	Binary	Assumptions	misc_time-of-day.s	
13	support	todcomp_2008HTS.dbf	Time of day model/factors	Binary	Assumptions	time-of-day.s	
14	support	toll_minutes.txt	Toll minutes equivalence file by Vehicle Type	Text	Assumptions	Highway_skims.s	
15	support	TPBMod_Jur_Boundary.shp	Jurisdictional boundaries	SHP	Network	Network editing with Cube Base	
16	support	Truck_Com_Trip_Rates.dbf	Truck and Commercial Vehicle Trip Rates	DBF	Calculated data	truck_com_trip_generation.s	
17	support	True_Shape_2045_link2050.shp	Used to display highway network with True Shape	SHP	Highway network	Network editing with Cube Base	
18	support	Ver23_f_factors.dbf	F-factors for trip distribution	DBF	Calculated data	trip_distribution.s	
19	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: v2.4.6\_inputs\_v1.xlsx

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 6.1 (“Parent batch files”) on page 75. The child batch files are the ones that 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 the input files located in the “inputs,” “controls,” and “support” folders are listed in Table 12. 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 the files shown in Table 12 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\_2045\_link2050.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 2021 has the same name as zone.dbf for the year 2045). Thus, the metadata that describes the scenario name is stored in the name of the output subdirectory (e.g., “2021”), not in the filenames themselves.

The SOFTWARE folder contains two C++ executable programs (ModeChoice.exe and Linesum.exe), one Fortran executable program (extrtab.exe.exe), and one dynamic-link library (DLL) file, as shown in Table 13.

**Table 13 Fortran and C++ executable files and dynamic-link library files required for running the Version 2.4.6 Travel Model**

Executable Name	Ver	Size (bytes)	Program Function	Requires a control file?
ModeChoice.exe <sup>70</sup>	7.5.8	1,955,328	Mode choice application program (C++, 64-bit)	yes
VoyagerFileAccess.dll		46,592	Required to run the ModeChoice program	no
extrtab.exe		464,559	Extracts sections from Cube Voyager report files (Fortran, 32-bit)	no
Linesum.exe	6.0.2	697,344	Creates reports summarizing transit loaded link files (C++, 32-bit)	yes

A listing of child batch files is provided in Table 14. 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 15. 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 for executing a model is addressed in the next section. The remaining chapters address the specific details of each modeling step.

**Table 14 Child batch files used to run the Version 2.4.6 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.
Walkshed_Process.bat	MWCOG_Prepate_Inputs_to_Walkshed_Proc ss_PT.s MWCOG_Prepate_Inputs_to_Walkshed_Proc ss_TRNBUILD.s	Run the automated/integrated GeoPandas-based 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.

<sup>70</sup> AECOM, "ModeChoice, Version 7.5.7" (Arlington, Virginia, September 15, 2021), <https://sourceforge.net/projects/transims/files/documentation/TRANSIMS%20Version%207.5%20User%20Guides.7z/download>.



Batch File	Scripts/Programs	Purpose
PP_Highway_Skims.bat (see also Highway_Skims.bat)	Highway_skims_am.s Highway_skims_md.s Modnet.s CheckStationAccess.s Highway_skims_mod_am.s Highway_skims_mod_md.s Joinskims.s Remove_PP_Speed.s	Create AM/off-peak highway skims. Check whether stations are accessible
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.
Mode_Choice_Parallel.bat	MC_purp.bat => ModeChoice.EXE mc_NL_summary.s	Execute the daily mode choice model (in P/A format).
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

Batch File	Scripts/Programs	Purpose
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	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.
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.
Shutdown_60s.bat	None	An optional function to shut down the computer automatically after a model run

Table 15 Sequence of the batch files used to run the Version 2.4.6 Travel Model

Batch File	Scripts/Programs	Speed Feedback Iteration				
		PP	1	2	3	4
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 CheckStationAccess.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	MC_purp.bat => ModeChoice.EXE mc_NL_summary.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	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 Installing software 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).

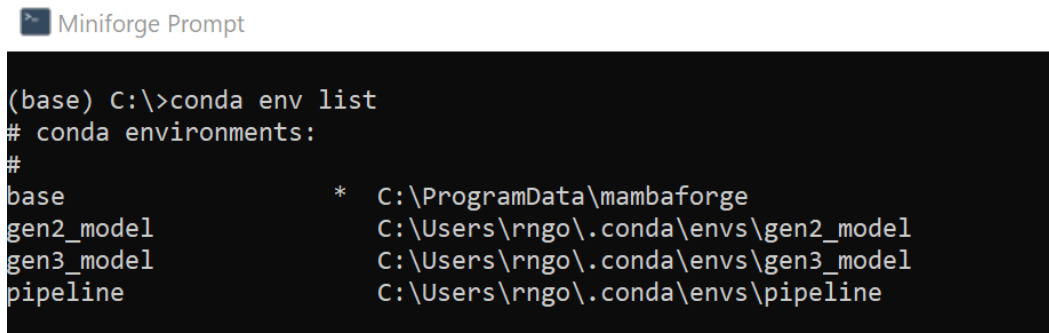
At COG, you will need to be a member of these two groups: "Administrators" and "SophosAdministrator" (the second group is associated with antivirus software). In other agencies, there may be other requirements. Additionally, based on testing done at COG, the mode choice application program (ModeChoice.EXE)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

**Step 2:** To run the transit walkshed process, which is necessary only if the model users alter the transit network inputs, you need to first install Mambaforge and the gen2\_model environment that includes the GeoPandas library.

- a) Install Mambaforge 64-bit from GitHub (<https://github.com/conda-forge/miniforge#mambaforge>). Note that during the installation, the user needs to select the "All Users" (requires admin privileges) option in the "Select Installation Type" window. After the installation, the software should be located on: C:\ProgramData\mambaforge. As part of the Mambaforge installation, a Miniforge command prompt will also be installed.
- b) Install the gen2\_model environment by opening the Miniforge Prompt as a regular user (NOT as administrator) and running the following command in the prompt: `mamba create -n gen2_model geopandas==0.12.2 -c conda-forge`. Press "Enter" after the prompt asks to proceed (with the installation of the downloaded packages).

The path of the gen2\_model environment and the location of Mambaforge will show up when typing the following command on the prompt: `mamba env list`.

The correct installation can be verified by ensuring that Mambaforge is located at "C:\ProgramData\mambaforge" and the gen2\_model environment's path follows this format: "C:\Users\%username%\conda\env\gen2\_model", where "%username%" represents the computer's username. An example is shown in Figure 23.



```
Miniforge Prompt
(base) C:\>conda env list
# conda environments:
#
base                * C:\ProgramData\mambaforge
gen2_model          C:\Users\rngo\.conda\envs\gen2_model
gen3_model          C:\Users\rngo\.conda\envs\gen3_model
pipeline            C:\Users\rngo\.conda\envs\pipeline
```

Figure 23 Example for the correct paths of Mambaforge and the gen2\_model environment

**Step 3:** 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. 2.4.6 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 4:** 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 processors (see Figure 24).

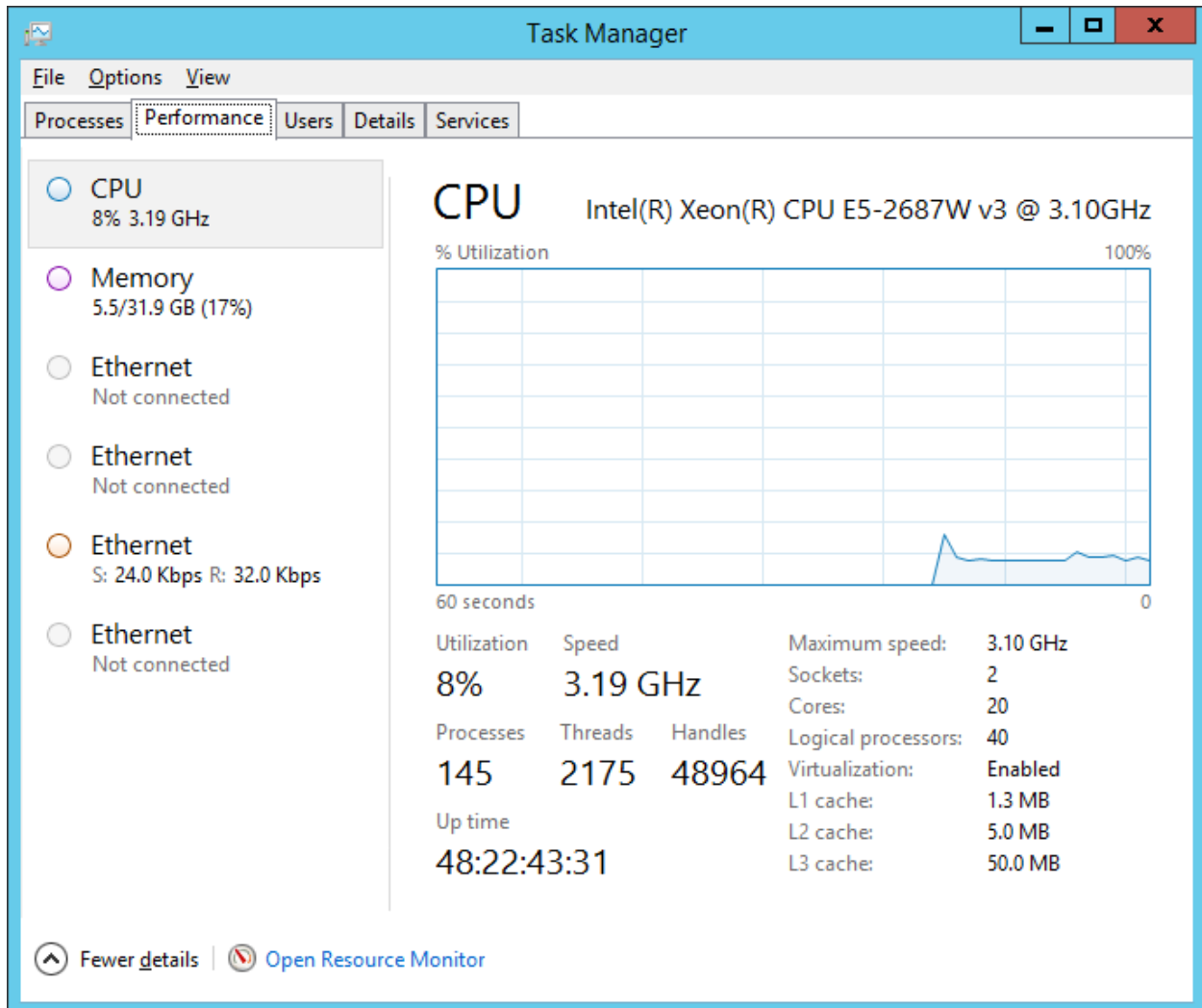


Figure 24 Number of cores and logical processors

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 many logical processors as the actual number of physical cores.

**Step 5:** Install Bentley Systems 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:

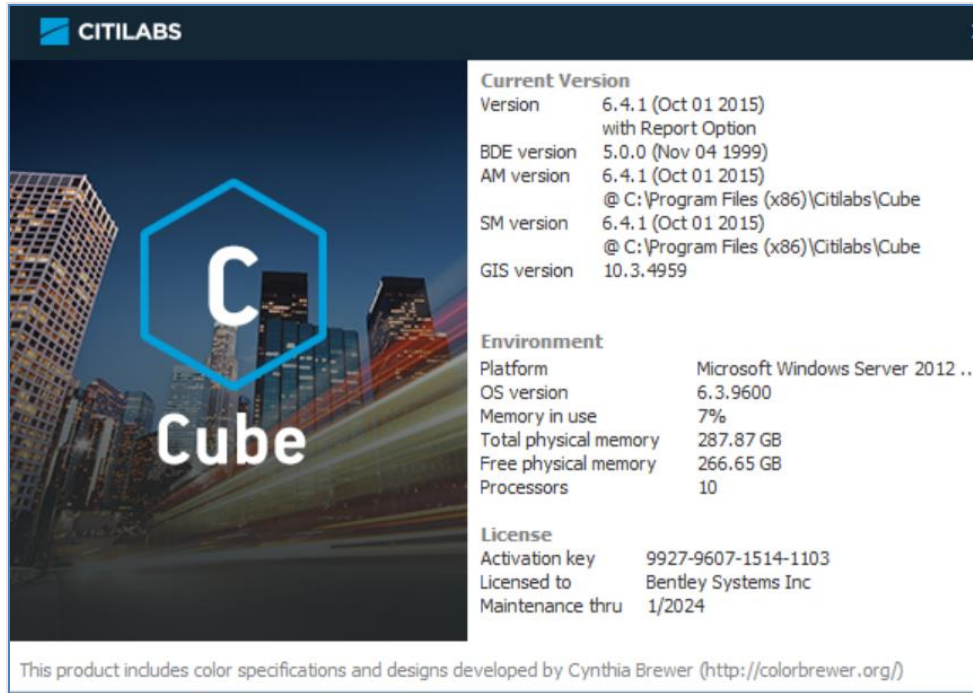


Figure 25 Bentley Systems Cube 6.4, "About" message window

From this window, we can see that we have installed Cube Base, version 6.4.1. The maintenance license expires in Jan. 2024. Cube sees 10 processors, even though this is on a computer with 20 cores and 40 logical processors.



Figure 26 Bentley Systems Cube 6.5, "About" message window



As shown in Figure 26 shows example of Cube Base, version 6.5. The maintenance license expires in July 2023 and it. Cube 6.5, sees 24 processors, even though this is on a computer with 20 cores and 40 logical processors.

If you are running the Ver. 2.4.6 Travel Model on a computer with fewer than 8 logical processors, you will need to follow the instructions in Table 20 (p. 101) before running the Ver. 2.4.6 Travel Model.

**Determining if you have Cube Cluster:** If you wish to determine whether a given computer includes a Cube Cluster license, open 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 27).

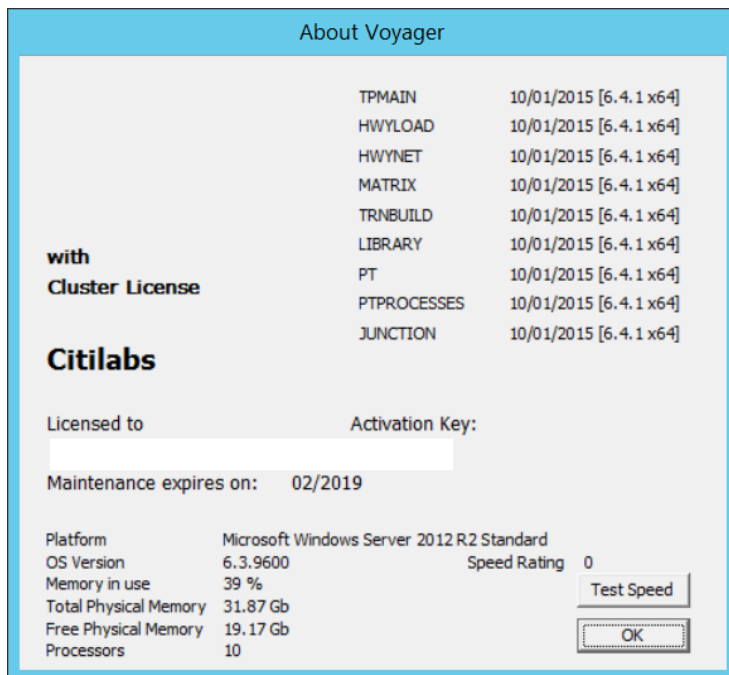


Figure 27 "About Voyager": Presence of Cube Cluster license

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:** Make sure that the version of PowerShell is 3.0 or higher. One may check their PowerShell version by using `$PSVersionTable` command in Windows PowerShell.

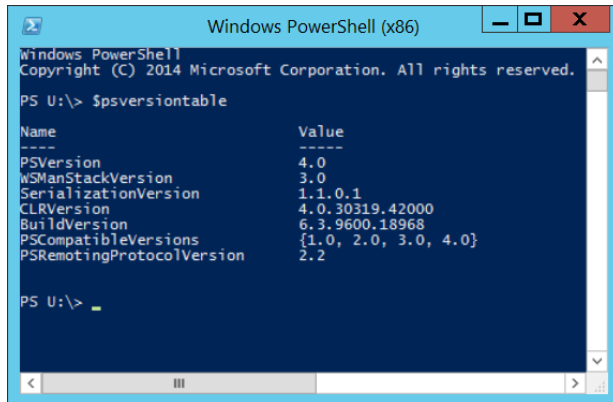


Figure 28 Windows PowerShell version

Most computers will likely be running Windows PowerShell 3.0 (or more recent), which comes preinstalled with Windows operating systems. If the version of PowerShell is 1.0 or 2.0, which may come with older versions of Windows, such as Windows XP or Windows Server 2008 SP1, the framework needs an upgrade to a more recent version. The steps to install a more recent Windows PowerShell version can be found in the Microsoft webpage "<https://docs.microsoft.com/en-us/powershell/scripting/setup/installing-windows-powershell?view=powershell-6>".

**Step 7:** 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.

**Step 8:** (Optional) Install Cygwin (<http://www.cygwin.com/>). **Like Ver. 2.4, the Ver. 2.4.6 Model does not need this software.** Nonetheless, some users may still choose to install this software, due to its ability to offer various Unix-like utility commands. 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. Another more recent, and perhaps more integrated way, to get access to Linux commands on a Windows computer is to install the Windows Subsystem for Linux (WSL).

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, in the past, we have found that, in the 64-bit version of Cygwin, the head and tail commands did not seem to work correctly. **Consequently, if you choose to install Cygwin, 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”.

**Step 9:** (Optional) 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
```

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
```

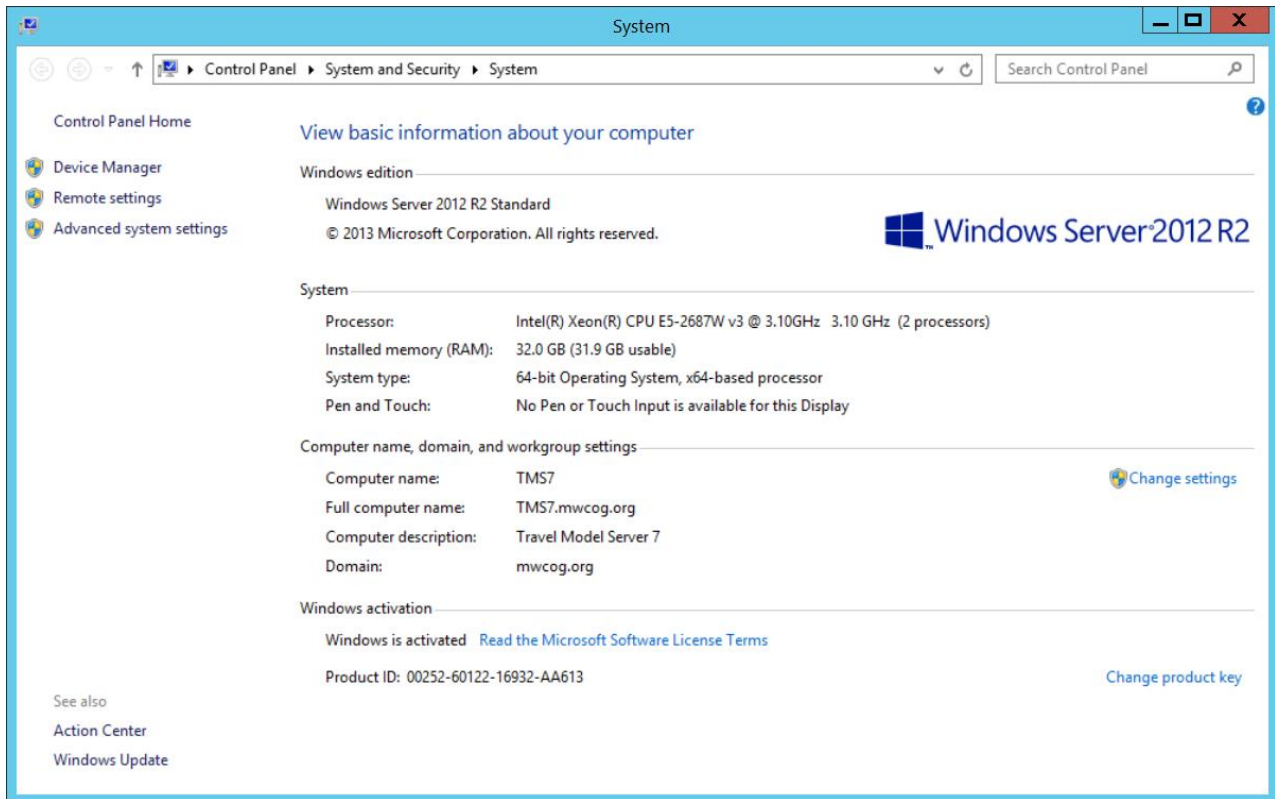
Additionally, the following paths might also be useful additions to your Windows PATH environment variable:

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
C:\cygwin\bin	Needed to run Cygwin from the command line

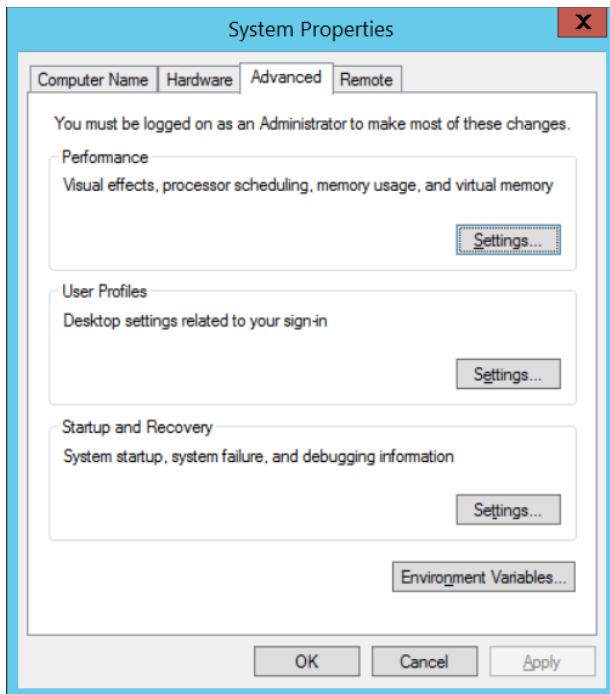
Here are instructions for updating the Windows PATH environment variable:

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

<Windows key><Pause/Break key>

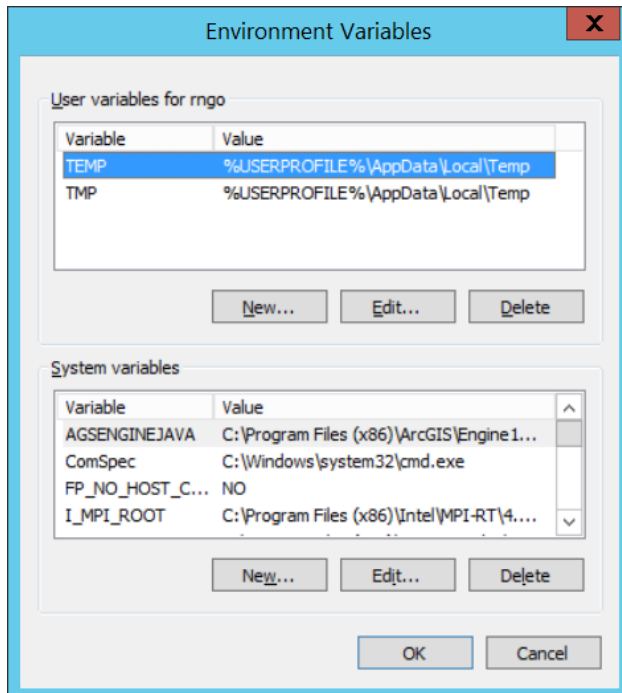


Click "Advanced system settings." Click the "Advanced" tab.

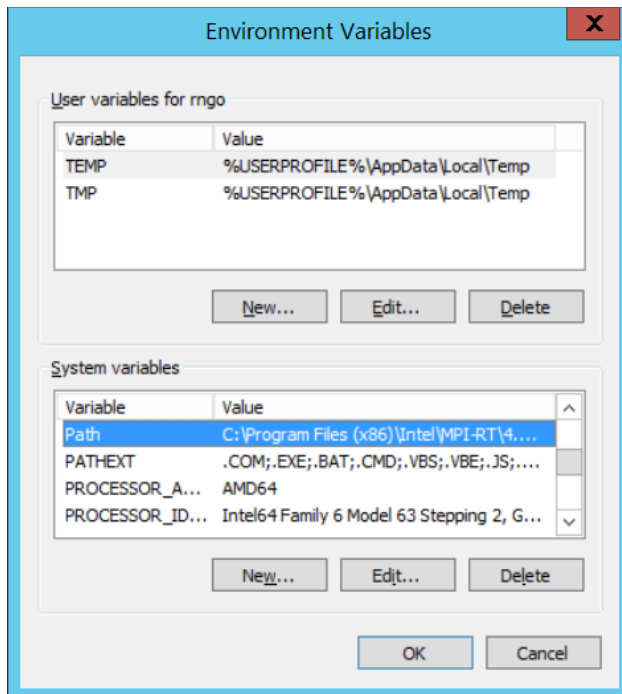


Click the "Environment Variables" button.

The lower half of this window contains "system variables."



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



Click "Edit."

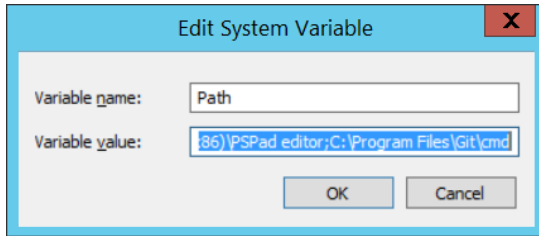
Add the Windows paths that you want. Add these to the end of the Path variable, using a semicolon (";") as the separator.

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 10:** (Optional) Install WinMerge and update the PATH environment variable to include:

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

**Step 11:** (Optional) 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 end 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 TAZ walk areas

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 Ver. 2.4 Travel Model and earlier versions, one of the first steps in the run\_modelSteps batch file was to run the automated transit walkshed process: "call ArcPy\_Walkshed\_Process.bat %1". Due to instabilities with ArcGIS and the ArcGIS runtime engine that is packaged with Cube, the automated transit walkshed process is one of the model steps that is most likely to fail (premature stop or crash). The ArcPy process requires either Cube's built-in ArcGIS Engine Runtime or ArcGIS Desktop, but each Cube version is compatible with only certain ArcGIS Desktop versions. For instance, Cube 6.4.1 is compatible with ArcGIS versions 9.3 through 10.3.1, while Cube 6.5 is compatible with versions 9.3 through 10.8.1. In cases where two incompatible Cube and ArcGIS Desktop versions are installed, the ArcPy-based process will crash. This means that every time we upgrade to a new Cube version, which typically comes with a new ArcGIS Engine Runtime and a different range of compatible ArcGIS versions, we need to make some minor code changes to make the ArcPy process work. But the code change itself does not guarantee a stable performance. Users have reported crashes of the COG/TPB Travel Model when running the process using compatible Cube and ArcGIS versions or using a computer with only Cube and Cube ArcGIS Engine Runtime installed. These crashes were associated with both Cube 6.4.5 and Cube 6.5.<sup>71</sup> The existing transit walkshed process<sup>72</sup> was developed by AECOM in 2014 based on ArcPy. ArcPy is Esri's Python site package used with ArcGIS for geographic data analysis, data conversion, data management, and map automation.<sup>73</sup>

In the Version 2.4.6 model, the process is migrated to GeoPandas,<sup>74</sup> a Python library that combines the capabilities of Pandas and Shapely to work with geospatial data. The GeoPandas-based process retains the Cube/Voyager scripts that prepare the inputs from the transit line files coded in Cube TRNBUILD or Public Transport (PT) format, and all the input files to the process remain the same. Similar to the ArcPy-based process, a working folder called "Transit\_Walksheds\_GIS" is also created with two sub-folders, "input" and "output", in it. When we transmit the model to end users, the automated transit walkshed process is turned off (commented out in the wrapper batch file (run\_Model\_year.bat). This is not a problem for most users since we provide in the inputs folder the primary output file (areawalk.txt) from the automated transit walkshed process. Users do not need to run the transit walkshed process if their

---

<sup>71</sup> Ray Ngo, Feng Xie, and Mark Moran, "User's Guide For the COG/TPB Gen2/Version 2.4 Travel Demand Forecasting Model" (Washington, D.C.: Metropolitan Washington Council of Governments, National Capital Region Transportation Planning Board, March 15, 2021), [https://www.mwcog.org/assets/1/6/mwcog\\_tpb\\_travel\\_model\\_v2.4\\_user\\_guide\\_final.pdf](https://www.mwcog.org/assets/1/6/mwcog_tpb_travel_model_v2.4_user_guide_final.pdf).

<sup>72</sup> David Roden and Krishna Patnam, "Task Order 11 of COG Contract 12-006 – Cube-Based Transit Walk Sheds," Memorandum, March 25, 2014

<sup>73</sup> "What Is ArcPy?—ArcGIS Pro | Documentation," accessed February 7, 2023, <https://pro.arcgis.com/en/pro-app/latest/arcpy/get-started/what-is-arcpy-.htm>.

<sup>74</sup> "GeoPandas 0.12.2," accessed February 7, 2023, <https://geopandas.org/en/stable/>.

model run does not involve a change to the transit inputs, especially the location of transit stops. If, however, a user wishes to make changes to the transit network, then we recommend setting `RUN_WALKACCESS=True` in line #30, of the main/wrapper batch file (`run_Model_[year].bat`), to allow the automated transit walkshed process to run. The new transit walkshed process is discussed in section 11 ("Building transit walksheds") of this report, beginning on p. 115.



## 6 Running the model

As noted in the “Hardware and software ” section, the Version 2.4.6 Travel Model is implemented using Bentley Systems Cube software (in 2020, Citilabs was acquired by Bentley Systems). 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 2.4.6 Travel Model is not launched using Cube Base’s Application Manager. Instead, the Version 2.4.6 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, although DOS no longer exists).

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.

### 6.1 Parent batch files

To run the Version 2.4.6 Travel Model, the user must edit two batch files and then run one of the batch files, which, in turn, will call the other file. These two batch files are known as the parent batch files, 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 29). The second parent batch file is called the “run model steps” (formerly “run all”) batch file (an example can be seen in Figure 30). 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 either directly or within a command prompt window that is pointing to the root directory. For example, if the root directory is “C:\modelRuns\fy23\Ver2.4.6\_aqc\_Vis2045”, 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_Model_2021.bat  
run_ModelSteps_2021.bat
```

```
run_Model_2023.bat  
run_ModelSteps_2023.bat
```

```
run_Model_2025.bat  
run_ModelSteps_2025.bat
```

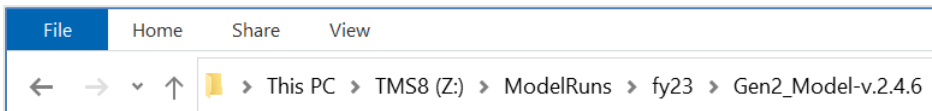
```
run_Model_2030.bat  
run_ModelSteps_2030.bat
```

`run_Model_2040.bat`  
`run_ModelSteps_2040.bat`

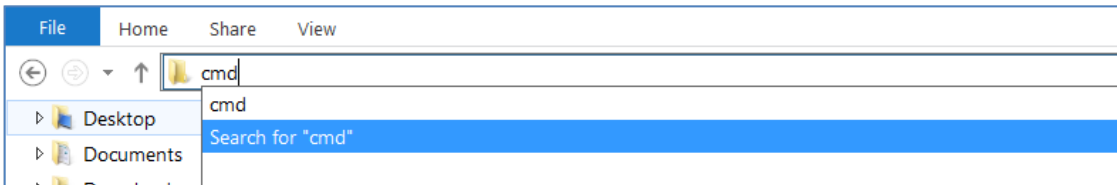
`run_Model_2045.bat`  
`run_ModelSteps_2045.bat`

**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. In earlier versions of Windows, one would select the folder in the left pane, and then, with nothing selected in the right pane, one would use the mouse to **shift-right-click** in the right pane, selecting “Open Command Window Here.” However, in newer versions of Windows, this action results in the option to “Open PowerShell window here.” Since the model is currently not run under Windows PowerShell, one should not select this option. Instead, one can do one of the following:

- Either, open a command window using the Windows Start button, and change the directory to the desired directory by using the change directory (CD) command. One can copy the desired path from the address bar of the file explorer:



- Or, one can put the cursor in the address bar and type “cmd”. This will open a command window whose current path is the path that had been in the address bar of the Windows file explorer:



The main drawback to using the second approach is that after one types cmd in the address box, the address box seems to no longer contain the original path, even though the Windows File Explorer still seems to show this location and the files stored in this location. Thus, after one types “cmd” in the address box, when one clicks in the box a second time, one will see this:

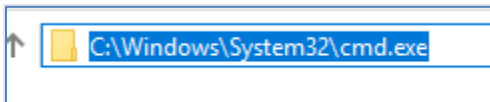


Figure 29 "Run model" batch file for 2021

```

1  :: Version 2.4.6
2
3
4  set root=.
5  set scenar=2021
6  set runbat=run_ModelSteps_2021.bat
7
8  :: STEP 1: Change distributed processing if the computer has less than 8 logical cores (threads)
9  :: Environment variables for (multistep) distributed processing:
10 :: Environment variables for (intrastep) distributed processing:
11 ::   use MDP = t/f (for true or false)
12 ::   use IDP = t/f (for true or false)
13 ::   Number of subnodes: 1-3 => 3 subnodes and one main node = 4 nodes in total
14 set useIdp=t
15 set useMdp=t
16 :: 05/10/2019 AMsubnode, MDsubnode, PMsubnode, NTsubnode are used in highway_assignment_parallel.bat/s
17 set AMsubnode=1-4
18 set MDsubnode=2-4
19 set PMsubnode=2-4
20 set NTsubnode=2-4
21 :: subnode used in transit fare and transit assignment
22 :: We no longer use IDP in transit skimming, since it would require 16 cores
23 set subnode=1-3
24
25 :: STEP 2: Set RUN_WALKACCESS=True in line 28 to include the walkshed process as part of the model run
26 :: Set the Python path by replacing USERNAME in line 30 as your computer's log in account. For example, if USERNAME
27 :: is rngo, change to C:\Users\rngo\.conda\envs\gen2_model\python.exe
28 :: If Mambaforge is installed for your account ONLY on the computer (NOT for all users), the path in line 30 should
29 :: be: C:\Users\%USERNAME%\AppData\Local\mambaforge\envs\gen2_model\python.exe
30 SET RUN_WALKACCESS=True
31 SET python_bindir=C:\Users\%USERNAME%\conda\envs\gen2_model\python.exe
32
33 :: STEP 3: Set AUTO_SHUTDOWN to True to automatically shut down a server at the end of a model run
34 :: This is useful when the model is run on an on-demand cloud server that is charged by hours used.
35 SET AUTO_SHUTDOWN=False
36
37 :: This command will
38 :: 1) time the model run (using timethis.exe and the double quotes)
39 :: 2) redirect standard output and standard error to a file
40 :: 3) Use the tee command so that stderr & stdout are sent both to the file and the screen
41
42 timethis "%runbat% %scenar%" 2>&1 | tee %root%\%scenar%\%scenar%_fulloutput.txt
43
44 :: Open up the file containing the stderr and stdout
45 if exist %root%\%scenar%\%scenar%_fulloutput.txt      start %root%\%scenar%\%scenar%_fulloutput.txt
46
47 :: Look four errors in the reports and output files
48 call searchForErrs.bat %scenar%
49 :: Open up the file containing any errors found
50 if exist %root%\%scenar%\%scenar%_searchForErrs.txt  start %root%\%scenar%\%scenar%_searchForErrs.txt
51
52 :: Open up other report files
53 if exist %root%\%scenar%\i4_Highway_Assignment.rpt  start %root%\%scenar%\i4_Highway_Assignment.rpt
54 if exist %root%\%scenar%\i4_mc_NL_summary.txt       start %root%\%scenar%\i4_mc_NL_summary.txt
55 if exist %root%\%scenar%\i4_Assign_Output.net       start %root%\%scenar%\i4_Assign_Output.net
56 cd %scenar%
57 start powershell.exe -noexit -Command get-content i4_ue*AM_nonHov*.txt -tail 1; get-content i4_ue*AM_hov*.txt -tail 1;
58 get-content i4_ue*PM_nonHov*.txt -tail 1; get-content i4_ue*PM_hov*.txt -tail 1; get-content i4_ue*MD*.txt -tail 1;
59 get-content i4_ue*NT*.txt -tail 1
60 cd ..
61 move_temp_files_v6.bat %scenar%
62
63 IF %AUTO_SHUTDOWN% EQU True call shutdown_60s.bat
64
65 :: Cleanup
66 set root=
67 set scenar=

```

```

68 set runbat=
69 set useIdp=
70 set useMdp=
71 set AMsubnode=
72 set MDsubnode=
73 set subnode=
    
```

Figure 30 "Run model steps" batch file for 2021

```

1  :: Version 2.4.6 TPB Travel Model on 3722 TAZ System
2
3  set _year_=2021
4  set _alt_=Ver2.4_2021
5  :: Maximum number of user equilibrium iterations used in traffic assignment
6  :: User should not need to change this.  Instead, change _relGap_ (below)
7  set _maxUeIter_=1000
8
9  :: Not set transit constraint path and files
10 set _tspath_=
11
12
13
14 :: UE relative gap threshold: Progressive (10^-2 for pp-i2, 10^-3 for i3, & 10^-4 for i4)
15 :: Set the value below
16
17 rem ===== Pump Prime Iteration =====
18
19 set _iter_=pp
20 set _prev_=pp
21 set _relGap_=0.01
22
23 IF %RUN_WALKACCESS% EQU True call Walkshed_Process.bat %1
24 REM call Set_CPI.bat %1
25 REM call PP_Highway_Build.bat %1
26 REM call PP_Highway_Skims.bat %1
27 REM call Transit_Skim_All_Modes_Parallel.bat %1
28 REM call Trip_Generation.bat %1
29 REM call Trip_Distribution.bat %1
30 REM call PP_Auto_Drivers.bat %1
31 REM call Time-of-Day.bat %1
32 REM call Highway_Assignment_Parallel.bat %1
33 REM call Highway_Skims.bat %1
34
35 :: rem ===== Iteration 1 =====
36
37 set _iter_=i1
38 set _prev_=pp
39
40 REM call Transit_Skim_All_Modes_Parallel.bat %1
41 REM call Transit_Fare.bat %1
42 REM call Trip_Generation.bat %1
43 REM call Trip_Distribution.bat %1
44 call Mode_Choice_Parallel.bat %1
45 call Auto_Driver.bat %1
46 call Time-of-Day.bat %1
47 call Highway_Assignment_Parallel.bat %1
48 call Highway_Skims.bat %1
49
50 :: rem ===== Iteration 2 =====
51
52 set _iter_=i2
53 set _prev_=i1
54
55 call Transit_Skim_All_Modes_Parallel.bat %1
56 call Transit_Fare.bat %1
57 call Trip_Generation.bat %1
58 call Trip_Distribution.bat %1
59 call Mode_Choice_Parallel.bat %1
    
```

```

60 | call Auto_Driver.bat          %1
61 | call Time-of-Day.bat        %1
62 | call Highway_Assignment_Parallel.bat %1
63 | call Average_Link_Speeds.bat %1
64 | call Highway_Skims.bat      %1
65 |
66 | :: rem ===== Iteration 3 =====
67 |
68 | set _iter_=i3
69 | set _prev_=i2
70 | set _relGap_=0.001
71 |
72 | call Transit_Skim_All_Modes_Parallel.bat %1
73 | call Transit_Fare.bat          %1
74 | call Trip_Generation.bat       %1
75 | call Trip_Distribution.bat     %1
76 | call Mode_Choice_Parallel.bat  %1
77 | call Auto_Driver.bat          %1
78 | call Time-of-Day.bat          %1
79 | call Highway_Assignment_Parallel.bat %1
80 | call Average_Link_Speeds.bat  %1
81 | call Highway_Skims.bat        %1
82 |
83 | :: rem ===== Iteration 4 =====
84 |
85 | set _iter_=i4
86 | set _prev_=i3
87 | set _relGap_=0.0001
88 |
89 | call Transit_Skim_All_Modes_Parallel.bat %1
90 | call Transit_Fare.bat          %1
91 | call Trip_Generation.bat       %1
92 | call Trip_Distribution.bat     %1
93 | call Mode_Choice_Parallel.bat  %1
94 | call Auto_Driver.bat          %1
95 | call Time-of-Day.bat          %1
96 | call Highway_Assignment_Parallel.bat %1
97 | call Average_Link_Speeds.bat  %1
98 | call Highway_Skims.bat        %1
99 |
100 | :: rem ===== Transit assignment =====
101 | @echo Starting Transit Assignment Step
102 | @date /t & time/t
103 |
104 | call Transit_Assignment_Parallel.bat %1
105 | call TranSum.bat %1
106 |
107 |
108 | @echo End of batch file
109 | @date /t & time/t
110 | :: rem ===== End of batch file =====
111 |
112 | REM cd %1
113 | REM copy *.txt MDP_%useMDP%\*.txt
114 | REM copy *.rpt MDP_%useMDP%\*.rpt
115 | REM copy *.log MDP_%useMDP%\*.log
116 | REM CD..
117 |
118 | set _year_=
119 | set _alt_=
120 | set _iter_=
121 | set _prev_=
122 | set _maxUeIter_=
123 | set _relGap_=

```

124

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

The first line of the “run model” batch file shown in Figure 29 is simply comment. Comments in batch files can be indicated using either a double colon (“::”) or the word REM at the start of the line.<sup>75</sup> In line #4, 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 #5, we define an environment variable called “scenar” (scenario) and set its value to the model scenario/year we want to run (in this case, 2021, but any string may be used, such as “2030\_lowGrowth”). In line #6, 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-2021 model run. Lines 8-23 is where one sets the environment variables that control distributed processing. Distributed processing is covered in more detail later in this report. Lines 25-31, highlighted in green, is where one enables the GeoPandas process as part of a model run by setting RUN\_WALKACCESS=True (line #30). In lines 33-35, highlighted in yellow, a toggle switch that turns on/off the automatic server shutdown is included. As shown in Figure 29 line #35, this command is turned off by default (set to False).

**Line 42 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 42 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 lasts 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 “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 45 simply opens 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 48 calls a batch file that searches reports and output files for certain errors. Line 48 simply opens 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**

---

<sup>75</sup> A single colon (“:”) before a word indicates a label, which is often the target of a GOTO statement.

**model users, the key file to review is the one that combines the standard output and error information (“\_fulloutput.txt”).**

Lines 53 through 55 contain commands which opens other report files, after the model run has completed. Line 57 (which is so long that it stretches over three lines in Figure 29) contains a PowerShell command that opens a window showing some summary convergence metrics for traffic assignment.

**Line 61 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 File 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.

**Lastly, Line 63, highlighted in yellow, calls a batch file (`shutdown_60s.bat`) which will automatically shut down a server** at the end of a model run if the `AUTO_SHUTDOWN` command is set to `TRUE` (line #35). One can set `AUTO_SHUTDOWN` to `True` to automatically shut down a server at the end of a model run, especially when running the model on an on-demand cloud server charged by hours used.

### 6.1.2 Description of the “run model steps” batch files

There is a “run model steps” batch file for each model run scenario/year. Before Ver 2.3.75, these “run model steps” batch files were structured to implement three special modeling procedures:

1. Metrorail constraint to and through the regional core.
2. HOT3+: HOT lanes with free access for HOV3+ (e.g., I-495 and I-95 Express Lanes).
3. HOT2+: HOT lanes with free access for HOV2+ (e.g., I-66 inside the Beltway for 2017-2020).

However, as explained in sections 1.3 and 2.4, starting with the Ver. 2.3.75 model, COG/TPB staff has eliminated the use of the Metrorail constraint to and through the regional core and also the HOV3+ skim substitution technique for modeling HOT lanes. These changes simplify the development of the “run model steps” batch files, since we now use the same batch file structure for all scenarios. The setup now excludes HOV2 and HOV3+ skim replacement and the Metrorail constraint procedures (in the past, the Metrorail constraint procedure required extra attention for modeled years after 2020, which had been the constraint year in the past).

Table 16 shows the key changes in three scenario representatives of Ver. 2.3.70 and Ver 2.4.6 (the same as Ver 2.3.78). For example, the HOV3+ skim replacement procedure, which was invoked in Ver 2.3.70 in the model run representing year-2019 conditions (since HOT lanes existed in that scenario), is not called in the Ver 2.4.6. Similarly, the Metrorail constraint and the HOV3+ skim replacement components are not invoked in the year-2040 model run from Ver 2.3.75. **Thus, a “base” scenario is not needed; only a “final” scenario is now needed to run any modeled year.** The final scenario can be called “2019”. As of Ver 2.3.75, the environment variable “\_tcpath\_” (transit constraint path) is set to blank/null for all scenarios since the Metrorail constraint path is not needed. Also, the HOV3+ skim

substitution/replacement technique is not used in the “run model steps” batch files of all scenarios, this means that the “\_HOV3PATH\_” environment variable is removed (see Figure 30). Although 2020 is not a conformity year in the Amendment to Visualize 2045, Table 16 still shows the differences between 2020 scenario model-step batch files of these two versions.



Table 16 Summary of differences to the “run model steps” batch files for the years 2019, 2020, and 2045 in Ver 2.3.70 and Ver 2.4.6 (or Ver 2.3.75, Ver 2.3.78)

	Year / Scenario model runs		Metrorail constraint through regional core?		HOV2+ skim substitution technique for modeling HOT lanes?		HOV3+ skim substitution technique for modeling HOT lanes?	
	V2.3.70	V2.3.75 V2.3.78 V2.4	V2.3.70	V2.3.75 V2.3.78 V2.4	V2.3.70	V2.3.75 V2.3.78 V2.4	V2.3.70	V2.3.75 V2.3.78 V2.4
2019	2019_base	N/A	Not used (2020 is constraining year)  i.e., “set _tspath_=”	N/A	N/A	N/A	No (Base HOV3+ skims are estimated)  i.e., “set _HOV3PATH_=”	N/A
	2019_final	2019_final	Not used (2020 is constraining year)  i.e., “set _tspath_=”	Not used “set _tspath_=”	N/A	N/A	Yes (Base HOV3+ skims are used from the “base” run)  i.e., “set _HOV3PATH_ =..\2019_base”	Removed
2020	2020_base	N/A	2020 is the year used to set the constraint, but no change is made to batch file  i.e., “set _tspath_=”	N/A	No (Base HOV2&3+ skims are estimated)  i.e., “set _HOV3PATH_=”	N/A	No (Base HOV3+ skims are estimated)  i.e., “set _HOV3PATH_=”	N/A
	2020_final	2020_final	2020 is the year used to set the constraint, but no change is made to batch file  i.e., “set _tspath_=”	Not used “set _tspath_=”	Yes (Base HOV2&3+ skims are used from the “base” run)  i.e., “set _HOV3PATH_ =..\2020_base”	Removed	Yes (Base HOV3+ skims are used from the “base” run)  i.e., “set _HOV3PATH_ =..\2020_base”	Removed
2040	2040_base	N/A	Yes  e.g., “set _tspath_ =..\2020_final”	N/A	N/A	N/A	No (Base HOV3+ skims are estimated)  i.e., “set _HOV3PATH_=”	N/A
	2040_final	2045_final	Yes  e.g., “set _tspath_ =..\2020_final”	Removed “set _tspath_=”	N/A	N/A	Yes (Base HOV3+ skims are used from the “base” run)  i.e., “set _HOV3PATH_ =..\2040_base”	Removed

Regarding the 2040 scenario, whose “run model steps” batch file is shown in Figure 31, Table 16 shows that starting from Ver. 2.3.75, the model no longer use the Metrorail constraint process. Thus, there are four changes to the batch file shown in Figure 31. These changes are highlighted in yellow. First, the “\_tcpath\_” environment variable is no longer set to the location containing the Metrorail trips for the constraint year, 2020 (see line 16 in Figure 31). Although we could have removed this command entirely from the batch file, we have chosen to leave it there, but with a blank argument, in case, in the future, there would be a need to re-apply the Metrorail constraint. Second, line 32 of Figure 31 is highlighted to indicate that *PP\_Highway\_Skims.bat* has been modified to adding a check to ensure that no rail stations are disconnected from the road network. Third, the lines that call the Metrorail constraint mode choice process (“call *Mode\_Choice\_TC\_V23\_Parallel.bat*”) have been changed to apply the mode choice process without constraint (“call *Mode\_Choice\_Parallel.bat*”). These changes have been highlighted in lines 50, 65, 82, and 99 in Figure 31. Fourth, regarding the modeling of HOT lanes, Table 16 shows us that, a “final” scenario is no longer needed to apply the HOV3+ skim substitution/replacement technique, so we no longer need to designate the location of the HOV3+ baseline skims. Thus, the “\_HOV3PATH\_” environment variable is removed and the *Highway\_Skims.bat* is used instead of *HSR\_Highway\_Skims.bat* (see lines 39, 54, 70, 87, and 104 in Figure 31). Line 29, highlighted in green has been updated to include the toggle switch in the wrapper batch file which turns on/off the walkshed process, as shown in Figure 29 line #30.

Figure 31 “Run model steps” batch file for 2040\_final

```

1  :: File location
2  :: Version 2.4.6
3  :: 2/20/2020 4:53:22 PM
4
5  :: Version 2.4 TPB Travel Model on 3722 TAZ System
6
7  set _year_=2040_final
8  set _alt_=Ver2.4_2040
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 :: Not set transit constraint path and files
14 :: Current year no longer used to set the constraint
15
16 set _tcpath_=
17
18
19
20 :: UE relative gap threshold: Progressive (10^-2 for pp-i2, 10^-3 for i3, & 10^-4 for i4)
21 :: Set the value below
22
23 rem ===== Pump Prime Iteration =====
24
25 set _iter_=pp
26 set _prev_=pp
27 set _relGap_=0.01
28
29 IF %RUN_WALKACCESS% EQU True call Walkshed_Process.bat %1
30 call Set_CPI.bat%1
31 call PP_Highway_Build.bat %1
32 call PP_Highway_Skims.bat %1
33 call Transit_Skim_All_Modes_Parallel.bat %1
34 call Trip_Generation.bat %1
35 call Trip_Distribution.bat %1

```

```

36 call PP_Auto_Drivers.bat      %1
37 call Time-of-Day.bat        %1
38 call Highway_Assignment_Parallel.bat  %1
39 call Highway_Skims.bat      %1
40
41 :: rem ===== Iteration 1 =====
42
43 set _iter_=i1
44 set _prev_=pp
45
46 call Transit_Skim_All_Modes_Parallel.bat %1
47 call Transit_Fare.bat          %1
48 call Trip_Generation.bat      %1
49 call Trip_Distribution.bat     %1
50 call Mode_Choice_Parallel.bat  %1
51 call Auto_Driver.bat          %1
52 call Time-of-Day.bat          %1
53 call Highway_Assignment_Parallel.bat  %1
54 call Highway_Skims.bat        %1
55
56 :: rem ===== Iteration 2 =====
57
58 set _iter_=i2
59 set _prev_=i1
60
61 call Transit_Skim_All_Modes_Parallel.bat %1
62 call Transit_Fare.bat          %1
63 call Trip_Generation.bat      %1
64 call Trip_Distribution.bat     %1
65 call Mode_Choice_Parallel.bat  %1
66 call Auto_Driver.bat          %1
67 call Time-of-Day.bat          %1
68 call Highway_Assignment_Parallel.bat  %1
69 call Average_Link_Speeds.bat  %1
70 call Highway_Skims.bat        %1
71
72 :: rem ===== Iteration 3 =====
73
74 set _iter_=i3
75 set _prev_=i2
76 set _relGap_=0.001
77
78 call Transit_Skim_All_Modes_Parallel.bat %1
79 call Transit_Fare.bat          %1
80 call Trip_Generation.bat      %1
81 call Trip_Distribution.bat     %1
82 call Mode_Choice_Parallel.bat  %1
83 call Auto_Driver.bat          %1
84 call Time-of-Day.bat          %1
85 call Highway_Assignment_Parallel.bat  %1
86 call Average_Link_Speeds.bat  %1
87 call Highway_Skims.bat        %1
88
89 :: rem ===== Iteration 4 =====
90
91 set _iter_=i4
92 set _prev_=i3
93 set _relGap_=0.0001
94
95 call Transit_Skim_All_Modes_Parallel.bat %1
96 call Transit_Fare.bat          %1
97 call Trip_Generation.bat      %1
98 call Trip_Distribution.bat     %1
99 call Mode_Choice_Parallel.bat  %1
100 call Auto_Driver.bat          %1
101 call Time-of-Day.bat          %1
102 call Highway_Assignment_Parallel.bat  %1
103 call Average_Link_Speeds.bat  %1

```

```

104 call Highway_Skims.bat %1
105
106 :: rem ===== Transit assignment =====
107 @echo Starting Transit Assignment Step
108 @date /t & time/t
109
110 call Transit_Assignment_Parallel.bat %1
111 call TranSum.bat %1
112
113
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_=

```

All two of the “run model steps” batch files (Figure 30 and Figure 31) apply the progressive relative gap procedure by using the “\_relGap\_” environment variable. For example, the relGap variable starts at a value of 0.01 ( $10^{-2}$ ) on line 27 in Figure 31, and then changes to 0.001 ( $10^{-3}$ ) on lines 76, before attaining the final value of 0.0001 ( $10^{-4}$ ) on line 93 in speed-feedback iteration 4. This is covered in more detail in the text surrounding both Table 9 and Table 86.

## 6.2 Running the model: An example

This section provides an example of how to run the travel model for the year 2021 (named “2021”) 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 specific modeling project (e.g. “Ver2.4.6\_aqc\_Update\_Vis2045”). The root folder contains all the batch files and modeling folders (shown in Figure 22). The folder/subdirectory called “2021” is referred to as the “outputs” folder or the “scenario-specific” folder. It should also be noted that, 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 20 on p. 101.

1. Ensure that the root folder has the two parent batch files:
  - a. *run\_Model\_2021.bat*, which is known as the wrapper batch file or the “run model” batch file.
  - b. *run\_ModelSteps\_2021.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 "2021"
  - c. The "runbat" environment variable should be set equal to "run\_ModelSteps\_2021.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 using the method described in section 6.1 ("Parent batch files"), which begins on page 75).

4. Type "run\_Model\_2021.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 can be provided, upon request, with about a dozen summary scripts, which may be helpful in analyzing the model output. These are listed in Table 17.

Table 17 Travel Model Summary Scripts

Summary script	Description	Folder
COMPARE_NL_MC.S	Compares <b>estimated</b> mode choice results between two different model runs.	summary
COMPARE_NL_MC_Cube61vsCube64.S	Compares <b>estimated</b> mode choice results between two different model runs.	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
Compare_Trip_Distribution.s	Compares <b>estimated</b> trip distribution to <b>observed</b> trip distribution from HTS	summary
Diff_Plots_Rev2.s	Plots volume differences between two input networks	summary
Retrieve_Proc_SubAreas.s	Summarize estimated productions and attractions by purpose and mode.	summary
Screen_Analysis.s	Performs analysis of traffic assignment volumes by screenline	summary
Summarize_2007_2040_Screenlines.s	Compares estimated screenlines volumes in 2007 and 2040	summary
Summarize_Est_Obs_Volume_Daily.s	Compares <b>estimated</b> daily traffic volumes on select links to <b>observed</b> counts. Also compares <b>estimated</b> and <b>observed</b> daily screenline volumes.	summary
Summarize_Est_Obs_Volume_Period.s	Compares <b>estimated</b> AM, MD, PM, and NT volumes on select links to <b>observed</b> counts.	summary
view_from_space.s	Creates global summary of demographic info, trips, and VMT.	summary
RMSE_Calc.s	Creates summaries of link counts and percent root mean squared error between estimated link volumes and observed counts.	assignment_summary
ScreenLine_Summary.s	Merges counts on to a network.	assignment_summary
TVOLDIF_Plot.s	Plots volume differences between two input networks	assignment_summary

Because of the changes to free-flow speeds in the highway speed look-up table in the Ver. 2.4 Model, staff updated the “view\_from\_space.s” script to work with the new model. The updated script will be included in the \Summary subfolder of the Ver. 2.4.6 Model transmittal package. It is worth noting that, **the updated “view\_from\_space.s” script can work with both Ver. 2.3 and Ver. 2.4 Models.**

The summary script Summarize\_Est\_Obs\_Volume\_Daily.s is used to compare estimated daily traffic volumes on select links to observed traffic counts. It also compares estimated and observed daily screenline volumes. Most freeway-class HOV lanes are paired with a set of general-purpose lanes. Similarly, most freeway-class HOT lanes are also paired with a set of general-purpose lanes. In the networks used by the travel model, such managed-lane road segments are typically dual coded, meaning that the road segment is represented with four one-way links (e.g., HOV northbound, LOV northbound, HOV southbound, and LOV southbound). Since this script summarizes estimated daily traffic volumes and observed traffic counts, this script must account for the different ways that the traffic counts are provided by the state DOTs. In some cases, such as the I-95 HOV lanes, the state DOT provides separate counts for the HOV and LOV lanes. In other cases, such as on the Capital Beltway HOT lanes, there are no separate counts – the HOT and LOV counts are grouped together as one set of counts. To process all counts in a consistent manner, the counts on links with separated counts need to be grouped together and placed on the LOV links only. This grouping is done using an LOV/HOV link equivalency file named HOV\_Links.txt, which is referenced in the Summarize\_Est\_Obs\_Volume\_Daily.s script. Thus, to run this script, a model user needs to ensure that the HOV\_Links.txt equivalency file is up to date with the network being summarized.

Figure 32 First five lines of the LOV/HOV link equivalency HOV\_Links.txt

40318	40149	40214	40330	;	<<<	LOVA	LOVB	A	B
40395	40268	40394	40396						
40268	40269	40396	40399						
35182	35183	34881	34883						
34889	35073	34884	34882						

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

## 7.2 Utilities

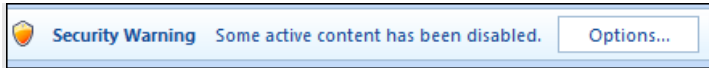
The Version 2.4.6 Travel Model requires many input files in various file formats. One of the file formats is dBase or DBF. Compared to space-delimited text files, DBF files have several advantages (e.g., fields do not mistakenly run together when values become large), but DBF files can also have some 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 several 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.

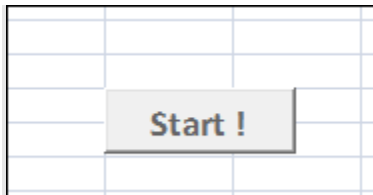
To begin the process, the user double clicks on the converter file/icon (DBF\_Converter\_v3.2.xls), which will open an Excel spreadsheet. In the center of the spreadsheet, there is a "Start" button. Before clicking on this button, the user has to enable the button by clicking on the "Options..." button:



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):

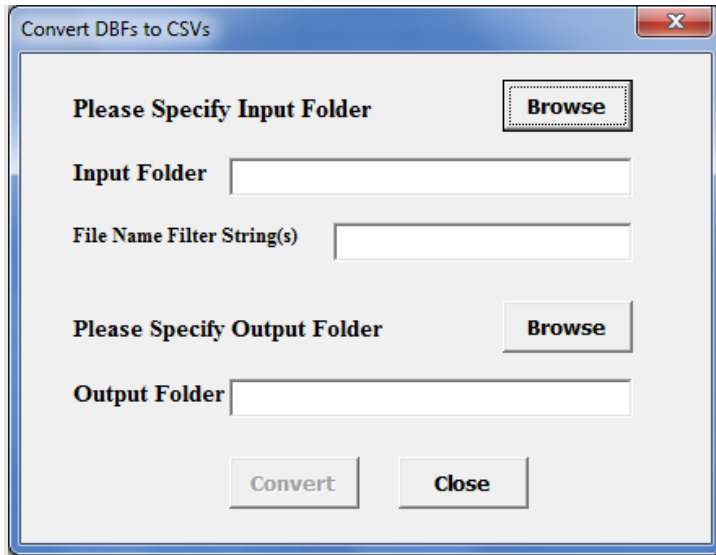
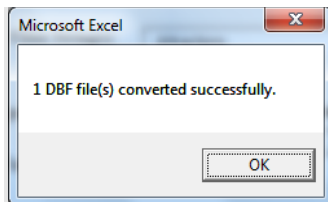


Figure 33 Convert DBFs to CSVs dialog box

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>76</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.



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

---

<sup>76</sup> Note, however, that this can take several minutes, since there are over 100 files.

## 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, Bentley Systems' implementation of distributed processing. On a newer travel model server, such as tms8 or tms7, the model run time is about 12 - 17 hours. Like the Ver. 2.3 Travel Model, the Ver. 2.4.6 Model also requires longer run times than the Ver. 2.2 Model.

There are several reasons why the Ver. 2.3 Travel Model, when it was first developed, had such long run times, compared to its predecessor, the Ver. 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 18. In the Version 2.3 and Version 2.4 Travel Models, 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 and Version 2.4.
- 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>77 78</sup>

---

<sup>77</sup> From a model run representing year-2002 conditions from the air quality conformity determination of the 2009 CLRP/FY 2010-2015 TIP.

<sup>78</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.

- 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. For travel model versions 2.3.52 through 2.4.6, we use a **progressively tightening relative gap** procedure, which is described in more detail later in this chapter.

Table 18 Five traffic assignments in the Version 2.2 Travel Model became six in the Version 2.3 and Version 2.4 Travel Models

Version 2.2 model: Five assignments	Version 2.3 and Version 2.4 models: 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 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, tms8, has a clock speed of 3.2 GHz, has two processors, each with 12 physical cores, which, with Hyper-Threading turned on, appears to the operating system as 48 logical processors (virtual cores).

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. Bentley Systems 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>79</sup>

<sup>79</sup> Citilabs, Inc., "Cube Voyager Reference Guide, Version 6.4.1" (Citilabs, Inc., September 30, 2015), 1124–25.

- “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 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>80</sup>

The Version 2.4.6 Travel Model uses both IDP and MDP and 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 2.4.6 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 2.4.6 Travel Model “out of the box” without making changes to the “run model”/wrapper batch file, since such a computer has eight virtual cores.**<sup>81</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. If one opens the Resource Manager within 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>82</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

---

<sup>80</sup> Citilabs, Inc., 1125.

<sup>81</sup> According to one external user who had a computer with only four cores (though it was not clear whether these were physical cores or virtual cores), the user 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 20 to get the model to run on the four-core computer.

<sup>82</sup> Citilabs, Inc., “Cube Base Reference Guide, Version 6.4.1” (Citilabs, Inc., September 30, 2015), 10–11.

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 computers, you would have a main computer, known as the “main node,” and 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>83</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 several 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. For example, in the Cube 6.4.1 Cube Voyager Reference Guide from 2015: “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>84</sup>

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

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 various 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

---

<sup>83</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>84</sup> Citilabs, Inc., “Cube Voyager Reference Guide, Version 6.4.1,” 1129.

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 introduced MDP to the travel model.<sup>85</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 2.4.6 Travel Models

The Version 2.3.52 Travel Model (and later models, including Ver. 2.4.6) 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*)
- 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:

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

---

<sup>85</sup> For more details, see AECOM and Stump/Hausman Partnership, "FY 2012 Report," chap. 5.

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 19 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. Prior to Version 2.3.78, the tokens used for IDP were AMsubnode and MDsubnode (more on this later in this chapter). As of Version 2.3.78, two more tokens, PMsubnode and NTsubnode, are added. 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 19, 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 (and used in subsequent versions of the model, such as Ver. 2.4.6). This section uses the 2019 "run model" batch file (Figure 29 on page 77) 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 102, describes how parallel processing (specifically IDP and MDP) has been implemented in one script: *Highway\_Assignment\_Parallel.s*. As noted in Table 19 (p. 99), 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 19 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 PMsubnode* NTsubnode*	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)		8
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

\* As of Version 2.3.78

\*\* 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 several user-defined Windows environment variables. Those environment variables that do not deal with distributed processing were discussed in section 6.1.1 (“Description of the “run model”/wrapper batch file”) on page 80. 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 13 and 14 of the “run model”/wrapper batch file (shown in Figure 29), 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>86</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 19 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***. 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>86</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.

Prior to Version 2.3.78, the distributed processing had only two environment variables (AMsubnode and MDsubnode). As discussed in Section 1.3.1, starting from Version 2.3.78, the model includes two more variables, PMsubnode and NTsubnode. Four environment variables dealing with distributed processing in the Version 2.4.6 can be found on in lines 16 to 23 of the “run model”/wrapper batch file (Figure 29):

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

More discussion on these variables and their usage in IDP and MDP of highway skimming and highway assignment is discussed in Section 1.3.1.1.

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 = PMsubnode = NTsubnode= 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>87</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 diagramed in Figure 34 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 2.4.6 Travel Model (Ver. 2.3.52 and later) is designed to run on a computer that has 8 or more cores. Table 20 shows the changes that a user should make in order to run the Version 2.4.6 Model on computers with fewer than 8 cores. See section 8.2.2 (“Effect of Cube Cluster on modeled results”) on p. 96 for a discussion about how modeled results can change slightly with the number of cores used.

**Table 20 Running the Version 2.4.6 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

<sup>87</sup> AECOM and Stump/Hausman Partnership, “FY 2012 Report,” 5–9 to 5–10.

Number of cores in your computer	Changes needed in the "run model"/wrapper batch file	Result
		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> <li>• Change "set PMsubnode=2-4" to "set MDsubnode=2"</li> <li>• Change "set NTsubnode=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 20 is based on testing done by COG/TPB staff using a virtual computer with Cube 6.0.2 installed.

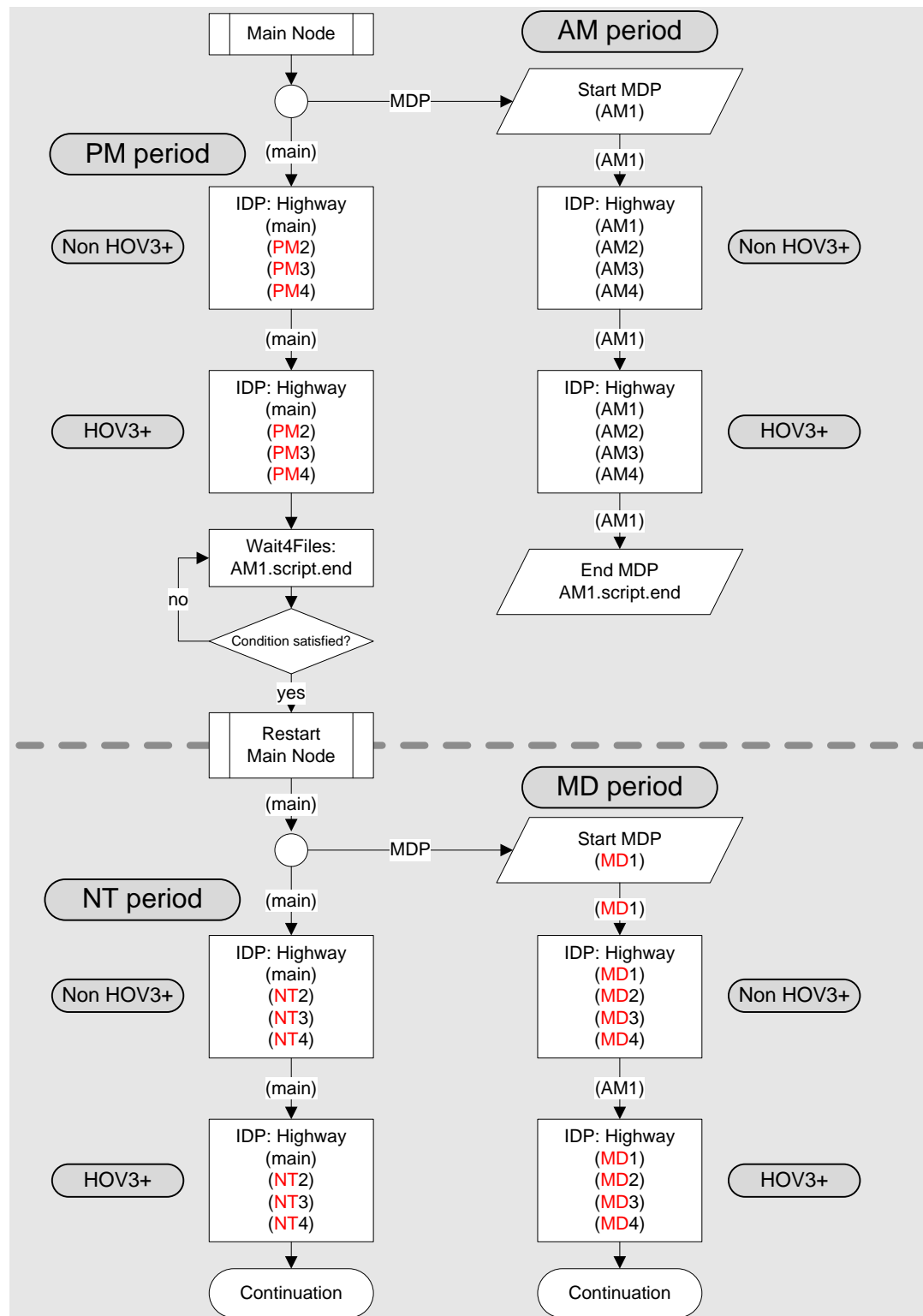
### 8.2.4.3 Parallel processing in the highway assignment script

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 (Ver. 2.3.52 and later) and Version 2.4.6 Travel Model. This section of the report describes how parallel processing (specifically IDP and MDP) have been implemented in one script: *Highway\_Assignment\_Parallel.s*. Figure 34 shows a schematic of how IDP and MDP have been implanted in the highway assignment script of Version 2.4.6, which includes more logical naming conventions (the changed sub-node names are indicated by using red font).

The actual highway assignment script relating to Figure 34 can be found in the "scripts" folder. 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 35, that focuses on the lines that are most relevant to IDP and MDP. Locations where code has been removed are indicated in Figure 35 by a triple ampersand ("&&&"). On line 4 of Figure 35, the statement "distribute intrastep=%useldp% multistep=%useMdp%" either turns IDP and MDP on or off, based on the value of the two tokens. The code in Figure 35 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 simply 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 34, 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 its work. The Wait4Files keywords have been highlighted in blue in Figure 35. So, as indicated in Figure 34, 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 34 Schematic of IDP and MDP in the highway assignment process of the Ver. 2.4.6 Travel Model (Highway\_Assignment\_Parallel.s)



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

Figure 35 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='PM', ProcessList=%PMsubnode%
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='PM', ProcessList=%PMsubnode%
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='MD', 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='MD', ProcessList=%MDsubnode%
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='NT', ProcessList=%NTsubnode%
97 FILEI NETI = @INPNET@ ; TP+ Network
98 &&&
99 ENDRUN
100
101 Wait4Files Files=MD1.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 ENDLLOOP ; 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   ENDLLOOP           ; 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

The four periods being processed are AM, PM, MD, and NT, and these are indicated in Figure 35 by pink/purple highlighting. 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 35), 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 35). As of Version 2.3.78, the mislabeled naming of sub-nodes beginning with the PM period assignment starting from line 45 is now corrected. For example, for the PM non-HOV3+ assignment, the sub-nodes are now named PM2, PM3, and PM4 (line 55 of Figure 35), since %MDsubnode% equals "2-4". The same misleading naming convention is fixed for the sub-node names in the PM HOV3+ assignment: PM2, PM3, and PM4 (line 64 of Figure 35), instead of MD2, MD3, and MD4. A similar issue occurs for the midday assignment (beginning on line 78 of Figure 35) and the nighttime assignment (beginning on line 91 of Figure 35).



## 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 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 36. 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>88</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. 2019\_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 37.

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 was to ensure there is a time delay (ca. 1 hour) between the start of two model runs that share the same root directory.**

---

<sup>88</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 36 An excerpt from an example of the "full output" text file that is created during a Ver.2.4.6 model run

```
1
2 Z:\ModelRuns\fy23\Gen2_Model_v.2.4.6>set _year_=2040
3 Z:\ModelRuns\fy23\Gen2_Model_v.2.4.6>set _alt_=Ver2.4.6_2040
4
5 Z:\ModelRuns\fy23\Gen2_Model_v.2.4.6>set _maxUeIter_=1000
6
7 Z:\ModelRuns\fy23\Gen2_Model_v.2.4.6>set _tcpath_=
8
9 Z:\ModelRuns\fy23\Gen2_Model_v.2.4.6>rem ===== Pump Prime Iteration =====
10 Z:\ModelRuns\fy23\Gen2_Model_v.2.4.6>set _iter_=pp
11
12 Z:\ModelRuns\fy23\Gen2_Model_v.2.4.6>set _prev_=pp
13
14 Z:\ModelRuns\fy23\Gen2_Model_v.2.4.6>set _relGap_=0.01
15
16 Z:\ModelRuns\fy23\Gen2_Model_v.2.4.6>IF True EQU True call Walkshed_Process.bat 2040_Final
17     Using Python from Directory = C:\Users\mseifu\.conda\envs\gen2_model\python.exe
18
19 1) Creating Subdirectories ...
20
21 2) Preparing Inputs ...
22
23     using TRNBUILD line files
24 3) Launching Walkshed Process ...
25
26 Current Path Z:\ModelRuns\fy23\Gen2_Model_v.2.4.6\2040\Inputs\Transit_Walksheds_GIS
27 Current Python C:\Users\mseifu\.conda\envs\gen2_model\python.exe
28
29
30 4) Copying AreaWalk.txt File ...
```

Ref: Z:\ModelRuns\fy23\Gen2\_Model-v.2.4.6\_Mes\2040\_Final\2040\_fulloutput.txt

Figure 37 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 2040_Final\i1_TRANSIT_SKIMS_AB.RPT:W(693): The following LINKO nodes do not have XY values:
13 2040_Final\i1_TRANSIT_SKIMS_AB.RPT:W(693): The following LINKO nodes do not have XY values:
14 2040_Final\i1_TRANSIT_SKIMS_AB.RPT:W(693): The following LINKO nodes do not have XY values:
15 2040_Final\i1_TRANSIT_SKIMS_AB.RPT:W(693): The following LINKO nodes do not have XY values:
16 2040_Final\i1_TRANSIT_SKIMS_AB.RPT:W(693): The following LINKO nodes do not have XY values:
17 2040_Final\i1_TRANSIT_SKIMS_AB.RPT:W(693): The following LINKO nodes do not have XY values:
```

Ref: Z:\ModelRuns\fy23\Gen2\_Model-v.2.4.6\_Mes\2040\_Final\ 2040\_Final\_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 96 for more details.

### 10.2 Model run stops before finishing

We have experienced some cases where a model run will prematurely stop (this is sometimes also referred to as a “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 stoppages, 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 109.

### 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/Bentley Systems, which began an investigation into the matter. However, since the Version 2.4.6 Model (like its predecessors, e.g., 2.3.57 - 2.4) 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 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, 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 on 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: **This no longer appears to be an issue.** In the past, using Cube 6.1 SP1, we had 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 stop prematurely or crash. However, **under Cube 6.5, we found that two or three users can submit concurrent model runs.**<sup>89</sup>
- Whether one runs the walkshed process: This is now turned off by default in the model transmittal package, but it can also be turned on by the user (RUN\_WALKACCESS=True) if the user is making changes to the transit network and wants to recompute the transit walksheds.
- Whether one introduces a time delay (lag time) between model runs: For example, **two model runs can be launched at 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 the runs were started at the same time or that there was some offset between the start times of the model runs.

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. Citilabs alludes to this issue in recent documentation: “However, when comparing two processors from the same family, assuming the processors are otherwise identical, an 8-core processor without Hyper-Threading will outperform a 4-core processor with Hyper-Threading, even though both processors are making 8 threads available to the operating system.”<sup>90</sup>

---

<sup>89</sup> Ngo to Moran et al., “Testing the COG/TPB Travel Model Servers: 1) Need for Admin Privileges; 2) Ability to Run Two or More Concurrent Model Runs by Two or More Users; 3) Experience with Malware,” June 6, 2017, 5.

<sup>90</sup> Citilabs, Inc., “Cube Base Reference Guide, Version 6.4.1,” 10–11.



## 11 Building transit walksheds using Geopandas

This section is largely extracted from a memo documenting the Migration of the Transit Walkshed Process from ArcPy to GeoPandas.<sup>91</sup>

### 11.1 Overview

In 2023, COG staff migrated the process to GeoPandas, a Python library that combines the capabilities of Pandas and Shapely to work with geospatial data. The process develops a series of transit walksheds and calculates the percentage of each zone that is within walking distance to transit stops. The zonal walkshed area file resulting from the process is used as an input to COG/TPB's Gen2 Regional Travel Model. 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. 187 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>92</sup> This second file is then an input to both the transit fare process (*prefarv23.s*) and the mode choice process.

**Note: For the associated Ver. 2.4.6 Model transmittal package, the automated transit walkshed process has been turned off** (in the main/wrapper batch file *run\_Model\_[year] batch files*). This is because if the user is not changing the transit network, there is no need to rerun this process, since we supply the needed areawalk.txt file with each network scenario.

### 11.2 Application Details

The Ver. 2.4.6 Travel Model continues using an automated/integrated transit walkshed process. The process is automated in the sense that it is run using a Python 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 ("IF %RUN\_WALKACCESS%= True call Walkshed\_Process.bat %1"). The transit walkshed process was migrated from ArcPy to GeoPandas process, due to the compatibility and stability issues associated with the ArcPy process, that was developed by AECOM in FY 2014.<sup>93</sup> The ArcPy process requires either Cube's built-in ArcGIS Engine

---

<sup>91</sup> Ray Ngo, "Migration of the Transit Walkshed Process from ArcPy to GeoPandas," Memorandum, March 6, 2023.

<sup>92</sup> See the modeling flowchart in Appendix A.

<sup>93</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

Runtime or ArcGIS Desktop, but each Cube version is compatible with only certain ArcGIS Desktop versions.

The migration to GeoPandas was driven by several key advantages of the GeoPandas library. First, GeoPandas is open-source and independent from Cube's ArcGIS Runtime Engine, making it more accessible to users and less prone to issues caused by ArcGIS Engine Runtime or ArcGIS Desktop. Second, the process that utilizes the GeoPandas library is compatible with any Cube 6 released version, potentially including future Cube releases. As such, GeoPandas was chosen as the replacement for the ArcPy package.<sup>94</sup> To run the new integrated walkshed process, you must install Mambaforge and the "Gen2\_Model" environment which are required to run the GeoPandas walkshed process as described in section 5.1 on page 63. Additionally, users need to enable the GeoPandas process as part of a model run by setting RUN\_WALKACCESS=True in line 26 of the main/wrapper batch file (**run\_Model\_[year].bat**) described in detail in section 6.1.1 on page 80.

Compared with the ArcPy-based method, the GeoPandas-based process has a few changes:<sup>95</sup>

- a. The ArcPy\_Walkshed\_Process.bat file that calls MWCOG\_ArcPy\_Walkshed\_Process.py is replaced by **Walkshed\_Process.bat** that calls **Walkshed\_Process.py** instead. Python (which is installed as part of the gen2\_model environment), instead of ArcGIS or ArcGIS Engine Runtime, will be used for the geoprocessing.
- b. Although the GeoPandas process generates all the intermediate shapefiles, including the buffers files, the ArcGIS/Cube MXD file that visualizes the walkshed buffers is no longer created. However, this is not a problem since the shapefiles under the "Transit\_Walksheds\_GIS" subfolder can be used to create visualization ad hoc using GIS tools such as ArcGIS or QGIS, as shown in Figure 39 on page 118.
- c. Geo-databases are no longer created. Shapefiles are written out separately.
- d. The file "PercentWalk.txt" is no longer generated because it is not used in the model.

The new GeoPandas-based transit walkshed process was tested using the buffering resolution of 32, 64, and 128 for the short (1/2 mile) and long (one mile) buffers around the transit stops. The buffering resolution of 128 in the GeoPandas-based process yields the best match of walk areas with those from the ArcPy-based process. At least 95 percent of the TAZ walk areas are identical, and less than 5 percent of TAZ walk areas have an absolute difference greater than 0.00011 square miles.<sup>96</sup>

---

Transportation Planning Board, Metropolitan Washington Council of Governments, August 18, 2014), <http://www.mwcog.org/uploads/committee-documents/Y11YWFZd20140922110646.pdf>.

<sup>94</sup> Ray Ngo, "Migration of the Transit Walkshed Process from ArcPy to GeoPandas," Memorandum, March 6, 2023.

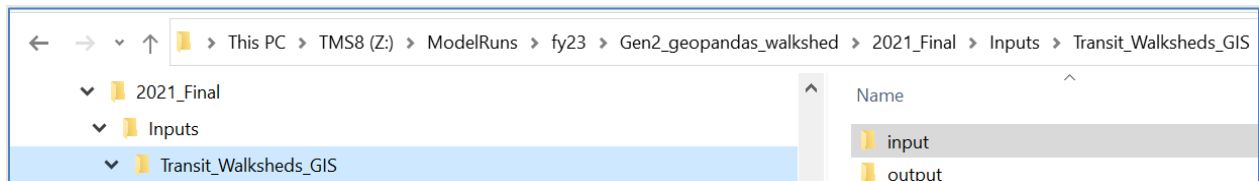
<sup>95</sup> Ibid, Page 2

<sup>96</sup> See page 3-5 Ray Ngo, "Migration of the Transit Walkshed Process from ArcPy to GeoPandas," Memorandum, March 6, 2023.

In the model transmittal package, this line has been turned off (by setting RUN\_WALKACCESS=False) for the reasons explained earlier in this section.

If this automated transit walkshed process is run, this step occurs at the start of the pump-prime (PP) speed feedback loop. Once the walkshed process has been run, a new folder will be created in the “inputs” folder, named “Transit\_Walkshed\_GIS,” as shown in Figure 38. Within this folder, one can find intermediate shapefiles (buff\_allpkshr.shp, buff\_allpkng.shp, \*.shp), which can be used to visualize the walkshed buffers, as shown in Figure 39.

Figure 38 Folder structure for the automated Geopandas 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 2021\_Final). The Geopandas walkshed process creates one file “areawalk.txt” which is used by the travel model. The final output file, AreaWalk.txt, is copied to the main scenario “Inputs” folder and replaces any existing copy.

As mentioned above the GeoPandas process generates all the intermediate shapefiles in the “Transit\_Walksheds\_GIS” subfolder but not the MXD (Map Exchange Document). Users can visualize the output shapefiles using ArcGIS as shown in Figure 39.

Based on testing the new transit walkshed process, staff concluded that the migration from ArcPy-based transit walkshed process to GeoPandas-based process with a buffering resolution of 128 significantly improves the runtime performance, reducing the runtime by a factor of 3, while maintaining the accuracy level of walkshed areas in the areawalk.txt file. Moreover, staff found this migration enhanced the process stability, as the process was run successfully with both old and new Cube versions.

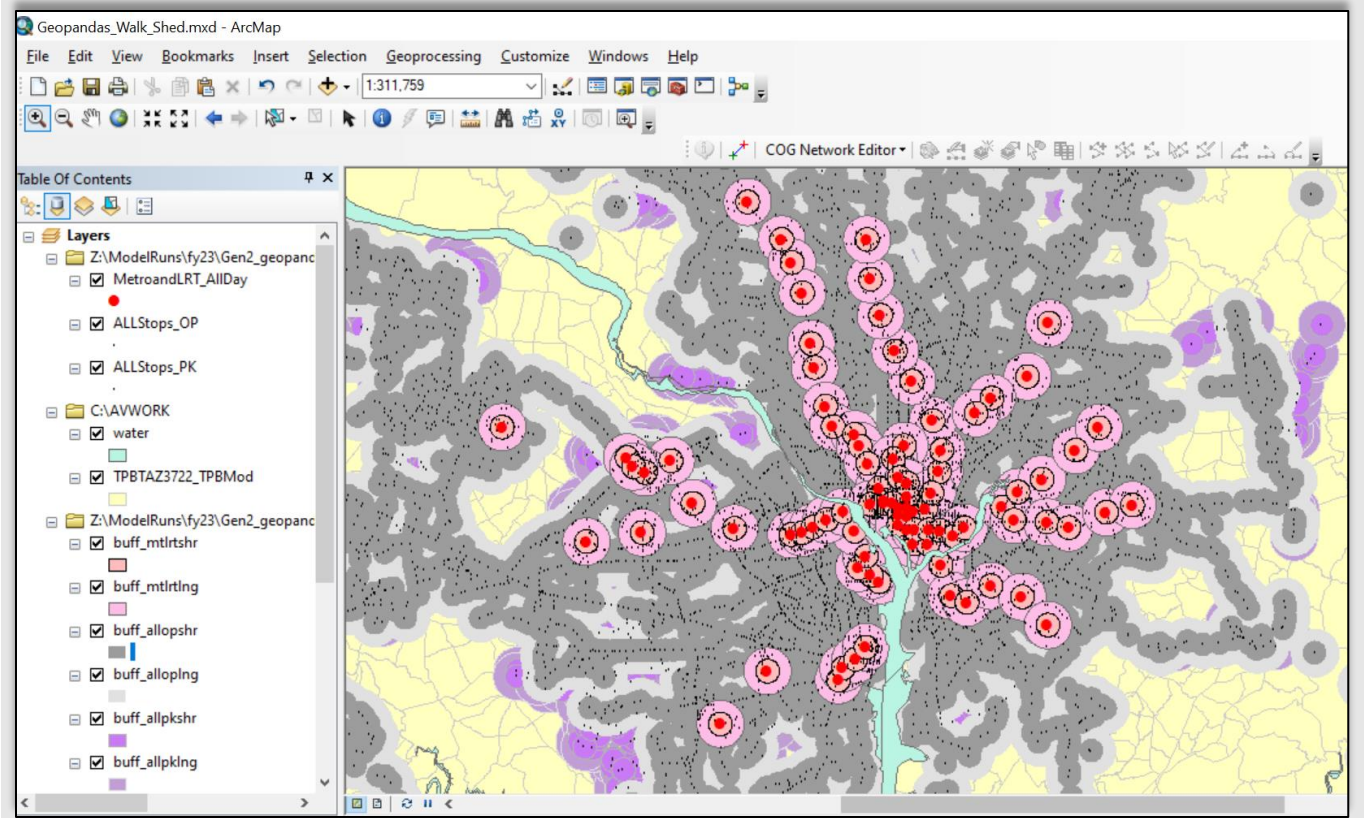


Figure 39 Walkshed buffers for a typical base-year scenario

## 12 Set-Up Programs and Highway Network Building

### 12.1 Overview

Following the generation of transit walksheds, the initial modeling steps of the Version 2.4.6 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 31). 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 21 and detailed in Table 22 through Table 26. The principal outputs are listed in Table 27, and are detailed in Table 28 and Table 29.

**Table 21 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 22 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 = 1.0) 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 23 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 24 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
	AMLIMIT	AM Peak Limit Code (0-9)

File Name	Variable Name	Description
	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	Geometric network link identifier
	LINKID	Logical network link identifier
	NETYEAR	Planning year of network link
	SHAPE_LEN	Geometric length of network link (in feet)
	PROJECTID	Project identifier
	TRANTIME	Unused place marker
	WKTIME	Unused place marker
	MODE	Unused place marker
	SPEED	Unused place marker

Notes:

- The mode choice model requires that all costs be in 2007 dollars, which was the calibration year.
- Link limit codes are shown in Table 25.

Table 25 Link limit codes

Limit Code	Description and Vehicles Allowed
0	All vehicles allowed
2	HOV 2+ occupant vehicles allowed
3	HOV 3+ occupant vehicles allowed
4	All vehicles allowed, except for trucks
5	Airport passenger auto driver trips allowed
9	Closed link or transit only link.

Example use cases:

- Transit only: Link closed to all traffic other than transit vehicles. If no transit routes traverse the link, then it is essentially closed to all vehicle traffic.
- Directional coding of managed-lane facilities, such as HOV and HOT, where some links are effectively closed to vehicles in some directions, during some periods of the day.
- Change in link directionality through time, e.g., if a road is 2-directional in some network years, but changes to a one-way street in the future, then limit 9 is used on the direction that is closed in the future.
- Reversible lanes, e.g., Rock Creek Parkway has limit code 9 in the off-peak direction, since it is closed for travel in that direction.
- Roads that do not exist in early years of the plan but are built in later years. For example, I-270 has future-year improvements in 2030 north of I-370. In the early years, these links are coded as limit code 9, since they do not yet exist.

Table 26 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 27 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 28 Zonal Area Type File (AreaType\_File.dbf)

Variable Name	Description
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)

Ref: "I:\ateam\docum\fy19\tpb\_tdfm\_gen2\ver2.3\travel\_model\_user\_guide\AreaType\_File.xlsx"

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

Table 29 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 40. Note that in the Version 2.4.6 Travel Model, the consumer price index file uses 2020 as the current CPI year, since the model currently uses the model inputs from the 2022 Update to Visualize 2045. The CPI file, along with other model inputs, will be updated when the Ver. 2.4.6 Model is used for the next LRTP update.

Figure 40 Consumer price index file (CPI\_File.txt)

```

1  ;; - MWCOC V2.3 Travel Model - Cost deflation Table
2  ;; - 6/15/2021 - RN
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  ;; https://www.bls.gov/regions/mid-atlantic/data/consumerpriceindexannualandsemiannual_table.htm
6
7  InflationFTR      = 1.0   ; Inflation Assumption (DEFAULT IS 1.0)
8  Defl_OverRide    = 0.0   ; Deflation Override (DEFAULT IS 0.0) If Non-zero it is used as deflator
9                    ; Used as deflator IF NON-ZERO
10 BaseCPIYear      = 2007   ; Base year of the CPI Table
11 CurrCPIYear     = 2020   ; Current year on CPI Table
12
13 ;=====
14 ; Establish historic CPI table and Deflation Factor      =
15 ;=====
16 ;
17 LOOKUP Name=CPI_Table,
18     LOOKUP[1]    = 1,Result = 2,          ;; CPI index (from US BLS)
19     LOOKUP[2]    = 1,Result = 3,          ;; Compounded Growth Rate From Base Year
20     LOOKUP[3]    = 1,Result = 4,          ;; Deflation Factor
21     Interpolate = N, FAIL=0,0,0,list=Y,
22
23 ;;
24 ;;          (((YrCPI/BsCPI)^(1/n))-1.0)*100   BsCPI/YrCPI)
25 ;;          Annual_Avg.                       Historic Deflation
26 ;;          Growth_Rate(%)                     Factor
27 ;; ----
28 R=' 2007, 207.342, 0.00, 1.0000 ', ; <--- BaseCPIYear
29 ' 2008, 215.303, 3.84, 0.9630 ', ;
30 ' 2009, 214.537, 1.72, 0.9665 ', ;
31 ' 2010, 218.056, 1.69, 0.9509 ', ;
32 ' 2011, 224.939, 2.06, 0.9218 ', ;
33 ' 2012, 229.594, 2.06, 0.9031 ', ;
34 ' 2013, 232.957, 1.96, 0.8900 ', ;
35 ' 2014, 236.736, 1.91, 0.8758 ', ;
36 ' 2015, 237.017, 1.69, 0.8748 ', ;
37 ' 2016, 240.007, 1.64, 0.8639 ', ;
38 ' 2017, 245.120, 1.69, 0.8459 ', ;
39 ' 2018, 251.107, 1.76, 0.8257 ', ;
40 ' 2019, 254.657, 1.76, 0.8110 ', ;
41 ' 2020, 258.811, 1.72, 0.8011 ' ; <--- Curr(ent)CPI Year
42 ; --- end of CPI File -----
43 ; -----

```

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 38).

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 2.4.6 Model. The highway elements are shown in Table 30.

Table 30 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 120000-160000 90000-90999**	Established ranges for each jurisdiction.
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>97</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.

\*\* Exclusively reserved for Montgomery County.

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,

<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),” October 2010, 6 and 10.

are reserved for future TAZ assignments.<sup>98</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 ranges: 20000 to 60000 and 120000 to 1600000. The unused nodes ranging from 90000-90999 are reserved exclusively for Montgomery County studies where more detailed coding is included to avoid the possibility of using the same node numbers in different locations. 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 31.

Table 31 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),

<sup>98</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.

- Floating\_LU.dbf (a zonal file containing the area, population, and employment within one mile),<sup>99</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.
- 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 based on the tollgrp code.
- 4) Speed and capacity classes are next defined as a two-digit integer, where the first digit represents the facility type and the second digit represents 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 <i>override</i> for toll group "t", as specified in Toll_Esc.dbf

---

<sup>99</sup> TAZ-level floating density is calculated by using the TAZ centroids and creating a 1-mile point buffer around each centroid.

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

The 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 32.

Table 32 zonehwy.net file Variables description

Variable Name	Description
A	A-Node
B	B-Node
DISTANCE	Link Distance in miles (x.xx)
SPDC	Not used
CAPC	Not used
JUR	Jurisdiction Code (0-23) <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 (1-3722)
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
<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)



<b>Variable Name</b>	<b>Description</b>
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 2.4.6 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>100</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. The *Remove\_PP\_Speed.s* script is executed in the pump-prime iteration only. The principal inputs and outputs are shown in Table 33 and Table 34, respectively.

**Table 33 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

<sup>100</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 34 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"

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

[Time rate]<sub>v</sub> = Time valuation (min/2007 dollar) of toll costs for vehicle class "V"  
 [Vehicle factor]<sub>vf</sub> = 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 toll\_minutes.txt (see below), which is located in the Support folder. 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>101</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 four tables**: 1) time (min), 2) distance in implied tenths of miles, 3) total toll (year-2007 cents), and 4) variably priced tolls (year-2007 cents).

<sup>101</sup> Jinchul Park to Team B Modelers, "Processes Related to Toll Setting in Version 2.3 Model (Draft)," Memorandum, October 12, 2012.

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 2.4.6 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 100, 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, 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 35. The outputs are shown in Table 36. 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 35 Inputs to auto driver trip development**

Pre-existing final iteration ModeChoice model output modal trip tables	???_NL_MC.MTT	Binary
Pump Prime iteration person trip tables	<iter>_???.PTT	Binary
Current iteration ModeChoice 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 36 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.s* (to update the bus speed/runtime degradation process, described in detail in section 1.3.3), 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 37. Specific file descriptions are shown in Table 38 through Table 41. The output files are shown in Table 42.

**Table 37** 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
CPI schedule and parameter file	CPI_File.txt	Text
Station mode-station type-max access dist. Lookup	StaAcc.dbf	DBF
Jurisdiction code- jurisdiction group lookup	Jur.dbf	DBF
List of zones connected to the Pentagon Metrorail station for the purpose of creating long-distance kiss-and-ride (KNR) links, which represent “slugging” or 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 38 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 39)
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 station file (station.dbf) is created by the create-station-file function of COGTools using transit nodes and transfer links. The input files for this procedure<sup>102</sup> are pre-existing transit support files listed at the top of p. A-5. STAN1, STAN2, STAN3, STAN4 represent transit stop nodes, which are used to generate station-to-transit-node connectors. A node could be a bus bay, bus stop, a light rail stop, a light

<sup>102</sup> Meseret Seifu to Files, "Create a Station File," Memorandum, July 20, 2011.

rail station, or a commuter rail station, etc. The information of these nodes is used in four scripts: *Autoacc5.s*, *Parker.s*, *Set\_Factors.s*, and *Refine\_Station\_File.s*. One transit station could have STAN1, or STAN1 and STAN2, or STAN1, STAN2, and STAN3. A station with four STANs could have other station connections beyond these four that exist in the network geodatabase, but these are not shown explicitly in the station file. It is worth noting that only the STAN1 information for the bus parking lots (“B”) actually get used in the model. All the information contained in “STAN2”-“STAN4” and contained in “STAN1” for non-bus modes in the station file (station.dbf) are not used in the model.

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 39 describes the meaning of each of the six access distance codes.

**Table 39 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 81. 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.

In the autoacc5.s script, the maximum station access distances correspond to the “AccDist” variable in the StaAcc.dbf file and are defined in the autoacc5.s script in an array called “STAD”. The following logic is used in autoacc5.s:

1. If a station (k) is not used, then STAD[k]=0.
2. If a station does not have a parking lot (STAPARK[k] != 'Y'), only KNR access links can be generated and STAD[k]=300 (i.e., 3.00 miles).
  - a. For stations that do not have a parking lot, the default value of STAD[k] is 300 (i.e., 3.00 miles).

- b. NCT=8 is a special NCT code reserved for the Pentagon Metrorail Station. As stated in Table 39, the Pentagon Metrorail Station allows for very long KNR links to represent “slugging.” Since the Pentagon station does not have a PNR parking lot for commuter use, its STAD[k]=300. However, the autoacc5.s script generates the KNR access links to this particular station in a separate process, in which there is no limit on the distance of those links (in other words, STAD[k] is not used for this particular station).
  - c. NCT=9 is a special NCT code reserved for stations that do not have a parking lot and prohibit KNR drop-offs, such as Dupont Circle, Farragut North, and Metro Center. As a result, although STAD[k] for those stations has the default value of 300, the model will not generate any PNR or KNR access link to them (also indicated in Table 39).
3. The model writes out the STAD values for all the stations listed in the station file in a text file called "debug1.asc" for debugging.

Due to the program logic, a model user does not need to worry about the “AccDist” values for stations with NCT=8 or 9 because the model automatically takes care of them: For NCT=8, KNR access links will be generated with no distance limitation and no PNR access links will be generated; for NCT=9, no PNR or KNR links will be generated at all.

Table 40 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 183.

Table 41 Walk Access Links (WalkAcc\_Lnks.dbf)

File Name	Variable Name	Description
WalkAcc_Links.dbf	A	A-Node
	B	B_Node
	DISTANCE	Link distance (in 1/100 <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 42 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 2.4.6 transit network system. The elements are listed in Table 43. 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 43 Overview of Version 2.4.6 Transit Network Elements**

Transit network element	Description	Numbering	Modes/Nodes
Bus stop nodes	Highway nodes that reflect bus stops	20000-60000 120000-160000	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: Metrorail 10001-10999: Metrorail
Rail links	Fixed guideway segments connecting stations (non-highway transit links)	-	Mode 3 = Metrorail Mode 4 = Commuter rail

Walk access links	TAZ-transit stop bike/pedestrian connections	-	Mode 5 = Light rail Mode 10 = BRT, Streetcar 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)		Mode = 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>103</sup> by four sub-mode groups (Bus only, Metrorail only, Bus-Metrorail combination, and commuter rail) by three access mode (walk, PNR, KNR).<sup>104</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 44 and Table 45, respectively.

**Table 44 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	<i>PATHTRACE.S</i>	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>103</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>104</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 45 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

\* TTT files: total transit time skims, which excludes wait time.

## 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. 2.4.6 travel demand model are shown in Table 46. 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 47.



Table 46 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 (no CR KNR file is created)	<iter>_OP_WK_CR.SKM <iter>_OP_DR_CR.SKM (no CR KNR file is created)
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 47 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 transfers (0 to N)
13	Drive-access time (0.01 min)
14	Drive-access 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 2016), 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., 2016). 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>105</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>106</sup> and has been retained in the Version 2.2, 2.3, and 2.4 Travel Models. In 2015, the local bus speed degradation factors were re-estimated,<sup>107</sup> and those re-estimated factors were part of the Ver. 2.3.57a travel demand model.

---

<sup>105</sup> Jeff Bruggeman and Bill Woodford to Ronald Milone, “Comments on MWCOG Modeling Procedures,” Memorandum, June 30, 2004.

<sup>106</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>107</sup> Meseret Seifu and Ronald Milone, “Update of Local Bus Speed Degradation Model,” Memorandum, March 19, 2015.

As of Ver. 2.3.66 Model, the process was significantly updated, as was described in section 1.3.4 of the Ver. 2.3.66 Model user’s guide<sup>108</sup> and a technical memo.<sup>109</sup> The major changes that were made to the process are also described in section 16.2.2 of the Ver. 2.4 Model user’s guide.<sup>110</sup>

In the Ver.2.4.6 model, Instead of updating the bus degradation factors every year when the baseline schedules change, the “Adjust\_Runtime.s” script was revised to read the “CurrCPIYear” variable (i.e., Current CPI Year) from the “CPI\_File.txt” file, which indicates the year when the baseline bus runtime information in transit mode files is extracted, looks up the degradation factors both for the “CurrCPIYear” and for the modeled year, and calculates new bus degradation factors between the current year and the modeled year using the current factors for the modeled year (relative to 2015) divided by the corresponding factors for the Current CPI Year (also relative to 2015). Let’s say, in the above example, that the bus degradation factor for the Current CPI Year of 2020 is 1.2 and that for the modeled year of 2045 is 1.8, then the revised script would compute a degradation factor of 1.5 (=1.8/1.2) and would use it to adjust the local bus runtimes coded in the year-2045 transit line files. 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 41. An example of a local bus route **after** the adjustment of its RUNTIME variable is shown in Figure 42.

```
;*          WMATA          *
;*****
LINE NAME="WMV04I",
OWNER="WMATA;CAPITOL HEIGHTS STATION & BUS BAY C;1ST SE & K ST SE;2015;0",
ONEWAY= Y,MODE= 01,FREQ[1]= 15,RUNTIME= 54,
N=  21188  21189  21977  21978  21190  21685  21684  21192  21968,
    21193  21196  21197  21204  21576  21203  21671  21202  21201,
    21182  20662  21303  21302  21704  21710  21705  21301  21300,
    21711  21299  -20967 -20969 -21417 -21416  21297  21657  21248,
    21254  21294  21410  21292  21732  21291  21290  21289  21286
```

Figure 41 A local bus route before its RUNTIME value is updated (excerpt from the Ver2.3.78 travel model Year 2045)

<sup>108</sup> Ronald Milone, Mark Moran, and Meseret Seifu, “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)” (Washington, D.C.: Metropolitan Washington Council of Governments, National Capital Region Transportation Planning Board, February 13, 2017), 7, <https://www.mwcog.org/transportation/data-and-tools/modeling/model-documentation/>.

<sup>109</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.

<sup>110</sup> Ray Ngo, Mark Moran, and Feng Xie, “User’s Guide for the COG/TPB Gen2/Version 2.4 Travel Demand Forecasting Model” (Washington, D.C.: Metropolitan Washington Council of Governments, National Capital Region Transportation Planning Board, March 15, 2021). <https://www.mwcog.org/transportation/data-and-tools/modeling/model-documentation/>

```

;*          WMATA          *
;* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
LINE NAME="WMV04I",
OWNER="WMATA;CAPITOL HEIGHTS STATION & BUS BAY C;1ST SE & K ST SE;2015;0",
ONEWAY= Y,MODE= 01,FREQ[1]= 15,RUNTIME= 56,
N= 21188 21189 21977 21978 21190 21685 21684 21192 21968,
   21193 21196 21197 21204 21576 21203 21671 21202 21201,
   21182 20662 21303 21302 21704 21710 21705 21301 21300,
   21711 21299 -20967 -20969 -21417 -21416 21297 21657 21248,
   21254 21294 21410 21292 21732 21291 21290 21289 21286
    
```

Figure 42 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 48.

Table 48 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.

## 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 49 and the outputs are shown in Table 50. 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 49 Inputs to transit fare development

Zonal transit walk percent	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 50 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 51 TAZ/Bus Fare Zone Equivalency File Format Description (TAZFRZN.ASC)

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

As shown in Table 51 above, the TAZ/Bus Fare Zone Equivalency File (TAZFRZN.ASC) essentially contains two look-up tables: the zonal data table includes all lines in the file, while the station data table includes only the first 150 lines of the file. Both look-up tables use Columns 1-8 as the index column, which represents TAZ Number (1-3675) for the zonal data and represents Station Number (1-150) for the station data. Station information contained in Columns 41-48 and 49-56 are populated in only the first 150 lines of the file (zeros are used as placeholders for Lines 151-3675).

## 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 latest 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 one 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 four 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 four 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**

$$\text{Transit fare} = [(\text{Bus Fare}(bi1/bj1) + \text{Bus Fare}(bi1/bj2))] / 2.0$$

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

$$\text{Transit fare} = [(\text{Bus Fare}(bi1/bj1) + \text{Bus Fare}(bi1/bj2) + (\text{Bus Fare}(bi2/bj1) + \text{Bus Fare}(bi2/bj2))] / 4.0$$

**Condition 4: Metrorail related paths**

$$\text{Transit Fare} = (\text{Bus Access fare} * (1.0 - \text{Origin Metrorail walk Pct.})) + \text{Metrorail fare}(si/sj) + (\text{Bus Egress fare} * (1.0 - \text{Destin. Metrorail walk Pct.}))$$

Bus Access Fare **Single** bus fare zone to **Single** Metrorail bus fare zone =

$$\text{Bus Fare}(bi1/mi1) - 0.5 \text{ Rail-Bus Discount}$$

Bus Access Fare **Single** bus fare zone to **Double** Metrorail bus fare zone =

$$\text{Min}[\text{Bus Fare}(bi1/mi1), \text{Bus Fare}(bi1/mi2)] - 0.5 \text{ Rail-Bus Discount}$$

Bus Access Fare **Double** bus fare zone to **Single** Metrorail bus fare zone =

$$[\text{Bus Fare}(bi1/mi1) + \text{Bus Fare}(bi2/mi1)] / 2.0 - 0.5 \text{ Rail-Bus Discount}$$

Bus Access Fare **Double** bus fare zone to **Double** Metrorail bus fare zone =

$$[\text{Min}[\text{Bus Fare}(bi1/mi1), \text{Bus Fare}(bi1/mi2)] + \text{Min}[\text{Bus Fare}(bi2/mi1), \text{Bus Fare}(bi2/mi2)]] / 2.0 - 0.5 \text{ Rail-Bus Discount}$$

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 48). 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 52 and Table 53.

**Table 52 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 53** 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 54.

Table 54 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 43 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” (other than the truck trip generation phase, where special generator TAZs are identified) 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>111</sup> These multiplicative adjustment factors included in

<sup>111</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>.

the Version 2.3 and Version 2.4 series of models are used to refine the trip productions and trip attractions that are normally calculated with land activity inputs and modeled trip rates. P-mods or A-mods may be applied either on a zonal basis or on an areawide basis at the discretion of the analyst. There are two principal reasons to use these types of adjustments:

- To more accurately account for the trip generation effects of special land activity conditions that are not adequately captured by the land activity or by the modeled trip rate. These types of conditions are typically known as special generators; or
- To refine the model calibration at higher geographic levels of aggregation such as at a district or a jurisdiction level of analysis. As trip generation is applied on a zone-by-zone basis, comparisons of estimated and observed results at higher levels of aggregation usually exhibit marginal disagreements. P-mods and A-mods may serve as adjustment tools for refining trip generation at more aggregate levels of analysis. While this adjustment is normally small in magnitude, it can substantially improve the downstream (trip distribution and mode choice) model performance.

Modification factors used for the purpose of refining the model calibration typically range between 0.90 and 1.10 (i.e., +/- 10%). In contrast, adjustments made for the purpose of reflecting special generators may be much larger in magnitude, yielding a result that is in line with observed traffic generation for a specific location. Admittedly, these types of external adjustments are sometimes viewed as ways to bias the model output or a way to undermine the logic of the model. It is important to include reasoned justifications for the use of these types of adjustments.

**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 43 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 55); 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>112</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>113</sup> (for which COG pays a license fee) and the work was performed in 2010 by COG/TPB staff.<sup>114</sup> Note that NAVTEQ is now known as HERE. 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>115</sup>
- Ramps (RAMP = Y)

Figure 44 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>112</sup> It has been found that areas with a higher density of street blocks are more walkable.

<sup>113</sup> NAVTEQ, "NAVTEQ's NAVSTREETS Street Data, Reference Manual v3.2," Proprietary and Confidential (Chicago, Illinois: NAVTEQ, April 1, 2009).

<sup>114</sup> Mary Martchouk to Mark S. Moran, "Developing GIS Walkability Measures," Memorandum, June 2, 2010, 6–7.

<sup>115</sup> NAVTEQ, "NAVTEQ's NAVSTREETS Street Data, Reference Manual v3.2", p. 4-5.



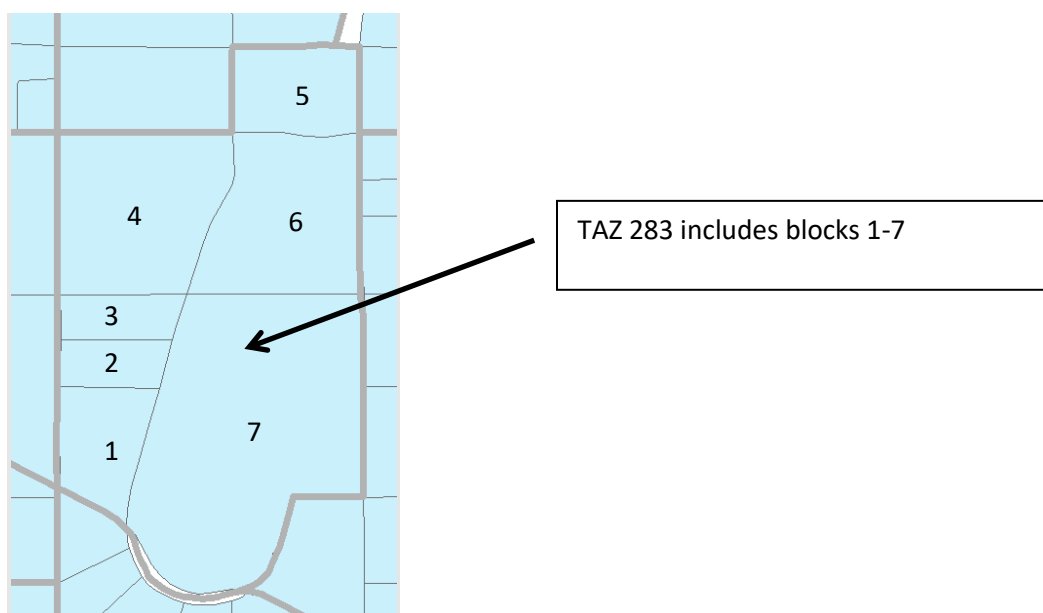


Figure 44 Example of seven NAVTEQ street blocks within TAZ 283 (Union Station)

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>116</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 55. 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>117</sup> The specific data elements are shown on Table 56. Note that, in the file *Ext\_PsAs.dbf* (Table 55), 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>116</sup> Milone et al., “Calibration Report for the TPB Travel Forecasting Model, Version 2.3,” 4–17 to 4–20.

<sup>117</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 55 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 56). 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 57.

Table 56 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 57 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 58.

Table 58 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 54. 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 59.

Table 59 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:

- *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.
- *Prepare\_Ext\_Auto\_Ends.s*, *Prepare\_Ext\_ComTruck\_Ends.s*: External trip-ends (Ps and As) are prepared.
- *Trip\_Distribution\_External.s*: External trip-ends are distributed, resulting in external trip tables, by purpose. Please note that the *Prepare\_Ext\_Auto\_Ends.s* and *Trip\_Distribution\_External.s* scripts have been further updated in the Ver. 2.4 Model based on the 2014 cellular origin-destination (O-D) data.
- *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>118</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

- *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>118</sup> This equation was developed by Bill Mann in the early 1990s.





## 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 60 and Table 61. 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 60 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 61 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<sub>i</sub> term calculation is specified in Table 62. 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 62 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% VOT)	Non-work (50% VOT)
\$ 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:

$$\text{IntAttrScaleFtr} = (\text{“Final” Intl Ps} + \text{Extl. Ps} - \text{Extl. As}) / \text{Intl. “Computed” As}$$

The internal Ps and As in the above equation include 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

As shown in Figure 45, the mode choice model in the Version 2.4.6 Travel Model was designed to have 15 choices, made up of the following modes:

- Three auto modes: Drive alone, shared ride 2 person, and shared ride 3+ person.
- Three transit access modes:
  - Drive to transit and park in a park-and-ride (PNR) lot;
  - Drive to transit and drop off passenger(s) at the kiss and ride (KNR) lot or station; and
  - Walk to transit.
- Four transit modes: Commuter rail, all bus, all Metrorail, and combined bus/Metrorail.

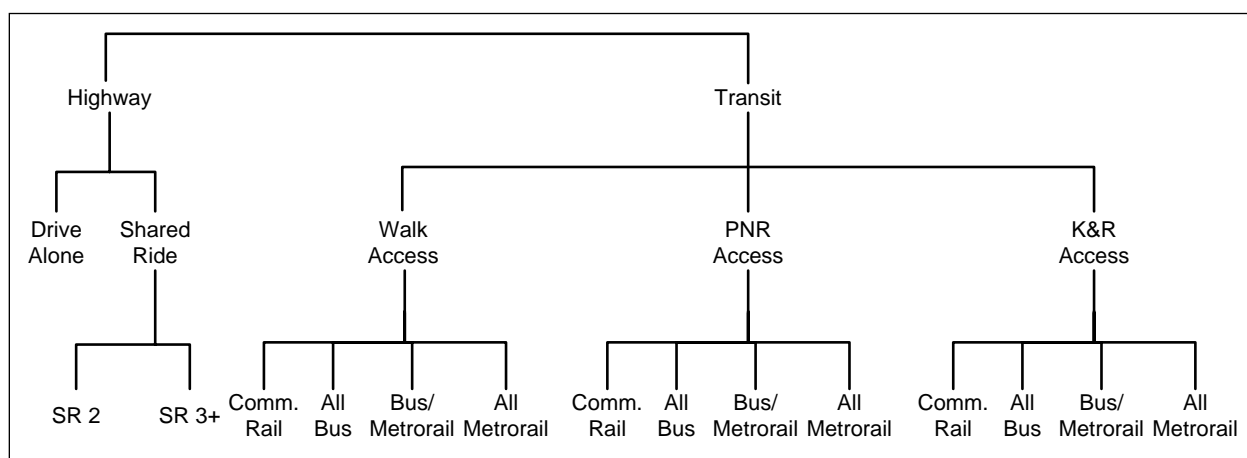


Figure 45 Designed nesting structure of the nested-logit mode choice model in the Version 2.4.6 Model

\* In model implementation, PNR and KNR access modes for commuter rail are combined into one choice, resulting in 14 choices, not 15.

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

Two important distinctions should be made. First, as per the design of AECOM, for the commuter rail mode, the model was implemented such that PNR and KNR commuter rail are combined as a single choice (in mode choice) or a single path (in path building), since, for commuter rail, the PNR- and KNR-access links are identical. Thus, instead of 12 access-mode/transit-mode choices, the model is implemented using 11 access-mode/transit-mode choices. Consequently, as implemented in the model, the mode choice model has 14 choices, not 15. This is difficult to portray in Figure 45, but is noted in a footnote on the figure. This combining of modes is also apparent in Table 67 (“Outputs from the ModeChoice mode choice application program”).

Second, regarding the three auto modes: 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 and 2.4.6 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.

### 21.1.1 Other issues relating to travel modes

Table 63 lists the ten transit modes that are handled by the Version 2.4.6 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).

Table 63 Transit sub-modes represented in the Version 2.4 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 streetcar	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 64.

Table 64 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 63 and Table 64 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 65. The mode interchanges where transfers are prohibited are denoted by “Y”.

**Table 65 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 Elimination of Metrorail constraint to and through the regional core

As of Ver 2.3.75, the Metrorail constraint to and through the regional core has been removed due to the stable long-term funding of \$500 million a year for Metro to support WMATA’s plans to implement all 8-car trains during peak periods in the Visualize 2045 Plan.<sup>119</sup> For the sake of documentation, below is a more detailed description of the constraint and its modeling-related aspects. This description came from previous model documentation.<sup>120</sup>

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

<sup>119</sup> Jane Posey, “Amendments to the Visualize 2045 Air Quality Conformity Scope of Work,” Memorandum, May 16, 2018.

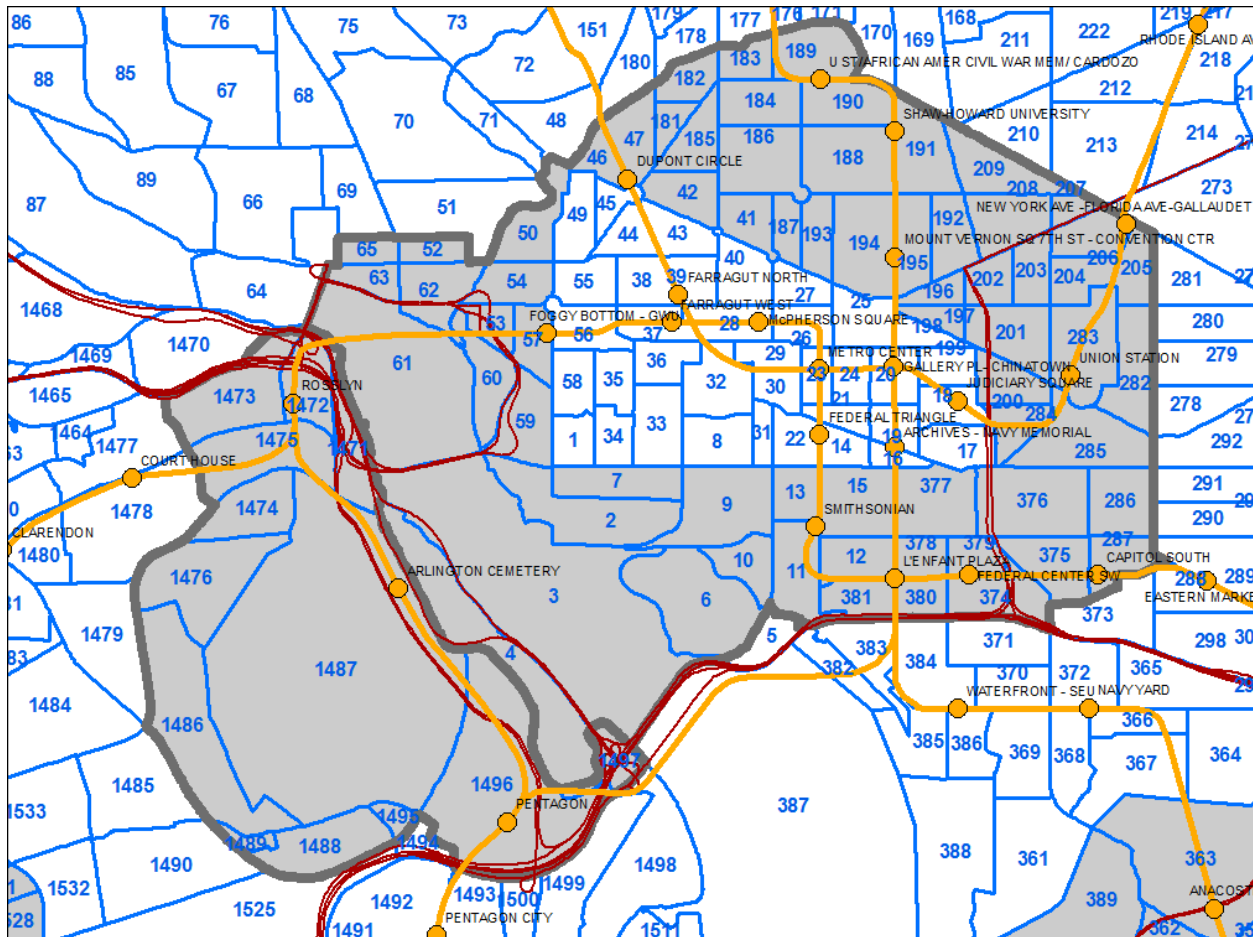
<sup>120</sup> Moran, Milone, and Seifu, “User’s Guide for the COG/ TPB Travel Demand Forecasting Model, Version 2.3.70. Volume 1 of 2: Main Report and Appendix A (Flowcharts).”

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>121</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 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 46, 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.

---

<sup>121</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).

Figure 46 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. Model runs representing any year following the binding year, e.g., 2030, are conducted using the *mode\_choice\_tc\_v23.bat* batch file, 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\_parallel.bat* batch file, which invokes the following:



- Mode choice model application program (ModeChoice.exe);
- Mode choice control files by purpose (<purp>\_ModeChoice.ctl)
- 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 (ModeChoice.exe);
  - Jurisdictional summary script (*MC\_NL\_Summary.s*);
  - Constraint adjustment script (*MC\_Constraint\_V23.s*);
- 

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

Note that the nesting structure of the TPB Version 2.4.6 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)

#### 21.2.1.1 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 an 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>122</sup> the modeler must add “transit-only” links to the transit network to represent the new service,

---

<sup>122</sup> Such as Metrorail (Mode 3), commuter rail (Mode 4), LRT (Mode 5), and BRT/streetcar (Mode 10).

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.

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>123</sup> though recent network documentation has designated the node number range of 7000-7999 (light rail/BRT PNR centroids), even though this range is not currently in use in the geodatabase.<sup>124</sup> Cube Voyager cannot combine headways for routes unless they are part of the same mode code, so, in cases where Mode 10 routes share a street segment with local bus (Mode 1), these two routes will not be represented with a combined headway.

#### **21.2.1.2 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>125</sup>

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

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

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

#### **21.2.1.4 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 67. 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

---

<sup>123</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),” October 2010, 6.

<sup>124</sup> Meseret Seifu, Ronald Milone, and Mark Moran, “Highway and Transit Networks for the Version 2.3.66 Travel Model, Based on the 2016 CLRP and FY 2017-2022 TIP,” Final Report (Washington, D.C.: Metropolitan Washington Council of Governments, National Capital Region Transportation Planning Board, March 17, 2017), 17, <https://www.mwcog.org/transportation/data-and-tools/modeling/model-documentation/>.

<sup>125</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),” October 2010, 10.

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.

### 21.3 Control/Support Files

In the Version 2.4.6 travel model, the nested-logit mode choice (NLMC) model is applied using a software program known as TRANSIMS ModeChoice,<sup>126</sup> which is an open-source C++ program. 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\_AM\_ModeChoice.CTL, HBS\_OP\_ModeChoice.CTL, HBO\_OP\_ModeChoice.CTL, NHW\_OP\_ModeChoice.CTL, and NHO\_OP\_ModeChoice.CTL.** A script file (**ModeChoice\_Script.txt**) and a geographic market segmentation file (**Segment\_Map.csv**), which are common for all trip purposes:

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 ModeChoice application program are shown in Table 66. The outputs are shown in Table 67.

Table 66 Inputs to the ModeChoice mode choice application program

Mode-specific constants and income-level adjustment constants	HBW_AM_ModeConstants.csv HBS HBO NHW NHO_OP_ModeConstants.csv HBW_AM_BiasConstants.csv HBS HBO NHW NHO_OP_BiasConstants.csv	CSV
Mode choice common keys control file	ModeChoice_CommonKeys.ctl	Text
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

<sup>126</sup> <https://sourceforge.net/projects/transims/>

All bus transit skims (INFILE 4)	trnam_ab.skm, trnop_ab.skm	Binary
Metrorail transit skims (INFILE 5)	trnam_mr.skm, trnop_mr.skm	Binary
Bus/Metrorail transit skims (INFILE 6)	trnam_bm.skm, trnop_bm.skm	Binary
Zonal data (INFILE 8)	zonev2.a2f	Text

Table 67 Outputs from the ModeChoice mode choice application program

Daily person trips, stratified by travel mode (14 tables): <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>	hbw_nl_mc.mtt, hbs_nl_mc.mtt, hbs_nl_mc.mtt, nhw_nl_mc.mtt, nho_nl_mc.mtt	Binary
	hbw_am_Summary.csv, hbs_op_Summary.csv, hbo_op_Summary.csv, nhw_op_Summary.csv, nho_op_Summary.csv	CSV

## 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 2.4 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>127</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 68.<sup>128</sup>

Table 68 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>129</sup>

1. DC core
2. VA core
3. DC urban

---

<sup>127</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>128</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>129</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 47 and in Table 69. Table 69 shows the equivalency between the seven NLMC superdistricts and the new 3,722-TAZ area system.

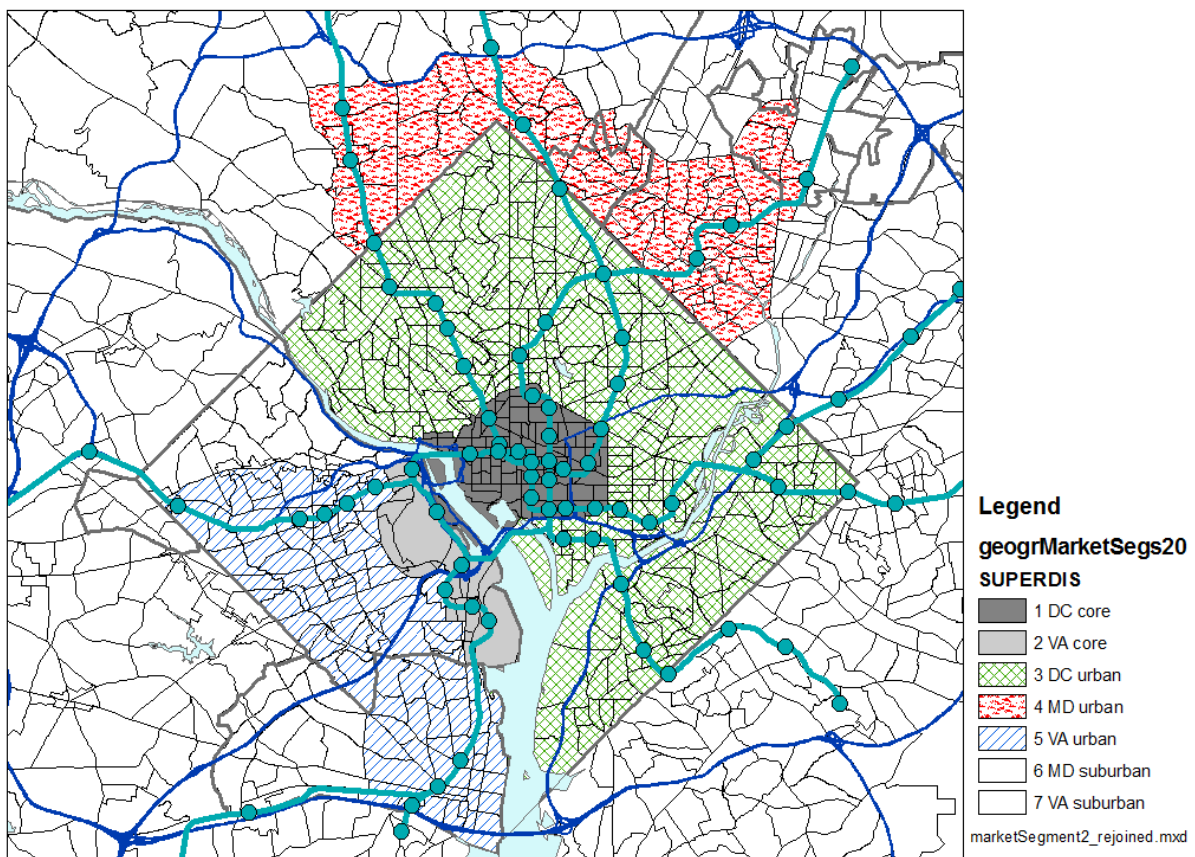


Figure 47 Seven superdistricts used in the Versions 2.3 and 2.4 nested-logit mode choice model

Ref: "I:\ateam\nest\_log\marketsegment2\_rejoined.tif"

Table 69 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 69 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 model development group reviewed the new zone system and found a few cases where the zone boundaries needed adjustment.<sup>130</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>131</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 70 and Table 71. Another way to view the 20 geographic market segments is shown in Table 72.

<sup>130</sup> Meseret Seifu, “Review of New Zone System: 3722 Transportation Analysis Zones (TAZ)” (January 22, 2010 meeting of the COG/TPB Travel Forecasting Subcommittee, held at the Metropolitan Washington Council of Governments, Washington, D.C., January 22, 2010), 4, <http://www.mwcog.org/uploads/committee-documents/Z15aV1dd20100122152445.pdf>.

<sup>131</sup> Seifu, 23.

Table 70 Production and attraction market segments used in the TPB Versions 2.3 and 2.4 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 71 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 72 Equivalency between seven super-districts and the 20 geographic market segments

	1 DC core	2 VA core	3 DC urban	4 MD urban	5 VA urban	6 MD suburban	7 VA suburban
1 DC core	1	2	3	3	3	4	4
3 DC urban	1	2	3	3	3	4	4
4 MD urban	5	6	7	7	7	8	8
2 VA core	9	10	11	11	11	12	12
5 VA urban	9	10	11	11	11	12	12
6 MD suburban	13	14	15	15	15	16	16
7 VA suburban	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>132</sup> The second subsection describes the more specific case of the six transit-access markets used in the mode choice model of the Version 2.4.6 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 TRANSIMS ModeChoice Version 7.5.8,<sup>133</sup> 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 2.4.6 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-

<sup>132</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).

<sup>133</sup> AECOM, “ModeChoice, Version 7.5.8” (Arlington, Virginia, September 15, 2021), <https://sourceforge.net/projects/transims/files/documentation/TRANSIMS%20Version%207.5%20User%20Guides.7z/download>.

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.
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>134</sup>

For many 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

---

<sup>134</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).

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.,

$$P(AB) = P(A) * P(B)$$

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(“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(“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(“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 73 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>135</sup>

**Table 73 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 74, Table 75, and Table 76 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 74 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

<sup>135</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.”

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 74 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 75 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 75 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 76 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 76 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 2.4.6 Travel Model and ModeChoice

Regarding the percent-walk-to-transit (PWT) values used by the mode choice model of the TPB Version 2.4.6 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>136</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.

<sup>136</sup> Seifu, Milone, and Moran, “Highway and Transit Networks for the Version 2.3.66 Travel Model, Based on the 2016 CLRP and FY 2017-2022 TIP,” 8.

The mode choice model in the TPB Version 2.4.6 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]);
- 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 45 (p. 174). The NLMC model is applied using the TRANSIMS **ModeChoice**, mode choice application program.

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>137</sup> or in section 21.2.1 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>138</sup>

- Percent of the zone within a short walk to Metrorail (Mode 3): PSWMET
- 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, in 2012, thanks to work done by Dusan Vuksan and Feng Xie, it was discovered 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

---

<sup>137</sup> Milone et al., "Calibration Report for the TPB Travel Forecasting Model, Version 2.3," 6–3 to 6–5.

<sup>138</sup> AECOM Consult, Inc., "Post MWCOG – AECOM Transit Component of Washington Regional Demand Forecasting Model: User's Guide" (AECOM Consult, Inc., March 2005), 11.

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.

As of the Ver. 2.3.57 model (and continued in subsequent models, including the Ver. 2.4 Model), these new definitions have been incorporated in the in the automated ArcPy transit walkshed process.

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-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.

In the Version 2.4.6 model the transit access markets were determined within the mode choice application program (ModeChoice) by combining information from a user script file that is common for all trip purposes (**ModeChoice\_Script.txt**). The model utility functions, and conditional overrides are defined in this script file. This script contains a range of variable names, lookup tables, and logical syntax statements (e.g., if-then-else-endif). Besides this script file, the ModeChoice is a C++ program, which requires a control file (\*.CTL) for each mode choice model (e.g., HBW\_AM\_ModeChoice.CTL, HBS|HBO|NHW|NHO\_OP\_ModeChoice.CTL) and constant files (e.g., HBW\_AM\_ModeConstants.csv, HBS|HBO|NHW|NHO\_OP\_ModeConstants.csv). Percent-walk-to-transit values are stored in a zonal data file (ZONEV2.A2F) that is read into ModeChoice.

In the ModeChoice script file six percent-walk-to-transit values are developed:

- Percent of the zone within a short walk to Metrorail (Mode 3) **or LRT (Mode 5)**:  
**SHORT\_WK\_METRO**
- Percent of the zone within a long walk to Metrorail (Mode 3) **or LRT (Mode 5)**:  
**LONG\_WK\_METRO**
- Percent of the zone within a short walk to any transit in the AM peak period:  
**SHORT\_WK\_ALL\_PK**
- Percent of the zone within a long walk to any transit in the AM peak period:  
**LONG\_WK\_ALL\_PK**
- Percent of the zone within a short walk to any transit in the off-peak period:  
**SHORT\_WK\_ALL\_OP**
- Percent of the zone within a long walk to any transit in the off-peak period:  
**LONG\_WK\_ALL\_OP**

The 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 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:

WK_MR_P	Percent of trips assumed to access Metrorail/LRT via walking at the production zone
WK_MR_A	Percent of trips assumed to access Metrorail/LRT via walking at the attraction zone
WK_METRO	Percent of all trips assumed to access Metrorail/LRT via walking

WK_ALL_P	Percent of trips assumed to access all transit via walking at the production zone, AM/OP peak per.
WK_ALL_A	Percent of trips assumed to access all transit via walking at the attraction zone, AM/OP peak per.
WK_ACCESS	Percent of all trips assumed to access all transit via walking

The production or attraction zone status is indicated using the letter “P” (production) or “A” (attraction). Percent-walk-to-transit values are calculated using buffering resolution of 128 around transit stop nodes (i.e., stations, bus stops, etc.) in the GeoPandas-based process. 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):

- WK\_MR\_P: Percent of trips assumed to access Metrorail/LRT via walking at the production zone
- WK\_MR\_A: Percent of trips assumed to access Metrorail/LRT via walking at the attraction zone
- WK\_ALL\_P: Percent of trips assumed to access all transit via walking at the production zone
- WK\_ALL\_A: 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 ModeChoice script file, one finds:

- WK\_MR\_P: Percent of trips assumed to access Metrorail/LRT via walking at the production zone  
 $\triangleright = \text{MIN} (\text{MAX} (((\text{SHORT\_WK\_P} + 0.25 * (\text{LONG\_WK\_P} - \text{SHORT\_WK\_P})) / 100.0), 0), 1)$
- WK\_MR\_A: Percent of trips assumed to egress from Metrorail/LRT via walking at the attraction zone  
 $\triangleright = \text{MIN} (\text{MAX} (((\text{SHORT\_WK\_A} + 0.25 * (\text{LONG\_WK\_A} - \text{SHORT\_WK\_A})) / 100.0), 0), 1)$
- WK\_ALL\_P: Percent of trips assumed to access “all transit” via walking at the production zone  
 $\triangleright = \text{MIN} (\text{MAX} (((\text{SHORT\_WK\_P} + 0.25 * (\text{LONG\_WK\_P} - \text{SHORT\_WK\_P})) / 100.0), \text{WK\_MR\_P}), 1)$
- WK\_ALL\_A: Percent of trips assumed to egress from “all transit” via walking at the attraction zone  
 $\triangleright = \text{MIN} (\text{MAX} (((\text{SHORT\_WK\_A} + 0.25 * (\text{LONG\_WK\_A} - \text{SHORT\_WK\_A})) / 100.0), \text{WK\_MR\_A}), 1)$

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 three sub-markets, as shown in Table 77. Similarly, in the

“must drive” market, there are two sub-markets, as shown in Table 78. Lastly, there is the “no access to transit” market, which is not part of either table.

Table 77 Four “can walk” sub-markets

Sub- mkt	Transit Service Available		Description
	Production TAZ	Attraction TAZ	
MODE_ACCESS_MARKET_1	MR/LRT	MR/LRT	Share that can use MR/LRT at both ends of the trip
MODE_ACCESS_MARKET_2	Other transit	MR/LRT	Share that can use “other transit” at production end and MR/LRT at attraction end of the trip
MODE_ACCESS_MARKET_3	MR/LRT	Other transit	Share that can use MR/LRT at production end and “other transit” at attraction end of the trip

Table 78 Two “must drive” sub-markets

Sub- mkt	Transit Service Available		Description
	Production TAZ	Attraction TAZ	
MODE_ACCESS_MARKET_4	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
MODE_ACCESS_MARKET_5	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

ModeChoice makes use of “MODE\_ACCESS\_MARKET” from the ModeChoice control keys (ModeChoice\_CommonKeys.ctl) file to allow the model users to specify subzone segmentation such as “can walk,” “must drive,” and “no transit,” so **the six sub-markets above, along with the “no transit” segment, are represented with seven WALK SEG (WS) variables defined in the segment map (Segment\_Map.csv ) control file**, and these six variables are calculated as follows from the previously-defined walk percentages:

**“Can walk” market (MODE\_ACCESS\_MARKET\_1)**

- MODE\_ACCESS\_MARKET\_1: Share of the “walk segment” that can use Metrorail/LRT at both ends of the trip  

$$= WK\_MR\_P * WK\_MR\_A$$
- MODE\_ACCESS\_MARKET\_2: Share of the “walk segment” that can use “other transit” at production end and MR/LRT at attraction end of the trip  

$$= (WK\_ALL\_P - WK\_MR\_P) * WK\_MR\_A$$

- MODE\_ACCESS\_MARKET\_3: Share of the “walk segment” that can use MR/LRT at production end and “other transit” at attraction end of the trip  

$$= WK\_ALL\_P * (WK\_ALL\_A - WK\_MR\_A)$$

**“Must drive” market**

- MODE\_ACCESS\_MARKET\_4: Share of the “walk segment” that must drive (to any transit) at the production, but can walk to Metrorail/LRT at the attraction  

$$= (1 - WK\_ALL\_P) * WK\_MR\_A$$
- MODE\_ACCESS\_MARKET\_5: 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)  

$$= (1 - WK\_ALL\_P) * (WK\_ALL\_A - WK\_MR\_A)$$

**“No transit” market**

- MODE\_ACCESS\_MARKET\_6: 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)  

$$= 1 - TRIP.ACCESS1 - TRIP.ACCESS2 - TRIP.ACCESS3 - TRIP.ACCESS4 - TRIP.ACCESS5$$

**Application of six 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 six transit-access markets are applied in ModeChoice to the travel modes represented in the Ver. 2.4.6 nested-logit mode choice model as shown in Table 79.

Table 79 Application of the six transit-access segments to travel modes represented in the Ver. 2.4.6 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_BUS_MR	WK_MR	PNR_CR	PNR_BUS	PNR_BUS_MR	PNR_MR	KNR_CR	KNR_BUS	KNR_BUS_MR	KNR_MR
1	MODE_ACCESS_MARKET_1	Can Walk MR/LRT	X	X	X	X	X	X	X	X	X	X	X	X	X	X
2	MODE_ACCESS_MARKET_2	Can Walk 1	X	X	X	X	X		X	X	X	X	X	X	X	X
3	MODE_ACCESS_MARKET_3	Can Walk 2	X	X	X	X	X		X	X	X		X	X	X	
4	MODE_ACCESS_MARKET_4	Must drive 1	X	X	X				X	X	X	X	X	X	X	X
5	MODE_ACCESS_MARKET_5	Must drive 2	X	X	X				X	X	X		X	X	X	
6	MODE_ACCESS_MARKET_6	No Transit	X	X	X											

Ref: "I:\ateam\docum\fy23\Ver.2.4.6\_User'sGuide\Ver2.4.6\_Pct\_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-BUS/MR = park-n-ride to bus and Metrorail/Light Rail
- KNR-BUS/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 2.4.6 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, streetcar 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 38 on page 140, 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 39 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 81. 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 80 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 require station centroids.

**Table 80 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>139</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 and Version 2.4 Travel Models, 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

<sup>139</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),” October 2010, 6 and 10.

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>140</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).

**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 48 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*.

---

<sup>140</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.

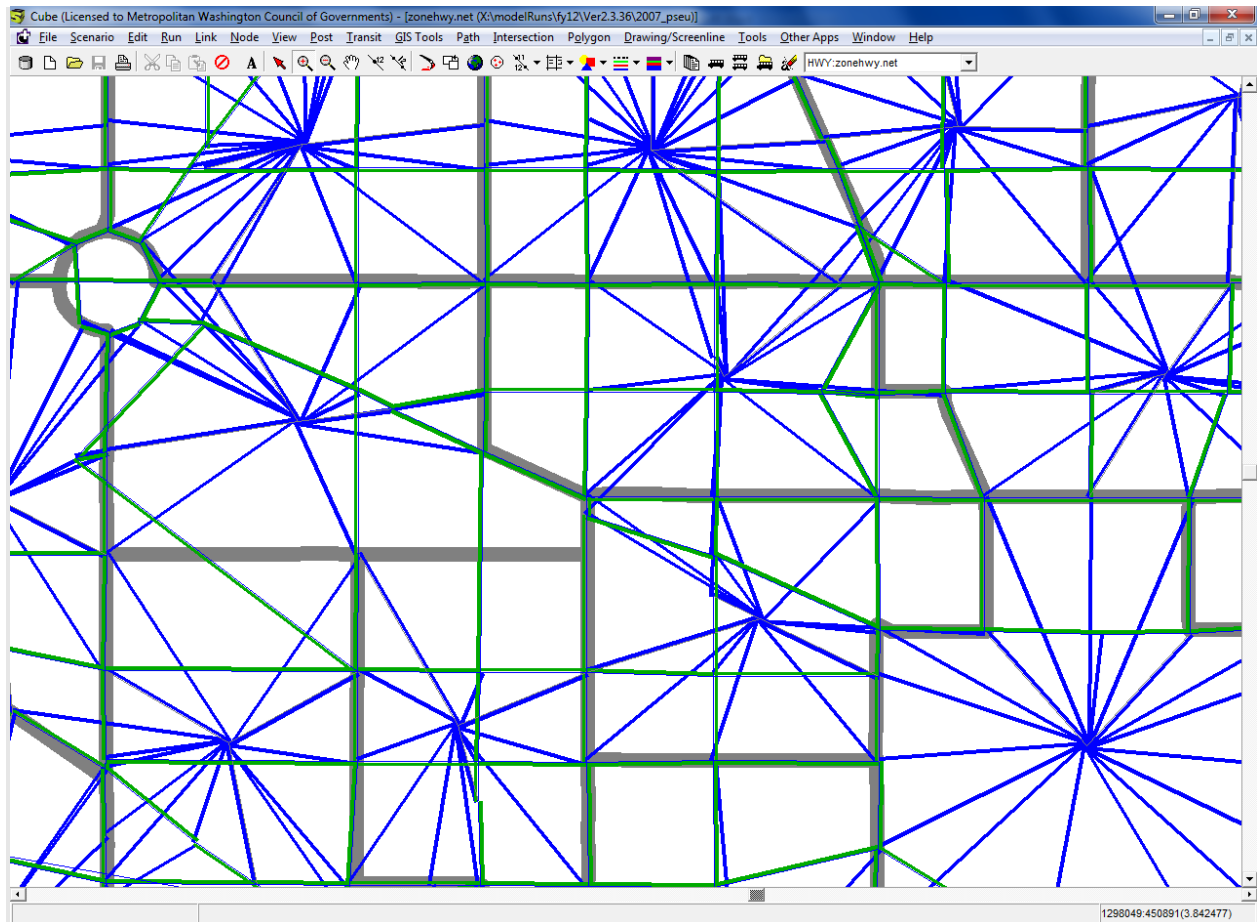


Figure 48 Zonal walk access links and sidewalk links in downtown DC near Farragut Square (Ver. 2.4.6 NL MC model)

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 (Table 81), which is a function of station type (e.g., terminal vs. non-terminal) and transit mode<sup>141</sup>;
- 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 81 Maximum link distances for drive-access-to-transit links: Ver. 2.4 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 49 shows kiss-and-ride (KNR) auto-access-to-transit links for the AM period associated with Metrorail stations in Northern Virginia.

<sup>141</sup> The maximum auto-access link distances in the Ver. 2.3/Ver. 2.4 Models are defined in the \Inputs\StaAcc.dbf file. In theory, a model user could alter maximum link distances by modifying this input file. It should be noted that, however, a significant change to this file may necessitate the re-calibration / re-validation of the transit models.

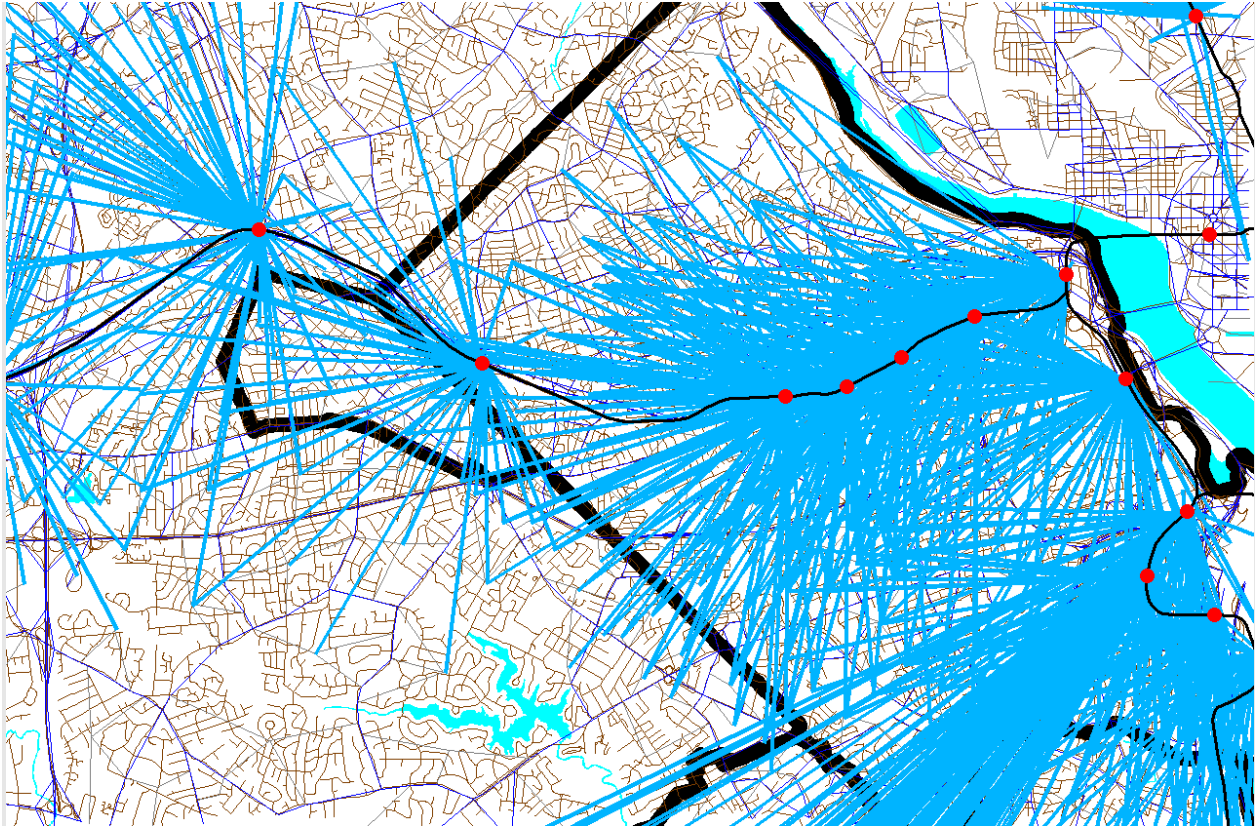


Figure 49 Kiss-and-ride (KNR) auto access links to Metrorail stations in Northern Virginia

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

Figure 50 shows park-and-ride (PNR) auto-access-to-transit links for the AM period associated with Metrorail stations in Northern Virginia. Notice that the Orange Line stations from Clarendon to Rosslyn do not have PNR-access links, since they do not have PNR lots. By contrast, these stations do have KNR-access links, since these stations can have KNR access. The Pentagon Metrorail station is another example of a station where the model does not allow travelers to have PNR access, but they may have KRN access. Notice that the KNR-access links and PNR-access links are not shaped like a circular “starburst,” but are somewhat flattened, due to the backtracking penalty. This was done to mimic the behavior of travelers who tend not to want to backtrack when driving to park at or be dropped off at a Metrorail or commuter rail station.

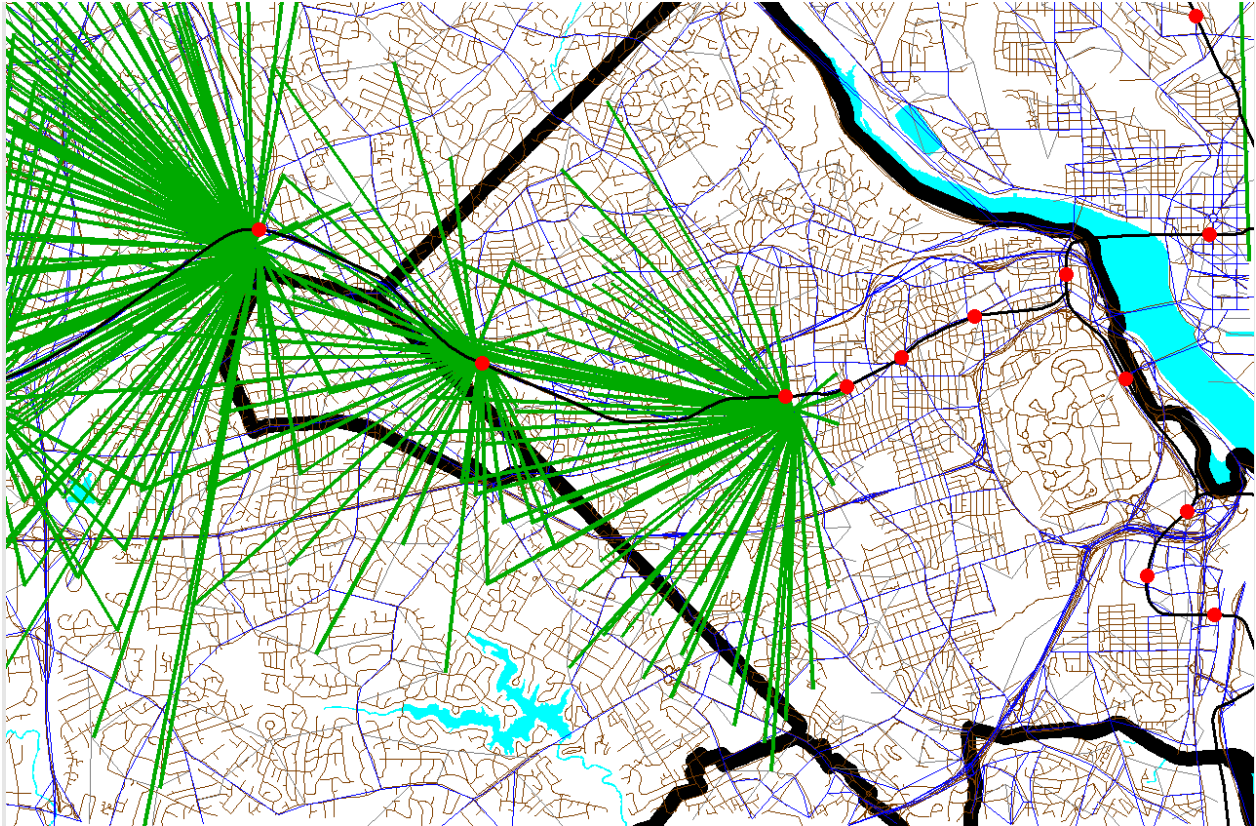


Figure 50 Park-and-ride (PNR) auto access links to Metrorail stations in Northern Virginia

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>142</sup>

The station file also includes shadow parking price variables (STAPKSHAD & STAOPSHAD, see Table 38) 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>143</sup>

<sup>142</sup> Jain to Milone and Moran, 6.

<sup>143</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 developed using GeoPandas-based transit walkshed process.

## 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 82 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>144</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 sub-modes (i.e., all transit modes have an IVT weight of 1.0). The maximum initial wait time for all ten

---

<sup>144</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 and later models, this adjustment is now done directly to the mode 1, 6, and 8 local bus line files (\*.TB) using the script Adjust\_Runtime.s.

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 82 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

## 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>146</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 2.4.6 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 2.4.6 Model includes a new parking cost model estimated based on the 2007/2008 HTS.<sup>147</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 51.

---

<sup>145</sup> AECOM Consult, Inc., *Post MWCOG – AECOM Transit Component of Washington Regional Demand Forecasting Model: User's Guide* (AECOM Consult, Inc., March 2005).

<sup>146</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),” October 2010, 6.

<sup>147</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.

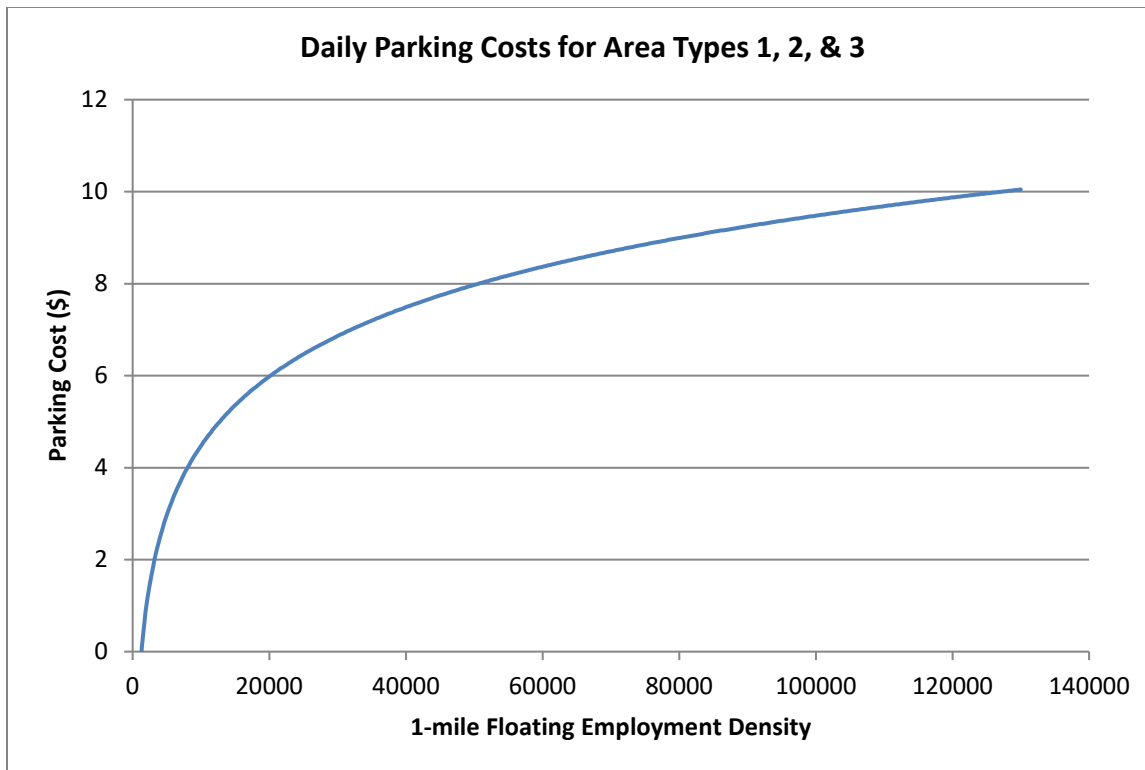


Figure 51 Non-transit-related, daily parking cost model used in the Version 2.4.6 Model

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 83.



**Table 83 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 (page A-9 of the flowchart in Appendix A) is applied to convert daily vehicle trips to time-of-day vehicle trips for 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 84 and Table 85.

Table 84 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 85 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 2.4.6 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>148</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>149</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 2.4.6 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 PCEs has not yet been implemented, but they will be considered in future developmental work.

Additionally, the Version 2.4.6 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>150</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>148</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, <https://doi.org/10.1680/ipeds.1952.11259>.

<sup>149</sup> Juan de Dios Ortúzar and Luis G. Willumsen, *Modelling Transport*, 2nd ed. (John Wiley & Sons, 1994), 304.

<sup>150</sup> Frank and Wolfe, “An Algorithm for Quadratic Programming.”

objective function. The process stops when a stopping criterion is met. Previously, the Version 2.2 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 9.

Table 86 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>151</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 2.4.6 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

### 23.2.1 Prior to 2008: 5 user classes

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 87). 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

<sup>151</sup> Moran and Milone, “Status Report on the Version 2.3 Travel Model: Updates to the Model and Year-2010 Validation,” 7–11.

5. Airport passenger vehicles.

Table 87 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>152</sup> In the revised process, shown in Table 88, 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>153</sup>

<sup>152</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>153</sup> Jinchul Park to Files, “Two Step Traffic Assignment for HOT Lane Modeling in 2008 CLRP,” Memorandum, December 2, 2008.

Table 88 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

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.

### 23.2.2 After 2008: 6 user classes

Recent versions of the regional travel demand model (e.g., from Version 2.3.52 to Version 2.4.6) continue to use the same two-step assignment, but there are now six assignments (not five) 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 89. Note that four of the six traffic assignments are multi-class, but two of the assignments contain only one user class (HOV3+ vehicles in the AM peak and HOV3+ vehicles in the PM peak).



Table 89 Traffic assignment in the Version 2.3.52 and later travel model: Six traffic assignments per speed feedback loop

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 2.4), including the traffic assignment step.

### 23.3.1 Generalized cost

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 90. 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 90 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)

### 23.3.2 Inputs and outputs

The inputs and outputs of the *Highway\_Assignment\_Parallel.s* script are shown in Table 91 and Table 95, respectively.

Table 91 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

Table 92 is a lookup table showing highway link capacities in free-flow conditions (vehicles per hour per lane). Table 93 is a lookup table showing the updated highway link speeds in free-flow conditions (mph).

Table 92 Lookup table: Highway link capacities in free-flow conditions (vehicles per hour per lane)

	Area Type					
	1	2	3	4	5	6
<b>0 Centroid Connectors</b>	3150	3150	3150	3150	3150	3150
<b>1 Freeways</b>	1900	1900	2000	2000	2000	2000
<b>2 Major Arterials</b>	600	800	960	960	1100	1100
<b>3 Minor Arterials</b>	500	600	700	840	900	900
<b>4 Collectors</b>	500	500	600	800	800	800
<b>5 Expressways</b>	1100	1200	1200	1400	1600	1600
<b>6 Ramps</b>	1000	1000	1000	1000	2000	2000

Ref: "I:\ateam\docum\fy19\tpb\_tdfm\_gen2\ver2.3\travel\_model\_user\_guide\ver2.3.75\_highway\_link\_lookupTables\_capacity\_speed.xlsx"

Table 93 Lookup table: Highway link speeds in free-flow conditions (mph)

	Area Type					
	1	2	3	4	5	6
<b>0 Centroid Connectors</b>	17	17	23	29	35	40
<b>1 Freeways</b>	63	63	69	69	75	75
<b>2 Major Arterials</b>	40	40	52	52	58	58
<b>3 Minor Arterials</b>	40	40	46	46	46	52
<b>4 Collectors</b>	35	35	35	40	40	40
<b>5 Expressways</b>	52	52	58	58	58	63
<b>6 Ramps</b>	23	23	35	35	40	58

Ref: "I:\ateam\docum\fy21\Version24Development\travel\_model\_user\_guide\ver2.4\_highway\_link\_lookupTables\_capacity\_speed.xlsx"

### 23.3.3 Multi-class assignment

As noted earlier, TPB travel forecasting model Ver. 2.3.52 (and later, including Ver. 2.4) perform six traffic assignments per speed feedback iteration (see Table 89). Four of these are multi-class assignments and two of them are single-class assignments. For the multi-user class assignments, two have five user classes (i.e., AM peak non-HOV3+ and PM peak non-HOV3+) and two have six user classes (i.e., midday and nighttime). The Cube Voyager PATHLOAD command is used to perform a traffic assignment, i.e., to load trips to a minimum-impedance path. For each of the traffic assignments, the number of PATHLOAD statements corresponds to the number of user classes (five or six, depending on the assignment). To perform a multi-user class assignment in Cube Voyager, a script must follow two steps:

1. First, in the LINKREAD phase, assign one or more links to a user group. To do this, one primarily uses the ADDTOGROUP (or ADDTOGRP) command, which sets group codes for a link.<sup>154</sup> Generally, one also makes use of link codes that indicate which vehicles are allowed or limited, such as our link LIMIT codes, whose values are shown in Table 94.
2. Second, when performing the traffic assignment with the PATHLOAD statement, one can then specify which groups are to be excluded from the traffic assignment.

Table 94 Link limit code, traffic assignment add group, and its meaning

Link Limit Code	Link Add Group	Definition
1	1	All vehicles accepted
2	2	Only HOV2 (or greater) vehicles accepted
3	3	Only HOV3 (or greater) vehicles accepted

<sup>154</sup> ADDTOGROUP is a subkey word of SETGROUP, although the key word SETGROUP does not need to appear in the script.

Link Limit Code	Link Add Group	Definition
4	4	Medium and heavy trucks are not accepted, but all other traffic is accepted
5	5	Airport passenger vehicle trips
6-8	6	(Unused)
9	7	No vehicles are accepted

So, for example, links that should be restricted to HOV2+ traffic can be added to group 2:

```
PHASE=LINKREAD
    IF (LI.@PRD@LIMIT==2) ADDTOGROUP=2
```

Then, when performing the traffic assignment with the PATHLOAD statement for HOV2+ trips, one can use the EXCLUDEGROUP command like this:

```
PATHLOAD PATH=LW.HV2@PRD@IMP, EXCLUDEGROUP=3,5,6,7, VOL[2]=MI.1.2 ; HOV 2
```

This means that HOV2 trips are excluded from using links that have been added to link groups 3 (HOV3+), 5 (airport passenger vehicles), 6 (unused), and 7 (unused).

### 23.3.4 Volume-delay functions

The Version 2.4 family of models use 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>155</sup>

### 23.3.5 Convergence of user equilibrium traffic assignment

When the traffic assignment process is run, the script creates a series of user equilibrium convergence report files, as shown in Table 95. 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.

---

<sup>155</sup> Milone et al., "Calibration Report for the TPB Travel Forecasting Model, Version 2.3."

Table 95 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

### 23.3.6 Loaded link highway network

Table 96 provides further details regarding the attributes of the final loaded highway network.

Table 96 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 LENG	Geometry length of network link (in feet)
PROJECTID	Project identifier
TAZ	Nearest TAZ centroid to midpoint of link (1-3,722)
ATYPE	Area Type (1-6)
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>	AM Peak Period (6:00-9:00 AM)
<Period>= AM	
MD	Mid-Day (9:00 AM - 3:00 PM)
PM	PM Peak Period (3:00 - 7:00 PM)
NT	All remaining hours

\* The link VMT does not account for centroid connector links (Ftype=0). Centroid connectors represent minor, local roadways that are not coded in the planning networks. This essentially means that our travel demand model simulates only average weekday “non-local roadway” VMT, or “network” VMT as we sometimes refer to it (i.e., Total Roadway VMT= Local Roadway VMT + Non-Local Roadway VMT, and the model only simulates the Non-Local Roadway VMT component). Please note that “local” in this case refers to the roadway type and not the trip origin or destination.

### 23.3.7 Averaging of link volumes

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.

### 23.3.8 Treatment of airport passenger auto driver trips on HOV and HOT lane facilities

Text for this section of the report come from or are derived from a recent memo on this subject.<sup>156</sup>

The terms “airport passenger trips” or “air passenger trips” refer to a motor vehicle carrying air passengers to or from one of the three commercial airports in the region: Reagan National (DCA), Dulles International (IAD), and Baltimore-Washington International (BWI). The focus is on highway assignment, not mode choice or transit assignment.

#### 23.3.8.1 Real world conditions

Regarding the use of HOV- and HOT-lane facilities by motor vehicles carrying passengers to the region’s three commercial airports (DCA, IAD, and BW), several real-world issues that make it challenging to reflect the corresponding occupancy restrictions in the travel model. There are many different HOV and HOT facilities with different restrictions on their use, as shown in Table 97.

Table 97 HOV and HOT-lane facilities in the Washington, D.C. area

Type of Facility	Use Restrictions	Examples
HOV2+	Vehicles must have two or more occupants (certain exemptions apply, including an airport-related exemption)	I-270, I-66, US 50 (MD)
HOV3+	Vehicles must have three or more occupants.	I-395, I-95 (VA)
HOT2+	Vehicles with two or more occupants can use the facility for free. Vehicles with one occupant may pay to use the facility. Users of the facility must have either an “E-Zpass” OR “E-ZPass Flex” tag/RFID transponder in vehicle. Users who want to gain free access to the facility due to meeting the occupancy requirement must have an “E-ZPass Flex” transponder.	I-66 Inside the Beltway starting in 2017
HOT3+	Vehicles with three or more occupants can use the facility for free. Vehicles with one or two occupants may pay to use the facility. Users of the facility must have either an “E-Zpass” OR “E-ZPass Flex” tag/RFID transponder in vehicle. Users who want to gain free access to the facility due to meeting the occupancy requirement must have an “E-ZPass Flex” transponder.	I-495 (VA), I-395 starting in 2019

Prior to 2017, when the I-66 inside the Beltway HOT lanes opened, there was an exception to the HOV occupancy restrictions regarding auto trips to/from one of the three commercial airports in the region: Motorists traveling to and from Dulles International Airport to board a flight or to pick someone up at

<sup>156</sup> Dusan Vuksan, Dzung Ngo, and Mark S. Moran, “Air Passenger Trips on HOV/HOT Lanes in the TPB Version 2.3 Travel Model: Discussion of Current Treatment and Recommendations for Modifications,” Memorandum, April 24, 2017.

the airport were permitted to use I-66 inside the Beltway (I-495) for free during HOV hours. This exception ended when the I-66 Inside the Beltway lanes converted from HOV to HOT in December 2017.

### 23.3.8.2 Treatment in the Ver. 2.4.6 Model

Before discussing how the model handles air passenger travel on HOV and HOT-lane facilities, this section of the report discusses the general way in which air passenger trips are handled in the travel demand model. Although air passenger travel on the road network is handled by the travel model, it is considered an exogenous input to the model. Other exogenous inputs to the travel model include taxi trips, visitor/tourist trips, school trips, through trips, and external trips. As noted in a recent memo:<sup>157</sup>

*The airport passenger auto driver trip tables are prepared based on base- and future-year trip tables that are developed as part of COG's Continuous Air System Planning (CASP) activities. The trip tables indicate local originations to the region's three major airports with commercial air service:*

- Ronald Reagan Washington National Airport (DCA)
- Washington Dulles International Airport (IAD)
- Baltimore/Washington International Thurgood Marshall Airport (BWI)

*The trip tables are stratified by mode and purpose (Home-Based and Non-Home-Base). The trip tables are developed by year and are prepared at the Airport Analysis Zone (AAZ) level of geography (p. 17).*

Airport passenger trips are stored in a binary trip-table file called airpax.adr. The current air passenger auto driver trip tables were developed using the COG 2011 Regional Air Passenger Survey. The 2013 and 2015 surveys were not used because they were conducted during a full and partial shutdown of the federal government. Although the most recent Airport Passenger Survey was conducted in 2017 without any issue, the reason that this survey was not used to develop the air passenger auto driver trip tables is explained in the previously mentioned memo as follow:

*Compared to the 2011 APS, the 2017 APS forecasts significant drops of auto driver travel (about 35%) and high increases in auto passenger travel. The changes are partly due to the shifts of trips from some other modes, including personal car and taxi, to Uber and Lyft, which did not exist in the 2011 APS. We acknowledge the importance of using the latest data available, but, in this case, we have decided to use the ground access trips from 2011 APS for processing airport passenger auto driver trip tables because:*

1. *In 2017 APS, Uber/Lyft trips were processed as auto passenger trips and the distinction of Uber/Lyft from other auto passenger trip modes requires the revision of SAS scripts used for processing the raw data. Unfortunately, the current budget does not allow this update.*

---

<sup>157</sup> Ray Ngo to DTP Technical Staff and Mark Moran, "Exogenous Demand Inputs to the TPB Travel Demand Model: Update for Round 9.1a Cooperative Forecasts," Memorandum, September 25, 2019.



*2. The total ground access trips to the three airports has a minor change between the two surveys. For example, the total trip forecast in the 2017 APS decreases 1% compared to the forecast in 2011 APS.*

In terms of the treatment of air passenger trips on HOV and HOT-lane facilities, the Ver. 2.3.70 model (and earlier) allowed airport trips to use any HOV facility regardless of the vehicle occupancy. For the HOV2+ lanes on I-66 Inside the Beltway, **this makes sense**, given the current policy for I-66 mentioned earlier. For other HOV facilities in the region, however, the model's representation does not reflect the real transportation system. For these other HOV facilities, vehicles carrying air passengers should be allowed to use the other HOV facilities only if the vehicles meet the occupancy requirements for the facility.

Table 98 shows how airport passenger trips are treated with respect to HOV facilities in the real world, the Ver. 2.3.66 model (and later, including the Ver. 2.4 Model). The three areas **highlighted in yellow** show where there was a mismatch between the real world and the Ver. 2.3.66 model.

Table 98 Use of HOV and HOT-lane facilities by autos serving airport passengers: Real world, Ver. 2.3.66 Model (and later, including Ver. 2.4.6 Model)

Auto Serving Airport Passenger (no. of occupants)	Case	HOV Facility			HOT-Lane Facility		
		2+	2+ I-66 Inside Beltway	3+	2+	2+ I-66 Inside Beltway	3+
<b>1 occupant</b>	<b>Real World</b>	No	Yes, VDOT exemption (1)	No	Yes (pay toll)*	Yes (pay toll)* (2)	Yes (pay toll)*
	<b>V. 2.3.66 Model</b>	Yes	Yes	Yes	Yes (pay toll)	Yes (pay toll)	Yes (pay toll)
	<b>V. 2.3.75 Model</b>	Yes	Yes	No	Yes (pay toll)	Yes (pay toll)	Yes (pay toll)
<b>2 occupants</b>	<b>Real World</b>	Yes	Yes	No	Yes (free)**	Yes (free)**	Yes (pay toll)*
	<b>V. 2.3.66 Model</b>	Yes	Yes	Yes	Yes (pay toll)	Yes (pay toll)	Yes (pay toll)
	<b>V. 2.3.75 Model</b>	Yes	Yes	No	Yes (pay toll)	Yes (pay toll)	Yes (pay toll)
<b>3+ occupants</b>	<b>Real World</b>	Yes	Yes	Yes	Yes (free)**	Yes (free)**	Yes (free)**
	<b>V. 2.3.66 Model</b>	Yes	Yes	Yes	Yes (pay toll)	Yes (pay toll)	Yes (pay toll)
	<b>V. 2.3.75 Model</b>	Yes	Yes	No	Yes (pay toll)	Yes (pay toll)	Yes (pay toll)

(1): Per VDOT: “Motorists traveling to and from Dulles International Airport to go to the airport to board a flight or to pick someone up at the airport are permitted to use I-66 inside the Beltway (I-495) during HOV hours. Motorists traveling to or from Dulles International Airport are not exempt from HOV restrictions on I-66 outside the Beltway (I-495). You are not permitted to use I-66 inside the beltway during HOV hours if you are going to the airport to eat, get coffee, get gas or any other reason other than boarding a plane or picking someone up at the airport.” (Source: “High Occupancy Vehicle (HOV) Lanes - Rules and FAQs.” *Virginia Department of Transportation*, February 1, 2017. <http://www.virginiadot.org/travel/hov-rulesfaq.asp>).

(2): Exemption for travelers to/from Dulles Airport will end when the I-66 Inside the Beltway HOT lanes open (Source: Thomson, Robert. “As Virginia Sets up I-66 HOT Lanes, Drivers Again Ask: What about Me?” *The Washington Post*. August 16, 2016. <https://www.washingtonpost.com/news/dr-gridlock/wp/2016/08/11/as-virginia-sets-up-i-66-hot-lanes-drivers-again-ask-what-about-me/>)

\* Must have an “E-Zpass” OR “E-ZPass Flex” tag/RFID transponder in vehicle.

\*\* Must have an “E-ZPass Flex” tag/RFID transponder in vehicle.

The only change made to the travel model (as of Ver. 2.3.75) regarding modeling airport trips was to prohibit airport trips from using HOV3+ facilities, since observed car occupancy for these types of trips is approximately 1.6 persons per vehicle, which is calculated from the Average Weekday Air Passenger Ground Access trip data documented in the 2013 Washington – Baltimore Regional Air Passenger Survey. These three areas are highlighted in green in Table 98.



## 24 Transit Assignment, Including Summary Process (LineSum)

### 24.1 Transit assignment process

Transit assignment is a new feature of the Version 2.3 and Version 2.4 Travel Models 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 20 and Figure 21).

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), 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 99 and the outputs are shown in Table 100.

Table 99 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 100 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 101.

**Table 101 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 52.

	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 52 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>158</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

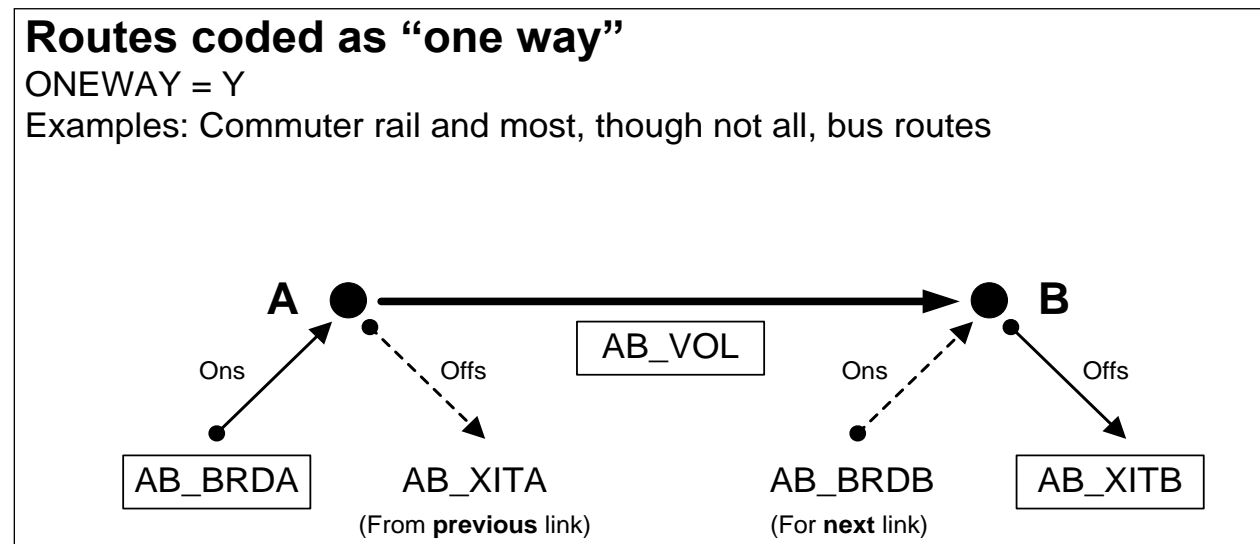
<sup>158</sup> Citilabs, Inc., "Cube Voyager Reference Guide, Version 6.0.2" (Citilabs, Inc., July 26, 2012), 958.



- 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 53 and Figure 54 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 53 is for the case of a one-way transit route, and Figure 54 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 53 Transit volumes from transit assignment using TRNBUILD: One-way route



Note: For a description of AB\_VOL, AB\_BRDA, AB\_XITA, etc., see page 1020, Cube Voyager Reference Guide, Version 6.4.1 Citilabs, Inc., September 30, 2015.

The simplest case is the one-way route (Figure 53). 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 53. The other two values shown in Figure 53 (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 54), 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)

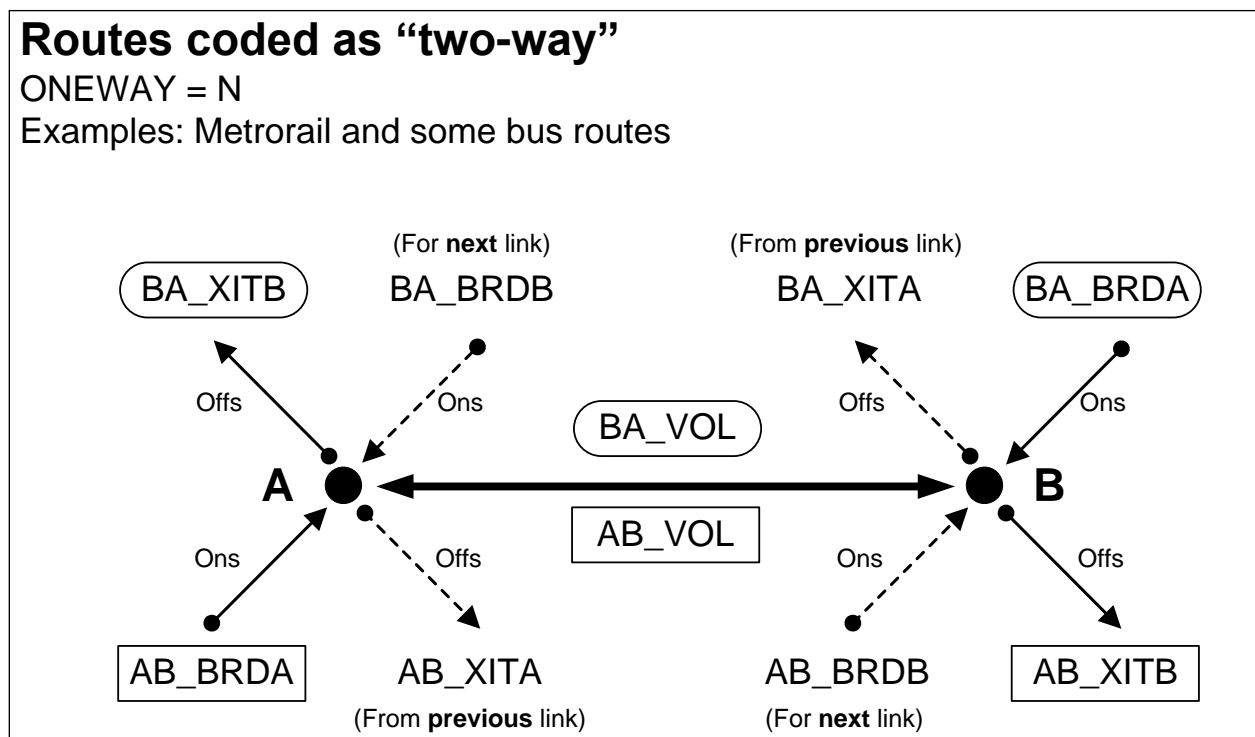
- AB\_XITB: Transit person alightings (“offs”) from the “to” node (node B in the figure)

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>159</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 54 with rectangular boxes that have rounded corners.

Figure 54 Transit volumes from transit assignment using TRNBUILD: Two-way route



Note: For a description of AB\_VOL, AB\_BRDA, AB\_XITA, etc., see page 1020, Cube Voyager Reference Guide, Version 6.4.1 Citilabs, Inc., September 30, 2015.

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>159</sup> Typically, the convention is that the “from” node is the A node and the “to” node is the B node. However, in Figure 54, 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>160</sup>

Some examples of the LINKO attribute values can be found in Figure 55 through Figure 57. For example, Figure 55 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 56 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 57 shows a portion of the AM walk-access to Metrorail LINKO file (i4\_WKMRAMlink.dbf) that has mode-12 links (walk transfer links).

---

<sup>160</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	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	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	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	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	0

Figure 55 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 56 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 57 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., 2019), the transit assignment summary process is run in the **transum** folder (e.g., 2019\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>161</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 58 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., 2019 \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 and subsequent models). 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.ct1
lineSum_MR_access.ct1
lineSum_MR_line.ct1
```

These files are described below, and the model user can always develop more control files to generate more reports.

<sup>161</sup> AECOM, *LineSum*, version 6.0.2 (Arlington, Virginia: AECOM, 2014).

At the beginning of the transum.bat batch file (line 8 in Figure 58), 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 58, 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 58 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 59) 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 59 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_KRBMAMlink.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 60 through Figure 64.



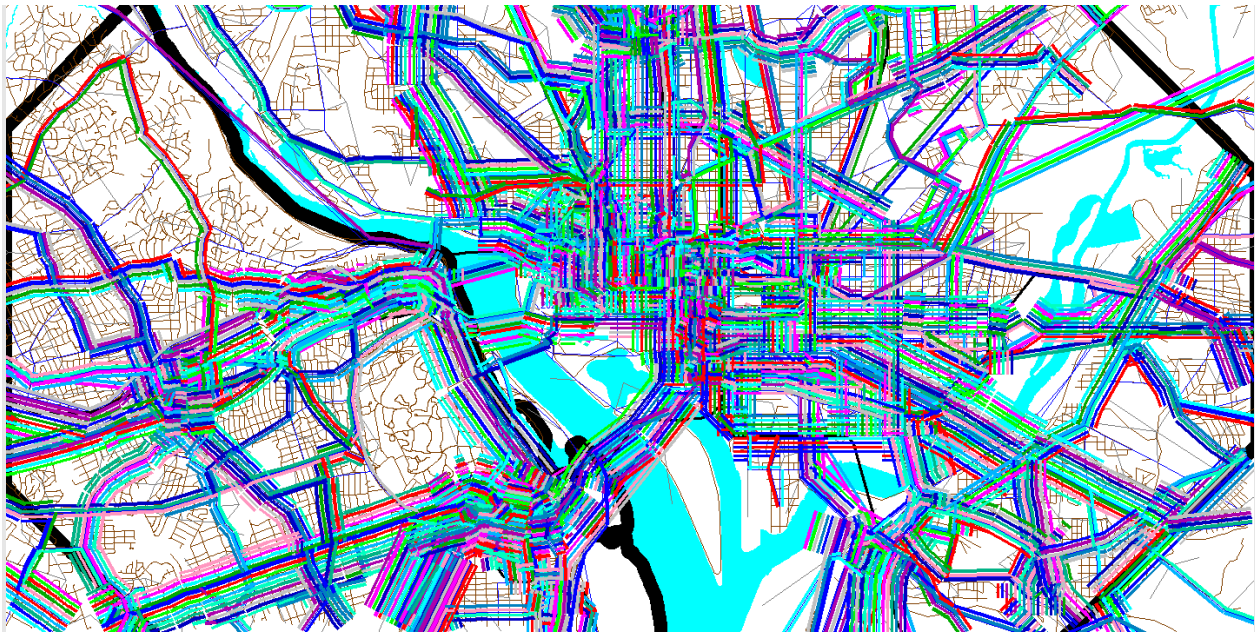


Figure 60 Using the pk\_vol.dbf file in Cube Base as the transit layer: All transit routes turned on, but non-transit links (modes 11-16) turned off

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

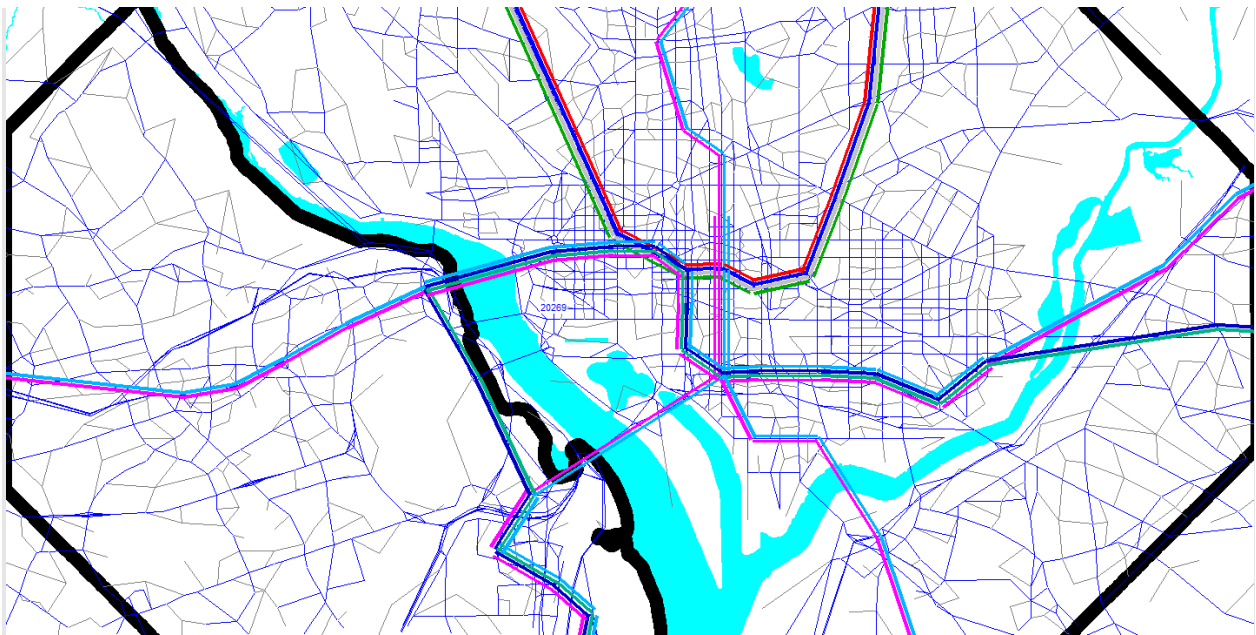


Figure 61 Using the pk\_vol.dbf file in Cube Base as the transit layer: Only mode-3 (Metrorail) links turned on

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

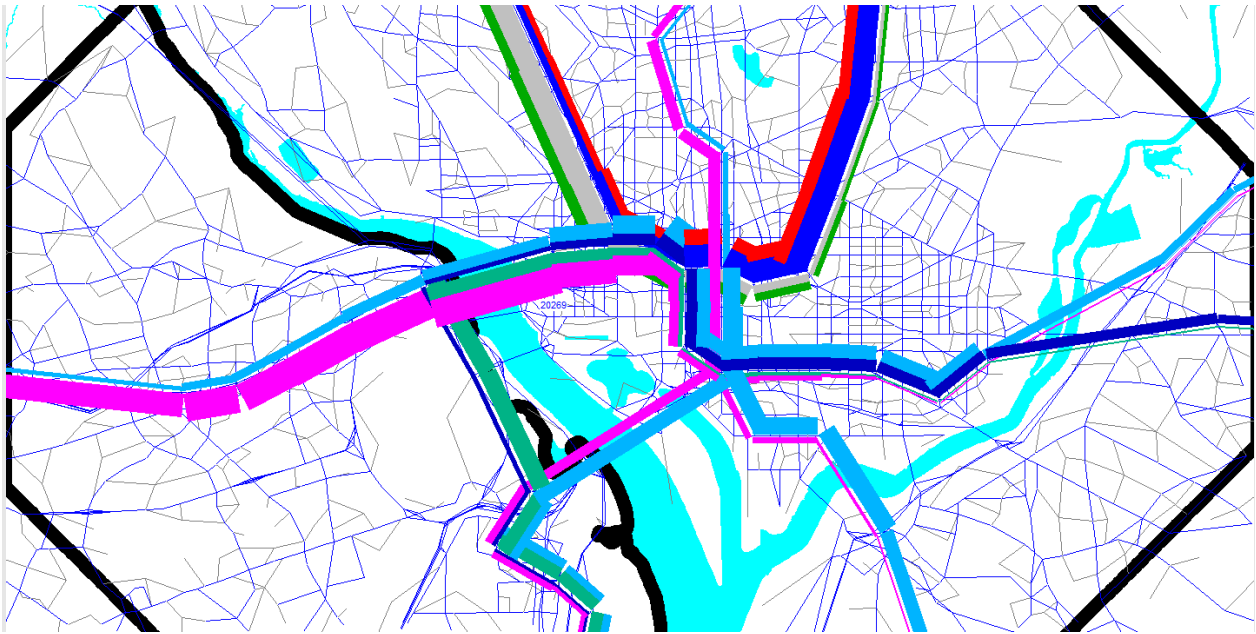


Figure 62 Using the pk\_vol.dbf file in Cube Base as the transit layer: Only mode-3 (Metrorail) links turned on; using multi-bandwidth to represent transit loads (ab\_vol): Arlington and DC

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

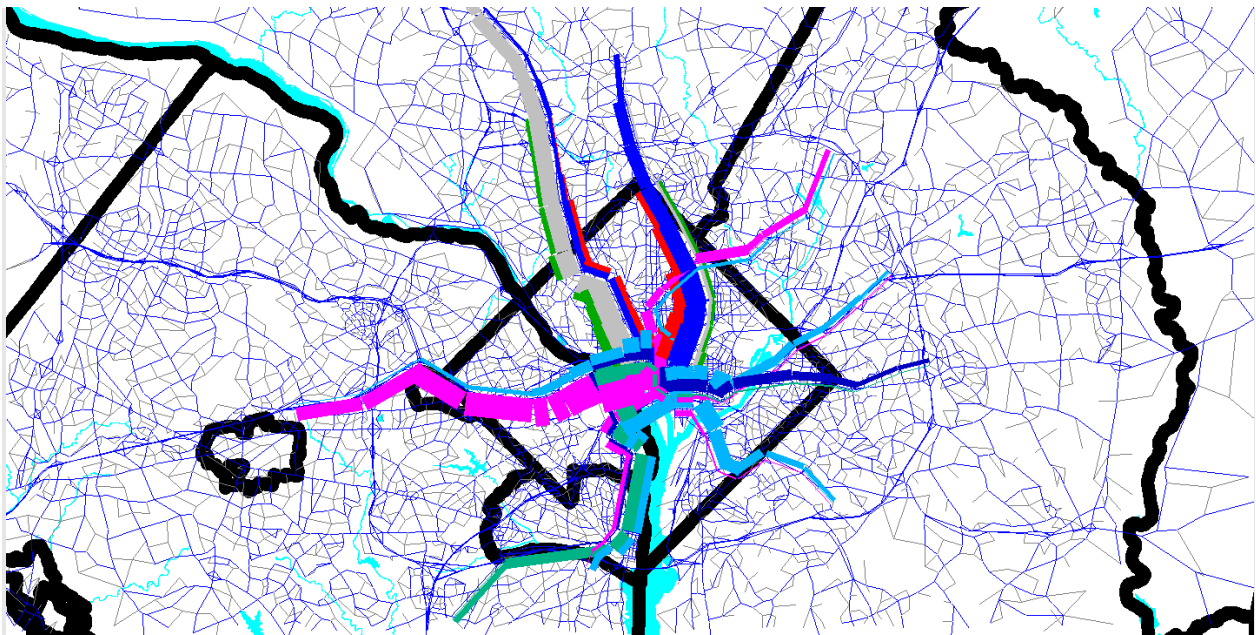


Figure 63 Using the pk\_vol.dbf file in Cube Base as the transit layer: Only mode-3 (Metrorail) links turned on; using multi-bandwidth to represent transit loads (ab\_vol): Metrorail system

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 type="text"/>	<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 64 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 65. The station\_names.dbf file is now stored in the “inputs” folder (not the “controls”) folder.

Figure 65 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 66. Once again, the station\_names.dbf file is now stored in the “inputs” folder (not the “controls”) folder.

Figure 66 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 67. Similarly, an example of the report generated by the lineSum\_MR\_\*\*line\*\*.ctl control file can be found Figure 68.

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 67 An excerpt from the report file generated by lineSum\_MR\_access.ct1

```

*****
|                                     |
|      LineSum - Version 6.0.2       |
|      Copyright 2014 by TRANSIMS Open-Source |
|      Tue Sep 11 04:41:33 2018     |
|                                     |
*****

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 = 446  Metrorail Station Access Summary
Tue Sep 11 04:41:34 2018  LineSum  page 2

Title: All
Modes: All

      ---- Peak ----  -- Offpeak ---  ---- Daily ---
Stop  Arrive Depart Arrive Depart Arrive Depart
8001  34719  2676  2252  705  36971  3381  Shady Grove
8002  12771  2899  1209  865  13980  3764  Rockville
8003  5500  4230  947  1400  6447  5630  Twinbrook
8004  6462  7816  1273  2706  7735  10522  White Flint
8005  9806  301  2003  587  11809  888  Grosvenor
8006  4902  7237  1085  1584  5987  8821  Medical Center
    
```

8007	15291	19297	4879	6070	20170	25367	Bethesda
8008	10843	6735	3402	2916	14245	9651	Friendship Heights
8009	10247	5849	2938	1708	13185	7557	Tenleytown
8010	5501	3360	1946	1368	7447	4728	Van Ness-UDC
8011	4685	889	1539	760	6224	1649	Cleveland Park
8012	8082	2629	2660	1537	10742	4166	Woodley Park-Zoo
8013	10939	30437	1946	6301	12885	36738	Dupont Circle
8014	3399	35046	1389	3722	4788	38768	Farragut North
8015	691	31251	295	4081	986	35332	Metro Center
8016	490	20397	707	3286	1197	23683	Gallery Place
8017	251	14986	212	1310	463	16296	Judiciary Square
8018	29588	39035	5116	6321	34704	45356	Union Station

Figure 68 The report file generated by lineSum\_MR\_line.ct1

```

*****
|                                     |
|      LineSum - Version 6.0.2        |
|      Copyright 2012 by TRANSIMS Open-Source |
|      Tue Sep 11 04:41:34 2018      |
|                                     |
*****

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
Tue Sep 11 04:41:35 2018      LineSum page 2

Title: All
Lines: All
Modes: 3

-----A->B Direction (Read Down) ----- B->A Direction (Read Up) -----Total-----
-----Peak----- -----Off-Peak----- -----Daily----- -----Peak----- -----Off-Peak----- -----Daily----- -----Daily-----
Stop      Dist  Time  On  Off  Ride  On  Off  Ride  On  Off  Ride  On  Off  Ride  On  Off  Ride  On  Off  Ride
(miles) (min)
Franconia- 3.49  6.29 12346  0 12346 1380  0 1380 13726  0 13726  0 2446 2446  0 602 602  0 3048 3048 13726 3048 16774
Van Dorn S 3.86  5.08 8410 174 20584 1997  59 3318 10407 233 23902 466 831 2810 76 704 1231 542 1535 4041 10949 1768 27943
King Stree 0.68  2.07 4386 1666 48952 1075 847 7331 5461 2513 56283 549 2619 11717 267 1248 5521 816 3867 17238 6277 6380 73521
Braddock R 1.21  1.98 5789 1541 53201 1657 1113 7872 7446 2654 61073 707 2841 13853 658 2133 6996 1365 4974 20849 8811 7628 81922
Potomac Ya 1.82  2.98 10851 2311 61738 3073 1448 9498 13924 3759 71236 1306 3903 16445 2000 1930 6928 3306 5833 23373 17230 9592 94609
National A 0.49  2.65  0 1198 60540 371 173 9695 371 1371 70235 0 2675 19124 73 686 7540 73 3361 26664 444 4732 96899
Crystal Ci 0.76  2.07 5140 5462 60219 3087 1532 11249 8227 6994 71468 949 17358 35532 1047 5514 12007 1996 22872 47539 10223 29866 119007
Pentagon C 0.61  1.01 8973 5450 63740 2039 825 12463 11012 6275 76203 2029 5116 38617 1118 2343 13230 3147 7459 51847 14159 13734 128050
Pentagon 1.24  2.99 2336 5842 18109 2063 1785 8540 4399 7627 26649 2693 2568 11286 1634 1278 5934 4327 3846 17220 8726 11473 43869
Arlington 0.99  2.14 105 0 18213 75 0 8615 180 0 26828 36 0 11249 25 0 5910 61 0 17159 241 0 43987
Rosslyn 1.35  3.19 5710 13813 88316 1830 6291 14363 7540 20104 102679 11195 14723 47980 5065 4051 13410 16260 18774 61390 23800 38878 164069
Foggy Bott 0.57  2.14 2148 11054 79410 1394 2638 13119 3542 13692 92529 814 29872 77039 841 5620 18193 1655 35492 95232 5197 49184 187761
Farragut W 0.38  0.99 1229 13797 66840 466 1567 12014 1695 15364 78854 834 24187 100391 354 2825 20660 1188 27012 121051 2883 42376 199905

```



McPherson	0.46	1.11	4155	11702	59294	1362	2010	11367	5517	13712	70661	2426	36361	134327	964	4082	23773	3390	40443	158100	8907	54155	228761
Metro Cent	0.29	0.94	18835	36658	41472	3476	5778	9065	22311	42436	50537	66870	22588	90043	10239	5363	18896	77109	27951	108939	99420	70387	159476
Federal Tr	0.41	2.15	0	6810	34665	30	805	8289	30	7615	42954	70	1264	91236	104	284	19078	174	1548	110314	204	9163	153268
Smithsonia	0.59	2.34	389	8833	26224	191	1748	6731	580	10581	32955	1206	2399	92434	577	759	19261	1783	3158	111695	2363	13739	144650
L'Enfant P	0.33	1.99	12507	12401	26326	3579	3179	7132	16086	15580	33458	46959	21081	66557	7352	6982	18895	54311	28063	85452	70397	43643	118910
Federal Ce	0.57	1.96	53	9530	16851	173	1854	5453	226	11384	22304	990	1976	67546	984	722	18632	1974	2698	86178	2200	14082	108482
Capitol So	0.50	1.99	70	8924	8000	248	1551	4149	318	10475	12149	811	3214	69946	903	985	18713	1714	4199	88659	2032	14674	100808
Eastern Ma	0.63	2.02	160	2580	5578	368	928	3587	528	3508	9165	3809	2248	68386	1950	539	17301	5759	2787	85687	6287	6295	94852
Potomac Av	0.66	0.99	413	1101	4892	268	815	3041	681	1916	7933	9501	434	59320	3706	406	14006	13207	840	73326	13888	2756	81259
Stadium Ar	2.69	3.17	687	515	2528	299	779	1410	986	1294	3938	5186	921	32604	1710	415	8440	6896	1336	41044	7882	2630	44982
Benning Ro	1.42	2.90	266	746	2045	201	608	1004	467	1354	3049	5890	376	27091	2586	362	6215	8476	738	33306	8943	2092	36355
Capitol He	0.97	2.95	119	477	1687	62	429	639	181	906	2326	5397	97	21790	2818	95	3491	8215	192	25281	8396	1098	27607
Addison Ro	1.77	3.13	155	327	1515	44	121	560	199	448	2075	8360	54	13485	1234	28	2282	9594	82	15767	9793	530	17842
Morgan Blv	1.23	2.78	141	466	1189	40	142	459	181	608	1648	3707	45	9823	637	34	1678	4344	79	11501	4525	687	13149
Largo Town				1189			459			1648		9823			1678			11501		11501	1648		
Greenbelt	2.44	2.88	15152	0	15152	1417	0	1417	16569	0	16569	0	1155	1155	0	225	225	0	1380	1380	16569	1380	17949
College Pa	1.94	3.02	5085	694	19543	1384	171	2630	6469	865	22173	210	3534	4476	54	1119	1289	264	4653	5765	6733	5518	27938
PG Plaza	1.24	3.14	5513	687	24364	1339	222	3749	6852	909	28113	481	2172	6163	178	487	1598	659	2659	7761	7511	3568	35874
West Hyatt	1.99	2.92	4343	140	28567	1998	245	5502	6341	385	34069	582	359	5943	251	302	1649	833	661	7592	7174	1046	41661
Fort Totte	1.62	2.89	6778	17364	17977	2895	3176	5221	9673	20540	23198	3378	3340	5902	920	1685	2413	4298	5025	8315	13971	25565	31513
Georgia Av	0.86	3.11	8768	3350	23396	2616	1117	6720	11384	4467	30116	2434	3916	7384	998	1168	2582	3432	5084	9966	14816	9551	40082
Columbia H	0.95	2.02	11346	1047	33692	2676	622	8772	14022	1669	42464	1029	2223	8580	495	1191	3280	1524	3414	11860	15546	5083	54324
U-Street-C	0.51	2.05	5194	2931	35959	2242	979	10038	7436	3910	45997	492	6235	14323	341	2402	5340	833	8637	19663	8269	12547	65660
Shaw-Howar	0.56	1.13	3186	685	38460	1462	380	11121	4648	1065	49581	734	2174	15764	333	1064	6070	1067	3238	21834	5715	4303	71415
Mt Vernon	0.49	1.66	2096	2600	37958	1203	1082	11242	3299	3682	49200	158	11171	26774	211	2561	8418	369	13732	35192	3668	17414	84392
Gallery Pl	0.36	1.92	37196	19641	55514	5690	5039	11892	42886	24680	67406	13001	45140	58913	2840	7040	12619	15841	52180	71532	58727	76860	138938
Archives	0.58	1.97	2411	3268	54655	156	656	11391	2567	3924	66046	4679	10749	64983	44	1454	14028	4723	12203	79011	7290	16127	145057
L'Enfant P	0.79	1.91	14484	21213	30763	2656	3917	6073	17140	25130	36836	10228	39128	66117	3302	11376	19284	13530	50504	85401	30670	75634	122237
Waterfront	0.59	1.80	816	5192	26386	657	911	5817	1473	6103	32203	4152	1394	63360	1867	1059	18477	6019	2453	81837	7492	8556	114040
Navy Yard	1.20	2.06	438	19198	7629	662	3025	3454	1100	22223	11083	6662	4978	61677	3773	1514	16217	10435	6492	77894	11535	28715	88977
Anacostia	1.38	2.98	369	5063	2934	470	1793	2128	839	6856	5062	14738	1087	48028	6810	466	9872	21548	1553	57900	22387	8409	62962
Congress H	0.97	1.78	85	1203	1815	113	642	1598	198	1845	3413	4877	372	43524	2138	195	7930	7015	567	51454	7213	2412	54867
Southern A	1.24	2.76	214	302	1725	112	484	1224	326	786	2949	14918	67	28674	3795	130	4266	18713	197	32940	19039	983	35889
Naylor Roa	1.48	2.34	184	619	1291	74	638	661	258	1257	1952	8961	119	19831	1317	67	3015	10278	186	22846	10536	1443	24798
Suitland	1.64	2.66	35	929	396	12	482	191	47	1411	587	4426	89	15495	1189	19	1844	5615	108	17339	5662	1519	17926
Branch Ave				396			191			587		15495			1844			17339		17339	587		
Vienna	2.39	3.69	21333	0	21333	1868	0	1868	23201	0	23201	0	1330	1330	0	596	596	0	1926	1926	23201	1926	25127
Dunn Lorin	2.49	3.99	6374	190	27519	1218	63	3023	7592	253	30542	113	1542	2757	57	762	1302	170	2304	4059	7762	2557	34601
West Falls	2.09	2.93	3279	54	30744	633	27	3629	3912	81	34373	122	342	2977	48	307	1562	170	649	4539	4082	730	38912
East Falls	2.51	3.96	12817	3272	81908	2585	835	10985	15402	4107	92893	632	1582	22154	434	1213	9219	1066	2795	31373	16468	6902	124266



User's Guide for the COG/TPB Gen2/Version 2.4.6 Travel Forecasting Model

L'Enfant P	2.35	5.15	18655	0	35818	5988	0	10048	24643	0	45866	0	39343	67109	0	2883	5702	0	42226	72811	24643	42226	118677
Pentagon			8612			3110			11722			24983		1501		26484			26484		26484	11722	
King Stree	0.64	1.89	1624	0	8459	573	0	3882	2197	0	12341	0	2425	28076	0	557	4341	0	2982	32417	2197	2982	44758
Eisenhower	0.55	1.08	119	6339	2238	124	2682	1323	243	9021	3561	4196	1355	25234	1363	783	3761	5559	2138	28995	5802	11159	32556
Huntington			2238			1323			3561		25234			3761		28995			28995		28995	3561	
Total	130.90	261.70		575826		136451		712277		633113		162008		795121		1507398							

Metrorail Line Summary

Tue Sep 11 04:41:35 2018 LineSum page 4

Stop	Dist (miles)	Time (min)	A->B Direction (Read Down)									B->A Direction (Read Up)									-----Total-----		
			-----Peak-----			-----Off-Peak-----			-----Daily-----			-----Peak-----			-----Off-Peak-----			-----Daily-----			On	Off	Ride
Max	5.83	7.70	37196	36658	96369	6140	6291	15636	42886	42436	112005	66870	67635	134327	10239	11376	23773	77109	78410	158100	99420	116626	228761
Passenger Miles			3932608			612906			4545514			3305303			732957			4038260			8583774		
Passenger Hours			141005			23187			164192			129617			28892			158509			322701		
Average Trip Length (miles)			6.8			4.5			11.3			5.2			4.5			9.7			21.1		
Average Trip Length (minutes)			14.7			10.2			24.9			12.3			10.7			23.0			47.9		

Tue Sep 11 04:41:35 2018 -- Process Complete with 9 Warnings (0:00:01)



## Appendix A. Flowcharts

Ref: Ver2.4.6\_flowchart.vsd

### Flowchart numbers associated with flowchart steps

This appendix contains a flowchart showing the flow of data through the TPB regional travel demand forecasting model (Ver. 2.4.6). The flowcharts are arranged on the basis of the 19 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 speed-feedback iteration. The flowcharts are numbered as indicated in the table below.

Version 2.4.6 - Flowchart Numbers Associated with Flowchart Steps

Batch File	Initial (Pump Prime) Iteration				
	PP	1	2	3	4
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		
Average_Link_Speeds.bat				11	
Highway_Skims.bat				12	
Transit_Fare.bat				13	
Mode_Choice_Parallel.bat				14	
Auto_Driver.bat				15	
Transit_Assignment_Parallel.bat					
Transit_Assignment_LineHaul_Parallel.bat					16
Transum.bat					17-18

Ref: Appendix\_A\_UB\_flowchart\_table\_v2.4.6.xlsx





**TITLE: Flowchart for the TPB trip-based, regional travel demand forecasting model, Ver. 2.4**

**COMPANY: COG/TPB**

Authors RJM, MS, MSM, RN,SKO

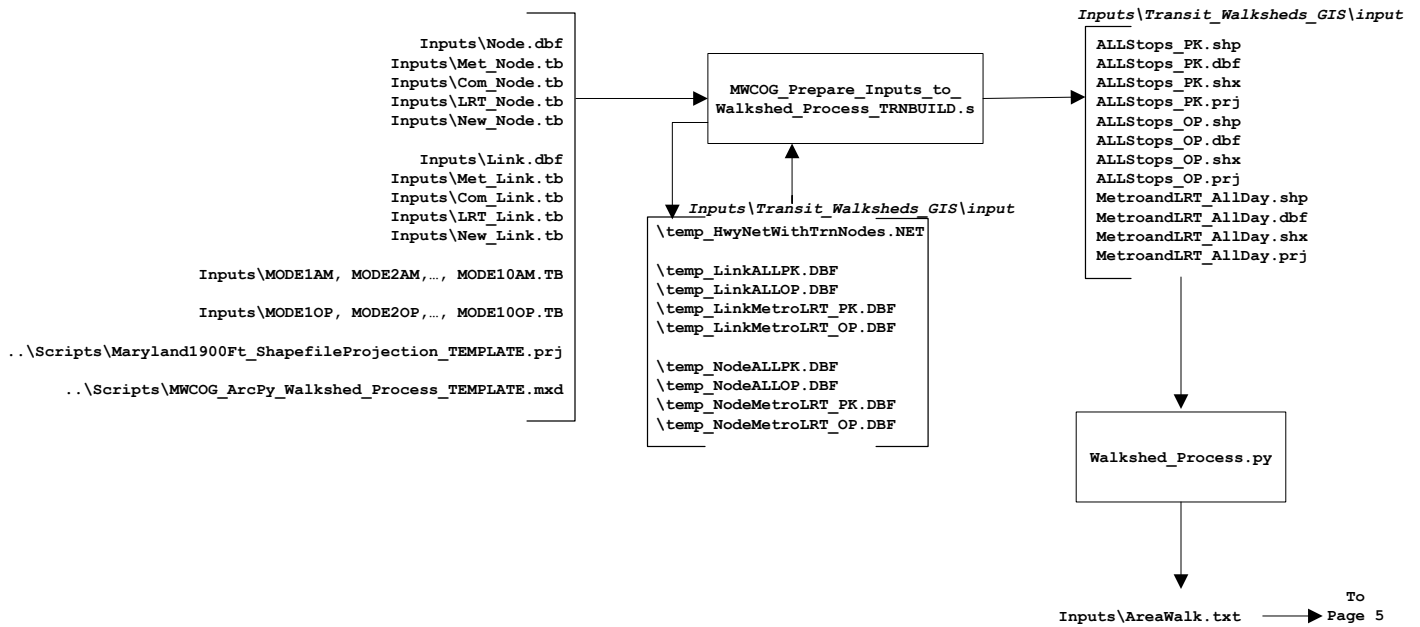
DATE: 5/15/2023

PG: 1

OF 18

FILENAME: Ver2.4.6\_flowchart.vsd

## Walkshed\_Process.bat





**TITLE: Flowchart for the TPB trip-based, regional travel demand forecasting model, Ver. 2.4**

**COMPANY: COG/TPB**

Authors RJM, MS, MSM, RN,SKO

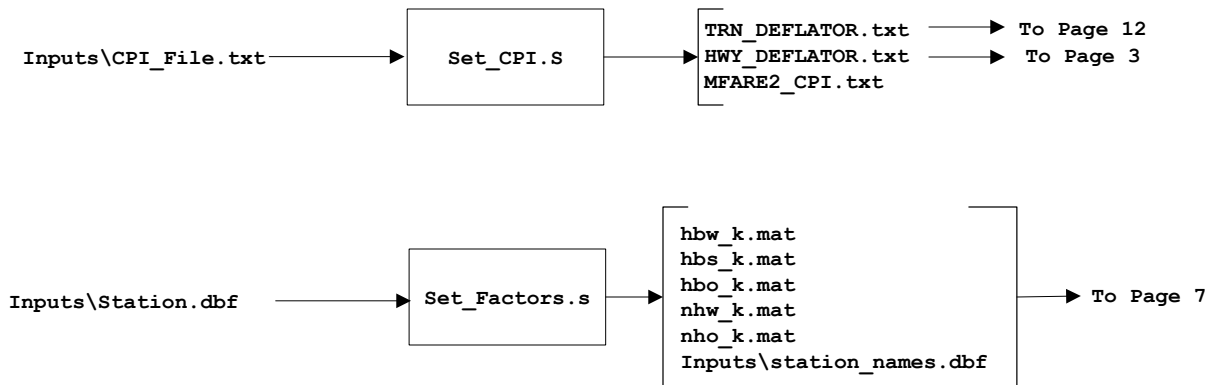
DATE: 5/15/2023

PG: 2

OF 18

FILENAME: Ver2.4.6\_flowchart.vsd

### Set\_CPI.bat: Develop CPI and K-factors



Report Files Generated by Set\_CPI.bat:

Set\_CPI.rpt  
Set\_Factors.rpt





**TITLE: Flowchart for the TPB trip-based, regional travel demand forecasting model, Ver. 2.4**

**COMPANY: COG/TPB**

**Authors RJM, MS, MSM, RN,SKO**

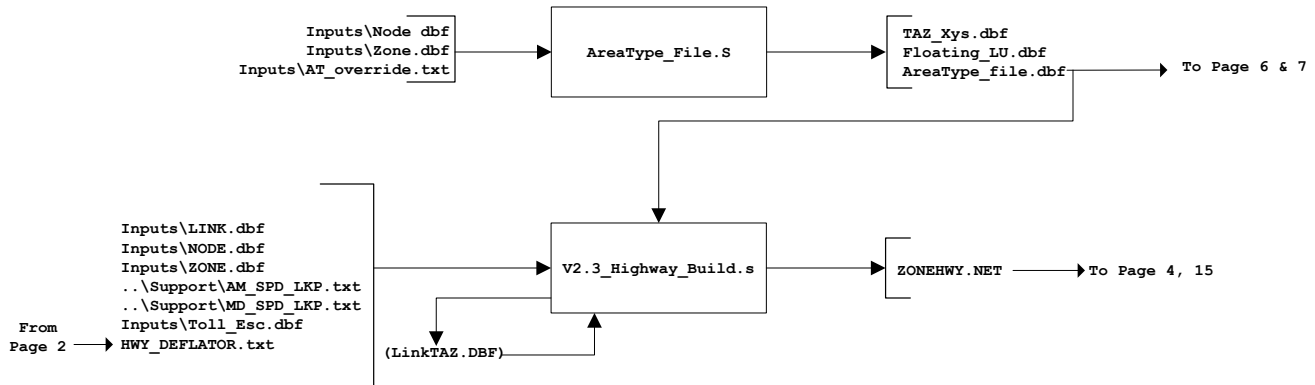
**DATE: 5/15/2023**

**PG: 3**

**OF 18**

**FILENAME: Ver2.4.6\_flowchart.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

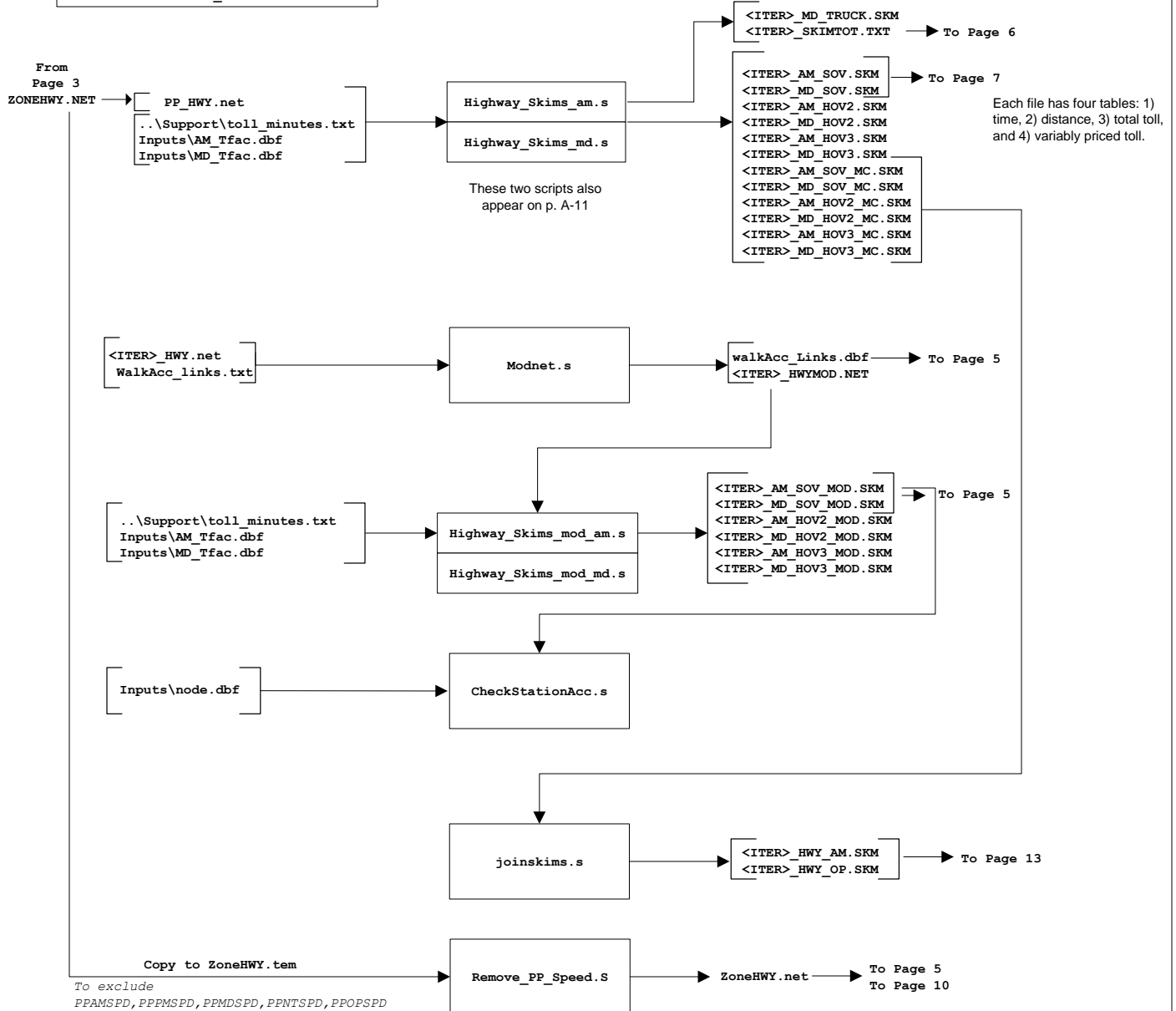


## PP\_Highway\_Skims.bat

(See also page A-11 for highway skimming process used in speed feedback iterations 1-4)

### PP\_Highway\_Skims.bat

COPY ZONEHWY.NET PP\_HWY.NET

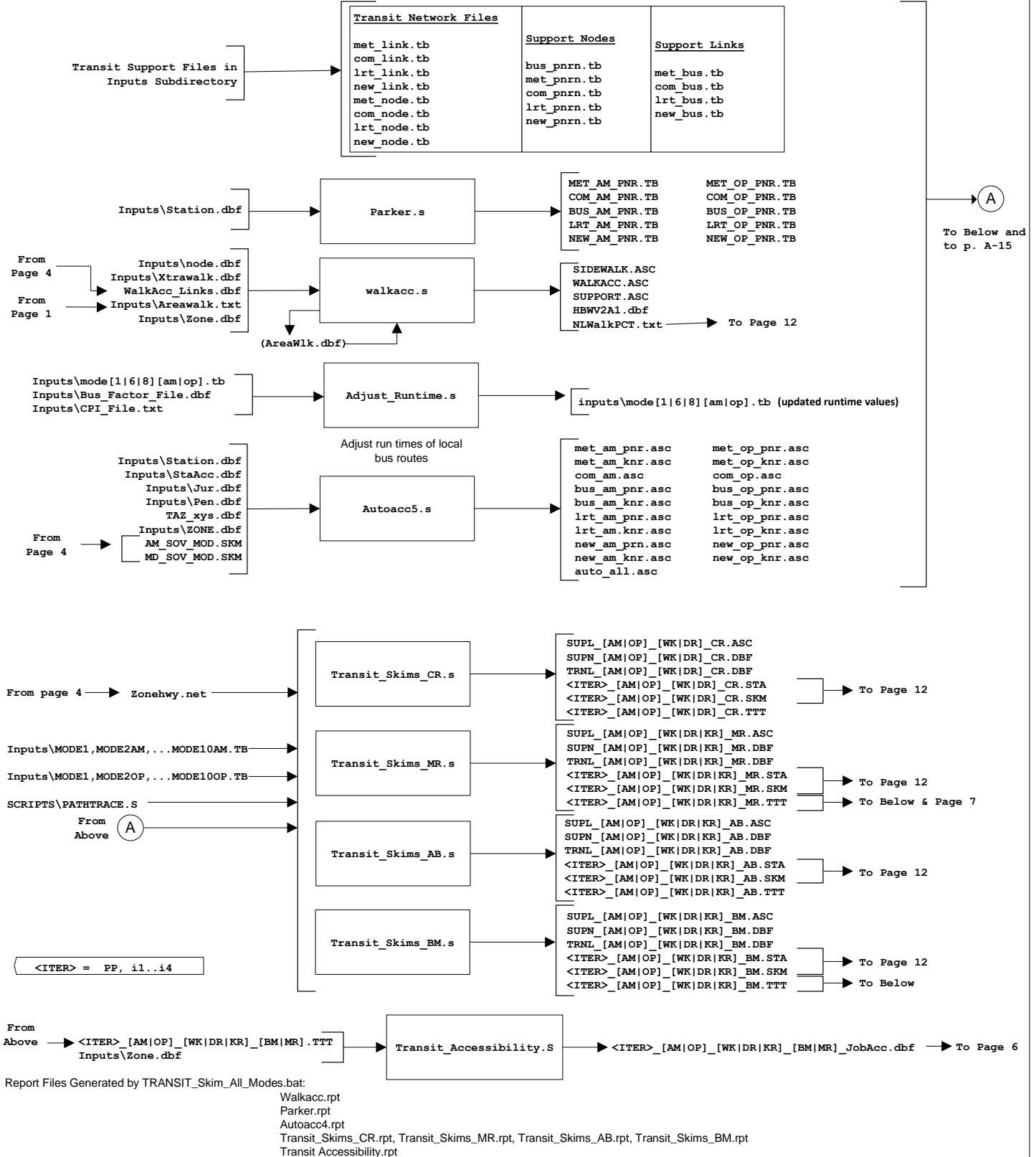


Report Files Generated by PP\_Highway\_Skims.bat:

pp\_Highway\_skims.rpt  
pp\_Joinskims.rpt  
pp\_Modnet.rpt  
pp\_Highway\_skims\_mod.rpt  
CheckStationAccess.rpt  
pp\_RemovePPSpeed.rpt

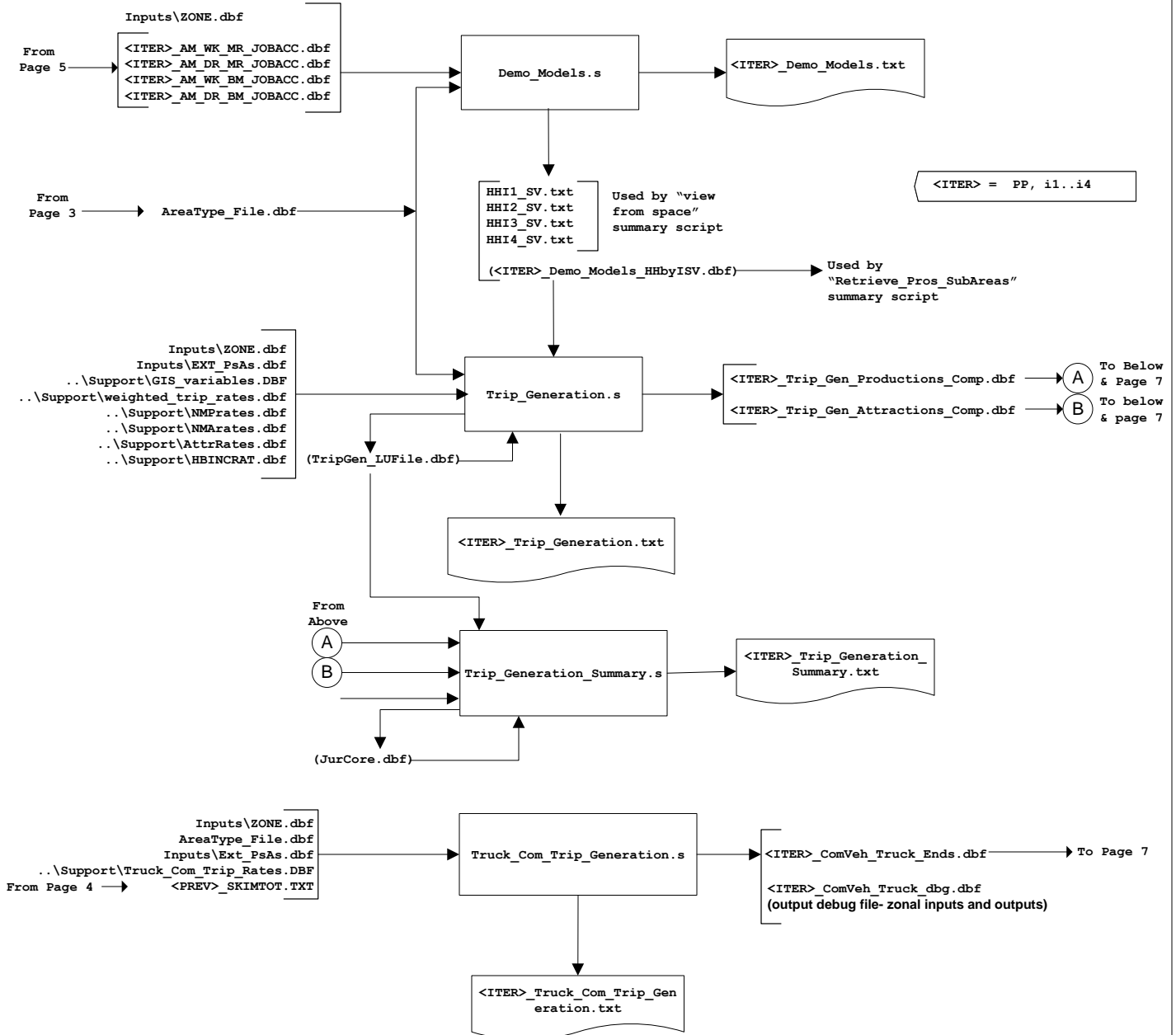


### Transit\_Skim\_All\_Modes\_Parallel.bat Transit\_Skim\_LineHaul\_Parallel.bat





### Trip\_Generation.bat

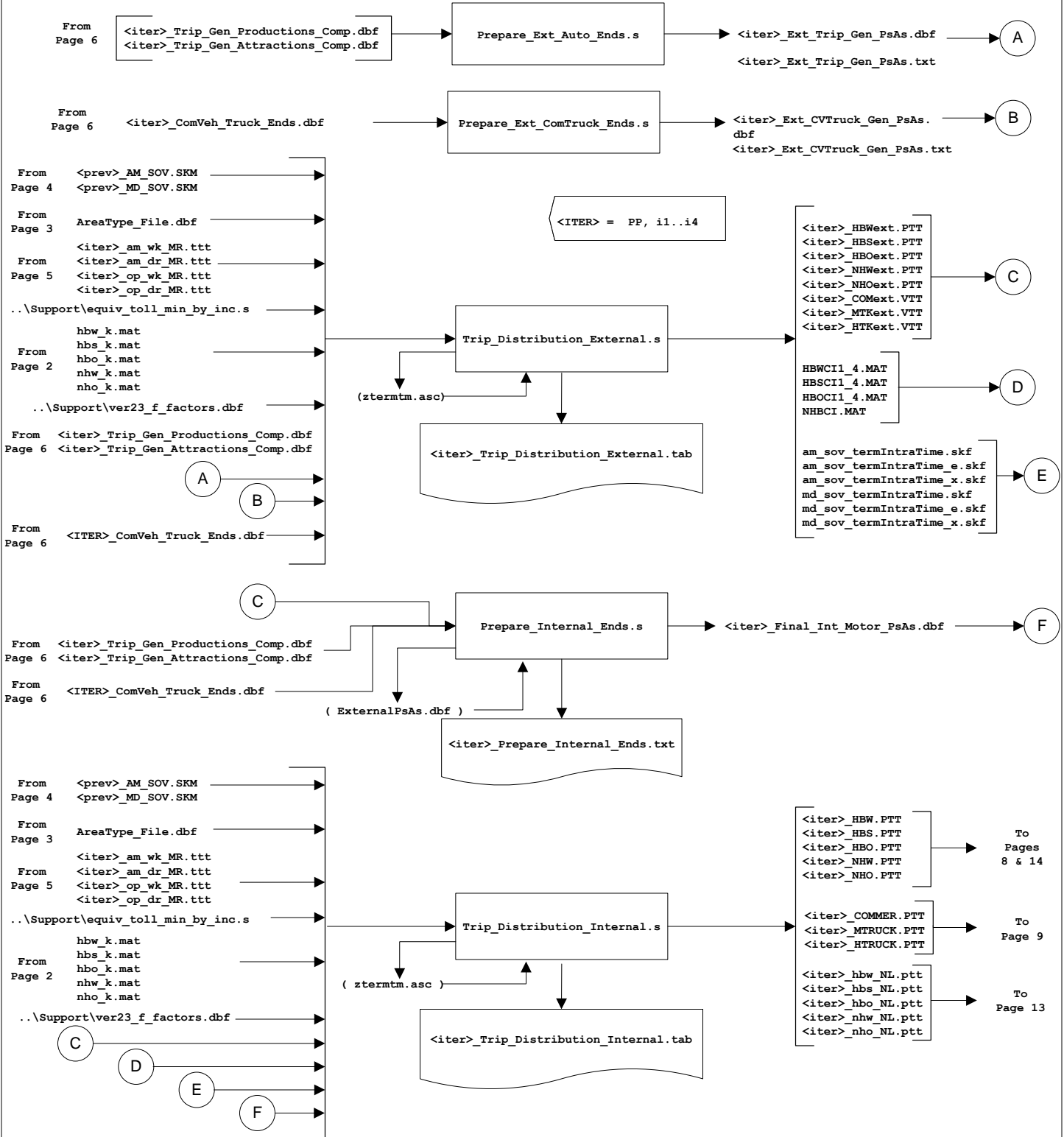


Report Files Generated by Trip\_Generation.bat:

- <ITER>\_Demo\_Models.rpt
- <ITER>\_Trip\_Generation.rpt
- <ITER>\_Trip\_Generation\_Summary.rpt
- <ITER>\_Truck\_Com\_Trip\_Generation.rpt



### Trip\_Distribution.bat





**TITLE: Flowchart for the TPB trip-based, regional travel demand forecasting model, Ver. 2.4**

**COMPANY: COG/TPB**

Authors RJM, MS, MSM, RN,SKO

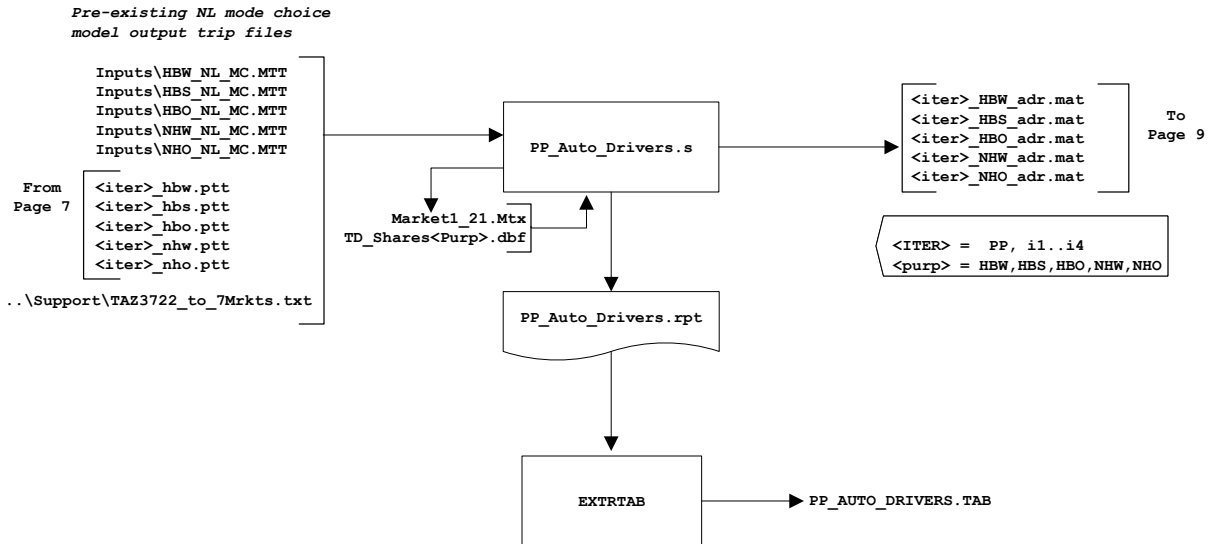
DATE: 5/15/2023

PG: 8

OF 18

FILENAME: Ver2.4.6\_flowchart.vsd

## PP\_Auto\_Drivers.bat

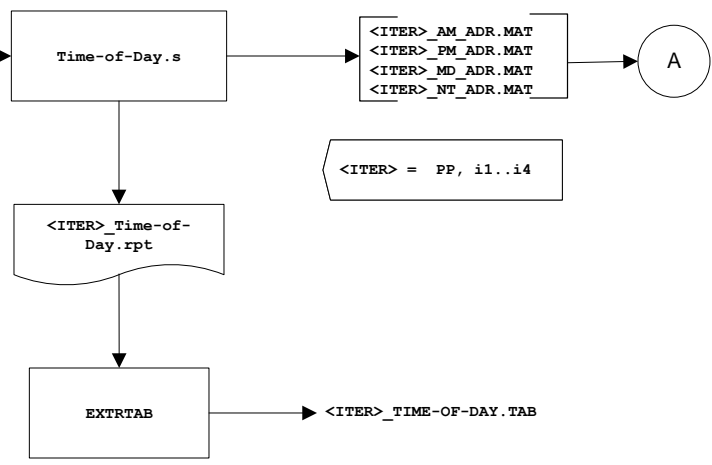




### Time-of-Day.bat

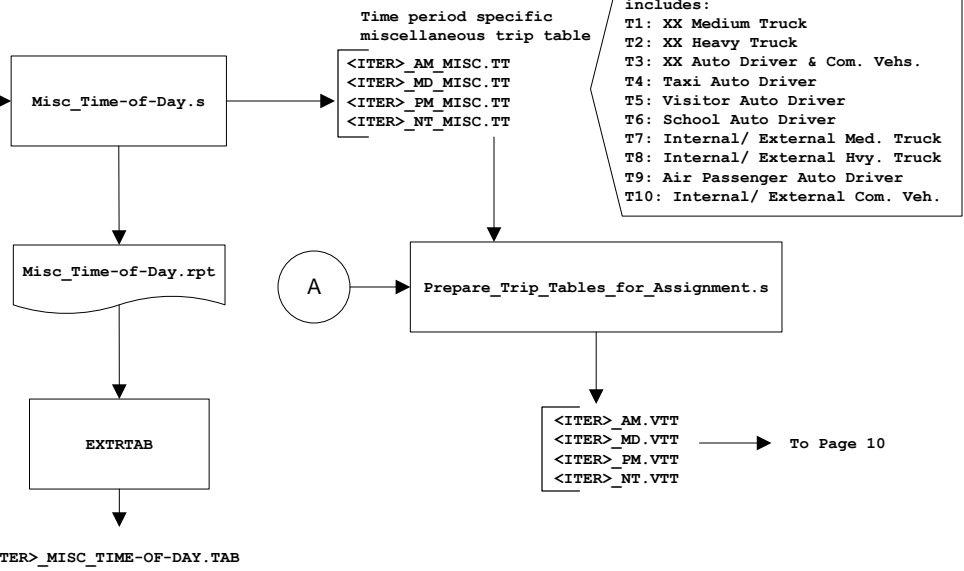
From Page 8 (pp Iiter)  
From Page 14  
(i1 to i4 iter)

```
<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\AIRPAK.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:** Flowchart for the TPB trip-based, regional travel demand forecasting model, Ver. 2.4

**COMPANY:** COG/TPB

**Authors** RJM, MS, MSM, RN,SKO

**DATE:** 5/15/2023

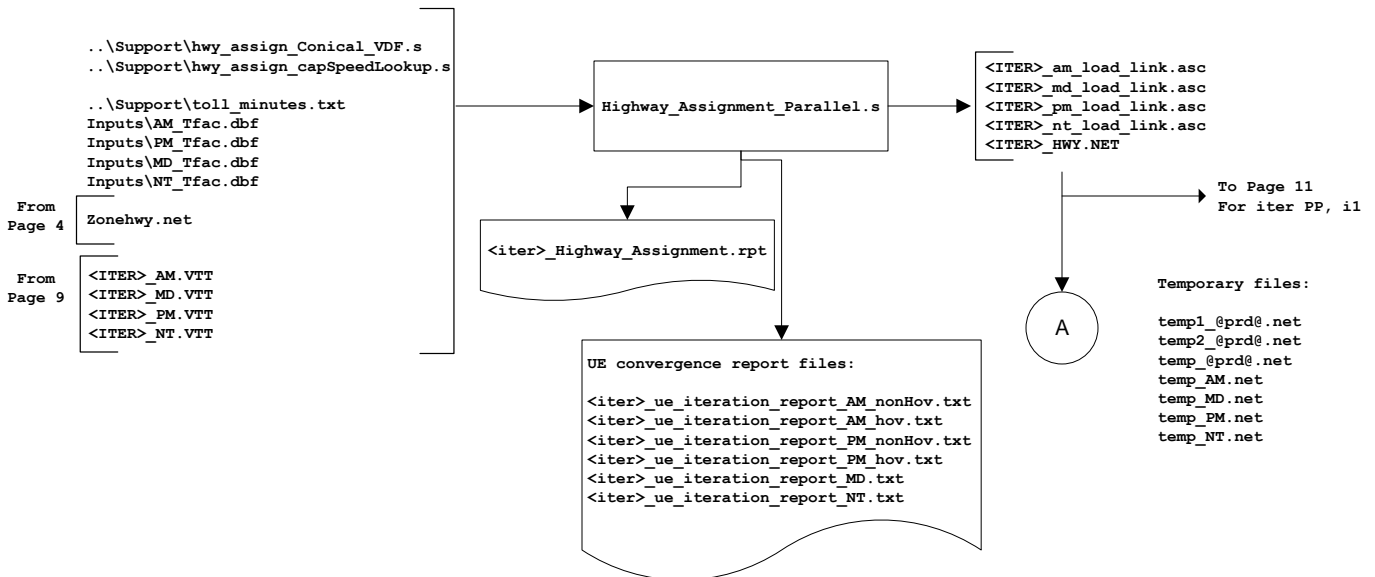
**PG:** 10

**OF** 18

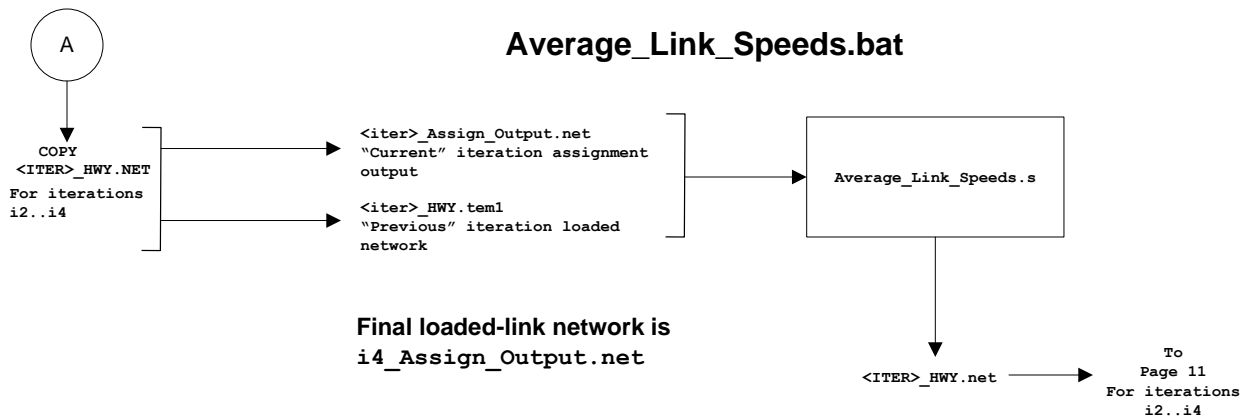
**FILENAME:** Ver2.4.6\_flowchart.vsd

## Highway\_Assignment\_parallel.bat

<ITER> = PP, i1..i4



## Average\_Link\_Speeds.bat



Report Files Generated by Highway\_Assignment.bat:

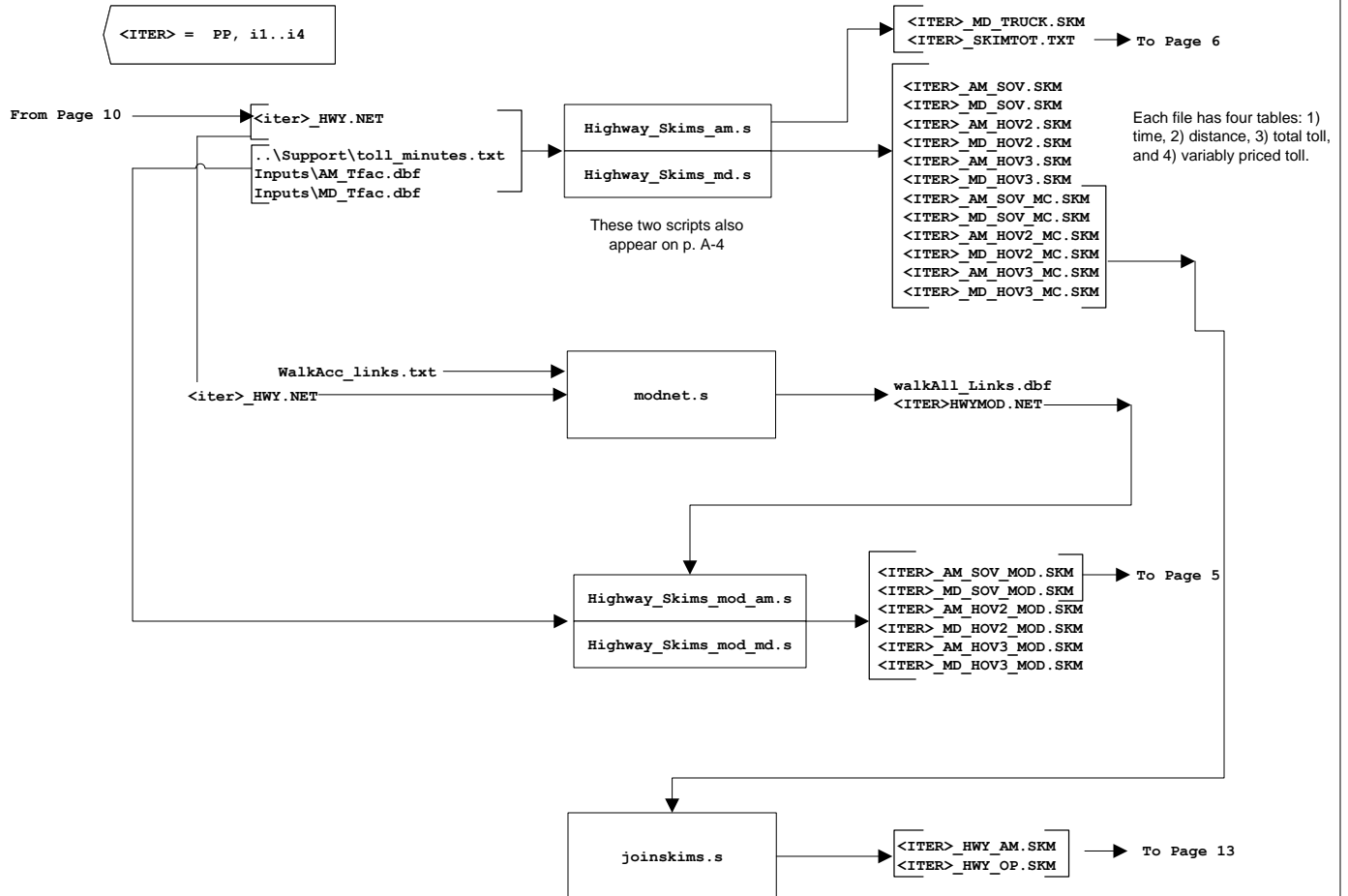
<ITER>\_Highway\_assignment.rpt  
Average\_Link\_Speeds.rpt





## Highway\_Skims.bat

(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: Flowchart for the TPB trip-based, regional travel demand forecasting model, Ver. 2.4**

**COMPANY: COG/TPB**

**Authors RJM, MS, MSM, RN,SKO**

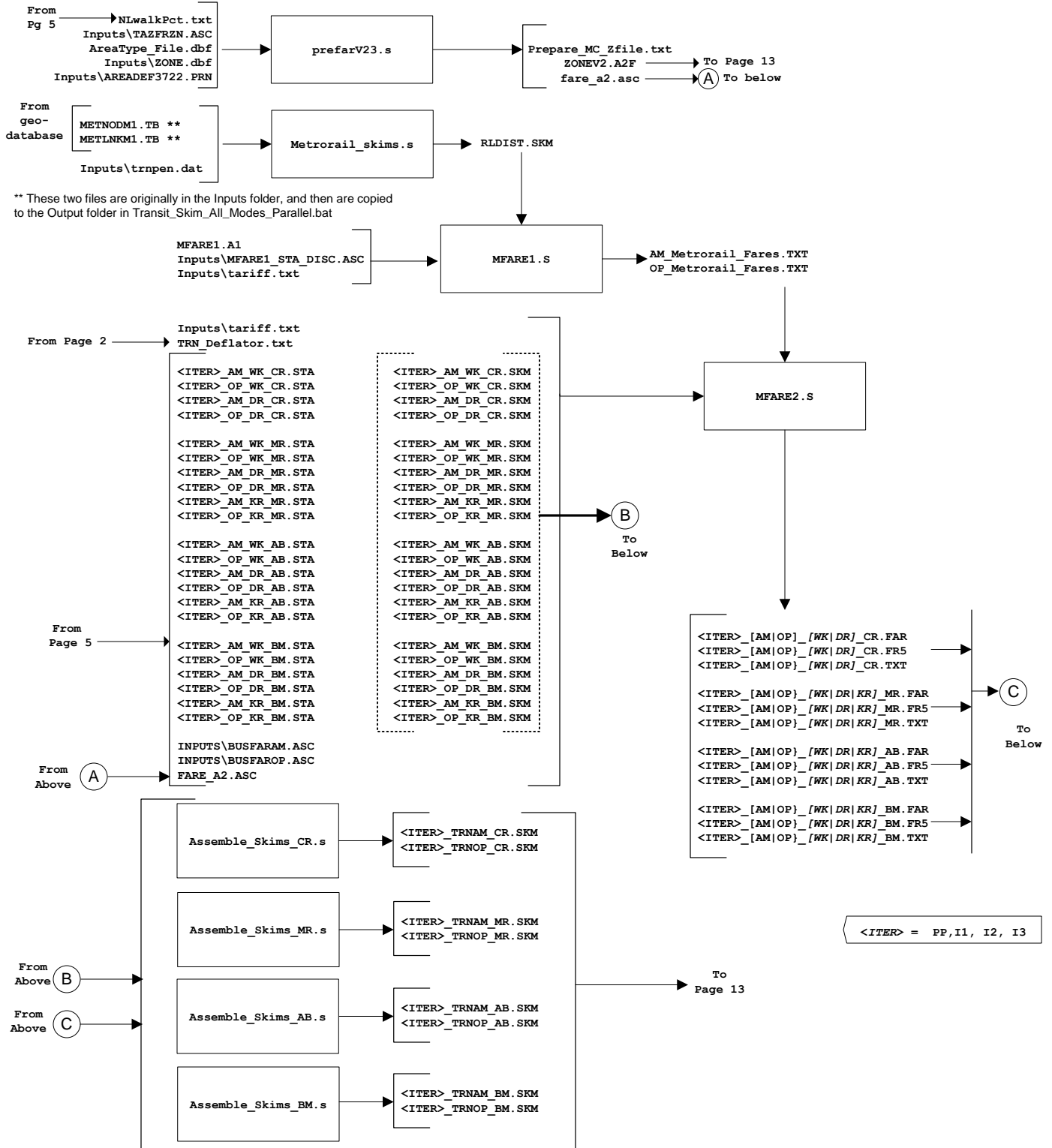
**DATE: 5/15/2023**

**PG: 12**

**OF 18**

**FILENAME: Ver2.4.6\_flowchart.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: Flowchart for the TPB trip-based, regional travel demand forecasting model, Ver. 2.4

COMPANY: COG/TPB

Authors RJM, MS, MSM, RN,SKO

DATE: 5/15/2023

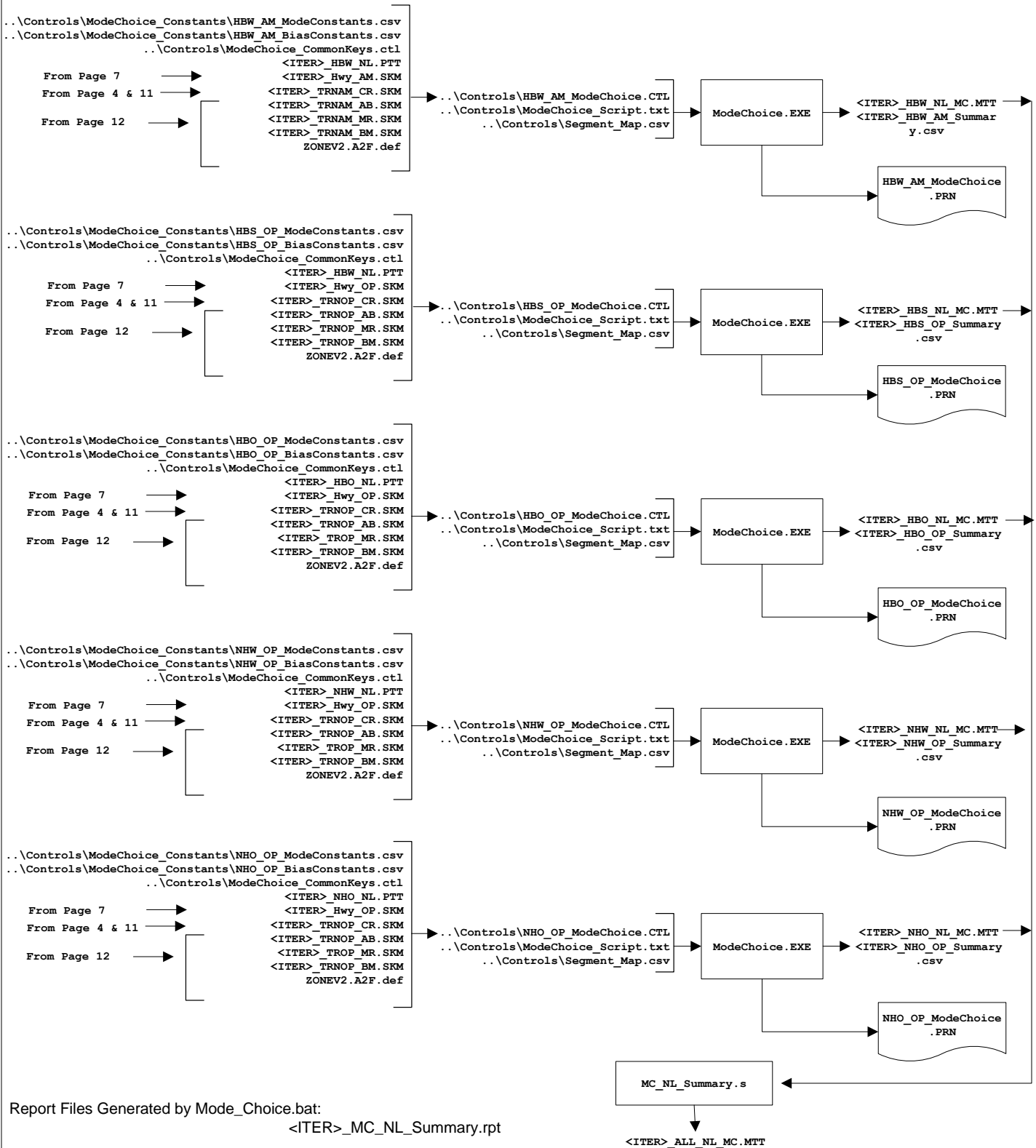
PG: 13

OF 18

FILENAME: Ver2.4.6\_flowchart.vsd

### Mode\_Choice\_parallel.bat MC\_purp.bat

<ITER> = i1..i4



Report Files Generated by Mode\_Choice.bat:

<ITER>\_MC\_NL\_Summary.rpt

<ITER>\_ALL\_NL\_MC.MTT



**TITLE: Flowchart for the TPB trip-based, regional travel demand forecasting model, Ver. 2.4**

**COMPANY: COG/TPB**

Authors RJM, MS, MSM, RN,SKO

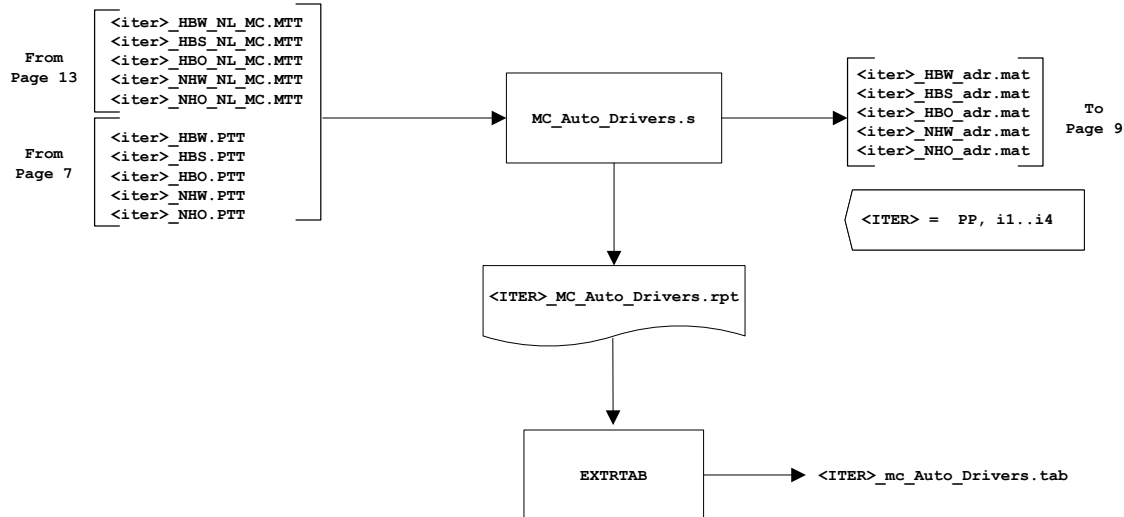
DATE: 5/15/2023

PG: 14

OF 18

FILENAME: Ver2.4.6\_flowchart.vsd

### Auto\_Driver.bat

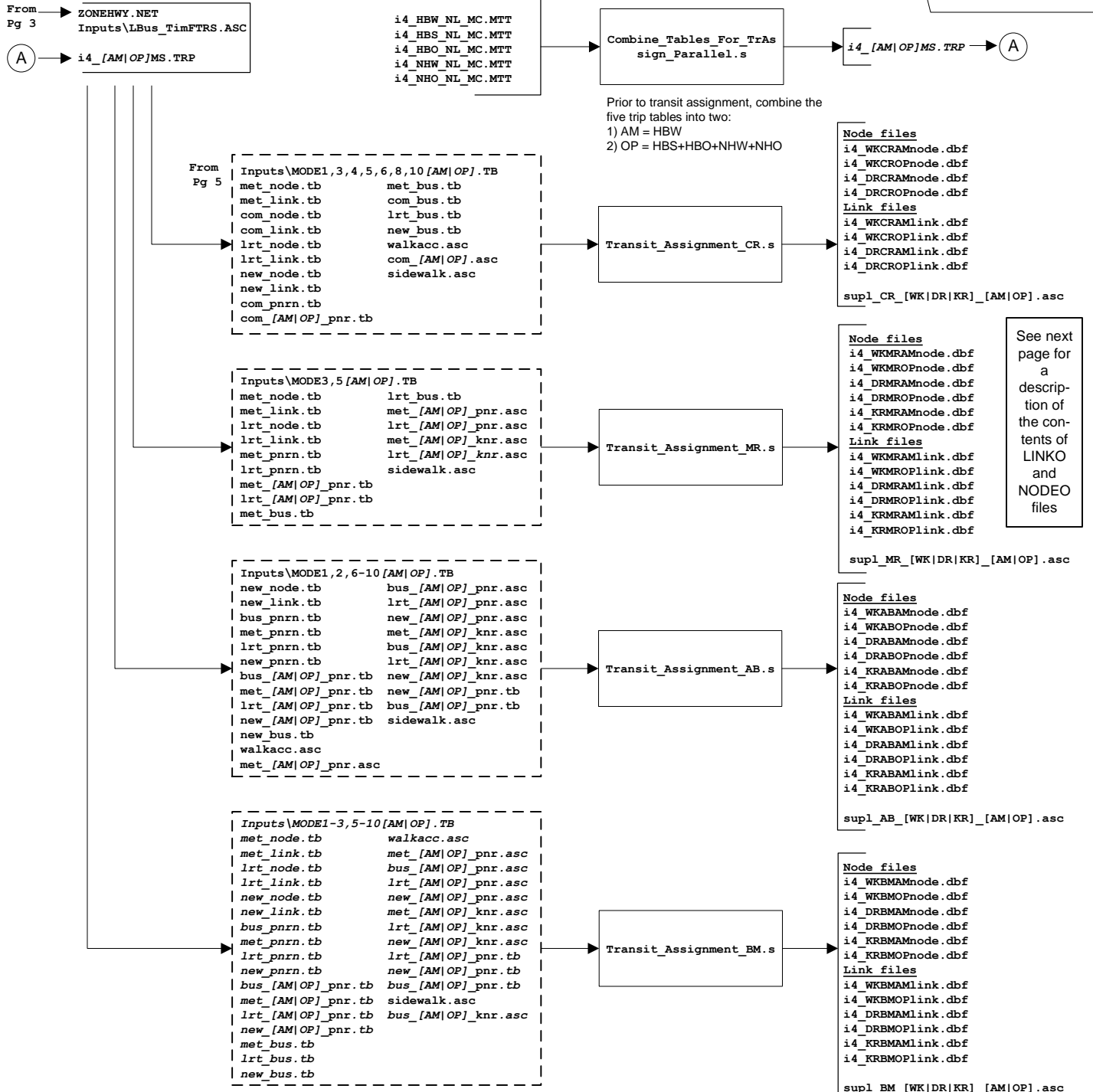


Report Files Generated by Auto\_driver.bat:  
    <ITER>\_MC\_Auto\_Drivers.rpt



### Transit\_Assignment\_Parallel.bat Transit\_Assignment\_LineHaul\_Parallel.bat

AA = WK, DR, KR  
(Note: No KR for CR)



See next page for a description of the contents of LINKO and NODEO files



**TITLE: Flowchart for the TPB trip-based, regional travel demand forecasting model, Ver. 2.4**

**COMPANY: COG/TPB**

**Authors RJM, MS, MSM, RN,SKO**

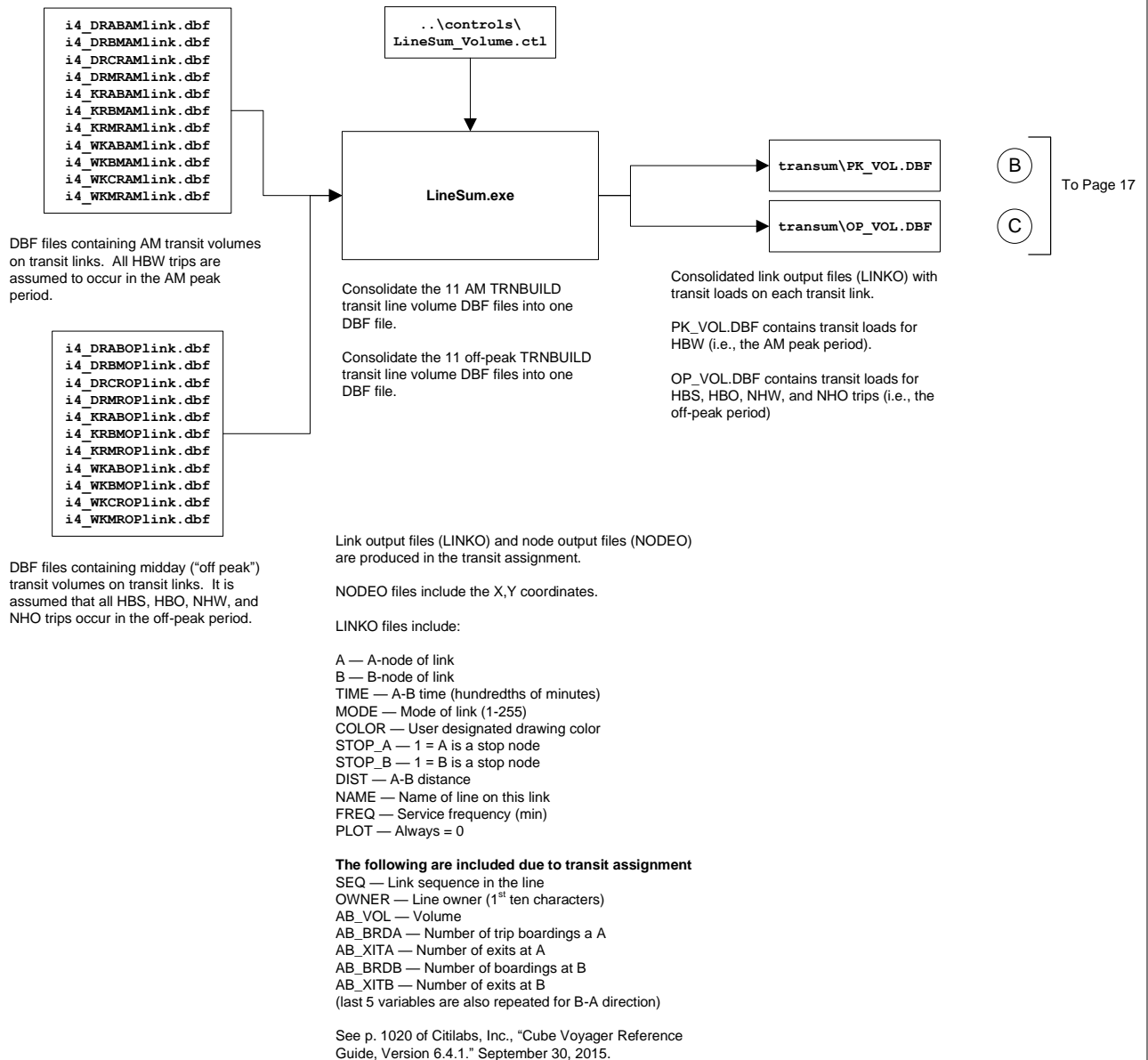
**DATE: 5/15/2023**

**PG: 16**

**OF 18**

**FILENAME: Ver2.4.6\_flowchart.vsd**

## Transum.bat (page 1 of 2)





**Transum.bat (page 2 of 2)**

