USER'S GUIDE

For the COG/TPB
Gen3/Version 1.1.0 Travel Demand Forecasting Model

April 16, 2026

User's Guide for the COG/TPB Gen3/Version 1.1.0 Travel Demand Forecasting Model

April 16, 2026

About the TPB

The National Capital Region Transportation Planning Board (NCRTPB 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, 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 (MWCOG or COG).

Credits

Director, Department of Transportation Planning (DTP) and Deputy Executive Director, Regional Planning: Kanti Srikanth
Editors: Feng Xie, Ray Ngo, and Mark Moran
Contributing Editors: Bahar Shahverdi, Glenn Lang and Meseret Seifu
Oversight: COG/TPB Travel Forecasting Subcommittee

In March 2025, COG/TPB's consultant (RSG) developed a user's guide for the Gen3 Travel Model, which corresponded to the state of the model at the end of Phase 2 of development (i.e., Gen3 Model, Version 1.0.0). That document was the culmination of a collaborative effort among COG, RSG, and Baseline Mobility Group (BMG).

In November 2025, COG/TPB staff updated and reformatted the user's guide for the beta release of the Gen3 Model (Version 1.0.5) to conform to COG style guidelines. Since COG took the ownership of the model, COG/TPB staff, with the on-call support from RSG and BMG, have made a range of updates to the model. Accordingly, COG staff have made substantial edits throughout this documentation to include those updates.

This third revision of the document includes additional edits in accordance with the post-beta model updates included in the official release of the Gen3 Model (Version 1.1.0), which is now deemed suitable for production use.

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 or call (202) 962-3300 or (202) 962-3213 (TDD).

TITLE VI Nondiscrimination Policy

The Metropolitan Washington Council of Governments (COG) operates its programs without regard to race, color, and national origin and 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 or call (202) 962-3300.

El Consejo de Gobiernos del Área Metropolitana de Washington (COG) opera sus programas sin tener en cuenta la raza, el color, y el origen nacional y cumple con el Título VI de la Ley de Derechos Civiles de 1964 y los estatutos y reglamentos relacionados que prohíben la discriminación en todos los programas y actividades. Para más información, presentar una queja relacionada con el Título VI, u obtener información en otro idioma, visite www.mwcog.org/nondiscrimination o llame al (202) 962-3300.

TABLE OF CONTENTS

1 INTRODUCTION 1
1.1 Development of the Gen3 Model 1
1.2 Phased development of the model 2
1.2.1 Recent updates to the model 3
1.3 What is an activity-based model (ABM) and how is it different from a trip-based model? 7
1.4 Purpose and need for an activity-based model for the metropolitan Washington region 8
1.5 ActivitySim and the ActivitySim Consortium 9
1.6 Description of Gen3 Model 11
1.6.1 Treatment of time 13
1.6.2 Treatment of space 14
1.7 Differences between the Gen2 Model and Gen3 Model 15
2 SYSTEM ARCHITECTURE 17
2.1 Hardware requirements 17
2.2 Software requirements 19
2.2.1 Package manager software 22
2.3 ActivitySim installation and model setup 23
2.3.1 Step 1: Download the Gen3 Model 23
2.3.2 Step 2: install and configure ActivitySim 23
2.3.3 Step 3: Configure the main batch file 24
2.4 Directory structure 25
3 DATA INPUTS & OUTPUTS 27
3.1 Model inputs 27
3.1.1 Auxiliary data 27
3.1.2 Highway data 28
3.1.3 Land use data 30
3.1.4 Synthetic population data 32
3.1.5 Transit data 35
3.1.6 External and visitor/tourist transit trip data 37
3.2 Model outputs 38
3.2.1 Model visualizer 39
3.2.2 ActivitySim 41
3.2.3 Auxiliary 53
3.2.4 Highway assignment 55
3.2.5 Highway network 55
3.2.6 Land use 56
3.2.7 Reports 57
3.2.8 Skims 59

3.2.9 Transit assignment 60
3.2.10 Transit network 62
4 RUNNING THE MODEL 63
4.1 What to prepare/change 63
4.2 Which scripts to run 63
4.3 Visualizer preparation 64
4.4 Running the model on an on-premises server versus in the cloud 64
4.5 Expected model warnings 65
4.5.1 Windows command line messages 65
4.5.2 ActivitySim messages 65
4.6 Debugging model crashes 65
4.6.1 Pre-run checks 66
4.6.2 ActivitySim crashes 67
4.6.3 Cube crashes 68
4.7 Model steps and speed-feedback loops 68
5 WORKING WITH ACTIVITYSIM 71
5.1 How ActivitySim runs 71
5.2 ActivitySim configuration files 71
5.2.1 Model settings files 71
5.2.2 Model specification files 72
5.2.3 Model coefficient files 72
5.2.4 Model coefficients template file 72
5.2.5 Annotation files 73
5.3 Working with ActivitySim model outputs 74
5.3.1 Working with household, person, tour, and trip files 74
5.4 Model calibration and validation 74
5.4.1 Calibration preparation 75
5.4.2 Model calibration 77
5.4.3 Model validation 77
6 MORE DETAILED DESCRIPTION OF MODEL SYSTEM 81
6.1 Highway skimming and assignment 81
6.1.1 Overview 81
6.1.2 Application details 82
6.2 Toll searching 83
6.3 Transit skimming and assignment 85
6.3.1 Background 85
6.3.2 Implementation 86
6.3.3 Implementing a partial fix to the hyperpath issue 87

6.4 PopulationSim inputs and process 89
6.4.1 Population Synthesis outputs 89
6.4.2 Data preparation and application 90
6.4.3 Scenario applications 92
6.4.4 Controls and settings 93
6.4.5 Validation 95
6.5 ActivitySim resident travel model 96
6.5.1 Long-term models 97
6.5.2 Daily models 100
6.5.3 Tour models 101
6.5.4 Trip models 105
6.6 Truck Model 107
6.7 Commercial vehicle model 108
6.8 Auxiliary model and input trip tables 109
6.8.1 Airport passengers, visitors, taxi trips, and auto-person External-External Trips 111
6.8.2 External-Internal (EI) and Internal-External (IE) auto trips model 111
6.9 External and Visitor transit travel model 113
7 APPENDIX 114
7.1 Detailed model flowcharts 114
7.2 Using the Gen3 Model for AV ownership 129
7.2.1 Simple studies 129
7.2.2 Complex studies 129
7.2.3 AV mode choice updates 130
7.2.4 Vehicle type data 130
7.2.5 Vehicle type choice setup 132
7.3 Example scripts 133
7.3.1 Model crosstab script 133
7.3.2 Model calibration scripts 133
7.4 Miscellaneous 137
7.4.1 Changing the disk space threshold 137

LIST OF FIGURES

Figure 1: COG/TPB Gen3 ActivitySim Flowchart (demand-side model) 13
Figure 2: Gen3 Model Transportation Analysis Zones 15
Figure 3: COG/TPB Gen3 Model File Structure 26
Figure 4: ABM Visualizer 41
Figure 5: Model Flow Summary 70
Figure 6: Example Highway Validation Spreadsheet Output 79
Figure 7: Gen3 Model Flow with an Integrated Toll Setting Process 84
Figure 8: Traffic Assignment Components and Auxiliary Models 109
Figure 9: Auxiliary Highway Model Process 111
Figure 10: Auto Person External-Internal and Internal-External Trip Flowchart 112
Figure 11: Example Plot From construct_veh_type_data.ipynb 132
Figure 12: Example Chart Output - AV Comparison 136

LIST OF TABLES

Table 1: Time Periods for Level-Of-Service Skims and Assignment 14
Table 2: Major Gen3 Model Updates relative to Gen2 Model 16
Table 3 Specifications Of Travel Model On-Premises and AWS Cloud Server 19
Table 4: Model Supporting Data Files 26
Table 5: Auxiliary Folder Input File List 27
Table 6: External Productions and Attractions (Ext_PsAs.DBF) File Format 28
Table 7: Highway Folder Input File List 28
Table 8: Toll Factor Input File Format 29
Table 9: Link.DBF File Format 29
Table 10: Node.DBF FIle Format 30
Table 11: Toll_Esc.DBF File Format 30
Table 12 Files Contained in the Land Use Data Folder 31
Table 13: Land Use and Zonal Data File Format 31
Table 14: Files contained in the Synthetic Population Data Folder 32
Table 15: Synthetic Households File Fields 32
Table 16: Synthetic Persons File Fields 33
Table 17: Transit Input Files 35
Table 18: Bus Factor File Format (Bus_Factor_File.DBF) 36
Table 19: Rail Link File Format (Rail_Links.DBF) 36
Table 20: Station Input File Format (STATION.DBF) 37
Table 21: External and Visitor Transit Input Files 38
Table 22: ActivitySim Model Output Files 41
Table 23: Final_Households.CSV Fields 42
Table 24: Final_Persons.CSV FIelds 44
Table 25: final_land_use.csv Fields 48
Table 26: Accessibility Impedance Values and Dispersion Parameters 49
Table 27: final_accessibility.csv Fields 49
Table 28: final_joint_tour_participants.csv Fields 51
Table 29: final_tours.csv Fields 51
Table 30: final_trips.csv Fields 52
Table 31: final_vehicles.csv Fields 52
Table 32: Auxiliary Model Output Files 53

Table 33: Highway Assignment Output Files 55
Table 34: Highway Network Output Files 55
Table 35: Land Use Output Files 56
Table 36: AreaType_File.dbf Fields 56
Table 37: Floating_LU.dbf Fields 56
Table 38: TAZ_XYs.dbf Fields 57
Table 39: ZONEV2.A2F Fields 57
Table 40: Reports Folder Output Files 57
Table 41: Transit Skim File Name Pattern Elements 59
Table 42: Highway Skim File Name Pattern Elements 59
Table 43 Transit Skim Matrix Tables 60
Table 44: LINKVOL.DBF Output Table Fields 61
Table 45: Station-To-Station Output Table Fields 61
Table 46: Visualizer Variables in Run_Model.Bat 64
Table 47: Example transit Validation Spreadsheet Table 80
Table 48: Highway Skimming Program Inputs 81
Table 49: Highway Skimming Output Files 81
Table 50. Data Sources for Seed Sample and Marginal Controls 91
Table 51: Example Household Size Control Adjustments 92
Table 52: COG PopulationSim Marginal controls, Residential 94
Table 53: COG PopulationSim Marginal Controls, Group Quarters 94
Table 54: Truck Trip Generation Rates as a Function of Truck Type, Area Type, and Land Use Category 108
Table 55: AV Mode Choice Adjustment Defaults 130
Table 56: AV interpolation Methods 131
Table 57: Example Script Output - Trips by Person Type Crosstab 133
Table 58: Python Libraries used in Calibration 133
Table 59: Example Script Output - AV Comparison 135

1 INTRODUCTION

The Metropolitan Washington Council of Governments (MWCOG 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 comprises 300 elected officials from 24 local governments, the Maryland and Virginia state legislatures, and U.S. Congress. The Board of Directors is COG's governing body and is responsible for its overall policies. The National Capital Region Transportation Planning Board (NCRTPB or TPB) is the federally designated metropolitan planning organization (MPO) for metropolitan Washington. TPB is responsible for developing and carrying out a continuing, cooperative, and comprehensive transportation planning process in the metropolitan area. COG is the administrative agent for the TPB, and the TPB is staffed by COG's Department of Transportation Planning (DTP). The COG/TPB staff, with some consultant assistance, develops, maintains, applies, and improves the TPB's family of regional travel demand forecasting models, which are used for regional, long-range transportation planning in the metropolitan Washington region. These regional travel demand models are developed under the guidance of the TPB Travel Forecasting Subcommittee (TFS).

This document is a user's guide for the COG/TPB Generation 3, or Gen3, Travel Demand Forecasting Model (Version 1.1.0), which covers the metropolitan Washington region, and which has been developed by RSG, Baseline Mobility Group (BMG), and COG/TPB staff.

The Gen3 Model is a state-of-the-practice, tour-based/activity-based travel model (ABM) that utilizes ActivitySim for modeling internal person travel demand and Bentley Systems' Cube Voyager for modeling auxiliary travel demand and transportation supply (e.g., trip assignment). ActivitySim is open-source software for developing and applying ABMs. The development of ActivitySim is guided by the ActivitySim Project Management Committee (a.k.a. ActivitySim Consortium), which currently consists of 14 public-sector member agencies, including COG.[1] The Association of Metropolitan Planning Organizations Research Foundation (AMPORF) was the administrative agent for the ActivitySim Consortium up to June 30, 2025. From July 1, 2025, the administrative agent is now the Zephyr Foundation.

Following the introduction, this document describes the system architecture, data inputs and outputs, how to run the model, and how to work with ActivitySim, and finally provides detailed model/sub-model descriptions.

1.1 Development of the Gen3 Model

Following a multi-year strategic plan that has been used to guide the development of the TPB's travel forecasting methods,[2] COG/TPB staff set out in late 2019 to develop a next-generation travel demand model. The project team, consisting of RSG and Baseline Mobility Group, recommended that COG transition from its current aggregate, trip-based travel demand model (i.e., Generation 2, or Gen2, Model) to an activity-based model (ABM) implemented in the open-source ActivitySim software platform. The new model is known as the Generation 3, or Gen3, Travel Model.

---

[1] Zephyr Foundation. "ActivitySim," 2026. https://zephyrtransport.org/ActivitySim/

[2] Cambridge Systematics, Inc., Draft Strategic Plan for Model Development, Task Order 15.2, Report 3 of 3, Final Report (Washington, D.C.: Metropolitan Washington Council of Governments, National Capital Region Transportation Planning Board, October 15, 2015).

1.2 Phased development of the model

The Gen3 Model was developed in three phases. Phase 1, which ended in February 2022, created a prototype model that was tested by the COG/TPB staff. Phase 2, which finished in March 2024, developed a model that was to be production ready or near production ready. Phase 3 was the usability testing phase, which evaluated and further improved the model's readiness for production work. The purpose of this phased approach to model development was to use the initial deployment and calibration efforts to inform the scope of subsequent model development and calibration/validation tasks, rather than scope the entire model development project at the project initiation. This allowed the project team to learn from the initial deployment in Phase 1 and prioritize resource allocation in Phase 2, and to ensure that the final delivered Gen3 Model at the end of Phase 3 meets the needs of COG/TPB, partner agencies, and decision-makers and is ready for production use.

In the Phase 1 development, a representative population (a.k.a. synthetic population) for the modeled region was created, and the ActivitySim model system was transferred from the Southeast Michigan Council of Governments (SEMCOG) region (Detroit, Michigan) to COG. Under the Phase 1 deployment, tour mode choice and tour destination choice models were estimated.[3] After implementation of the estimated tour mode choice and tour destination choice models, several model components such as auto ownership, tour mode choice, trip mode choice, individual non-mandatory tour frequency and intermediate stop frequency models were calibrated to the observed distributions from the 2017-18 COG/TPB Regional Travel Survey (RTS) and the 2018-19 Maryland Statewide Household Travel Survey (MTS) data. The processing of RTS/MTS data was documented in the Phase 1 data development report.[4] The 2018 traffic counts and transit boarding counts that are used for Phase 1 model validation were provided by COG/TPB staff and were documented separately.[5] Thus, the preparation of the count data is not covered in this document. Some important documentation for the Phase 1 development can be found on the COG website.[6]

In the Phase 2 development, model refinements were made based on what was learned in Phase 1 and a few new components (including an autonomous vehicle ownership model and a vehicle type choice model) were added. The project team (RSG, BMG and COG) estimated additional models and calibrated more than a dozen of the component models to base-year (2018) conditions. At the end of Phase 2 development, highway and transit validations were performed, comparing base-year model output to observed data. The project team found that the validation metrics of the Gen3 Model conformed to the federal and state benchmarking standards, and the validation performance was either comparable to or better than that of the current adopted, production-use Gen2 Model. The calibration and validation results were documented in a technical report that can be found on the

---

[3] RSG, "Tour Mode Choice and Destination Choice Model Estimation", Technical Report Prepared for Metropolitan Washington Council of Governments, January 19, 2022.

[4] RSG, "Gen3 Data Development." Metropolitan Washington Council of Governments, National Capital Region Transportation Planning Board. December 29, 2021 (https://app.box.com/s/xe5vb28daox1aqtw895iy2r5ocy584w8)

[5] See, for example, *Metrorail Average Weekday Passenger Boardings (Station Level): 1977 to 2018*, Summary Table (Washington Metropolitan Area Transit Authority, 2018), https://www.wmata.com/about/records/public_docs/upload/2018_historical_rail-ridership_May-weekday-avg.pdf; Meseret Seifu, "2018 Daily and Hourly Traffic Counts," Memorandum to Feng Xie, June 22, 2021.

[6] Metropolitan Washington Council of Governments. "Travel Demand Forecasting - Gen3 Travel Model". https://www.mwcog.org/transportation/data-and-tools/modeling/gen3-model/

COG website.[7] In addition, sensitivity tests that were conducted in Phase 1 were repeated in Phase 2 to examine the reasonableness of model responses to system changes. A scenario study was also conducted to examine the model responses in a hypothetical scenario involving Autonomous Vehicles (AVs). The project team found the outcomes of these tests to be satisfactory.[8] At the end of Phase 2 development, RSG delivered a production-capable model system labelled as Gen3 Model Version 1.0.0.

Following delivery of the Gen3, Phase 2 Model, the project team immediately started the Phase 3 development in March 2024. The Phase 3 development, which has been led by COG/TPB staff with on-call technical support from RSG and BMG, primarily focused on the evaluation of model usability for production work. Based on the findings from the usability testing, the project team also continued to improve the model through new features, feature enhancements, and bug fixes. In November 2025, COG/TPB staff released a beta version of the Gen3 Model (specifically, Version 1.0.5) for testing. In April 2026, COG released Version 1.1.0, COG/TPB's first production-use Gen3 Model, marking the completion of the Phase 3 development.

1.2.1 Recent updates to the model

Since the delivery of Gen3 Model Version 1.0.0, six version-flagged updates have been made to the model:

- Gen3 Model Version 1.0.1 integrated COG/TPB's heuristic toll setting process in the Gen3 Model flow.[9] It is worth noting that COG/TPB's toll setting process does NOT simulate tolls at the operational level. Instead, it provides "planning-grade" toll estimates that serve as an indicator of travel demand patterns on the regional highway system of variably priced tolling facilities. A toggle switch is provided in the main batch file to turn toll-setting on or off depending on the user preference. When the toggle is switched off, which is the default, a set of COG/TPB pre-estimated toll rates, usually from the latest Air Quality Conformity analysis, will be used for travel demand modeling. When the toggle is switched on, on the other hand, the model will re-estimate the toll rates on variably priced tolling facilities and run travel forecasting based on them.
- Gen3 Model Version 1.0.2 implemented a bug fix that addressed the model's mishandling of negative values in the synthetic population data. These negative values are legitimate. For example, a negative household income value ("hhincadj") indicates a net loss of annual income reported from the American Community Survey (ACS) Public Use Microdata Sample (PUMS) data. However, these negative values were not initially properly handled in the Gen3 Model. RSG instituted a fix and conducted a model run to evaluate the effects of this fix on

[7] RSG and BMG, "Gen3 Model Calibration and Validation Report", February 7, 2024, https://www.mwcog.org/assets/1/6/Gen3_Model_Calibration_and_Validation_Report.pdf

[8] The results from both the Phase 2 sensitivity tests and the scenario study were documented in a separate report: RSG, BMG and MWCOG, "Gen3 Model Phase 2 Sensitivity Testing Results", May 22, 2024, https://www.mwcog.org/assets/1/6/Gen3Model_Phase2_SensitivityTests_Report_FINAL_Updated_May_22_20241.pdf

[9] Xie, Feng and Bahar Shahverdi, "Integrating Toll Setting in Gen3 Model Flow: Testing and Recommendations", COG/TPB Memorandum, March 15, 2024.

model results. According to RSG's testing results, the effects of this fix on model output were very limited.[10]

- Gen3 Model Version **1.0.3** included the following three updates:
    - COG/TPB staff replaced Mambaforge with MiniForge3 in the Gen3 Model, as Mambaforge was retired as of December 2024. MiniForge3 is an installer for the Conda package and is also a software environment manager.
    - RSG staff updated the Internal-External (IE) and External-Internal (EI) trip models to split the IE and EI auto-driver trip tables into different auto-occupancy groups. Previously, all the IE and EI trips were simulated as Single-Occupant Vehicles (SOVs).
    - COG/TPB staff fixed a glitch in the AreaType_File.s script file that had previously led to unexpected changes in Area Type values.
- Gen3 Model Version **1.0.4** included the following major updates:
    - RSG re-calibrated the model to address several issues identified during the model usability testing, such as under-estimation of household trip rates, overestimation of SOV share and underestimation of High-Occupancy Vehicle (HOV) share in mode choice, and under-representation of trips in midday and evening periods.
    - COG/TPB staff included tolls in commercial vehicle (CV)/truck trips impedances used in CV/truck trip distribution. Previously, the trip distribution model for CV/truck trips used only mid-day SOV/truck travel time as impedances, without considering tolls.
    - RSG reverted the telecommute frequency coefficients in the Coordinated Daily Activity Pattern (CDAP) model to their Phase 1 (Version 1.0.1 Model) values after COG/TPB staff found that, with the telecommute frequency coefficients from Phase 2 (Version 1.0.3 Model), the reduction in work travel resulting from increased telecommuting was inadequate.
    - RSG also reversed the global 15% increase in estimated truck and CV trips in the model due to an unrealistic increase of truck and CV VMT as compared to the empirical data.
    - COG/TPB staff re-validated the Gen3/Ver.1.0.4 Model to 2018 conditions, and the consultant updated the Gen3 Model Calibration and Validation Report.[11]
- Gen3 Model Version **1.0.5** (beta release) included the following updates:
    - Staff implemented an update that allows users to run the Gen3 Model with OpenPaths Cube 2025 Version 25.00.01, in addition to the default option of Cube CE Version 6.5.1. No other Cube versions are currently supported. While the Gen3

[10] Xie, Feng, "COG/TPB Gen3 Travel Model Status Report", Presentation at TPB's Travel Forecasting Subcommittee Meeting on July 12, 2024.
[11] RSG and BMG, "Gen3/Ver. 1.0.4 Model Calibration and Validation Report", August 25, 2025,

/Version 1.0.5 Model can run on OpenPaths Cube 2025 with an approximately 8% runtime reduction, we recommend using Cube 6.5.1 for several reasons:

- The associated networks remain in the Cube 6 (.NET) format rather than OpenPaths Cube's preferred database formats, such as .SQLite CUBE Databases. OpenPaths Cube supports visualization of CUBE .NET network files, so converting output networks to the CUBE 2025 format is only necessary if further editing of the output networks is required.[12]
- The model outputs using OpenPaths Cube 2025 differ slightly from those produced using Cube 6.5.1 due to the rounding differences in OpenPaths Cube 2025.

- COG/TPB staff updated the model to work with ActivitySim 1.3.4 and utilized the new explicit chunking mechanism to dynamically adjust the number of chunks based on household sample size and computer specifications, eliminating the need to conduct chunk training.[13] Different RAM configurations do not affect the ActivitySim outputs, but different processor configurations do.
- Staff implemented a series of changes that substantially improved the model runtime, including:
  - Expanding the transit skimming process from 4 to 16 logical computer processing cores to enhance parallel processing efficiency.[14]
  - Removing the 5-second delays in moving some unnecessary output files, which can be deleted to free up space, to the temp_files folder.
  - Expanding the CUBE to OMX matrix conversion from 1 to 16 cores.
  - Removing unused skims from skims.omx to reduce the processing time.
  - Utilizing 16 or 32 logical cores for highway assignment automatically, depending on available computer resources. For a computer with 32 cores or more, cores are allocated as follows: 14 cores for AM peak, 4 cores for Midday (MD), 12 cores for PM peak, and 2 cores for Nighttime (NT). For a computer with 16 to 31 cores, the allocation is 7 cores for AM, 2 cores for MD, 6 cores for PM, and 1 core for NT. After making this change, COG/TPB staff now consistently conduct their Gen3 Model runs using the 32-core configuration, unless otherwise specified. Due to Cube Cluster architecture, using a different number of cores would result in slight variations in model

[12] Bentley Systems, Inc. Setting up Pre/Post processing networks. 2025. https://docs.bentley.com/LiveContent/web/OpenPaths-v2025/Help/en/topics/2867622/GUID-6DB5AAB7-8892-4D1D-B1EA-9FBCB450714E.html

[13] ActivitySim version 1.3.4. March 6, 2025. https://github.com/ActivitySim/activitysim/releases/tag/v1.3.4

[14] Skimming refers to the process of extracting and summarizing network attributes — such as travel time, distance, cost, or the number of transfers — between origin-destination (O-D) pairs across a transportation network.

outputs (for example, regional VMT differed by 0.02% in one of our tests). For the toll-setting procedure, processing the MD toll search first, followed by running the AM and PM processes in parallel, each using half of the available cores each. Previously, the AM, PM, and MD processes each utilized 4 cores.

  - Staff fixed the incorrect intermediate stop location choice model calculations on the inbound leg of the tour.[15]

- Gen3 Model Version **1.1.0** included the following major updates:
  - Staff upgraded the ActivitySim software from v.1.3.4 to v.1.5.1. There were minimal changes in model results due to this software update.
  - Staff replaced Anaconda/MiniForge3 with UV as the package manager software in the Gen3 Model. UV enables a substantially faster installation of the "Gen3_Model" environment and largely simplifies the model setup. UV also slightly reduces the model runtime.
  - In the previous model versions, the transit subsidy availability model and free parking eligibility model were inherited from the doner trave model from SEMCOG without being calibrated to local data. In this model version, COG/TPB staff recalibrated both models to COG's 2019 State of the Commute (SOC) survey data. Staff subsequently re-calibrated the tour and trip mode choice models. The additional model calibration work is documented in a memo dated February 20, 2026,[16] which amends the Gen3 Model Calibration and Validation Report dated August 25, 2025.
  - Staff implemented a bugfix in the transit skimming process to properly consider the parking cost associated with Park-and-Ride (PNR) transit trips. After this fix, staff conducted a sensitivity test that verified the reasonableness of the model response to changes to the PNR parking cost.
  - Staff re-organized and cleaned up the Visualizer input files that significantly reduced the size of the scenario input folder.
  - Staff updated the 2018 validation of the Gen3 Model with all the latest model changes, which is documented in a memo dated April 6, 2026.[17]

[15] "Intermediate stop location choice model calculations are incorrect for stops on the inbound leg of a tour." 2021. https://github.com/ActivitySim/activitysim/issues/491

[16] Bahar Shahverdi to Feng Xie, "Additional Model Calibration Included in the Gen3/Version 1.1.0 Travel Model", COG/TPB Memorandum, February 20, 2026.

[17] Glenn Lang and Ray Ngo to Feng Xie, "Gen3/Ver. 1.1.0 Travel Model Validation," COG/TPB Memorandum, April 6, 2026.

1.3 What is an activity-based model (ABM) and how is it different from a trip-based model?

A trip is a one-way movement of a person or vehicle from an origin to a destination. A tour is a series of trips (a trip chain) beginning and ending at a home or work anchor location.[18] An activity-based travel model is a model that simulates the travel-related choices made by individual travelers throughout a typical day on the surface transportation system. These choices include long-term choices, daily activity scheduling choices, daily tour-level choices, and daily trip-level choices. Long-term choices include choices like a person's work or school location (if applicable) and how many vehicles a household will own. Daily activity scheduling choices include whether a person will make mandatory tours, non-mandatory tours, or stay home during the day, as well as number of work and school tours during the day. Tour-level choices include tour mode, tour timing, and number of stops on a tour. Finally, trip-level choices determine specific travel modes, times, locations, and purposes for each segment of each tour. All of these choices are linked in the travel model system.

The main difference between trip-based and activity-based models is the level of detail and granularity in modeling travel behavior. Trip-based models focus on individual trips, while activity-based models represent travel as both trips and tours, and consider the entire daily activity pattern of individuals and households. Activity-based models represent a 24-hour time window, at the person level, and use it as a constraint on travel, recognizing that one person can be in only one place at a time. By contrast, trip-based models are aggregate – often making uses of zonal averages – and, thus, do not explicitly account for a traveler's time budget. In fact, most trip-based models represent travel by market segments rather than individual households and persons.

Activity-based models are generally considered to be more realistic in terms of responses to inputs. Compared to trip-based models, ABMs can consider many more variables, including constraints and linkages between travel decisions. However, ABMs are also more complex and computationally more intensive than trip-based models and require additional and different skills to work effectively with model outputs.

Assuming that the model user has a basic knowledge of activity-based modeling theory and model frameworks, this user's guide is focused on the setup and use of the Gen3 Model. For those who want to learn more about activity-based modeling, please refer to Activity-Based Travel Demand Models: A Primer[19] and the Transportation Model Improvement Program (TMIP) Activity Based Model Tutorial.[20]

[18] Castiglione, Joe, Mark Bradley, and John Gliebe. "Activity-Based Travel Demand Models: A Primer." SHRP 2 Capacity Project C46. Washington, D.C.: Transportation Research Board of the National Academies, 2015. http://www.trb.org/main/blurbs/170963.aspx.

[19] National Academies of Sciences, Engineering, and Medicine. 2014. Activity-Based Travel Demand Models: A Primer. Washington, DC: The National Academies Press. https://doi.org/10.17226/22357.

[20] Transportation Model Improvement Program. 2012. TMIP Activity Based Model Tutorial. Online video series. https://tmip.org/content/tmip-activity-based-model-tutorial.

1.4 Purpose and need for an activity-based model for the metropolitan Washington region

During the initial phase of development of the Gen3 Model, COG outlined three objectives for the Gen3 Model:[21]

1. To ensure that the COG/TPB travel demand forecasting methods are either state of the practice or state of the art with respect to the modeling practices of peer MPOs.
2. To address current shortcomings with the TPB's adopted, production-use, trip-based travel demand model (currently the Gen2/Ver. 2.4.6 Model).
3. To ensure that the new model has the capability to address the most pressing regional transportation planning issues in the Washington, D.C. region.

An activity-based travel model (ABM) is the current state of the practice for large urban areas in the U.S. For example, according to a survey of 23 peer MPOs to the TPB, conducted in 2015,[22] 70% of the MPOs had developed or were developing a production-use ABM. At the time of the survey, which was done for TPB's strategic plan for modeling, TPB was neither developing nor using an ABM. Since the survey was conducted, one of the surveyed MPOs that did not have an ABM in development now has an ABM, using ActivitySim, under development (SEMCOG). It is therefore the conclusion of the project team that the state of the practice for peer MPOs is a standard activity-based model, and state of the art is an activity-based model that is "advanced" compared to peer MPO ABMs in one or more ways - treatment of space, time, behavior, special travel markets, integration with dynamic traffic assignment, etc. RSG and COG believe that ActivitySim represents the state of the practice in terms of activity-based model form and function.

One of the key shortcomings of the Gen2/Ver. 2.4.6 Travel Model is aggregation bias[23] due to the aggregate, trip-based model structure. By contrast, ActivitySim is a disaggregate activity-based model system, which eliminates some aggregation bias. The structure of an ABM, such as ActivitySim, also allows the use of any number of explanatory variables without significantly increasing the computational burden.

In addition, the ABM, implemented in ActivitySim, effectively responds to some of the key shortcomings of the current trip-based model. For example, time-of-day choice is explicitly represented at the tour level. ABMs, such as those implemented in ActivitySim, consider accessibility and therefore respond to changes in congestion. As peak-period travel experiences more congestion relative to off-peak periods, the utility and probability of travel in peak periods decreases, all else being equal, and the time-of-day choice model subsequently shifts some travel away from peak periods (i.e., peak spreading, which is not captured in the Gen2 Model). In addition to this, specific features of ActivitySim and the Gen3 Model that address shortcomings in the Gen2 Model include:

- Non-motorized modes

[21] RSG, Inc. and BMG. Gen3 Model Design Plan. July 2, 2020.

[22] Cambridge Systematics, Inc., Status of Activity-Based Models and Dynamic Traffic Assignment at Peer MPOs, Task Order 15.2, Report 2 of 3, 10–11.

[23] See, for example, Bobbit, Zach. "What is Aggregation Bias? (Explanation & Example)," September 24, 2020. https://www.statology.org/aggregation-bias/ for an explanation of aggregation bias.

- Transportation Network Companies (TNCs) and other shared mobility modes, such as Uber and Lyft
- Teleworking, including periodic telecommute (with a regular out-of-home workplace) and working from home (without a regular out-of-home workplace)
- Employer-based transit/parking subsidies
- Connected/autonomous vehicles (CAVs)
- Improved sensitivity to time, pricing, and income
- All households and people in the ABM use disaggregated travel times and costs and use individual household incomes and values of time from the representative population for the basis of all travel decisions.

Activity-based models, including those implemented in ActivitySim, can be used to address the most pressing regional transportation issues in the metropolitan Washington region. The Coordinated Travel - Regional Activity-based Modeling Platform (CT-RAMP) model system, upon which ActivitySim is developed, has been used successfully by other MPOs -- e.g., the San Francisco Bay area's Metropolitan Transportation Commission (MTC) and the Atlanta Regional Commission (ARC) -- for long-range regional transportation planning, transportation improvement program and air quality conformity analyses for approximately 10 years. The model system has been used to analyze the impacts of land use on transportation demand, the effects of highway capacity increases, planning for priced infrastructure including toll roads and managed lanes, demand for at-grade and grade-separated transit investments, and many other relevant projects and policies.

1.5 ActivitySim and the ActivitySim Consortium

ActivitySim is an open platform for activity-based modeling. It is the product of a collaborative activity-based travel behavior modeling software development project that is supported both financially and managerially by a consortium of public-sector agencies. The consortium is led by the ActivitySim Project Management Committee (PMC), which is sometimes referred to as the ActivitySim Consortium. The travel modeling capabilities that are currently implemented in ActivitySim are based on a fully functional activity-based model that was originally designed for MTC.[24] The modeling capabilities currently implemented in ActivitySim are those from the Coordinated Travel Regional Activity-Based Modeling Platform (CT-RAMP) family of models.[25] The system relies on logit choice models to represent travel decisions (how frequently to travel, where to travel, by what mode, etc.) and was designed to achieve behavioral realism within a practical system of components. The existing ActivitySim/CT-RAMP modeling framework addresses many of the limitations noted above with respect to the COG/TPB Gen2 trip-based model.

The original CT-RAMP model was developed jointly for both MTC and ARC and implemented in the Java programming language. In 2014, a consortium of transportation planning agencies, including

[24] Metropolitan Transportation Commission. Plan/Bay Area: Technical Summary of Predicted Traveler Responses to First Round Scenarios Technical Report. March 22, 2011, available https://mtcdrive.app.box.com/s/3qj8egg1esg01ac68qtnlq8e0c4l4h6s.

[25] Davidson, Vovsha, Freedman, and Donnelly. CT-RAMP Family of Activity-Based Models. Australasian Transport Research Forum 2010 Proceedings. 29 September – 1 October 2010, Canberra, Australia. Publication website: http://www.patrec.org/atrf.aspx

Metropolitan Planning Organizations (MPOs), state Departments of Transportation (DOTs), and municipal planning agencies created the ActivitySim project to "create and maintain advanced, open-source, activity-based travel behavior modeling software based on best software development practices for distribution at no charge to the public." [26] The consortium decided to adopt the MTC Travel Model One (TM1) activity-based model as the basis for the new software tool, and subsequently contracted for consultant services, using the Association of Metropolitan Planning Organizations (AMPO) Research Foundation (AMPORF), to convert the model code to Python, enhance, and maintain the software code. The new Python-based software is very flexible, configurable, and easier to administer.

In 2025, the administrative agent for the consortium changed from AMPORF to the Zephyr Foundation.[27] As of 2025, there are currently 14 member agencies in the consortium:

- Atlanta Regional Commission (ARC)
- Central Transportation Planning Staff (CTPS Boston)
- Chicago Metropolitan Agency for Planning (CMAP)
- Maricopa Association of Governments (MAG)
- Metropolitan Council (Met Council)
- Metropolitan Transportation Commission (MTC)
- Metropolitan Washington Council of Governments (MWCOG)/National Capital Region Transportation Planning Board (NCRTPB) (Washington, D.C.)
- Puget Sound Regional Council (PSRC)
- San Diego Association of Governments (SANDAG)
- San Francisco County Transportation Authority (SFCTA)
- Southeast Michigan Council of Governments (SEMCOG)
- TransLink (Vancouver)
- Transport for New South Wales (Sydney)
- Victoria Department of Transport and Planning (Melbourne)

The MTC Travel Model One has been fully implemented in ActivitySim. Model deployments are currently underway for member agencies like ARC (there are minor differences between the MTC and ARC models that are being implemented), SEMCOG, SANDAG, MTC, Met Council, PSRC, COG, and TransLink, as well as several agencies that are not members of the consortium, including Transport for New South Wales (Sydney, Australia), Southeast Regional Planning Model (SERPM, Southeast Florida), Perth Australia, Melbourne Australia, and the Lawrence Livermore National Laboratory in their Behavior, Energy, Autonomy, and Mobility (BEAM) model.[28] As can be gleaned from the above information, ActivitySim has a robust and active user community.

The ActivitySim model framework has the following characteristics:

---

[26] "ActivitySim: An Open Platform for Activity-Based Travel Modeling," 2020. https://activitysim.GitHub.io/, accessed April 23, 2020.

[27] Zephyr Foundation, "ActivitySim," with Joe Castiglione et al., 2026, https://zephyrtransport.org/ActivitySim/.

[28] Membership can change over time and is not limited to US-based transportation agencies. Examples are current as of September 2025.

- Utilizes tours (sequences of trips beginning and ending at an anchor location such as home or work) as an organizing principle for the generation of travel and to ensure consistency of information across trips within a tour.
- Utilizes demand micro-simulation for modeling travel choices, in which a representative population is generated, and explicit mobility and travel choices are simulated for each decision maker in the population according to contextual probability distributions.[29]
- Addresses both household-level and person-level travel choices including limited intra-household interactions between household members.
- Schedules tours into time-windows to ensure there are no overlapping travel episodes.
- Reflects and responds to detailed demographic information including household structure, household income, age, and other key attributes.

1.6 Description of Gen3 Model

The Gen3 Model runs in three phases – a preparation phase, a feedback loop phase, and a final phase. The preparation phase prepares pricing conversion data, highway network data, highway skim data,[30] and transit fares. The feedback loop phase prepares transit network and skim data, runs ActivitySim to model the internal person travel demand, prepares auxiliary travel demand data for special transportation markets (e.g., ground access to airports, visitors, trucks, external-internal, internal-external, and external/through traffic), assigns highway trips, and then prepares new highway skim data and feeds it back to the beginning of this process. This feedback-loop phase repeats four times with increasing sampling proportions of the region's households (specifically, with 250,000 households, 500,000 households, 1 million households, and finally all households for speed-feedback loop iterations 1-4), and iterations 2-4 use the highway skim data from the prior iteration as inputs. After the final feedback loop iteration (i.e., Iteration 4), the model assigns transit trips (including internal transit trips from ActivitySim and external transit trips prepared in an exogenous process), prepares a transit summary file, and prepares a summary of the ActivitySim output in a visualizer and in a "view-from-space" summary file.

Regional travel demand forecasting models (TDFMs) are typically divided into two parts: a demand-side model and a supply-side model. The demand-side model estimates or forecasts the demand for travel on the surface transportation network, typically for an average day or an average weekday. In aggregate, trip-based demand models, demand is typically estimated for the entire day. By contrast, in disaggregate, activity-based models, such as ActivitySim, the travel demand may be estimated for subsets of the day, such as 30-minute increments, even though those subsets are often aggregated to larger periods, such as the AM peak period or the midday period (see below for more discussion of this topic), for subsequent trip assignment.

The supply-side model represents the transportation networks that are available to the traveling public, typically a highway network and a transit network, and the conditions that would be found on the network for various times of day, such as the AM peak period or the midday period. There is

[29] In the travel modeling literature, a representative population is usually referred to as "synthetic population." In this document, the two terms are used interchangeably.

[30] "Skims" refer to travel time and cost information that has been accumulated from zone-to-zone paths that traverse the transportation network.

typically feedback between the demand and supply models, known as speed feedback, to ensure that the travel speeds used as an input to the demand model are the same as, or similar to, the travel speeds coming out of the supply-side model, which includes the effects of traffic congestion. For the Gen3 Travel Model, the demand-side model is primarily implemented in the open-source ActivitySim software platform. By contrast, the supply-side model is implemented in the proprietary Bentley Systems Cube software. It is common practice in the U.S. for urban areas that maintain TDFMs to use proprietary software for one or more components of the model, even though some components may also be open source.

Like many other activity-based travel models, the ActivitySim portion of the Gen3 Model is implemented as a model system that simulates interconnected travel-related choices made by individual households/persons throughout a typical day, such as long-term mobility choices, daily activity scheduling, tour-level choices, and trip-level choices. Figure 1 shows the general structure of the ActivitySim portion (demand side) of the Gen3 Travel Model. As noted in the figure legend, boxes with the red borderline indicate the seven models that were estimated as part of the Phase 1 development, and boxes with the yellow borderline highlight another seven models estimated as part of the Phase 2 development. Section 6.5 details the components of ActivitySim resident travel model.

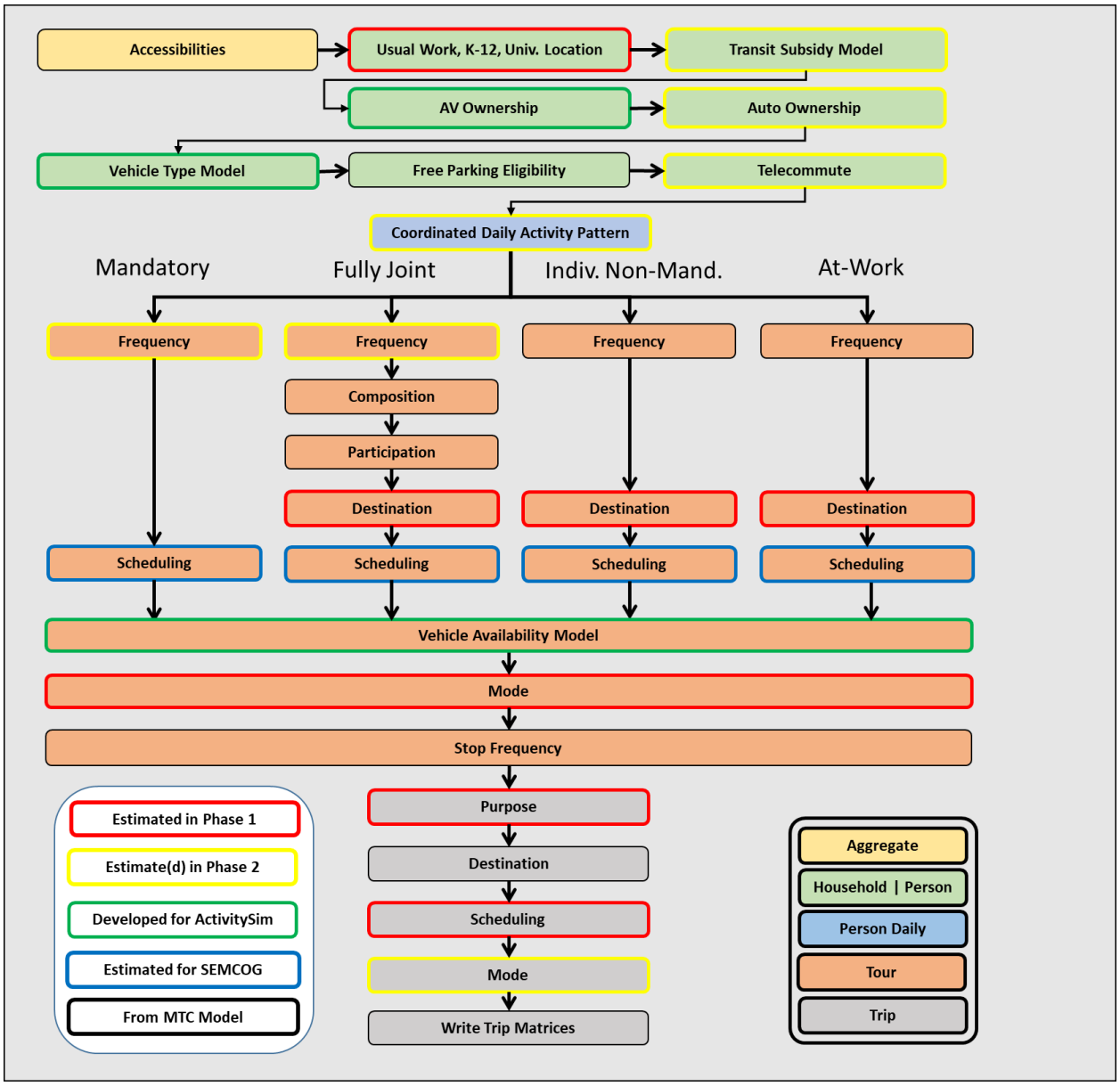


Figure 1: COG/TPB Gen3 ActivitySim Flowchart (demand-side model)

1.6.1 Treatment of time

ActivitySim in the Gen3 Model is set up with a 30-minute temporal resolution that begins at 3 A.M. and ends at 2:59 A.M. the following day. By contrast, the Gen2 (trip-based) Travel Model uses an average weekday divided into four time-of-day periods.

In the Gen3 Model, temporal integrity is ensured so that no activities are scheduled with conflicting time windows, except for short activities/tours that are completed within an hour increment. For example, a person may have a short tour that begins and ends within the 8 AM – 9 AM period, as well as a second longer tour that begins within this time period but ends later in the day.

A critical aspect of the model system is the relationship between the temporal resolution used for scheduling activities, and the temporal resolution of the network simulation (skimming and assignment). Although each activity generated by the model system is identified with a start time and end time in 30-minute increments, level-of-service matrices are created for four aggregate time-of-day periods consistent with the current Gen2 (trip-based) Model: AM Peak, Midday, PM Peak, and Night-Time (Evening/Early AM). Trips occurring in each time period can reference the appropriate level-of-service (skim) information on the transport network depending on their trip mode and time period (determined by the mid-point trip time). The definition of time periods for level-of-service matrices is given in Table 1, below.

Table 1: Time Periods for Level-Of-Service Skims and Assignment

NUMBER	DESCRIPTION	BEGIN TIME	END TIME
1	AM Peak	6:00 AM	8:59 AM
2	Midday	9:00 AM	2:59 PM
3	PM Peak	3:00 PM	6:59 PM
4	Night Time (Evening/Early AM)	3:00 AM & 7:00 PM	5:59 AM 2:59 AM next day

Although the temporal resolution of ActivitySim models is different from that of network simulation in the current Gen3 Model, a 30-minute temporal resolution will allow a more effective integration with a regional dynamic traffic assignment (DTA) model in the future.

1.6.2 Treatment of space

The Gen3 Model is set up to utilize Transportation Analysis Zones (TAZs) as the unit of geography in the model. The TAZs come from the current zone system used in the Gen2/Ver. 2.4.6 Model, which includes 3,722 TAZs. There are 3,669 internal TAZs in the Gen3 Model, numbered 1 through 3,675 (TAZs 2555, 2629, 3103, 3478, 3482, and 3495 are unused) and 47 external stations (3676 to 3722), which represent locations where trips enter and exit the modeled area, i.e., internal-to-external trips, external-to-internal trips, and external-to-external (through) trips. The TAZs are shown in Figure 2, with stars representing the external stations. Although ActivitySim can also operate at a finer geography, such as Micro-Analysis Zones (MAZs), it was the project team's decision to keep TAZs as the smallest geography in the Gen3 Model, mainly because COG's Cooperative Land Use Forecasts data are currently generated at the TAZ level.

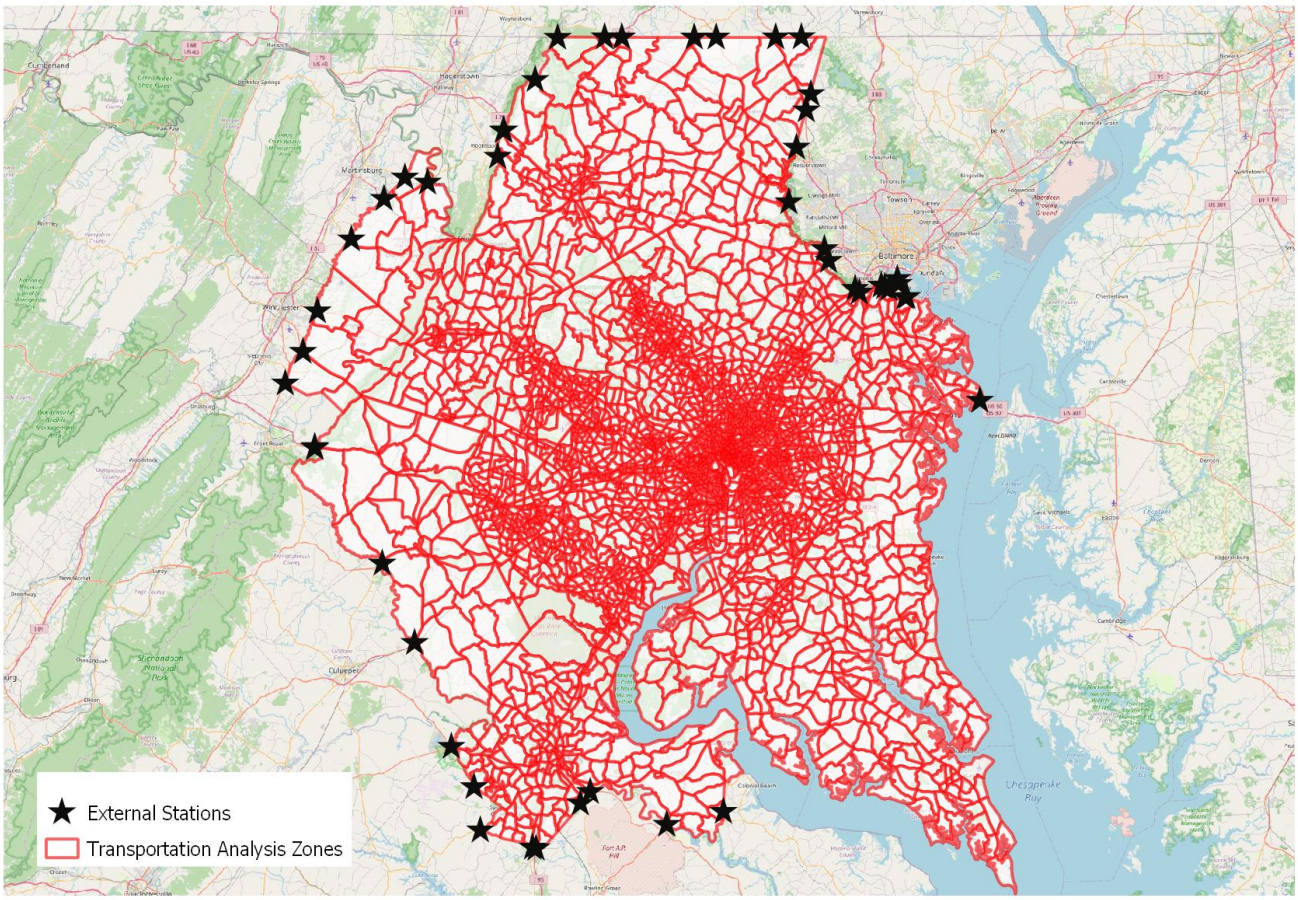


Figure 2: Gen3 Model Transportation Analysis Zones and External Stations

1.7 Differences between the Gen2 Model and Gen3 Model

The primary difference between the TPB's current production-use Gen2/Ver. 2.4.6 Model and the Gen3 Model is that the Gen2/Ver. 2.4.6 Model is a trip-based model that forecasts the trips from each zone, based on household characteristics, and aggregates trips for the purposes of trip distribution and mode choice. By contrast, the Gen3 Model is an activity-based model that micro-simulates travel at the person level ("demand micro-simulation") and associates travel-related decisions with more specific household- and person-level data such as household income, ages of people in the household, person type (i.e., full-time worker, part-time worker, university student, retired person, nonworking adult, driving-age child, school-age pre-driving-age child, and preschool-age child), work or school locations, if a worker is telecommuting that day, and even interactions among household members. The TPB Model can have four types of updates: a bug fix, a new feature, a feature enhancement, and updated documentation. Major Gen3 Model updates relative to the Gen2 Model are shown in Table 2.

Table 2: Major Gen3 Model Updates relative to Gen2 Model

#	DESCRIPTION	TYPE OF UPDATE	FURTHER DETAILS AND BENEFIT(S)	CHANGES MODEL RESULTS?
1	Replacement of trip-based demand model for residents with ActivitySim	New Feature	New model sensitivities in terms of household and person variables, sensitivity to changes in network level of service by time-of-day, representation of telecommuting, transit subsidies, park-and-ride versus kiss-and-ride, ride-hail modes and autonomous vehicles.	Yes
2	Representation of external-internal transit travel	New Feature	Used transit on-board survey data to generate trip tables representing non-resident travel into and out of modeled area.	Yes
3	Updated transit skimming procedures	New Feature	Replaced Cube TRNBUILD with Cube Public Transport (PT) module to generate multi-path transit skims with post-processing to ensure transit paths are more consistent with mode choice model alternatives.	Yes
4	Representation of external-internal highway travel	Feature Enhancement	Used a new methodology to estimate internal-external and external-internal trips.	Yes
5	Addition of Activity-Based Model Visualizer	New Feature	Created a dashboard showing model results and compares them to the expanded household survey or a base model run.	No

2 SYSTEM ARCHITECTURE

This section discusses the computer hardware and software requirements for running the travel model. In general, a powerful computer server is required to run the model. Specific hardware specifications are discussed in section 2.1. The main software required for the model includes Cube Voyager, ActivitySim, Python, and R. Cube Voyager is a commercial transportation modeling platform that must be purchased from Bentley Systems and requires a computer with a Windows operating system. Python is an open-source cross-platform programming language that is currently one of the most popular programming languages. Python is the core language of ActivitySim. R is an open-source cross-platform statistical analysis and scripting language. In the Gen3 Model, R is used for the model visualizer. The required software is discussed further in section 2.2.

2.1 Hardware requirements

The COG/TPB Gen3 Model runs on a Microsoft Windows workstation or server, with the minimum and recommended system specifications described as follows:

- Minimum specifications:
  - Operating System: 64-bit Windows 8 (8.1), 64-bit Windows 10, 64-bit Windows Server 2019
  - Processor: 16-core CPU processor (16 logical/virtual cores)
  - Memory: 206 GB RAM
  - Disk space: 500 GB
- Recommended specifications:
  - Operating System: 64-bit Windows 10, 64-bit Windows 11, or 64-bit Windows Server 2019[31]
  - Processor: 3rd Generation Intel Xeon Scalable (Skylake) or a comparable CPU processor with at least 32 logical/virtual cores.
  - Memory: 256 GB RAM
  - Disk space: 1200 GB

A model run conducted on a computer with the minimum specifications (using 16 cores) will yield slightly different results compared to the same run on a recommended high-spec computer (32 cores), due to the different number of cores utilized by Cube Cluster. This variation occurs because Cube Cluster's parallel processing, conducted via distributed processing, introduces minor differences in outputs when the number of processing cores varies.[32] The model limits highway assignment processing to a maximum of 32 cores, even on computers with more available cores, to minimize output variations. Additionally, increasing the number of cores beyond this point yields diminishing returns in highway assignment run times.

---

[31] Windows Server also comes in versions for 2022 and 2025, but COG/TPB staff have not tested them with the Gen3 Model.

[32] This is noted in the Bentley Systems Cube Voyager documentation. See for example, p. 1080 of "Cube Voyager Reference Guide, Version 6.5.1." Bentley Systems, Inc., March 17, 2023: "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." (p. 1080).

In general, a higher number of CPUs and a larger amount of memory (RAM) will result in faster runtimes, as long as ActivitySim is configured to utilize additional processors and RAM. Note that the model is unlikely to run on servers that have less than 206 GB of RAM. Prior to 2023, COG/TPB staff conducted all of the modeling work on COG's on-premises travel modeling servers, such as "tms5" and "tms8." As of 2023, COG/TPB staff had moved all the modeling data and the majority of the Gen2/Gen3 modeling work to cloud computing, specifically Amazon Web Services (AWS) on-demand cloud servers. The computer specifications of COG travel modeling servers (including both on-premises and cloud servers) are shown in Table 3 and in a technical memorandum.[33] The on-premises server (tms8) is reserved as a backup server.

[33] Ray Ngo, "Computer Specifications of Travel Model Servers (on-Premises and Cloud Servers)," Memorandum to Feng Xie et al., April 3, 2025.

Table 3 Specifications Of Travel Model On-Premises and AWS Cloud Server

HOST NAME:	TMS8 (ON-PREMS)	AWS CLOUD HIGH-INSTANCES
System Model:	ProLiant DL380 Gen10	R8i.8xlarge
Cost (As of 4/1/2025)	NA	$3.6950 hourly
Purchase date	May or June 2018	
OS Name:	Windows Server 2019 Standard	Windows Server 2019 DataCenter
OS Version:	10.0.17763	10.0.17763
System Manufacturer:	HP	NA
System Type:	64-bit	64-bit
Number of processors:	2	1
Processor name(s):	Intel Xeon Gold 6146	3rd Generation Intel Xeon Scalable (Skylake)
Clock speed of processor (GHz):	3.2	3.9
No. of physical cores per processor:	12	16
No. of threads (virtual cores) per processor:	24	32
Total number of physical cores:	24	16
Total number of threads:	48	32
Hyper-Threading Technology:	yes	yes
Total Physical Memory (GB):	256	256
Hard Disk Drives: No. of Drives	6	NA

2.2 Software requirements

The installation of ActivitySim will be discussed in the next section as part of the model setup. In addition to ActivitySim, two software applications, Cube and a Python Package Manager (e.g., UV for Gen3/Version 1.1.0 and Miniforge3/Anaconda for prior model versions) should be installed on the computer that will be used to run the model.

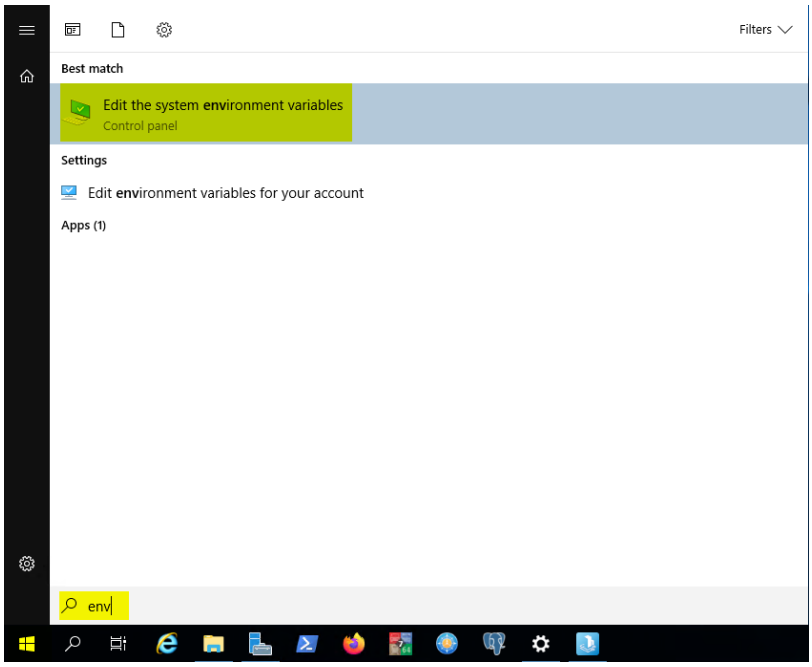
The Gen3 Model system is an integrated model that is controlled by and primarily runs in the Cube transportation planning software platform. Bentley Systems, Inc. produces and markets Cube. Cube is a proprietary transportation planning software platform that includes Cube Base and Cube Voyager. Cube Base allows for viewing and editing highway and transit network files and viewing of matrix files. These files are critical for the transportation-supply side of the model. The Gen3 Model currently uses Cube Connect Edition, version 6.5.1[34] as default, but can also run on OpenPaths Cube 2025, version 25.00.01. Note that, with Cube Connect and OpenPaths Cube versions of the Cube software (i.e., versions of Cube downloaded from Bentley's website after Bentley purchased Cube), users will need to log into and activate the Bentley Connect Edition software after any server reboots.

[34] Bentley Communities Legacy Member, "Check out what's new in CUBE CONNECT Edition (v 6.5)!", Bentley Communities Blog, 2020, https://bentleysystems.service-now.com/community?id=community_blog&sys_id=cd3acbd347718690e3378d53636d435e

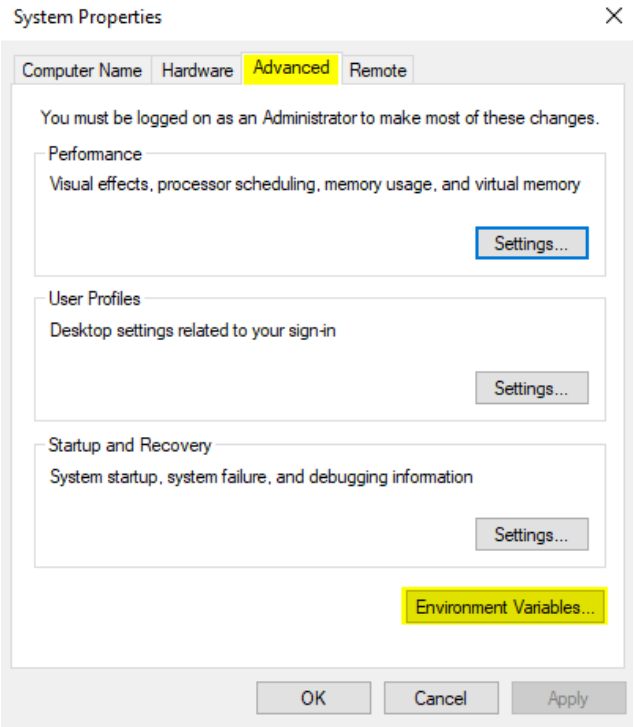
A further caveat of using OpenPaths Cube involves active ClusterManager.exe instances. The Gen3 Model makes use of the Cube Cluster utility during Highway Assignment to reduce the model runtime. This process is managed by the ClusterManager.exe software. However, staff noticed that ClusterManager.exe is not automatically closed after using OpenPaths Cube software. This would result in a crash in the Highway Assignment step if one model user uses Cube and leaves a ClusterManager instance open and then another user launches a Gen3 Model run on the same computer that tries to access the instance created by the previous user. Staff shared this finding with Bentley staff. As a fix, the model now includes a function that attempts to close any existing ClusterManger.exe process at the beginning and at the end of a run. However, this function may not be able to close ClusterManger.exe instances started by other users on the same server if the user launching the model run does not have administrator privileges. In this case, running the batch file from an elevated Command Prompt (Run as Administrator) or having ClusterManger.exe closed by an admin or its owner should be done before running the model.

If users choose to conduct their Gen3 Model runs with OpenPaths Cube 2025, version 25.00.01, it is required to add the software path (*C:\Program Files\Bentley\OpenPaths\CUBE 25.00.01\Voyager)* to the computer's PATH system environment variable because it is not automatically done. Below are the steps needed to add the software path:

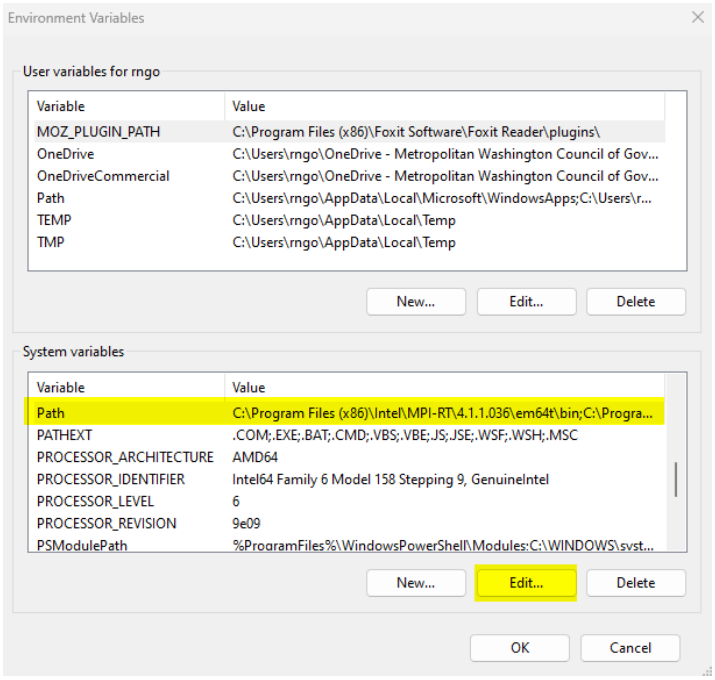
Step 1: Open system properties by pressing the Windows key, type *env*, and select *Edit the system environment variables* (or right-click This PC or Computer on your desktop (or in File Explorer) and choose Properties. Then click Advanced system settings.)



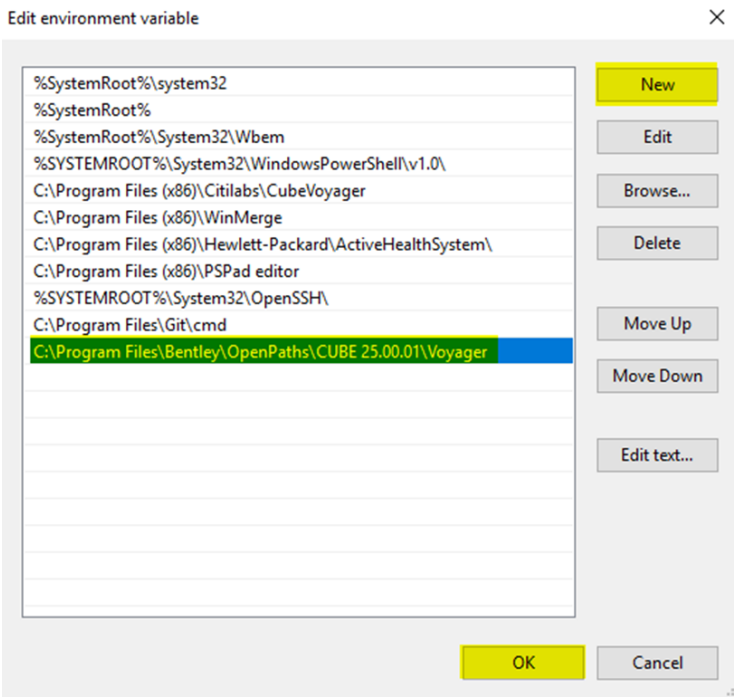
Step 2: Open environment variables: In the System Properties window, click the Environment Variables



Step 3: Find system Path: In the System variables section, scroll to find the variable named Path. Select it and click Edit



Step 4: Add OpenPaths Cube path: In the Edit Environment Variable window, click New and paste *C:\Program Files\Bentley\OpenPaths\CUBE 25.00.01\Voyager, then OK on all dialogs to save your changes and close out*



2.2.1 Package manager software

A Python package manager is software that creates a software environment for an instance of Python. Instance refers to the libraries used by the Python processor in the computing environment. ActivitySim and related Python processes in the Gen3 Model are executed in an environment that is setup with a specific version of Python and specific library versions. This ensures that changes outside of the Python environment will not cause errors or alter model results, and that the specific versions of Python and specific libraries needed by the Gen3 Model and ActivitySim do not cause errors or changes to other Python software installations on the server. The Python libraries needed by ActivitySim extend the base functionality of Python.

Prior to Version 1.1.0, the Gen3 Model used Conda as the package and environment manager. Two optional Conda distributions are provided in the Gen3 Model configuration: Miniforge3 and Anaconda. In general, Miniforge3 is preferred because it is free for organizations of all sizes, while Anaconda requires a paid subscription for organizations of more than 200 employees.[35] The Gen3 Model was initially developed with Anaconda but was later changed to work with Miniforge3 and, at the time of the Gen3 Model beta release (Version 1.0.5), both package managers are supported (while Miniforge3 is the default, a user can switch to Anaconda by changing the setting in the main batch file). Model results are not affected by the choice of package manager, only a small increase in runtime was observed with Miniforge3.

[35] Anaconda, Inc. “Anaconda Terms of Service.” 2025. https://www.anaconda.com/pricing/terms-of-service-faqs

In the Gen3/Version 1.1.0 Model, COG/TPB staff has replaced MiniForge3 and Anaconda with a new package manager called UV. Compared to Miniforge3 and Anaconda, UV enables faster installation and makes the model setup much more straightforward. Both the UV software and the "gen3_model" environment can be installed by executing a batch file following the instruction in the "Read_Me.md" file. More installation information can be found in the next section.

2.3 ActivitySim installation and model setup

To set up the Gen3 Model, a user needs to download the travel model along with the model inputs and dependencies, install ActivitySim, and configure the main batch file in a scenario folder. These steps are described in detail below.

2.3.1 STEP 1: Download the Gen3 Model

Download the model from the link provided by COG/TPB staff. Unzip the model and browse to the /Gen3_Model_* folder. There are at least four sub-folders in this directory: /documentation , /source, /support. and /2018_base. The detailed description of these folders can be found in Section 2.4.

There may be additional folders related to specific model scenarios. As part of the validated model, the 2018_base folder is expected to exist since 2018 was the model base year for calibration and validation. Other folders may exist as well, such as "2025" or "2050". There is no requirement that the model folder name must include the year, but it is a good practice to do so since the files included in the inputs folder are generically named. Using the year and a scenario description as part of the folder name (e.g., "2025_Build") reduces the potential for confusion related to what inputs and outputs are included in a scenario folder.

The downloaded model package should contain all the model files that are needed to run the Gen3 Model, including the inputs files and dependencies.

2.3.2 STEP 2: install and configure ActivitySim

Install the Python Package Manager UV by launching the gen3_model151_installation.bat file located in the */source/software/gen3_model_uv* folder, per the instruction in the "Read_Me.md" file. If the installation fails, please contact the IT department to resolve potential security policy restrictions. Each user must run the batch installation individually, as the setup is per-user.

The batch file that is used to install UV also creates a specific computer environment used to run ActivitySim. The environment is a configuration of Python that is created specifically for ActivitySim - this environment allows ActivitySim to use specific software libraries without interfering with the server's installed version of Python (if one exists, which is not required) and keeps other Python installations from interfering with ActivitySim.

Once Cube, UV and the environment are installed, the model user should be able to run the Gen3 Model out of the box using the default configurations. To modify model configurations to your needs, please refer to the instruction in the next section.

2.3.3 STEP 3: Configure the main batch file

The main batch file, run_model.bat, which can be found under the root directory of each scenario folder, can be configured before executing a Gen3 Model run. The batch file is divided into four sections: USER SPECIFICATIONS, VISUALIZER SETTINGS, DEFAULT MODEL SPECIFICATIONS, and MODEL EXECUTION. A model user usually needs to change only the configurations in the first section to set up a model run. The user can also change the configurations in VISUALIZER SETTINGS to compare the results from two model runs in the Visualizer program. The model configurations in the last two sections are not intended to be modified by the user.

Currently, the **USER SPECIFICATIONS** section provides the following configuration options. Some of those options are scenario specific, while others are one-time configurations that are not dependent on scenario.

- cubeversion: Look for this option in ":: STEP ONE: Choose the version of Cube". This option allows model users to choose between Cube CE 6.5.1 and OpenPaths Cube 2025 Version 25.00.01 to run the Gen3 Model. Note that the current Gen3 Model is designed to primarily work with Cube CE 6.5.1. Although it can now run with Cube 2025, its input and output files are not optimized with the new cube software.
- _year_ Look for this option in ":: STEP TWO: Set model year variable". This option is used to specify the model year associated with a specific scenario.
- PYENVNAME: Look for this option in ":: STEP THREE: Set Python environment name". This option specifies the Python environment used to run the model. The specified environment needs to be installed following the instructions in Step 2 (or README.md). Keep the default value (gen3_model151) for this variable.
- PYTHON: Look for this option in ":: STEP FOUR: Set PYTHON path". This variable points to the Python executable that is installed as part of the Python environment to run the Gen3 Model. Keep the default value (C:\Users\%USERNAME%\%PYENVNAME%\.venv\Scripts\python.exe) for this variable.
- RUN_MINI: Look for this option in ":: STEP FIVE: Set RUN_MINI to True to skip Iterations 2 to 4 of the model". This option is used for a short model run that runs only the first feedback loop out of the four feedback loops and can be used for debugging and functionality testing. By default, this is set to False, set to True to activate this feature.
- AUTO_SHUTDOWN: Look for this option in ":: STEP SIX: Set AUTO_SHUTDOWN to True to automatically shut down a server at the end of a model run". This option activates an automatic shutdown of the computer where a model is run when the model completes the run, which is ideal when using on-demand cloud servers, where charges accrue by the time used. By default, this is set to False, set to True to activate this feature. This option will shut down the server regardless of whether the model run was successful. It is up to the analyst to check the model output logs to ensure that the model run was successfully completed. It is also up to the analyst to ensure that other users are not using the server prior to activating an automatic shutdown (Although an on-demand cloud server precludes the possibility of concurrent users, an on-premises server does not).

- TOLL_SETTING: Look for this option in ":: STEP SEVEN: Set TOLL_SETTING to True to turn on the toll setting processing". This option turns on toll setting, which re-estimates the toll rates on variably priced tolling facilities in an iterative process. When the option is set to False, which is the default, the model will use COG-provided toll inputs. Note that turning on this option will at least double the model runtime, as it involves a "pump prime" run that sets up for toll setting, an iterative toll setting process, and a "final" run that produces final travel forecasts based on estimated toll rates.

VISUALIZER SETTINGS can also be configured for ActivitySim output visualization:

- IS_BASE_SURVEY: Look for this option in ":: STEP EIGHT: Set IS_BASE_SURVEY to Yes or No to compare the ActivitySim output to the survey or a previous run". By default, this option is set to "Yes" to compare the current model run to the household survey and Census data for calibration and validation purposes. If it is set to "Yes" by default, STEP NINE will be skipped; If this option is changed to "No", the related variables will need to be specified in STEP NINE, as instructed below.
- BASELINE_SCENARIO_PATH, BASELINE_SCENARIO_NAME, and ALTERNATIVE_SCENARIO_NAME: Look for these options in ":: STEP NINE: Comparison setting". In the case that the model output should be compared to a different base condition from another model run (for example, comparing the model output from a Build Scenario to that of a No Build Scenario), there are four environment variables that need to be updated in the run_model.bat file. First, set IS_BASE_SURVEY in STEP EIGHT to "No". Then proceed to STEP NINE and specify the three remaining variables: BASELINE_SCENARIO_PATH, BASELINE_SCENARIO_NAME, and ALTERNATIVE_SCENARIO_NAME. BASELINE_SCENARIO_PATH should be a path to the scenario of a previously completed model run, for example, "D:\ModelRuns\MWCOG_Gen3_Model\2018_base". BASELINE_SCENARIO_NAME is the name of the existing run, for example, "2018Base". And ALTERNATIVE_SCENARIO_NAME is the name of this current run, for example, "2018Build". Please note that the two scenario name variables are used to define the filename of the HTML visualizer output file.

2.4 Directory structure

The contents of the Gen3 Model folders are described in more detail below.

- **/.github**: This is a folder used with Git to manage source code. For model users who want to work with the Gen3_Model GitHub repository, DO NOT modify any files in this folder.
- **/2018_base**: This is the 2018_base scenario folder. This folder can be renamed by the user. It can also be copied within the /Gen3_Model directory to create additional scenarios. This folder contains an inputs folder, an outputs folder, and two batch scripts to run the model. Since 2018 is the model base year, the inputs folder contains input files that were used for model calibratioin/validatioin.
- **/Documentation**: Model documentation is included in this folder.
- **/Source**: This folder contains the scripts and settings files to run various components of the model. There are several subfolders in this folder. The configs subfolder contains the ActivitySim configuration files. The scripts subfolder contains the batch file scripts, Cube scripts and Python scripts used to run the model. The software subfolder contains the

supporting software executables. The visualizer subfolder contains the ABM Visualizer setup and survey data.

- **/Support**: This folder contains supporting data such as speed lookup files, volume-delay function parameters, toll equivalency parameters, and other parameters used in the model. These files are listed in Table 4. The user does not usually change any of these files.

Figure 3: COG/TPB Gen3 Model File Structure

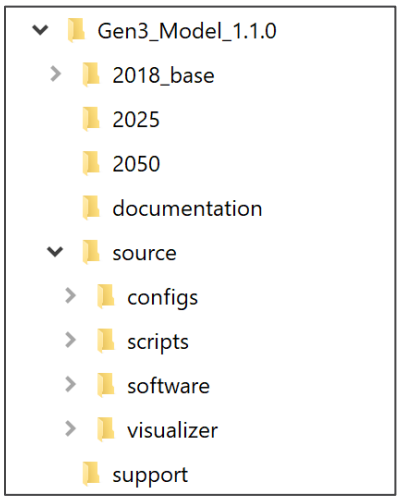


Table 4: Model Supporting Data Files

FILENAME	DESCRIPTION
AM_SPD_LKP.txt	Peak-period speed lookup file
cvdelta_3722.trp	Commercial vehicle calibration adjustment tables (used in auxiliary time-of-day script)
equiv_toll_min_by_inc.s	Equivalent toll minutes by income script file
hwy_assign_capSpeedLookup.s	Script to apply speed-capacity lookup
hwy_assign_conical_vdf.s	Script to apply volume-delay functions
MD_SPD_LKP.txt	Off-peak period speed lookup file
tkdelta_3722.trp	Truck calibration adjustment tables (used in auxiliary time-of-day script)
todixxi_2018HTS.dbf	Time-of-day factors by purpose, mode, direction, and time period
Toll_Minutes.TXT	Equivalent toll minutes lookup script
Truck_Com_Trip_Rates.DBF	Truck vehicle calibration adjustment tables (used in auxiliary time-of-day script)
True_Shape_2050Links_Viz2050.shp (and dbf, shx, sbx, sbn, prj)	Network true-shape display file used in Cube when viewing network files. This file is solely for viewing the network and is not used during the model run. This file may change names (e.g. True_Shape_2045) based on COG/TPB's long-range plan.
Ver23_f_factors.dbf	Trip distribution friction factors (used in some of the auxiliary trip models)

3 DATA INPUTS & OUTPUTS

3.1 Model inputs

The model inputs folder contains six sub-folders that organize the inputs into auxiliary data, highway, land use, representative population (synthetic population), and transit. These folders are discussed in detail below.

3.1.1 Auxiliary data

The auxiliary input folder, “auxiliary”, contains travel demand inputs that are not simulated directly by the travel demand model. This auxiliary travel data, also known as exogenous or residual travel, includes airport passenger auto-driver trips, external trips, truck trips, commercial vehicle trips, special-generator taxi trips, visitor/tourist trips, and external/through trips (X-I, I-X, and X-X). In all cases except external trips, TPB staff use separate, pre-processing procedures to forecast the demand for these markets. External transit trips are derived from base-year transit on-board survey data and are held constant in the forecast years until new survey data becomes available. External auto-driver trips, truck trips, and commercial vehicle trips are estimated in the model based on inputs in this folder.

Table 5: Auxiliary Folder Input File List

FILENAME	DESCRIPTION
airpax.adr	Air passenger auto-driver trips. Cube matrix with one unnamed table representing vehicle trips.
Ext_PsAs.dbf	External productions and attractions (autos, trucks, commercial vehicles). See Table 6.
taxi.adr	Special-generator taxi auxiliary trips. Cube matrix with one table named TaxiAdr_{modelyear} representing vehicle trips.
visi.adr	Visitor/tourist auxiliary trips. Cube matrix with one table named VisiAdr_{modelyear} representing vehicle trips.
xxaut.vtt	External/through auto-driver trips. Cube matrix with one unnamed table representing vehicle trips.
xxcvt.vtt	External/through trips for commercial vehicles and trucks. Cube matrix with three unnamed tables representing vehicle trips for commercial vehicles (CVs), medium trucks, and heavy trucks.

Table 6: External Productions and Attractions (Ext_PsAs.DBF) File Format

NAME	LENGTH	DESCRIPTION
TAZ	Integer	Transportation Analysis Zone Number (3676-3722)
FACILITY	Character	Facility Name
AAWT_CTL	Integer	Average Annual Weekday Traffic
CNTFTR	Decimal	(Unused)
AUTO_XI	Integer	Auto External-Internal Vehicle Trips
AUTO_IX	Integer	Auto Internal-External Vehicle Trips
AUTO_XX	Integer	Auto External Through Vehicle Trips
CV_XX	Integer	Commercial Vehicle External Through Vehicle Trips
HBW_XI	Integer	Home-based Work External-Internal Vehicle Trips
HBS_XI	Integer	Home-based Shopping External-Internal Vehicle Trips
HBO_XI	Integer	Home-based Other External-Internal Vehicle Trips
NHB_XI	Integer	Nonhome-based External-Internal Vehicle Trips
CV_XI	Integer	Commercial-Vehicle External-Internal Vehicle Trips
HBW_IX	Integer	Home-based Work Internal-External Vehicle Trips
HBS_IX	Integer	Home-based Shopping Internal-External Vehicle Trips
HBO_IX	Integer	Home-based Other Internal-External Vehicle Trips
NHB_IX	Integer	Non-home-based Internal-External Vehicle Trips
CV_IX	Integer	Commercial-Vehicle Internal-External Vehicle Trips
TRCK_XX	Integer	Total Truck External Through Vehicle Trips
TRCK_XI	Integer	Total Truck External-Internal Vehicle Trips
TRCK_IX	Integer	Total Truck Internal-External Vehicle Trips
MTK_XI	Integer	Medium Truck External-Internal Vehicle Trips
HTK_XI	Integer	Heavy Truck External-Internal Vehicle Trips

3.1.2 Highway data

The highway data contained in the “hwy” folder provides the input highway (road) network data for use with highway path building, highway skimming, highway assignment, and transit network processing. More details associated with the highway and transit network input data can be found in the network report that accompanies this User’s Guide.

Table 7: Highway Folder Input File List

FILENAME	DESCRIPTION
AM_Tfac.dbf	AM Period Toll Factors. See Table 8.
CPI_File.txt	Consumer Price Index (CPI) inflation factors. This file is a script that is input as source code during the model run.
Link.DBF	Database file of all links in the COG highway network. See Table 9.
MD_Tfac.dbf	Mid-day (MD) Period Toll Factors. See Table 8.
Node.dbf	Database of all nodes in the COG highway network. See Table 10.
NT_Tfac.dbf	Night-time (NT) Period Toll Factors. See Table 8.
PM_Tfac.dbf	PM Period Toll Factors. See Table 8.
Toll_Esc.dbf	Toll escalation factors. See Table 11.

Xtrawalk.txt	Extra walk links added to the highway network for use with the transit network processes.

Table 8: Toll Factor Input File Format

NAME	TYPE	DESCRIPTION
TOLLGRP	Decimal	Toll Group Number
[AM| MD| PM| NT]SOVTFTR	Decimal	Single Occupant Vehicle Toll Factor
[AM| MD| PM| NT]HV2TFTR	Decimal	Shared-Ride 2 Person Toll Vehicle Factor
[AM| MD| PM| NT]HV3TFTR	Decimal	Shared-Ride 3+ Person Toll Vehicle Factor
[AM| MD| PM| NT]COMTFTR	Decimal	Commercial Vehicle Toll Vehicle Factor
[AM| MD| PM| NT]TRKTFTR	Decimal	Truck Toll Vehicle Factor
[AM| MD| PM| NT]APXTFTR	Decimal	Airport Toll Vehicle Factor

Table 9: Link.DBF File Format

NAME	TYPE	DESCRIPTION
OBJECTID	Decimal	Internal Indexing Field
A	Decimal	A (From) Node Number
B	Decimal	B (To) Node Number
DISTANCE	Decimal	Length (in Hundredths of Miles)
JUR	Decimal	Jurisdiction Number (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	Decimal	Screenline Number (if applicable) (1-38)
FTYPE	Decimal	Facility Type 0 Centroid Connectors 1 Freeway 2 Major Arterial (Non-Freeway) 3 Minor Arterial 4 Collector 5 Expressway 6 Ramp
TOLL	Decimal	Toll Value in Cents (Current-Year Prices)
TOLLGRP	Decimal	Toll Group Code (1-9999)
AMLANE	Decimal	Number of lanes during the AM period
AMLIMIT	Decimal	AM Peak Limit Code (0-9)
PMLANE	Decimal	Number of lanes during the PM period
PMLIMIT	Decimal	PM Peak Limit Code (0-9)
OPLANE	Decimal	Number of lanes during the MD and NT period
OPLIMIT	Decimal	MD and NT Peak Limit Code (0-9)
EDGEID	Decimal	Geometric network link identifier
LINKID	Decimal	Logical network link identifier
NETYEAR	Decimal	Planning year of network link
SHAPE_LENG	Decimal	Geometric length of network link (in feet)

PROJECTID	Character	Project identifier
TRANTIME	Decimal	Transit time in minutes, used only in Cube Public Transport (PT) processes
WKTIME	Decimal	Walk time in minutes, used only in PT
MODE	Decimal	Travel mode identifier, used only in PT
SPEED	Decimal	Travel speed, used only in PT
STREETNAME	Character	Name of street

Table 10: Node.DBF FIle Format

NAME	TYPE	DESCRIPTION
N	Integer	Node Number
X	Integer	X Coordinate of node
Y	Integer	Y Coordinate of node
MRFZONE	Integer	Metrorail Station Fare Zone
CRMFZONE	Integer	MARC Station Fare Zone
CRVFZONE	Integer	VRE Station Fare Zone

Table 11: Toll_Esc.DBF File Format

NAME	TYPE	DESCRIPTION
TOLLGRP	Integer	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	Decimal	Deflation factor override
DSTFAC	Decimal	Distance (per mile) based toll factor in present year cents/dollar (optional)
AM_TFTR	Decimal	AM period toll factor
PM_TFTR	Decimal	PM period toll factor
OP_TFTR	Decimal	MD and NT period toll factor
AT_MIN	Decimal	Area Type minimum override (optional)
AT_MAX	Decimal	Area Type maximum override (optional)
TOLLTYPE	Decimal	Toll Type 1 = Fixed tolls 2 = Variably priced tolls

3.1.3 Land use data

The land use data folder, “landuse”, includes land use and socioeconomic data inputs for the model.

Table 12: Files Contained in the Land Use Data Folder

FILENAME	DESCRIPTION
AT_override.TXT	File for overriding the computed area types of certain zones (e.g., used for Arlington National Cemetery and The Pentagon)
land_use.csv	Zonal data file for ActivitySim. See Table 13.

Table 13: Land Use and Zonal Data File Format

NAME	TYPE	DESCRIPTION
TAZ	Integer	Zone ID (1-3722)
HH	Integer	Number of households
HHPOP	Integer	Household population
GQPOP	Integer	Group quarters population (non-institutional GQ population only)
TOTPOP	Integer	Total population
TOTEMP	Integer	Total employment
INDEMP	Integer	Industrial employment
RETEMP	Integer	Retail employment
OFFEMP	Integer	Office employment
OTHEMP	Integer	All other employment
JURCODE	Integer	Jurisdiction code (county or city)
LANDAREA	Decimal	Land Area of TAZ in square miles
TAZXCRD	Integer	TAZ X Coordinate
TAZYCRD	Integer	TAZ Y Coordinate
K_8	Integer	Kindergarten - 8th grade enrollment
G9_12	Integer	High school enrollment
COLLEGE	Integer	College/university enrollment
Park_Acres	Decimal	Park area of TAZ in acres
GC_Acres	Decimal	Golf course area of TAZ in acres
PRKCST	Decimal	8-hour parking cost in TAZ (daily parking for work)
OPRKCST	Decimal	Hourly off-peak parking cost in TAZ
TERMINAL	Integer	Terminal time for trip-ends in TAZ, in minutes
AREATYPE	Integer	Area type of TAZ

The information in the first fourteen columns ("TAZ" to "TAZYCRD") is derived from COG's Cooperative Forecasts of Land Use data. School enrollment forecasts by grade level ("K_8", "G9_12") are developed by COG/TPB using an approach that was suggested by the consultant.[36] College enrollment ("COLLEGE") is derived from the model-base-year college enrollment data and assumed to be constant in the future. Park areas and golf course areas ("Park_Acres", "GC_Acres") are developed by COG/TPB's GIS team using a base-year geo-spatial database and are assumed to be constant in the future. Parking costs at primary tour destination ("PRKCST", "OPRKCST"), highway terminal time ("TERMINAL") and zonal area type ("AREATYPE") are not really model input data. They

[36] Feng Xie, "SCHOOL ENROLLMENT PROJECTION FOR GEN3, PHASE 2, MODEL: Proposed Methodologies", COG Travel Forecasting Subcommittee (TFS) Meeting, Item 3b, March 20, 2022. https://www.mwcog.org/file.aspx?&A=5%2fmBWnNDbJvWrpRQ5T59Si7SWqXRzTDsq2FEVgSy3Ws%3d

are land use related variables that are computed and appended to the land use input file during a model run.

3.1.4 Synthetic population data

The synthetic population folder, "popsyn", includes two data files of the synthesized population and household input files. These are inputs to ActivitySim.

Table 14: Files contained in the Synthetic Population Data Folder

FILENAME	DESCRIPTION
combined_synthetic_hh.csv	Synthetic residential and group quarters household listing from the population synthesizer. See Table 15.
combined_synthetic_per.csv	Synthetic residential and group quarters person listing from the population synthesizer. See Table 16.

Activity-based travel models, such as those implemented in the Gen3 Travel Model, require a representative population (a.k.a., synthetic population) as an input. The synthetic population data includes a household file and a person file. The attributes in these files are provided in Table 15 and Table 16 and are a result of the best fit for the controlled attributes of the population. For more information, see the COG Population Synthesizer Report.[37] The procedure for generating a synthetic population is not run by default but should be re-run any time the user changes the number of households or persons in one or more TAZs. The procedure for generating the synthetic population is described in Section 6.4.

Table 15: Fields used in the Synthetic Households File

NAME	TYPE	DESCRIPTION
household_id	Integer	Household ID number
puma_geoid	Integer	Household Public Use Microdata Area (PUMA) Geographic ID
TAZ	Integer	Household TAZ ID
TYPE	Integer	Type of housing unit 1 = Household 2 = Institutional group quarters 3 = Noninstitutional group quarters
NP	Integer	Number of people in household
VEH	Integer	Number of vehicles in household

[37]RSG. MWCOG Population Synthesizer. Final Report. Metropolitan Washington Council of Governments, National Capital Region Transportation Planning Board, August 4, 2021.https://www.mwcog.org/assets/1/6/MWCOG_Population_Synthesizer_COG_final.pdf

HHT	Integer	The type of household - Census PUMS HHT variable: N/A: Not applicable (group quarters or vacant) 1: Married couple household 2: Other family household: Male householder, no spouse present 3: Other family household: Female householder, no spouse present 4: Nonfamily household: Male householder: Living alone 5: Nonfamily household: Male householder: Not living alone 6: Nonfamily household: Female householder: Living alone 7: Nonfamily household: Female householder: Not living alone
hhincadj	Integer	The annual household income in the household, adjusted to the model base year
workers	Integer	The number of workers in the household
has_children	Boolean	1 if household has children

Table 16: Fields used in the Synthetic Persons File

NAME	TYPE	DESCRIPTION
PUMA_GEOID	INTEGER	HOUSEHOLD PUMA GEOGRAPHIC ID
TAZ	INTEGER	HOUSEHOLD TAZ ID
HOUSEHOLD_ID	INTEGER	HOUSEHOLD ID NUMBER (MUST AGREE WITH HOUSEHOLD FILE)
PER_NUM	INTEGER	NUMBER OF PERSONS IN HOUSEHOLD
AGEP	INTEGER	AGE OF PERSON IN YEARS
SEX	INTEGER	SEX OF PERSON IN HOUSEHOLD (1 = MALE, 2 = FEMALE)
WKHP	INTEGER	USUAL HOURS WORKED PER WEEK IN PAST 12 MONTHS
WKW	INTEGER	WEEKS WORKED DURING PAST 12 MONTHS N/A: UNDER 16 YEARS OLD OR DID NOT WORK IN PAST 12 MONTHS 1: 50-52 WEEKS WORKED 2: 48-49 WEEKS WORKED 3: 40-47 WEEKS WORKED 4: 27-39 WEEKS WORKED 5: 14-26 WEEKS WORKED 6: 13 OR FEWER WEEKS WORKED
ESR	INTEGER	EMPLOYMENT STATUS RECODE N/A: UNDER 16 YEARS OLD 1: CIVILIAN EMPLOYED, AT WORK 2: CIVILIAN EMPLOYED, EMPLOYED BUT NOT AT WORK 3: UNEMPLOYED 4: ARMED FORCES, AT WORK 5: ARMED FORCES, EMPLOYED BUT NOT AT WORK 6: NOT IN LABOR FORCE
RAC1P	INTEGER	RACE IDENTIFIER 1 White alone 2 Black or African American alone 3 American Indian alone

		4	Alaska Native alone
		5	American Indian and Alaska Native tribes specified; or American Indian or Alaska native, not specified and no other races
		6	Asian alone
		7	Native Hawaiian and Other Pacific Islander alone
		8	Some other race alone
		9	Two or more major race groups
HISP	INTEGER	HISPANIC IDENTIFIER	
		1	Not Spanish/Hispanic/Latino
		2	Mexican
		3	Puerto Rican
		4	Cuban
		5	Dominican
		6	Costa Rican
		7	Guatemalan
		8	Honduran
		9	Nicaraguan
		10	Panamanian
		11	Salvadoran
		12	Other Central American
		13	Argentinean
		14	Bolivian
		15	Chilean
		16	Colombian
		17	Ecuadorian
		19	Peruvian
		20	Uruguayan
		21	Venezuelan
		22	Other South American
		23	Spaniard
		24	All Other Spanish/Hispanic/Latino
SCHG	INTEGER	SCHOOL GRADE N/A: NOT IN SCHOOL 1: PRESCHOOL 2: KINDERGARTEN 3-14: GRADES 1-12 15: UNDERGRAD COLLEGE/UNIVERSITY 16: COLLEGE/UNIVERSITY BEYOND BACHELOR'S DEGREE	

MIL	INTEGER	MILITARY SERVICE N/A: UNDER 17 YEARS OLD 1: ON ACTIVE DUTY 2: ACTIVE IN PAST, BUT NOT CURRENTLY 3: ONLY ACTIVE FOR TRAINING IN RESERVES/NATIONAL GUARD 4: NEVER SERVED IN MILITARY
NAICSP	INTEGER	NAICS ID OF WORK OCCUPATION (CENSUS CODE)
INDP	INTEGER	WORK INDUSTRY CODE (CENSUS CODE)
PERSON_ID	INTEGER	PERSON ID NUMBER USED AS INDEX

3.1.5 Transit data

Transit (public transportation) data contained in the "trn" folder includes all of the transit model input files. These files define the alignment of bus routes, rail routes, stops/stations, fare inputs, "system" inputs, and "factor" inputs. Most of the transit input files are coded in the Cube Public Transport (PT) format as the Gen3 Model uses PT for transit modeling. The PT "system" and "factor" input files define which fare systems belong to which routes, and how people can move within the transit network - including where transfers can be made, how long transfers take, and how the transit systems appear compared to driving. The transit input files, including transit network input files (.lin), are listed in Table 17. The fields contained in three particular transit input files, the bus factor file, the rail link file, and the station file, are specified in Table 18, Table 19, and Table 20, respectively. The PT based transit skimming and assignment processes are explained in Section 6.5. Transit assignment is conducted in origin-destination (O-D) format for four time-of-day periods.

Table 17: Transit Input Files

FILENAME	DESCRIPTION
AM_TRN.FAC	AM transit factor input
AM_TRN_BM.FAC	AM bus-metro transit factor input
AM_TRN_CR.FAC	AM commuter rail transit factor input
Bus_Factor_File.dbf	Local Bus Time Degradation Factors. See Table 18.
fare_am.dat	AM fare setup file
fare_md.dat	MD fare setup file
fare_nt.dat	NT fare setup file
fare_pm.dat	PM fare setup file
MARC_Fare.dat	MARC train fare setup file
MD_TRN.FAC	AM transit factor input
MD_TRN_BM.FAC	AM bus-metro transit factor input
MD_TRN_CR.FAC	AM commuter rail transit factor input
MetLnkM1.TB	Metrorail link file
MetNodM1.TB	Metrorail node file
MFare1.a1	Metrorail Station XYs scaled to 1/100ths of miles
mfare1_Sta_Disc.ASC	Metrorail Station fare discount array in cents
MODE{m}{p}.LIN	Transit line input files, four periods ({p} - AM, MD, PM, and NT) for each mode ({m}): 1: WMATA local bus (Metrobus)

	2: WMATA express bus (Metrobus) 3: WMATA rail (Metrorail) 4: Commuter rail (Amtrak, MARC, VRE) 5: Light rail 6: Other local bus in WMATA service area 7: Other express bus in WMATA service area 8: Other local bus outside of WMATA service area 9: Other express bus outside of WMATA service area 10: Bus Rapid Transit (BRT) and streetcar
NT_TRN.FAC	NT transit factor input
NT_TRN_BM.FAC	NT bus-metro transit factor input
NT_TRN_CR.FAC	NT commuter rail transit factor input
PM_TRN.FAC	PM transit factor input
PM_TRN_BM.FAC	PM bus-metro transit factor input
PM_TRN_CR.FAC	PM commuter rail transit factor input
Rail_Links.DBF	Metrorail link input file. See Table 19.
STATION.DBF	Station data input file. See Table 20.
station_names.dbf	List of station ID and name
tariff.txt	Metrorail tariff script input
TRNPEN.DAT	Transfer penalty input file
TRN_Deflator.txt	Transit deflation factor input file
TSYSD.PTS	Transit system input file
VRE_Fare.dat	Virginia Rail Express (VRE) fare input file

Table 18: Bus Factor File Format (Bus_Factor_File.DBF)

NAME	TYPE	DESCRIPTION
YEAR	Integer	Year of analysis
AMIBFTR	Decimal	AM inbound time degradation
AMOBFTR	Decimal	AM outbound time degradation
MDIBFTR	Decimal	MD inbound time degradation
MDOBFTR	Decimal	MD outbound time degradation
PMIBFTR	Decimal	PM inbound time degradation
PMOBFTR	Decimal	PM outbound time degradation
NTIBFTR	Decimal	NT inbound time degradation
NTOBFTR	Decimal	NT outbound time degradation

Table 19: Rail Link File Format (Rail_Links.DBF)

NAME	TYPE	DESCRIPTION
A	INTEGER	A NODE OF LINK
B	INTEGER	B NODE OF LINK
MODE	INTEGER	TRANSIT MODE (ALWAYS 3)
SPEED	INTEGER	SPEED IN MPH
DISTANCE	DECIMAL	DISTANCE IN MILES

TRANTIME	INTEGER	TRANSIT TIME IN MINUTES

Table 20: Station Input File Format (STATION.DBF)

NAME	TYPE	DESCRIPTION
OBJECTID	INTEGER	UNIQUE ID NUMBER
SEQNO	INTEGER	SEQUENCE NUMBER
MM	TEXT	MODE CODE (M FOR METRORAIL, C FOR COMMUTER RAIL, B FOR BUS, L FOR LIGHT RAIL, N FOR BRT/STREETCAR)
NCT	INTEGER	STATION ACCESS DISTANCE CODE (0, 1, 2, 3, 8, 9) 0: ONLY KNR-ACCESS LINKS GENERATED (E.G., BRADDOCK ROAD, NATIONAL AIRPORT, CLARENDON) 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) 8: PENTAGON METRO STA., ALLOWS FOR VERY LONG KNR LINKS, TO REPRESENT “SLUGGING” (INFORMAL CARPOOL) 9: METRORAIL STA. IN USE, BUT NO PNR/KNR ACCESS (E.G., DUPONT CIRCLE, FARRAGUT NORTH, METRO CTR.)
STAPARK	TEXT	IF STATION HAS PARKING (Y OR N)
STAUSE	TEXT	IS THE STATION IN USE (Y OR N/BLANK)
SNAME	TEXT	NAME OF STATION
STAZ	INTEGER	STATION ZONE NUMBER
STAT	INTEGER	STATION STOP NODE NUMBER (CORRESPONDS TO RAIL NETWORK FOR MR AND CR)
STAP	INTEGER	STATION PARKING HIGHWAY NODE NUMBER
STAN1	INTEGER	STATION LINK ACCESS NODE NUMBER
STAN2	INTEGER	STATION LINK ACCESS NODE NUMBER
STAN3	INTEGER	STATION LINK ACCESS NODE NUMBER
STAN4	INTEGER	STATION LINK ACCESS NODE NUMBER
STAPCAP	INTEGER	STATION PARKING CAPACITY
STAX	INTEGER	STATION LOCATION X COORDINATE
STAY	INTEGER	STATION LOCATION Y COORDINATE
STAPKCOST	INTEGER	STATION PARKING COST
STAOPCOST	INTEGER	STATION OFF-PEAK PARKING COST
STAPKSHAD	INTEGER	STATION PEAK SHADOW PRICE (ALL 0)
STAOPSHAD	INTEGER	STATION OFF-PEAK SHADOW PRICE (ALL 0)
FIRSTYR	INTEGER	FIRST YEAR OF OPENING
STA_CEND	INTEGER	STATION PROJECT ID
STWALKTM	DECIMAL	STATION WALK TIME IN MINUTES

3.1.6 External and visitor/tourist transit trip data

External and visitor/tourist transit trip data are located in the trn\External_Visitor_Transit folder. There are 8 OMX files in the folder, all have a naming pattern that includes the time-of-day period,

external or visitor, and a file extension of OMX. The model will convert the OMX files to TRP (Cube Voyager Matrix) files, which will be used in the transit assignment process in Cube. These files are listed in Table 21. Each matrix has 18 matrix tables, which represent 18 different combinations of access mode and line haul modes.

The access modes are:

- WALK (walk access)
- PNR (park-n-ride)
- PNRE (park-n-ride egress)
- KNR (kiss-n-ride)
- KNRE (kiss-n-ride egress)

The line-haul transit modes are:

- AB (all-bus)
- MR (Metrorail only)
- BM (bus and Metrorail)
- CR (commuter rail)

Note that for commuter rail, PNR and KNR modes are combined.

These files are not adjusted in the model to account for growth - that must be done prior to the model run.

Table 21: External and Visitor Transit Input Files

FILE	DESCRIPTION
{period}_External.OMX	Input external transit trip tables
{period}_Visitor.OMX	Input visitor transit trip tables

3.2 Model outputs

The model outputs written into the “Outputs” folder are divided into 10 sub-folders based on the model component that generated the outputs. These folders are described in more detail below. Note that, at the end of a model run, some less important model outputs are automatically moved to a separate ”temp_files” folder to be deleted later to save storage space. A typical Gen3 Model run generates around 30 GB of data in the “Outputs” folder and around 456 GB of data in the "temp_files” folder. Furthermore, during a model run, a “runtime” folder is created in the outputs folder and then deleted at the end of a successful run. This folder acts as the working folder for certain multithreaded processes, such as transit skimming, transit assignment, and skm-to-omx conversion. In the event that any of these three steps crashes, the Cube report files can be found either in the runtime folder or one of its subfolders.

The model outputs folder also includes a visualizer html file. All of these are described in this section.

When “TOLL_SETTING” is turned on/enabled in the main batch file, all the intermediate model outputs associated with the toll setting process will be written to a “TOLL_SETTING” folder under the scenario root directory, which is separate from the “Outputs” folder. The toll escalation file (toll_esc.dbf) under \inputs\hwy will be updated with the toll setting results during the model run.

The model outputs reflect the outputs of a series of modeling processes based on model inputs and the parameters calibrated to various data sources. As such, output data reliability is a function of input data reliability and model reliability from the calibration data. There are many sources of uncertainty associated with model outputs and users should exercise caution when using these outputs for future-year forecasts.

Like most ABMs, the Gen3 Model is implemented using Monte Carlo simulation, which means that the model outputs are subject to some degree of variation associated with random seeds used in the simulation. ActivitySim uses a fixed set of random seeds for Monte Carlo simulations, ensuring that model outputs can always be replicated with the same model inputs. However, ActivitySim also allows users to alter the random seeds. In theory, the variations associated with Monte Carlo simulation can be minimized by averaging model outputs from multiple runs using different sets of random seeds. In practice, however, this is not recommended due to significantly increased run time and complexities. COG/TPB staff conducted an experiment executing the same Gen3 Model run using three different sets of random seeds[38] and found that, at the regional and jurisdictional levels, the differences in model outputs between the three runs were basically negligible; but at a finer geographic or demographic summary level, the differences became more pronounced. This experiment supported the premise that the random effects associated with Monte Carlo simulation do not play a significant role in aggregated model results, **but it also indicated a need to consider these effects when interpreting the model outputs at a disaggregate level**.

3.2.1 Model visualizer

In the root of the model output folder is the ABM visualizer .html file (e.g., "\2018_base\outputs\SURVEY_vs_2018_base.html"). This HTML file is built by an R script with RMarkdown and Flexdashboard libraries to show an interactive dashboard of ActivitySim outputs. The visualizer opening screen is shown in Figure 4. This visualizer is organized into five main sections: Welcome, Overview, Long Term, Tour Level, and Trip Level. The welcome screen is a simple map that shows the modeled region and the regional transportation network. The Overview section contains a screen that includes basic summary information of the model and compares it to the survey (which is the default) or a user specified base model (Section 2.3.3 describes how "Visualizer Settings" in the main batch file can be revised to compare the current model run to an existing base run). The Long-Term section shows various comparisons of long-term choices, such as auto-ownership, the distance to work, university, or school, and where zero-auto households are located. The Tour-Level section displays comparisons related to daily schedule models (such as the Coordinated Daily Activity Pattern model output) as well as those for tours (such as tour mode choice). The trip model section shows trip model comparisons.

The visualizer is automatically created at the end of the model run. In addition to the .html file, the visualizer folder (outputs\visualizer under each scenario run) contains the summarized ActivitySim outputs used to create the dashboard. There are 156 files in this output folder: 72 jpegs, 1 pdf, 4 shapefile data files (.dbf, .prj, .shp, .shx), and 79 csv files. The jpegs depict a series of graphs highlighting relationships between demographic and land use data. The pdf contains a series of histograms depicting counts of tours and trips for different categories, such as tour purpose. The shapefile holds the map depicting the number of zero-auto households per TAZ shown under the Long-Term tab of the dashboard. Finally, all the csv files but one hold the summarized data used to

[38] Feng Xie, Email to Bahar Shahverdi, "Re: setting random number seed in ActivitySim", July 23, 2024.

produce the graphs shown in the dashboard. The “rm_gq.csv” file is the only one which holds model run information and depicts whether group quarters were removed from the summarized data. This file is used when using the visualizer to compare the outputs to a previous run.

The Visualizer summaries comparing a base-year model run to the survey data are extremely useful for base-year model calibration, while the summaries comparing the current model run to an existing run are often used in model applications such as alternatives analysis and scenario analysis. The Visualizer summaries are focused on the percentage distributions rather than absolute values of model metrics. The absolute values for most of these metrics can instead be found in the “View from Space” summaries, which will be discussed in Section 3.2.7.

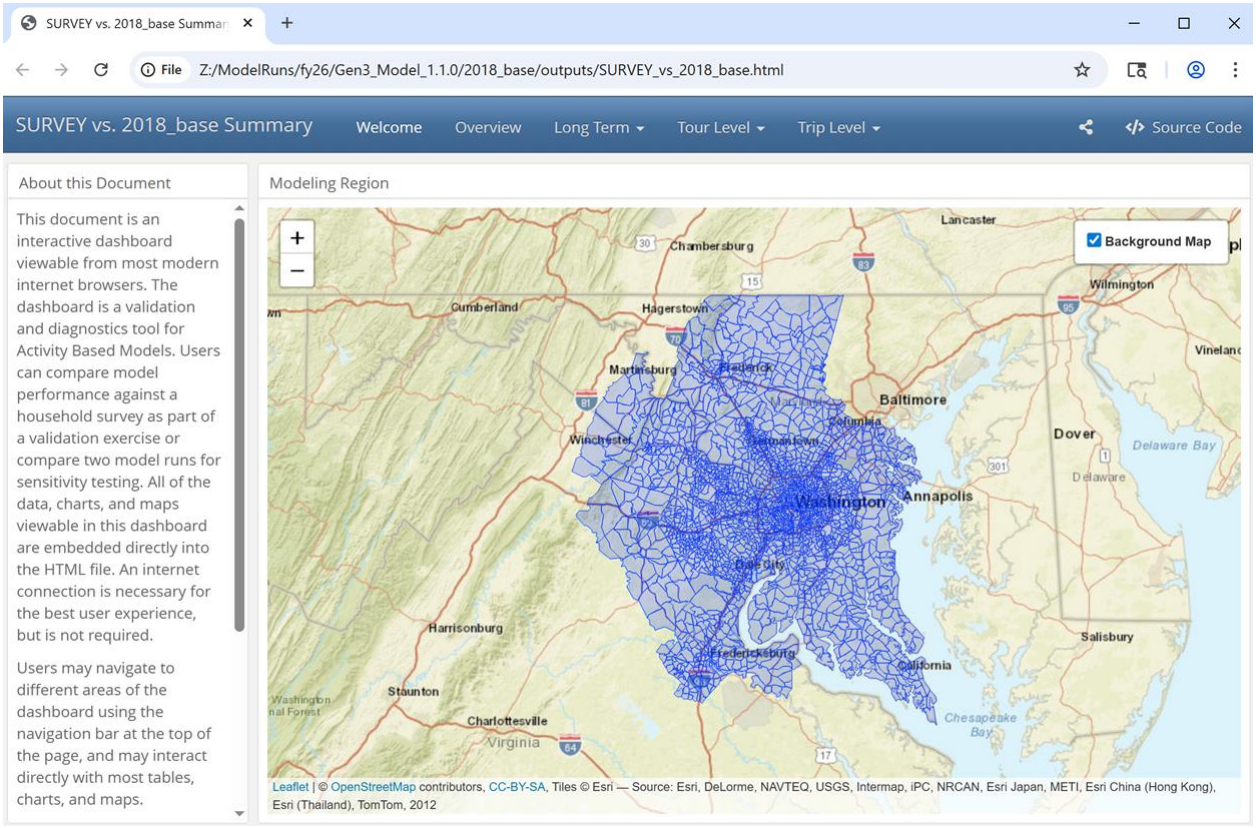


Figure 4: ABM Visualizer

3.2.2 ActivitySim

The ActivitySim folder ("activitysim") contains output from running ActivitySim. Table 22 shows the contents of the folder in more detail. In addition to these output files, ActivitySim also generates intermediate data storage folders known as "pipeline" folders, which hold the state of various tables (households, persons, tours, trips, etc.) at given points in the simulation run as Parquet files.[39] Beyond these files, the model does not exploit Parquet in its full capacity due to interface limitations with CUBE, as the software does not support the format.

Table 22: ActivitySim Model Output Files

FILE	DESCRIPTION
activitysim.log	Model run progress terminal output
data_dict.csv	List of variables (data dictionary) in each model dataframe
final_accessibility.csv	Accessibility measures created by ActivitySim. See Table 27.
final_checkpoints.csv	This is an internal ActivitySim file that relates to the pipeline folders
final_households.csv	Synthetic households file with the additional variables created during model run. See Table 23.

[39] Apache Parquet is a free, open-source column-oriented data storage format designed for efficient data processing. Unlike row-based formats like CSV, Parquet organizes data by columns, which can enhance performance for some types of computations.

final_joint_tour_participants.csv	Joint tour participants output file. See Table 28.
final_land_use.csv	Land use file with the additional variables created during model run. See Table 25.
final_persons.csv	Synthetic persons file with the additional variables created during model run. See Table 24.
final_tours.csv	All the tours created by ActivitySim and their attributes. See Table 29.
final_trips.csv	All the trips created by ActivitySim and their attributes. See Table 30.
final_vehicles.csv	Modeled household vehicle output file. See Table 31.

The final_households.csv table contains all the original synthetic population household attributes as well as the ones created in ActivitySim.[40] Table 23 below shows all household attributes.

Table 23: Final_Households.CSV Fields

Name	Type	Description
household_id	Integer	Unique ID number of household (from population synthesizer)
puma_geoid	Integer	GEOID of Census Public Use Microdata Area where the household resides
home_zone_id	Integer	TAZ ID of home location
TYPE	Integer	Type of housing unit 1 = Household 2 = Institutional group quarters 3 = Noninstitutional group quarters
hhsize	Integer	Household size (number of persons in household)
auto_ownership	Integer	Auto ownership level
HHT	Integer	The type of household - Census PUMS HHT variable: N/A: Not applicable (group quarters or vacant) 1: Married couple household 2: Other family household: Male householder, no spouse present 3: Other family household: Female householder, no spouse present 4: Nonfamily household: Male householder: Living alone 5: Nonfamily household: Male householder: Not living alone 6: Nonfamily household: Female householder: Living alone 7: Nonfamily household: Female householder: Not living alone
hhincadj	Integer	Household income

[40] The tabular file data from ActivitySim are written to CSV files, which can easily be input into R, Python, and many other software packages. Excel is generally not recommended, however, since the files tend to be larger than what Excel allows. Excel allows 1,048,576 rows, and a full ActivitySim run has several output tables that drastically exceed this limit. See Sections 5.3.1 and 0 for more information.

Name	Type	Description
workers	Integer	Number of workers in household
has_children	Integer	If the household has children
sample_rate	Decimal	Sampling rate of household (set to 1.0 for all records)
income	Integer	Household income (hhincadj with NA values converted to 0)
income_in_thousands	Integer	Household income in thousands
income_segment	Integer	4-level segmented income: low(<$50K), medium (>$50K, <$100K), high (>$100K, <$150K), very high (>$150K)
median_value_of_time	Decimal	Median household value of time (according to income)
hh_value_of_time	Decimal	Household value of time
num_workers	Integer	Number of workers in the household
num_non_workers	Integer	Number of non-workers in the household
num_drivers	Integer	Number of drivers in the household
num_adults	Integer	Number of adults in the household
num_children	Integer	Number of children in the household
num_young_children	Integer	Number of young children in the household
num_children_5_to_15	Integer	Number of children aged 5-15 yrs in the household
num_children_6_to_12	Integer	Number of children aged 6-12 yrs in the household
num_children_16_to_17	Integer	Number of children aged 16-17 yrs in the household
num_college_age	Integer	Number of college age children in the household
num_young_adults	Integer	Number of young adults in the household
non_family	Integer	Non-family household
family	Integer	Family household
home_is_urban	Integer	Home is in urban area
home_is_rural	Integer	Home is in rural area
home_jurisdiction	Integer	Jurisdiction that the home resides in
TAZ	Integer	TAZ ID of home location
num_predrive_child	Integer	Number of pre-driving age children in household
num_nonworker_adults	Integer	Number of adult nonworker
num_fullTime_workers	Integer	Number of full-time workers in the household
num_partTime_workers	Integer	Number of parttime workers in the household
retired_adults_only_hh	Integer	Households with only retired persons
hh_work_auto_savings_ratio	Decimal	Time saved driving vs. taking transit to work
av_ownership	Integer	If the household owns an automated vehicle
num_under16_not_at_school	Integer	Number of under-16 persons not in school in the household

Name	Type	Description
num_travel_active	Integer	Number of persons who travel in the household (with M or N CDAP pattern)
num_travel_active_adults	Integer	Number of adults who travel in the household (with M or N CDAP pattern)
num_travel_active_preschoolers	Integer	Number of preschoolers who travel in the household (with M or N CDAP pattern)
num_travel_active_children	Integer	Number of children who travel in the household (with M or N CDAP pattern)
num_travel_active_non_preschoolers	Integer	Number of non-preschoolers who travel in the household (with M or N CDAP pattern)
participates_in_jtf_model	Integer	Household with joint tours
joint_tour_frequency	Integer	Joint tour in frequency of the household
num_hh_joint_tours	Integer	Number joint in tours a household makes

The final_persons.csv table contains all the original synthetic population person attributes as well as the ones created in ActivitySim. Table 24 shows the person attributes contained in the final_persons.csv table.

Table 24: Final_Persons.CSV FIelds

Name	Type	Description
person_id	Integer	Person ID number used as index
puma_geoid	Integer	Household PUMA Geographic ID
TAZ	Integer	Household TAZ ID
household_id	Integer	Household ID number (must agree with household file)
per_num	Integer	Number of persons in household
age	Integer	Age of person in years
SEX	Integer	Sex of person in household (1 = male, 2 = female)
WKHP	Integer	Usual hours worked per week in past 12 months
WKW	Integer	Weeks worked during past 12 months N/A: under 16 years old or did not work in past 12 months 1: 50-52 weeks worked 2: 48-49 weeks worked 3: 40-47 weeks worked 4: 27-39 weeks worked 5: 14-26 weeks worked 6: 13 or fewer weeks worked

Name	Type	Description
ESR	Integer	Employment Status Recode N/A: Under 16 years old 1: Civilian employed, at work 2: Civilian employed, employed but not at work 3: Unemployed 4: Armed forces, at work 5: Armed forces, employed but not at work 6: Not in labor force
RAC1P	Integer	Race identifier (not used in model)
HISP	Integer	Hispanic flag (not used in model)
SCHG	Integer	School grade N/A: Not in school 1: Preschool 2: Kindergarten 3-14: grades 1-12 15: Undergrad college 16: College beyond bachelor's degree
MIL	Integer	Military Service N/A: under 17 years old 1: On active duty 2: Active in past, but not currently 3: Only active for training in Reserves/National Guard 4: Never served in military
NAICSP	Integer	NAICS ID of work occupation (Census Code)
INDP	Integer	Work industry code (Census code)
age_0_to_5	Integer	Person's age is between 0 and 5
age_6_to_12	Integer	Person's age is between 6 and 12
age_16_to_19	Integer	Person's age is between 16 and 20
age_16_p	Integer	Person's age more than 16
adult	Integer	Person is an adult
young	Integer	Person is 25 or younger
old	Integer	Person is 65 or older
male	Integer	Person is male
female	Integer	Person is female
esr	Integer	ESR (above) with NA values set to zero
wkhp	Integer	WKHP (above) with NA values set to zero
wkw	Integer	WKW (above) with NA values set to zero
schg	Integer	SCHG (above) with NA values set to zero
mil	Integer	MIL (above) with NA values set to zero
pemploy	Integer	Employment type
pstudent	Integer	Student type
ptype	Integer	Person type category

Name	Type	Description
has_non_worker	Integer	Person's household has nonworker
has_retiree	Integer	Person's household has retiree
has_preschool_kid	Integer	Person's household has preschooler
has_driving_kid	Integer	Person's household has driving age child
has_school_kid	Integer	Person's household has school age child
has_full_time	Integer	Person's household has full time worker
has_part_time	Integer	Person's household has part time worker
has_university	Integer	Person's household has a university student
student_is_employed	Integer	Person is a student and is employed
nonstudent_to_school	Integer	Person is a non-student ptype and pstudent indicates student
is_student	Integer	Person is a student
is_gradeschool	Integer	Persons is a grade schooler
is_highschool	Integer	Person is a high schooler
is_university	Integer	Person is a university student
school_segment	Integer	School segment type (1: grade school, 2: high school, 3: university)
is_worker	Integer	Person is a worker
home_zone_id	Integer	TAZ ID of home location
PNUM	Integer	Member ID of person
income	Integer	Household income
income_in_thousands	Integer	Household income in thousands
income_segment	Integer	4-level segmented income: low(<$50K), medium (>$50K, <$100K), high (>$100K, <$150K), very high (>$150K)
is_fulltime_worker	Integer	Person is full-time worker
is_parttime_worker	Integer	Person is part-time worker
is_out_of_home_worker	Integer	Person works out-of-home
value_of_time	Integer	Persons's value of time
is_income_less25K	Integer	Household income <$25K
is_income_25K_to_60K	Integer	Household income >$25K & <$60K
is_income_60K_to_120K	Integer	Household income >$60K & <$120K
is_income_greater60K	Integer	Household income >$60K
is_income_greater120K	Integer	Household income >$120K
is_non_worker_in_HH	Integer	Persons's household has a non-worker
is_all_adults_full_time_workers	Integer	All adults are full-time workers in the household
is_pre_drive_child_in_HH	Integer	Person's household has a predriving age child
has_young_children	Integer	Person's household has young children
has_children_6_to_12	Integer	Person's household has children aged between 6 and 12
hh_child	Integer	Number of children in the household

Name	Type	Description
home_jurisdiction	Integer	Jurisdiction home is located in
school_zone_id	Integer	TAZ ID of school location (where applicable)
distance_to_school	Decimal	Distance in miles to school
roundtrip_auto_time_to_school	Decimal	Round trip auto travel time to school
noUsualSchoolLocation	Integer	Person has no usual school location
work_from_home	Integer	Person works from home
workplace_zone_id	Integer	TAZ ID of workplace location (where applicable)
workplace_location_logsum	Decimal	Workplace location logsum
distance_to_work	Decimal	Distance in miles to work
workplace_in_cbd	Integer	Person's workplace zone is in CBD
work_zone_area_type	Integer	Area type of workplace zone
roundtrip_auto_time_to_work	Decimal	Round trip auto travel time to work
work_auto_savings	Decimal	Time saved driving vs. taking transit to work
work_auto_savings_ratio	Decimal	Scaled time saved driving vs. taking transit to work (-1,1)
usualWorkLocationIsHome	Integer	Person's usual work location is home
noUsualWorkLocation	Integer	Person has no usual work location
work_jurisdiction	Integer	Jurisdiction workplace is located in
work_park_cost	Decimal	Parking cost of work zone
transit_pass_subsidy	Integer	Person receives transit subsidy
free_parking_at_work	Integer	Person gets free parking at work
telecommute_frequency	String	Telecommute frequency
cdap_activity	String	CDAP pattern (M = Mandatory, N = Non-mandatory, H = Home/out of region)
travel_active	Integer	Person travels out of home
under16_not_at_school	Integer	Person is under 16 and does not go to school
has_preschool_kid_at_home	Integer	Person has preschool kid at home
has_school_kid_at_home	Integer	Person has school kid at home
mandatory_tour_frequency	Integer	Frequency of mandatory tours
work_and_school_and_worker	Integer	Persons is a worker and goes to work and school
work_and_school_and_student	Integer	Persons is a student and goes to work and school
num_mand	Integer	Number of mandatory tours for each person
num_work_tours	Integer	Number of work tours for each person
has_pre_school_child_with_mandatory	Integer	Presence of preschool kid with mandatory tours
has_driving_age_child_with_mandatory	Integer	Presence of driving age school children with mandatory tours
num_joint_tours	Integer	Number of joint tours for each person
non_mandatory_tour_frequency	Integer	Non-mandatory tours for each person
num_non_mand	Integer	Number of non-mandatory tours

Name	Type	Description
num_escort_tours	Integer	Number of escort tours
num_eatout_tours	Integer	Number of eat-out tours
num_shop_tours	Integer	Number of shopping tours
num_maint_tours	Integer	Number of maintenance tours
num_discr_tours	Integer	Number of discretionary tours
num_social_tours	Integer	Number of social tours
num_non_escort_tours	Integer	Number of non-escort tours
num_shop_maint_tours	Integer	Number of shopping and maintenance tours
num_shop_maint_escort_tours	Integer	Number of shopping and maintenance and escort tours
num_add_shop_maint_tours	Integer	Number of additional shopping and maintenance tours
num_soc_discr_tours	Integer	Number of social and discretionary tours
num_add_soc_discr_tours	Integer	Number of additional social and discretionary tours

The `final_land_use.csv` table contains all the original land use fields as well as the ones created in ActivitySim. The table below shows the additional land use attributes created by ActivitySim.

Table 25: final_land_use.csv Fields

FIELD	TYPE	DESCRIPTION
zone_id	Integer	Zone ID
HH	Integer	Number of households
HHPOP	Integer	Household population
GQPOP	Integer	Group quarters population
TOTPOP	Integer	Total population
TOTEMP	Integer	Total employment
INDEMP	Integer	Industrial employment
RETEMP	Integer	Retail employment
OFFEMP	Integer	Office employment
OTHEMP	Integer	All other employment
JURCODE	Integer	Jurisdiction code
LANDAREA	Decimal	Land Area of TAZ in square miles
TAZXCRD	Integer	TAZ X Coordinate
TAZYCRD	Integer	TAZ Y Coordinate
K_8	Integer	Kindergarten - 8th grade enrollment
G9_12	Integer	High school enrollment
COLLEGE	Integer	College/university enrollment
Park_Acres	Decimal	Park area of TAZ in acres
GC_Acres	Decimal	Golf course area of TAZ in acres
PRKCST	Integer	8-hour parking cost in TAZ (daily parking for work)
OPRKCST	Integer	Hourly off-peak parking cost in TAZ
TERMINAL	Integer	Terminal time for trip-ends in TAZ, in minutes
AREATYPE	Decimal	Area type of TAZ
household_density	Decimal	Total household divided by zone area

employment_density	Decimal	Total employment divided by zone area
density_index	Decimal	Density index defined as (household_density * employment_density)/(employment_density + household_density)
TOPOLOGY	Integer	Topology code (not used)
emp_adjust	Decimal	Employment adjustment factor accounting for work-from-home and in-migration
ADJ_INDEMP	Decimal	Adjusted industrial employment accounting for work-from-home and in-migration used in destination choice
ADJ_RETEMP	Decimal	Adjusted retail employment accounting for work-from-home and in-migration used in destination choice
ADJ_OTHEMP	Decimal	Adjusted other employment accounting for work-from-home and in-migration used in destination choice
ADJ_OFFEMP	Decimal	Adjusted office employment accounting for work-from-home and in-migration used in destination choice
ADJ_TOTEMP	Decimal	Adjusted total employment accounting for work-from-home and in-migration used in destination choice

The `final_accessibility.csv` table contains all the accessibility measures created in ActivitySim. Table 27 shows accessibility measures created by ActivitySim. The accessibility measure used in ActivitySim is a decay function that is shown in Equation 1.

Equation 1: ActivitySim Accessibility Formula

$$A_i = ln\left(\sum_{j=1}^{Zones} size_j * e^{(I_{ij} * d)}\right)$$

Where:

A is the accessibility
Size is a size parameter, such as retail employment or total employment
I is an impedance parameter from zone i to zone j (usually time, see Table 26)
d is a dispersion parameter (see Table 26)

Table 26: Accessibility Impedance Values and Dispersion Parameters

MODE	IMPEDANCE VALUE	DISPERSION PARAMETER
Auto	Auto Time	-0.05
Transit	Transit Weighted Time	-0.05
Non-motorized	Walk Distance	-1.0

Table 27: final_accessibility.csv Fields

FIELD	TYPE	DESCRIPTION
zone_id	Integer	MAZ zone ID
auPkRetail	Decimal	Destination retail accessibility by auto at peak time
auPkTotal	Decimal	Destination total accessibility by auto at peak time
auOpRetail	Decimal	Destination retail accessibility by auto at off-peak time
auOpTotal	Decimal	Destination total accessibility by auto at off-peak time
trPkRetail	Decimal	Destination retail accessibility by transit at peak time
trPkTotal	Decimal	Destination total accessibility by transit at peak time
trPKHH	Decimal	Destination residential accessibility by transit at peak time
trOpRetail	Decimal	Destination retail accessibility by transit at off-peak time

trOpTotal	Decimal	Destination total accessibility by transit at off-peak time
nmRetail	Decimal	Destination retail accessibility by non-motorized modes
nmTotal	Decimal	Destination total accessibility by non-motorized modes
auShare	Decimal	Regional auto share constant
trShare	Decimal	Regional transit share constant
nmShare	Decimal	Non-motorized share constant
TotalAcc	Decimal	Total accessibility

The `final_joint_tour_participants.csv` table contains data on the joint tours in ActivitySim. Table 28 shows these data.

Table 28: final_joint_tour_participants.csv Fields

FIELD	TYPE	DESCRIPTION
participant_id	Integer	Person ID of participant
tour_id	Integer	Tour ID
household_id	Integer	Household ID
person_id	Integer	Person ID of participant
participant_num	Integer	Number of persons on tour

The `final_tours.csv` table contains all the information on the tours created in ActivitySim. Table 29 shows the attributes of ActivitySim tours.

Table 29: final_tours.csv Fields

Name	Type	Description
tour_id	Integer	Tour ID
person_id	Integer	Person id
tour_type	String	Tour type: work, school, othmaint, social, eatout, shopping, othdiscr, escort, eat, maint, business
tour_type_count	Integer	Number of tours of tour_type parent has (parent's max tour_type_num)
tour_type_num	Integer	If there are multiple of the same type tours, they will be numbered
tour_num	Integer	Index of tour (of any type) for parent tour
tour_count	Integer	Number of tours of any type for parent (parent's max tour_num)
tour_category	String	Category of tour. One of 'mandatory', 'non_mandatory', 'atwork', or 'joint'
number_of_participants	Integer	Number of participants on a tour
destination	Integer	TAZ ID for tour destination
origin	Integer	TAZ ID for tour origin
household_id	Integer	Household ID of the person making the tour
tdd	Integer	ID of the in-outbound bin combination
start	Integer	Start half hour bin of tour, ranging 1-48
end	Integer	End half hour bin of tour, ranging 1-48
duration	Integer	Tour duration in half hour unit
composition	String	Composition of a joint tour: all adults, all children, mixed
destination_logsum	Decimal	Tour destination logsum
tour_mode	String	Tour mode: TNC_SINGLE, WALK_MR, SHARED2, DRIVEALONE, WALK, WALK_BM, WALK_AB, SCHOOLBUS, BIKE, KNR_MR, TAXI,KNR_BM, SHARED3, WALK_CR, PNR_MR, TNC_SHARED, PNR_AB, KNR_AB, PNR_BM, PNR_CR, KNR_CR
mode_choice_logsum	Decimal	Mode choice logsum

selected_vehicle	String	Selected vehicle for tour
atwork_subtour_frequency	Integer	Frequency of atwork subtours for workers
parent_tour_id	Integer	ID of the parent tour if current tour is a subtour
stop_frequency	Integer	Number of stops on a tour
primary_purpose	String	Primary purpose of the tour: work, school, univ, othmaint, social, eatout, shopping, othdiscr, escort, atwork

The `final_trips.csv` table contains all the information on the trips created in ActivitySim. Table 30 shows the attributes of ActivitySim trips.

Table 30: final_trips.csv Fields

FIELD	TYPE	DESCRIPTION
trip_id	Integer	Trip ID
person_id	Integer	ID of the person that the trip belongs to
household_id	Integer	ID of the household that the person belongs to
primary_purpose	String	Purpose of the tour that the trip belongs to: work, school, univ, othmaint, social, eatout, shopping, othdiscr, escort, atwork
trip_num	Integer	Index of trip in the tour
outbound	Integer	Outbound trip direction
trip_count	Integer	Number of trips on a tour
destination	Integer	TAZ ID for trip destination
origin	Integer	TAZ ID for trip origin
tour_id	Integer	ID of tour trip belongs to
purpose	Integer	Trip purpose: othmaint, home, othdiscr, atwork, work, social, eatout, shopping, escort, school, univ, parking
destination_logsum	Decimal	Trip destination logsum
depart	Integer	Departure half hour bin: 1-48
trip_mode	String	Trip mode: DRIVEALONE, SHARED2, SHARED3, WALK, BIKE, WALK_AB, WALK_BM, WALK_MR, WALK_CR, PNR_AB, PNR_BM, PNR_MR, PNR_CR, KNR_AB, KNR_BM, KNR_MR, SCHOOLBUS, TAXI, TNC_SINGLE, TNC_SHARED
mode_choice_logsum	Decimal	Trip mode choice logsum

Table 31: final_vehicles.csv Fields

FIELD	TYPE	DESCRIPTION
vehicle_id	Integer	Vehicle ID
household_id	Integer	ID of the household that the vehicle belongs to
vehicle_num	Integer	ID number of the vehicle in the household
vehicle_type	String	Type of vehicle. These are listed as BODY_AGE_FUEL, where BODY indicates the type of vehicle (Car, SUV, Truck, etc), AGE is the age in years, and FUEL is the fuel source (Gas, Hybrid, Diesel, BEV, PEV)
auto_operating_cost	Integer	The operating cost of the vehicle
Range	Integer	The range of the vehicle
MPG	Integer	The fuel efficiency (in miles per gallon)
is_av	Integer	If the vehicle is an autonomous vehicle

3.2.3 Auxiliary

The “auxiliary” folder includes model outputs for miscellaneous trips, which include truck trips, commercial vehicle trips, auto-driver external-internal trips, auto-driver internal-external trips, through trips (auto driver, commercial vehicle, and trucks), airport auto-driver trips, supplemental taxi trips, and visitor trips. The Auxiliary output folder contains only the results from the final iteration (i4). Outputs associated with pump prime and iterations i1, i2, and i3 are moved to the temp_files folder at the end of the model run. The auxiliary output files for this model are listed in Table 32.

Table 32: Auxiliary Model Output Files

Filename	Description
autopaxixxi.trp	Auto driver Internal-external and external-internal trip matrix
COM.SQZ	Jurisdiction-Jurisdiction commercial vehicle trip matrix
COM.TEM	Temporary commercial vehicle trip matrix
COMext.TEM	Temporary external-internal and internal-external commercial vehicle trip matrix
DJ.EQV	Jurisdiction - TAZ equivalency file (script input file)
ext.tem	Temporary external auto driver trip matrix
ExternalPsAs.dbf	External Productions and Attractions by zone
HTK.SQZ	Jurisdiction-Jurisdiction heavy truck trip matrix
HTK.TEM	Temporary heavy truck trip matrix
HTKext.TEM	Temporary external heavy truck trip matrix
i4_{pp}_adr.mat	Auto-driver external-internal and internal-external trips for iteration 4 and period {pp}. Includes three matrix tables, {pp}_ADRs_1 (auto driver trips), {pp}_ADRs_2 (unused), {pp}_ADRs_3 (unused).
i4_{pp}_misc.tt	Miscellaneous trips for iteration 4and period {pp}. Includes 9 matrix tables: {pp}_XXTrk (truck through trips), {pp}_XXAdr (auto through trips), {pp}_TxAde (supplemental taxi trips), {pp}_VtAdr (visitor trips), {pp}_ScAdr (supplemental school trips, not used), {pp}_MedTk (medium truck trips for assignment), {pp}_HvyTk (heavy truck trips for assignment), {pp}_APAdr (airport trips), {pp}_ComVe (commercial vehicle trips for assignment).
i4_COMext.VTT	External-internal and internal-external commercial vehicle trip matrix for iteration {iter}
i4_COMMER.PTT	Internal commercial vehicle trip matrix for iteration 4}. Includes three tables, COM_Int (internal trips), COM_Ext (internal-external and external-internal trips), and COMAllVeh (sum of both matrices)
i4_ComVeh_Truck_dbg.dbf	Output debug file for iteration4, includes zonal inputs and outputs
i4_ComVeh_Truck_Ends.dbf	Truck and commercial vehicle trip ends by zone for iteration 4
i4_Ext_CVTruck_Gen_PsAs.dbf	External-internal and internal-external commercial vehicle and truck trip ends for iteration 4

Filename	Description
i4_Ext_CVTruck_Gen_PsAs.txt	Summary report of external truck and commercial vehicle trip productions and attractions for iteration 4
i4_Final_Int_Motor_PsAs.dbf	Final internal truck and commercial vehicle trip ends by zone for iteration 4
i4_HTKext.VTT	External-internal and internal-external heavy truck matrix for iteration 4
i4_HTRUCK.PTT	Internal heavy truck matrix for iteration 4. Includes three tables, HTK_Int (internal trips), HTK_Ext (internal-external and external-internal trips), and HTKAllVeh(sum of both matrices)
i4_MTKext.VTT	External-internal and internal-external medium truck matrix for iteration 4
i4_MTRUCK.PTT	Internal medium truck matrix for iteration 4. Includes three tables, MTK_Int (internal trips), MTK_Ext (internal-external and external-internal trips), and MTKAllVeh (sum of both matrices)
i4_Prepare_Internal_Ends.txt	Internal PA trip report for truck and commercial vehicle trips for iteration 4}
i4_Truck_Com_Trip_Generation.txt	Internal truck and commercial trip end report for iteration {iter}
ixxi_access.dbf	Auto driver Internal-external and external-internal accessibility
ixxi_prod.dbf	Auto driver Internal-external and external-internal production and attraction file
ixxidcutils.mat	Auto driver Internal-external and external-internal destination choice utilities matrix
Misc_Daily.VTT	Daily miscellaneous trips. Includes 9 matrix tables: XXTrk (truck through trips),_XXAutocv (auto through trips), taxi (supplemental taxi trips), visi (visitor trips), schl (supplemental school trips, not used),_Mtk (medium truck trips for assignment), Htk (heavy truck trips for assignment), AirPx (airport trips), CV (commercial vehicle trips for assignment).
MTK.SQZ	Jurisdiction-Jurisdiction medium truck matrix
MTK.TEM	Temporary medium truck matrix
MTKext.TEM	Temporary external medium truck matrix
SUMMARY_EXT_EXO_TRIPS.csv	External and exogenous trip table summary
TA_Vehs.VTT	All vehicles assigned
trip_distribution_extTrk.RPT	External truck and commercial vehicle trip distribution report
trip_distribution_intTrk.RPT	Internal truck and commercial vehicle trip distribution report

3.2.4 Highway assignment

The highway assignment outputs are written in the “hwy_assign” folder. The files in this folder are the output for each iteration and include vehicle trip tables and text-based loaded network files. This output folder contains only the results from the final iteration (i4). Outputs associated with pump prime and iterations i1, i2, and i3 are moved to the temp_files folder at the end of the model run. The files are listed in Table 33.

Table 33: Highway Assignment Output Files

Filename	Description
i4_{pp}_load_link.ASC	Output loaded link files for iteration 4and period {pp}
person_trips_by_tour_purp_and_trip_mode.csv	Aggregated value of person trip tables by tour purpose (atwork, eatout, escort, othdiscr, othmaint, school, shopping, social, univ, work and all purposes) and by trip modes.
Trip_tables_at_jur_level.csv	Jurisdictional value of person trip tables by tour purpose (atwork, eatout, escort, othdiscr, othmaint, school, shopping, social, univ, work and all purposes) and by trip modes.
i4_ue_iteration_report_{AM/PM}_hov.txt	Traffic assignment report for AM/PM HOV assignment
i4_ue_iteration_report_{AM/PM}_nonHov.txt	Traffic assignment report for AM/PM non-HOV assignment
i4_ue_iteration_report_{MD/NT}.txt	Traffic assignment report for MD/NT assignment
final_trips_{pur}.TRP	Zonal value of person trip tables by tour purpose (atwork, eatout, escort, othdiscr, othmaint, school, shopping, social, univ, work and all purposes). Each includes 20 matrix tables categorized by trip modes.
i4_{pp}.VTT	Vehicle trip tables for assignment, iteration 4and period {pp}
i4_Asim_{pp}.VTT	Output vehicle trip tables from ActivitySim for iteration 4 and period {pp} (converted from ActivitySim OMX files). Includes three matrix tables, SOV, SHARED2, and SHARED3.

3.2.5 Highway network

Several key network outputs from the highway network building and highway assignment procedures are written in the “hwy_net” folder.

Table 34: Highway Network Output Files

Filename	Description
linkTAZ.dbf	Nearest Taz to each link
i4_Assign_Output.csv	Loaded highway network from Iteration 4 in csv format
i4_Assign_Output.NET	Loaded highway network from Iteration 4

i4_Averaged_HWY.net	Identical to i4_HWY.net
zonehwy.net	Base highway network which excludes transit and transit support links
zonehwy_unbuild.net	Base network that can be “unbuild” to generate link.dbf and node.dbf
i4_HWY.net	Loaded highway network from Iteration 4 with averaged link volumes
i4_Average_Link_Speeds.txt	Average link speeds from iteration 4

3.2.6 Land use

The land use outputs are in the “landuse” folder. These files are outputs of land use processing scripts. The contents of these land use files are described in Table 35 through Table 39.

Table 35: Land Use Output Files

Filename	Description
AreaType_File.dbf	Computed area types. See Table 36.
AreaType_File.txt	Computed area type summary report.
Floating_LU.dbf	Computed 1-mile floating population and employment density. See Table 33.
TAZ_XYs.dbf	TAZ X and Y locations. See Table 37.
ZONE.dbf	Final zonal data
ZONEV2.A2F	Zonal parking and terminal times. See Table 38.
ztermtm.asc	Computed zonal terminal time report file.

Table 36: AreaType_File.dbf Fields

Name	Type	Description
TAZ	Integer	Zone ID
POP_10	Integer	Population within 1 mile
EMP_10	Integer	Employment within 1 mile
AREA_10	Decimal	Land area within 1 mile
POPDEN	Decimal	Floating 1 mile population density
EMPDEN	Decimal	Floating 1 mile employment density
POPCODE	Integer	Population density classification
EMPCODE	Integer	Employment density classification
ATYPE	Integer	Area type

Table 37: Floating_LU.dbf Fields

Name	Type	Description
TAZ	Integer	Zone ID
HH00	Integer	Households within 0 miles (zonal households)
POP00	Integer	Population within 0 miles (zonal population)
EMP00	Integer	Employment within 0 miles (zonal employment)
AREA00	Decimal	Area within 0 miles (zonal area)
HH10	Integer	Households within 1 mile
POP10	Integer	Population within 1 mile

EMP10	Integer	Employment within 1 mile
AREA10	Integer	Land area within 1 mile

Table 38: TAZ_XYs.dbf Fields

Name	Type	Description
N	Integer	Zone ID
X	Integer	X coordinate of TAZ centroid
Y	Integer	Y coordinate of TAZ centroid

Table 39: ZONEV2.A2F Fields

Name	Type	Description
I	Integer	Zone ID
HBWParkCost	Integer	Home-based Work trip parking cost in cents
HBSParkCost	Integer	Home-based Shopping trip parking cost in cents
HBOParkCost	Integer	Home-based Other trip parking cost in cents
NHBParkCost	Integer	Non-home-based trip parking cost in cents
HB_TermTime	Integer	Terminal time for home-based trips
NHB_TermTime	Integer	Terminal time for nonhome-based trips

3.2.7 Reports

The “reports” folder includes various reports for the Cube processes of the model, as well as a copy of all messages sent to the screen during the model run. The Reports output folder contains only the results from the final iteration (i4). Outputs associated with pump prime and iterations i1, i2, and i3 are moved to the temp_files folder at the end of the model run. These files are listed in Table 40. In general, if there is a problem with the non-ActivitySim portion of the model, the last files written to this folder will show what happened. The {scenario_dir}_fulloutput.txt file should display an error message or an error location.

Table 40: Reports Folder Output Files

Filename	Description
{scenario_dir}_fulloutput.txt	Copy of full output to the screen
Adjust_Runtime.RPT	Cube report
AreaType_File. RPT	Cube land use processing report
HWY_Deflator.txt	Cube highway network report
i4_Average_Link_Speeds. RPT	Cube highway network report
i4_Combine_Tables_For_TrAssign_Parallel. RPT	ActivitySim generated transit trip tables to TRP format conversion and fixed external and visitor trips tables combination report
i4_Highway_Assignment.RPT	Cube highway assignment report
i4_Highway_Skims_4TOD.RPT	Cube highway skimming report
i4_IXXI_TripGeneration.RPT	Cube auxiliary model report
i4_Misc_Time-of-Day.RPT	Cube auxiliary model report
i4_Prepare_Ext_ComTruck_Ends.RPT	Cube auxiliary model report

i4_Prepare_Int_ComTruck_Ends.RPT	Cube auxiliary model report
i4_Prepare_Trip_Tables_for_Assignment.RPT	Cube highway assignment preparation report
i4_Prepare_Trip_Tables_For_Assignment.txt	Cube highway assignment preparation report
i4_Transit_Assgn_AB.RPT	Cube transit assignment report for all bus
i4_Transit_Assgn_BM.RPT	Cube transit assignment report for bus-metro
i4_Transit_Assgn_CR.RPT	Cube transit assignment report for commuter rail
i4_Transit_Assgn_MR.RPT	Cube transit assignment report for metrorail
i4_Transit_Assgn_AB_{pp}.RPT	Cube transit assignment report for all bus for period {pp}
i4_Transit_Assgn_BM_{pp}.RPT	Cube transit assignment report for bus-metro for period {pp}
i4_Transit_Assgn_CR_{pp}.RPT	Cube transit assignment report for commuter rail for period {pp}
i4_Transit_Assgn_MR_{pp}.RPT	Cube transit assignment report for metrorail for period {pp}
i4_Trip_Distribution_ExtTrk.RPT	Cube auxiliary model report
i4_Trip_Distribution_IntTrk.RPT	Cube auxiliary model report
i4_Truck_Com_Trip_Generation.RPT	Cube auxiliary model report
MFARE2_CPI.TXT	Cube transit network report
mod2.RPT	Cube transit network report
Prepare_non_motorized_skims.RPT	Cube non-motorized network skimming report
PT_NetProcess_PT.RPT	Cube transit network report
set_CPI.RPT	Cube CPI processing report
Summarize_trip_tables_at_jur_level.RPT	Cube trip tables at jurisdictional level report
Transit_assignment.RPT	Cube transit assignment report
TRN_Deflator.txt	Cube transit network report
Truck_Com_Trip_Generation.RPT	Cube auxiliary model report
V2.3_highway_build.RPT	Cube highway network report
V2.5_PTNet_Build.RPT	Cube transit network report
V2.5_PTNet_Build_Iteration.RPT	Cube transit network report
view_from_space_gen3_external_exogenous_trips.RPT	External and exogenous trip table summary report
View_from_Space_Summary_Gen3.csv	Regional and subregional summary in a spreadsheet. The summary includes (1) Overview; (2) Long-Term Model Summaries; (3) Tour-Level Summaries; (4) Trip-Level Summaries; (5) Exogenous Trip Summary; (6) Highway Assignment Summaries and (7) Transit Assignment Summaries. Some values are shared between this spreadsheet and the visualizer. For certain measures, the View_from_Space_Summary_Gen3.csv file reports values in absolute terms, while the

	visualizer displays the same measures as percentages.

3.2.8 Skims

The “skims” folder contains nearly 1,300 skim files. These files are all named with one of two patterns. The first pattern is i4_{pp}_{am}_{tm}_{em}.FAR/SKM/TTT. These are transit skims, and the naming follows the pattern in Table 41. A second pattern is used for auto skims, i4_{mode}{_option}.skm, and the pattern elements are listed in Table 42. The highway skim files include four tables, time in minutes, distance in tenth miles, toll in cents, and variable-price toll in cents.

A subfolder of this folder is OMX_Skims and includes a copy of each skim file that has been converted to the Open Matrix format for use in ActivitySim. In addition to these copies, there is a county.omx file, which is a matrix of destination jurisdiction code and is built on-the-fly from the land use input data. Additionally, a file called skims.omx is created in this folder and is a consolidated copy of all skim files in this folder for the current iteration. At the end of a model run, all the .omx files in the OMX_Skims folder are automatically moved to the temp_files folder (to be removed in the future to save storage space), and thus the OMX_Skims folder will be empty.

Table 41: Transit Skim File Name Pattern Elements

Token	Data Element
{pp}	Time period (AM, MD, PM, NT)
{am}	Access Mode (WK = Walk, DR = PNR, KR = KNR)
{tm}	Transit Mode (AB = All-bus, MR = Metrorail only, BM = Bus and Metrorail combined path, CR = Commuter rail)
{em}	Egress Mode (WK = Walk, DR = PNR, KR = KNR)
File Extension	FAR = Fare Skim SKM = Skim TTT = Sum of transit time

Table 42: Highway Skim File Name Pattern Elements

Token	Data Element
{mode}	Mode (sov, hov2, hov3)
{_option}	If there is no _option, it is a skim with time in minutes, distance in miles, and tolls in dollars. If the option is _MC, it is a skim for mode choice.

The transit skim files include several tables, which are listed in Table 43.

Table 43 Transit Skim Matrix Tables

Name	Type	Description
IVTLB	Decimal	Local bus in-vehicle time (minutes)
IVTXB	Decimal	Express bus in-vehicle time (minutes)
IVTMR	Decimal	Metrorail in-vehicle time (minutes)
IVTCR	Decimal	Commuter Rail in-vehicle time (minutes)
IVTLR	Decimal	Light rail in-vehicle time (minutes)
IVTBR	Decimal	Bus Rapid Transit in-vehicle time (minutes)
IWAIT	Decimal	Initial wait time (minutes)
XWAIT	Decimal	Transfer wait time (minutes)
WALKT	Decimal	Walk time (minutes)
DRVACCT	Decimal	Drive access time (minutes)
FARE	Decimal	Fare (dollars)
DRVACCD	Decimal	Drive access distance (miles)
PNRCOST	Decimal	PNR cost (not used)
BRDLB	Decimal	Local bus boardings (not used in skims)
BRDXB	Decimal	Express bus boardings (not used in skims)
BRDMR	Decimal	Metrorail boardings (not used in skims)
BRDCR	Decimal	Commuter Rail boardings (not used in skims)
BRDLR	Decimal	Light Rail boardings (not used in skims)
BRDBR	Decimal	Bus Rapid Transit boardings (not used in skims)
TOTTT	Decimal	Total travel time (minutes)
WALKACCT	Decimal	Walk access time (minutes)
WALKOTHT	Decimal	Walk transfer time (minutes)
XFERS	Decimal	Number of transfers
XFERPEN	Decimal	Transfer penalty (minutes)
PNRT	Decimal	PNR time (minutes)
PNRC	Decimal	PNR cost (cents)
TOTIVT	Decimal	Total in-vehicle time (minutes)
PROB	Decimal	Probability of using bus + Metrorail routes

3.2.9 Transit assignment

The transit assignment outputs are located in the “trn_assign” folder. These files follow the same nomenclature as the skim files shown in Table 41. For each iteration, time period, access mode, transit mode, and egress mode, there are four files:

- LINKVOL.DBF: a link volume file. See Table 44.
- S2SVOL.DBF: a station-to-station volume file. See Table 45.
- PRN file: Cube transit assignment report file.
- TRP file: Transit trip O/D table. Includes one matrix, which represents transit person-trips for a specific submarket.

Table 44: LINKVOL.DBF Output Table Fields

Name	Type	Description
A	Integer	A node of link
B	Integer	B node of link
MODE	Integer	Mode number
OPERATOR	Integer	Operator number
NAME	String	Line name (ID value)
LONGNAME	String	Line name (longer)
DIST	Decimal	Distance of link
TIME	Decimal	Transit travel time of link
LINKSEQ	Integer	Number of links in route
HEADWAY_1	Integer	Route headway
STOPA	Integer	If the A node is a stop (1) or non-stop (0)
STOPB	Integer	If the B node is a stop (1) or non-stop (0)
VOL	Integer	Volume for link, mode, operator, name
ONA	Integer	Boardings in current direction at A node
OFFA	Integer	Alightings in current direction at A node
ONB	Integer	Boardings in current direction at B node
OFFB	Integer	Alightings in current direction at B node
REV_VOL	Integer	Volume for reverse link (same mode, operator, name)
REV_ONA	Integer	Boardings in reverse direction at A node
REV_OFFA	Integer	Alightings in reverse direction at A node
REV_ONB	Integer	Boardings in reverse direction at B node
REV_OFFB	Integer	Alightings in reverse direction at B node

Table 45: Station-To-Station Output Table Fields

Name	Type	Description
I	Integer	Trip origin zone id
J	Integer	Trip destination zone id
FromNode	Integer	Trip origin station number
ToNode	Integer	Trip destination station number
Mode	Integer	Trip mode
Accum	Integer	Movement type code (all 5, indicating adjacent-by-mode analysis)
VOL	Decimal	Volume from station FromNode to station ToNode

3.2.10 Transit network

The transit network outputs are in the “trn_net” folder. There are over 600 files in this folder following a few patterns:

- LEG and NT files: These are network link files (text-based) for access and egress Cube Public Transit (PT) Non-Transit (NT) legs. The file names include the time period, access mode, transit mode, and egress mode (if applicable)
- LIN files: These are transit line files in Cube PT format (text-based)
- I4_{pp}_{acc}_{trn}_{egg}_LINK|LINKVOL.DBF (using the pattern listed in Table 41): The “_LINK” and “_LINKVOL” files are identical and use the fields listed in Table 44 except the boarding, alighting, and volume fields.
- NET files: This file is the transit network with the mode combination. This can be read in Cube to show routes, connections (access, egress, and transfer links), etc.
- RTE files: This is the Cube route file. This file can be read in Cube to trace transit paths.
- Skim.PRN: This is a skim report file

4 RUNNING THE MODEL

4.1 What to prepare/change

The initial model inputs were prepared for the 2018 base year model. The model package may include inputs for other analysis years, such as 2025 and 2050. Inputs to the Gen3 Model, such as the highway network, transit network, land use, and/or other files, tell the model about the transportation and land use conditions that the model will be simulating. Some model files can be altered to reflect policy decisions that affect if, how, when, or where people travel. For example, some model files can be modified to increase the number of persons working from home, which would cause fewer people to make work-related trips.

For updates to the Long-Range Transportation Plan (LRTP)/Metropolitan Transportation Plan (MTP), air quality conformity analysis, and traffic impact studies, changes to model inputs are generally limited to the input highway network, input transit network, land use input files, and auxiliary model input files. These files should be updated to reflect the horizon year of the study.

The travel model can also be used to conduct project planning studies, regional studies, policy analyses, and exploratory planning studies on various topics. For example:

- Corridor studies
- Deployment of Autonomous Vehicles (AVs)
- Telecommute behavior changes
- Tolling and congestion pricing studies

Note that the Gen3 Model is a regional travel model that is calibrated and validated mainly at the regional level. The application of the Gen3 Model in a corridor- or subarea-level transportation planning study usually requires re-calibrating/re-validating the model at a finer geographic level. Often, post-processing of travel demand model outputs, such as those found in NCHRP Report 765,[41] is also involved in project-level studies.

4.2 Which scripts to run

Similar to the setup in the Gen2 Travel Model, the primary model script contained in a scenario folder, run_model.bat, runs the entire model based on the scenario-specific inputs. Specifically, run_model.bat calls run_ModelSteps.bat which subsequently executes all the individual model steps. Model users should review Chapter 2 prior to running this script. By editing these two batch files, users may also conduct a “partial” model run that executes only certain model steps. It is recommended to create a copy of those batch files and rename them specifically for the partial run (e.g., run_model_part.bat and run_ModelSteps_part.bat). In that case, one needs to replace “run_ModelSteps.bat” with “run_ModelSteps_part.bat” in the run_model_part.bat file to call the appropriate partial ModelSteps file.

The main batch file run_model.bat is divided into four sections: USER SPECIFICATIONS, VISUALIZER SETTINGS, DEFAULT MODEL SPECIFICATIONS, and MODEL EXECUTION. A model user usually needs

[41] CDM Smith, Horowitz, A., Creasey, T., Pendyala, R., & Chen, M. (2014). NCHRP Report 765: Analytical travel forecasting approaches for project-level planning and design. Transportation Research Board of the National Academies.

to change the configurations in only the first two sections to set up a model run. The parameters in the last two sections usually do not require any change. The fourth section executes model steps in sequence for four speed-feedback loops. The user does not need to make any change to them unless a partial run needs to be executed. Please refer to Section 2.3.3 for more information.

4.3 Visualizer preparation

At the end of a model run, a Visualizer file in html format is automatically prepared to compare the model run either to the household survey data (which is the default) or to a previous model run. The former comparison is ideal for model base-year calibration and validation purposes, while the latter comparison would be more useful for a scenario analysis.

Specifically, the Visualizer program first summarizes the current model run to the folder {scenario}\outputs\visualizer, and then compares that data to the data from the household travel survey or from a previous model run in the path defined in the run_model batch file by the key BASELINE_SCENARIO_PATH variable. To make a comparison with a previous model run, the visualizer variables in Run_Model.bat need to be modified **prior to** executing the current model run. Example specifications for these variables are shown in Table 46. Refer to Section 2.3.3 for more details. Alternatively, COG/TPB staff also developed a "post-processing" visualizer program that allows model users to compare a model run to a previous model run after both model runs are executed.

Table 46: Visualizer Variables in Run_Model.Bat

VARIABLE NAME	DESCRIPTION	EXAMPLE
IS_BASE_SURVEY	Tells the Visualizer script to look for certain files – must be 'No' when comparing to a prior model run.	No
BASELINE_SCENARIO_PATH	Path to the scenario to use as the baseline scenario.	D:\ModelRuns\MWCOG_Gen3_Model\2025_baseline
BASELINE_SCENARIO_NAME	Baseline/previous scenario name	2025_NoBuild
ALTERNATIVE_SCENARIO_NAME	Alternative/current scenario name	2025_Build

4.4 Running the model on an on-premises server versus in the cloud

As discussed in Section 2.3.3, there is an automatic shutdown option for use with cloud servers where a cost might be accumulated for a server that is running (even if it is not being actively used by the user). However, if you are running your model run on an on-premises server, you probably do not want to shut down the on-premises server at the conclusion of your model run, since that could cause problems for other users of that on-premises server.

In general, COG/TPB staff can run the Gen3 Model on both an on-premises server and a cloud server without any problems. Occasionally, however, COG/TPB staff have encountered random model run errors that were associated only with cloud servers, but most of the time, the model run in the cloud is as stable as on an on-premises server. By random model run error, we mean that it was difficult or impossible to replicate the model run error.

4.5 Expected model warnings

Several warning messages that are printed to the screen are expected. Some of these are Windows command line error messages that are frequently seen when running a subsequent run on the same file structure. Others are Python messages and warnings that look ominous but are known. COG/TPB staff intend to fix these "false" error messages in the future.

4.5.1 Windows command line messages

"A subdirectory or file already exists": this message is shown on the screen when the model script attempts to create a folder that already exists. This happens during skimming in the second, third, and fourth iterations where the script attempts to create the omx_skims subfolder that is already created in the first iteration.

"SQLalchemy is not installed. No support for SQL output.": This message means that a library called SQLachemy is not installed, which is not used in any of the model processes. The message can be ignored.

4.5.2 ActivitySim messages

"PerformanceWarning: DataFrame is highly fragmented." This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using pd.concat(axis=1) instead. To get a de-fragmented frame, use "newframe = frame.copy()": This is a warning related to some of the ActivitySim models and can be ignored. It is not an error.

"mp_households_9 WARNING - activitysim.abm.models.trip_scheduling - trip_scheduling.i100.outbound.num_2 coercing 75 depart choices to most initial": This is a warning that some trips failed to be scheduled in the scheduling model, it is normal and should be a small number of trips.

4.6 Debugging model crashes

Troubleshooting a Gen3 Model crash can be challenging, as there are many moving parts. The first step to debugging a crash is to determine if there is an obvious reason - such as insufficient disk space, since the Gen3 Model requires a lot of disk space.

The next step is to determine where the model crashed. At times, the model may continue running after an initial crash and then finally stop after a subsequent crash. One of the best places to check for this is {scenario}\outputs\reports\{scenario}_fulloutput.txt, which is a log of everything written to the screen. From this, a user should be able to determine:

- Did the model crash in the initial iteration of the model application (known as the "pump prime" iteration) or one of the subsequent speed feedback loops (e.g., iterations 1-4)? A model crash that occurred during feedback loop iterations I2, I3 or I4 may indicate a random error. Random errors are errors that cannot be or are hard to duplicate. An example of a random error would be when two model runs are being conducted concurrently and both model runs attempt to use the same file at the same time, since the first model run could begin using the file and lock access to it, resulting in the second model run not being able to access the file, resulting in the second model run crashing.

- Did the model run ActivitySim correctly? Did the model start ActivitySim? If the model crashes at the ActivitySim step, does the ActivitySim log file ({scenario}\outputs\activitysim\log\activitysim.log) provide more information on the crash?
- Is there any fatal error (e.g., "ReturnCode = 2") in the last written voya*.prn file that is generated from a Cube Voyager process under the scenario root directory?
- Is there an error message written to the screen that tells what crashed?

4.6.1 Pre-run checks

Before executing any of the model steps, the main batch file ("run_Model.bat") will do the following checks:

- Check the root and/or scenario paths contain one or more hyphens, which is incompatible with Cube. If yes, the model run will stop with the following warning:

  The root and/or scenario paths contain one or more hyphens, which is incompatible with Cube.
  Please remove or exchange the hyphens and restart the model run.
  Exiting model run.

- Check if the user specified path to Python (%PYTHON%) exists. If not, the model run will also stop with the following error message:

  The specified Python path does not exist. Please verify that the gen3_model environment is correctly installed under C:\Users\%USERNAME%
  Exiting model run.

- Another check that the main batch file will perform is to verify that the "cubeversion" variable is set to one of the two allowed values "6.5.1" or "25.00.01". In the case that they are not set appropriately, the following text will be printed to the screen:

  Invalid cubeversion set! Possible values are 6.5.1 or 25.00.01.
  Exiting model run.

The main batch file will also perform three checks to verify that the visualizer settings are properly set. First, it will check whether the data for the baseline scenario is available, be that the survey or a previous run. In case the data is not available, it will display the first of the error messages below. Next, two checks are performed if the %IS_BASE_SURVEY% variable is set to "No". These verify that the %BASELINE_SCENARIO_NAME% and the %ALTERNATIVE_SCENARIO_NAME% variable are set properly. In the case these are not set, the second and third messages will be displayed.

1. BASELINE_SUMMARY_DIR: %BASELINE_SUMMARY_DIR%
   Baseline summary to compare to does not exist. Please verify parameters.
2. Comparing against an existing run, please specify BASELINE_SCENARIO_NAME.
3. Comparing against an existing run, please specify ALTERNATIVE_SCENARIO_NAME.

In the case that the selected version for Cube is "25.00.01", the main batch file will also check that there are no unmodifiable instances of ClusterManager.exe open. If there are, the model run will stop with the following text:

There is a remaining ClusterManager instance running - please close it before trying to run the model again.
Exiting model run.

Finally, three checks are performed to verify that the system used meets the requirements for running the model. First, the available disk space is checked and, if less than 500 GB is available, the first error message below will be displayed. Next, the number of logical CPU cores and available RAM are checked. If these are below the required 16 cores and 206 GB, the second and/or third of the below error messages are displayed.

1. INSUFFICIENT DISK SPACE TO CONTINUE!
2. Insufficient logical cores: {available_cpu} (minimum is 16)
3. Insufficient RAM: {available_ram} GB (minimum is 206 GB)

4.6.2 ActivitySim crashes

A way to determine what happened in ActivitySim is to check the output log. If ActivitySim crashes, the output log ({scenario}\outputs\activitysim\log\activitysim.log) will either show the error or show that an error occurred in a process log (mp_households_n-activitysim.log, where n is the process number). If activitysim.log has no errors, the last line will likely be:

{date} {time} INFO - activitysim.core.tracing - Time to execute all models: {seconds} ({minutes})

In this case, ActivitySim ran correctly.

The following ActivitySim crashes are the most common in our experience.

4.6.2.1 No space left on device

As the error implies, there is no space left on the drive where the output folder is stored. Additionally, sometimes the HDF5 library will throw an exception that looks similar to the output below:

```
  File "tables\hdf5extension.pyx", line 1297, in tables.hdf5extension.Array._
create_array
tables.exceptions.HDF5ExtError: Problems creating the Array.
```

4.6.2.2 Probabilities do not add up to 1.0

This error happens when one or more choosers have no available alternative. Generally, this happens when there are restrictions on all alternatives for that chooser. The best way to handle this issue is to trace one of the households for that model to investigate if there is a data problem or a model problem.

This error should not be seen in the production version of the Gen3 Model. However, during the model development phase, it can be seen during estimation data bundle preparation for various reasons if data shows an impossible combination. One example is if a person is marked as works-from-home and has a work location TAZ included in the person file used to create the estimation data bundles.

4.6.2.3 Expected {x} fields, found {y} fields

This happens when a configuration file is formatted incorrectly. Frequently the specific line where the error occurs is printed on the screen. If there are fewer fields found than expected (y < x), locate the line in the configuration file and check to see if there is a missing field. If there are more fields than expected (y > x), locate the line and ensure that there isn't an additional field and that an expression that includes a comma is always in quotes.

4.6.2.4 KeyError: {data field}

This means that the model specification or chooser annotation is referencing a field that it cannot find. One of the causes of that is if the model has a specific list of chooser fields. This is found in the model's settings file (yaml file) under the key SIMULATE_CHOOSER_COLUMNS, and there is a list of fields that are available to the model. Another possible cause is that the field does not exist for the chooser in a specific model step. For example, the household variable auto_ownership is not available to other model steps without being added in an annotation file.

4.6.2.5 Could not find {output_dir, data_dir, configs_dir} {path}

This message indicates that ActivitySim cannot find the folder (output_dir, data_dir, or configs_dir) at the path specified. Check all input and output folder paths and the configuration of ActivitySim and/or the model.

4.6.3 Cube crashes

There are two places to check Cube outputs to determine why Cube crashed. One of them is the report files in {scenario}\outputs\reports - this can be sorted by date/time and checked in the order of file creation time (newest to oldest). Additionally, some steps may not finish and may leave temporary print log files in the root directory of {scenario}. The filenames of such files will generally look like voyaNNN.prn, where NNN is a sequential number. Note that "VoyagerCrashDump_NNNNN.log" files rarely contain usable information and are quite common to see even in model scripts that successfully run.

Cube print log files can be difficult to understand since Cube frequently will attempt to run an entire step, even if an error happens in the middle of this step. For this reason, it's best to search for all lines that begin with "F(" [where "F" stands for failure and "W" stands for warning]- these will be followed by a number that indicates the error. It is generally best to fix the first error and re-run the step since errors will frequently cascade (e.g. a missing file will cause an error saying that it can't find the file, and additional errors will be displayed when parts of that file - matrices, field names, etc. - are referenced).

In the case of an error indicating that no path can be found in transit assignment - this error should not happen unless ActivitySim is set to cache the skim files. Ensure that read_skim_cache and write_skim_cache in {scenario}\source\configs\activitysim\configs\network_los.yaml are both set to False. Those should never be set to True in a production model. This ensures that ActivitySim is using the correct skim files and not old data ("old" can mean prior-iteration skim data in this case).

4.7 Model steps and speed-feedback loops

The travel model runs in three phases - the first phase is running a "pump prime" iteration of the speed feedback loop/process ("pp" Iteration), the second phase is running four speed-feedback loops (Iterations "I1" to "I4"), and the third phase is performing the final transit assignment and post-

processing. The purpose for the speed feedback iterations is to ensure consistency in travel speeds between the demand-side model and the supply-side model. In other words, the congested travel speeds coming out of trip assignment should be consistent with the travel speeds input to various demand models, such as destination choice. These steps are shown in Figure 5.

The initial pump-prime phase includes four steps. The first step sets pricing factors using Consumer Price Index data. The second step builds the highway network from the input data. The third step prepares initial highway skim data to be used in ActivitySim. The final step prepares transit fare data, which only needs to be done once.

The second phase implements the speed feedback loop, which is run four times and feeds back congested travel times (highway skims) based on ActivitySim output.

The feedback loop includes six steps. The first is transit path building and skimming, which prepares the transit skim data for use in ActivitySim. The second step is ActivitySim, which models the movements of internal (resident) trips. The third step is auxiliary trips, which prepare truck, commercial vehicle, and other miscellaneous trips that are not handled by ActivitySim, including airport passengers, visitors, through traffic, external-internal, and internal-external trips. The fourth step is highway assignment preparation, which prepares the vehicle trip tables to be assigned in high assignment. The fifth step is highway assignment, which assigns all the auto trips from ActivitySim and the auxiliary models onto the highway network. The sixth step conducts highway path building and skimming and update highway link speeds.

The final phase includes transit assignment, which only needs to be run at the end of the model, and two post-processing steps to prepare a dashboard of the ActivitySim output and a spreadsheet-style “View from Space” summary of aggregated model outputs. The final phase also includes other procedures such as moving less important model output files to the “temp_files” folder, etc.

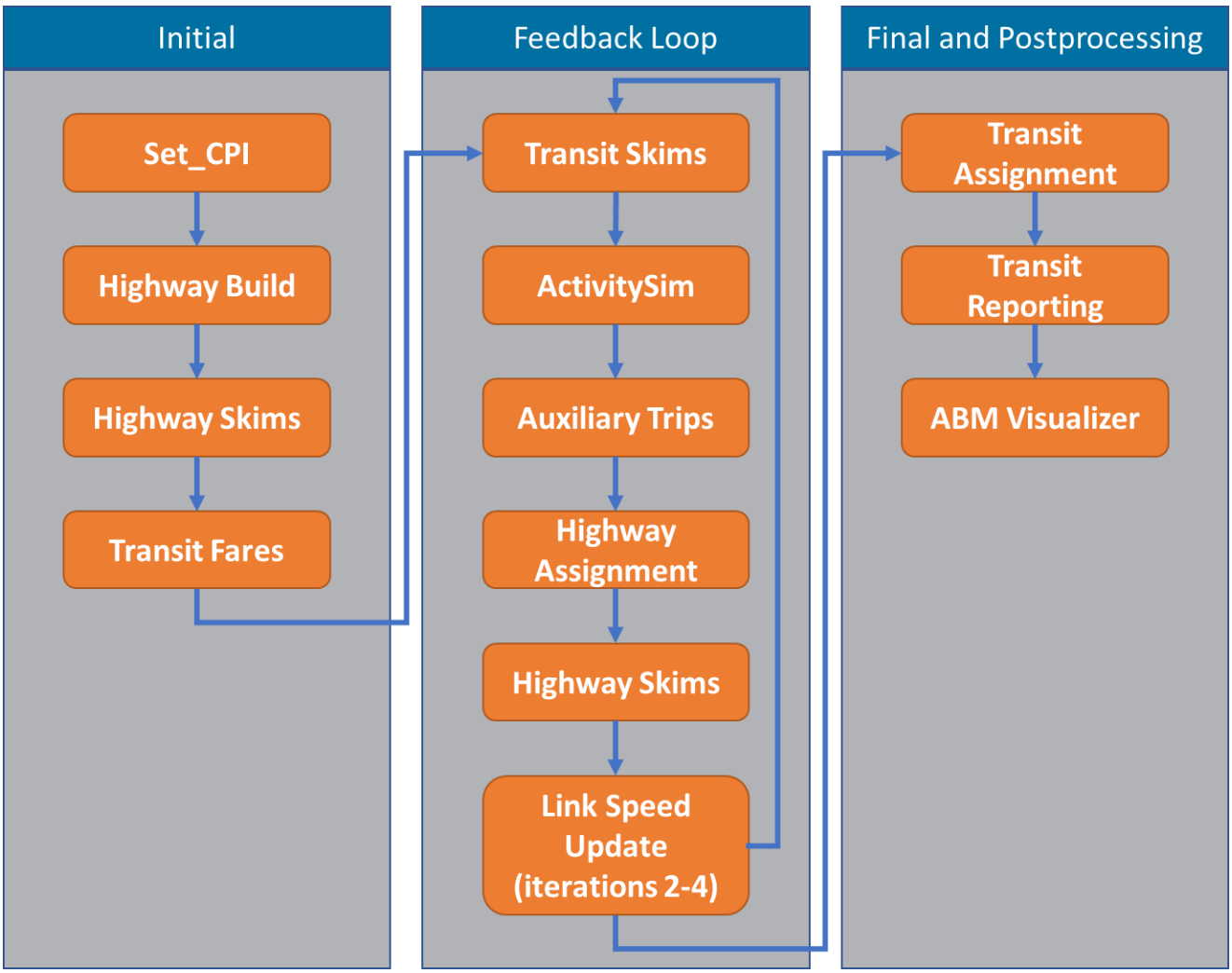


Figure 5: Model Flow Summary

5 WORKING WITH ACTIVITYSIM

ActivitySim is a complex modeling system that uses open-format files and attempts to standardize as much of the formatting as possible in terms of configuration and data tables across model steps.

5.1 How ActivitySim runs

ActivitySim starts by reading the input files listed in the settings file - which are the land use file, synthetic population and household files, and skim matrices. Once the files are read into memory, ActivitySim starts to execute the models listed in the settings file in sequence.

The first three model steps set up the data needed for the subsequent choice models. First, the `initialize_landuse` step runs through annotations on the land use. The annotation files are described later in this section, but generally they add variables to the data. The second step is `compute_accessibility`. This model computes zonal accessibility data that is used throughout the model. The third step is `initialize_households`, which runs through annotations on the households (In ActivitySim, annotation is used to add results from a model step to the ActivitySim data tables). The next several steps (everything between initialize_households and write_data_dictionary) are specific model steps. Each of these steps has a model settings file, a model specification, and a coefficients file. Some of these steps have an annotation file, and mode choice models include an additional coefficient template file. These files specify how each model step will run. Annotation files are described in Section 5.2. The model settings, specification, coefficient, and coefficient template files are all discussed in Section 5.2.

The final four steps of the model, including `write_data_dictionary`, `track_skim_usage`, `write_tables`, and `write_trip_matrices`, direct ActivitySim to write the outputs. The most important two steps are `write_tables` and `write_trip_matrices`: `write_tables` directs ActivitySim to write the tables listed in the settings file (under `output_tables`) and `write_trip_matrices` directs ActivitySim to write out OMX trip matrix files that can be input into Cube.

5.2 ActivitySim configuration files

As noted earlier, most ActivitySim model steps use three files - a settings file, a specification file, and a coefficient file for configurations. Mode choice models use a fourth file, a coefficient template file, that bridges the alternatives in the specification file with the coefficients by purpose. The four types of configuration files are explained in turn below.

5.2.1 Model settings files

The specification files - named as a model step and with a yaml extension - tell ActivitySim which files to use for the model step and other pertinent information about the model specification. These are mostly simple but can become complex at times.

There are two keys that are required for each model, SPEC and COEFFICIENTS. SPEC is the file name of the specification file, and COEFFICIENTS is the file name of the coefficients list. These two files are discussed later in this section.

Sometimes these files will include a line that says annotate_choosers. This directs ActivitySim to run an annotation file on the choosers BEFORE running the model simulation.[42] This allows additional variables to be included for that model. This is common with mode choice models where things like the availability of a mode may be defined in an annotation file to make the model spec easier to read.

Sometimes these files will include a line that says annotate_<table>, where table is persons, households, or another table in ActivitySim. This directs ActivitySim to run an annotation AFTER the model simulation. This is used to add results from a model step to the internal and output ActivitySim tables.

Sometimes these files will include CONSTANTS, which defines constants used in that model step. The mode choice versions of these files also define the nest structure of the model since mode choice in ActivitySim is usually a nested logit model.

5.2.2 Model specification files

The model specification files determine how the model calculates the utilities for each alternative. These files are CSV files with fields for description, fields for expression, and fields for each alternative. The expression defines the data, and the fields for each alternative define the alternative-specific coefficient, which can be a coefficient name (variable) or a number. The best practice is to use -999 as the coefficient value when an expression is used to limit the availability of an alternative. For all other applicable coefficients, the name should be used. The coefficient name needs to be identical to that used in either the coefficients file or the coefficients template file (mode choice models only).

5.2.3 Model coefficient files

The model coefficient files simply list the coefficient name, value, and a field that indicates if the value should be constrained by estimation software or calibration scripts. The coefficient names are case-sensitive and should not contain any space. The values should only be numbers, and the constrain indicator field should only be either F (for false) or T (for true). The constrain field does not affect model results. It is only used in model estimation and calibration.

5.2.4 Model coefficients template file

In the case of mode choice models, the coefficient template file links alternative coefficients to purposes. The file is a CSV file and has a field for the coefficient name and fields for each purpose in the model. The coefficient name field should match the coefficient names used in the model specification file. The fields for each purpose should match the coefficients file. When the mode choice model is simulated, ActivitySim will look at the specification file for the coefficient, then look for that coefficient in the template file. Then it will select the coefficient name for the tour/trip purpose it is simulating and select the corresponding coefficient value from the coefficients list.

[42] A chooser the decision-making agent in a particular model step, such as a person in the mandatory_tour_frequency step, or a tour in the tour_mode_choice step. A chooser is represented as a row in the Pandas DataFrame (table).

5.2.5 Annotation files

The annotation files are files that add variables to a table. In some cases, this is to format data that ActivitySim is expecting, such as the ptype annotations in annotate_persons.csv. In some cases, it involves getting data from another table. There are two types of annotation files in ActivitySim: preprocessors (usually ending with "_preprocessor" in the filename) and annotators (usually starting with "annotate_" in the filename). A model preprocessor is run right before a specific model is run and adds temporary columns to the chooser table that only that model can access. An annotator runs right after a model is run and appends permanent columns to the table of interest.

Each annotation file is a CSV file with three fields:

- Description: This is a text description of the data. It is not used by ActivitySim.
- Target: This is the variable name. This is case-sensitive and cannot contain any space.
- Expression: This is a Python expression that defines the data in the target.

5.2.5.1 Expression examples

Example 1: Get number of retired adults in household

This example uses the _PERSON_COUNT function (see Example 3 for more information) to count the number of persons in the household that are ptype 5, which is retired adults. This is placed in the variable _num_retired_adults. This variable is not written to the output file - beginning a variable name with an underscore tells ActivitySim that it should not be written to the output file. Note that the expression is in quotes because it includes commas in the expression - the quotes will prevent ActivitySim from parsing the expression into several fields (and ultimately throwing an error).

Target: `_num_retired_adults`

Expression: `"PERSON_COUNT('ptype == 5', persons, households)"`

Example 2: Determine if a household is retired adults only

This example looks at the household size (household.hhsize) and compares it to the _num_retired_adults variable to determine if the household has only retired adults. This variable will be written to the output file.

Target: `retired_adults_only_hh`

Expression: `(households.hhsize > 0) & (households.hhsize == _num_retired_adults)`

Example 3: Person Count Query

This is a query that is used in other annotation lines. The target begins with an underscore, which tells ActivitySim to not include this variable in the output. The expression uses a lambda, which is essentially a function that uses query, persons, and households as input arguments and then queries the persons table based on the query argument, groups by household_id (which is a reserved term in ActivitySim), gets the size, reindexes to the order of the households index (note that

this is a Pandas Series operation,[43] NOT the ActivitySim operation), fills NA values with 0, and then returns the values as an integer.
Target: `_PERSON_COUNT`
Expression: `"lambda qry, persons, households: persons.query(qry).groupby('household_id').size().reindex(households.index).fillna(0).astype(np.int8)"`

Example 4: Using Reindex to Assign a Variable from Another Table

This example gets the jurisdiction code (JURCODE) from the land use file and sets it as the home_jurisdiction variable based on the household's home_zone_id. Different from the last example, the reindex() function here is an ActivitySim function that takes the destination variable as the first argument and the destination index as the second argument. The destination index needs to be the same data value-type as the table being reindexed - in this case, land use data is indexed on the TAZ ID, and households.home_zone_id is the TAZ ID of the home zone, so reindex returns the JURCODE from the land use file for each home zone.

Target: `home_jurisdiction`
Expression: `"reindex(land_use.JURCODE, households.home_zone_id)"`

5.3 Working with ActivitySim model outputs

This section provides some additional guidance about using the ActivitySim output files. ActivitySim outputs matrix files, which are summaries of trips compatible with software to assign these matrices (such as Cube Voyager). ActivitySim also outputs several tabular files in comma-separated values (CSV) format, which are described in detail in Section 3.2.

5.3.1 Working with household, person, tour, and trip files

Tabular file data dictionaries are listed in Section 3.2. In all files, there is an id field that can be used to join to other files. The id values (household_id, person_id, tour_id, trip_id) are unique such that only one needs to be used to join (e.g. if joining a person value to the tour table, only the person_id needs to be used to join. All tables include as many id values as possible. For example, in the trip table, the household_id, person_id, tour_id, and trip_id fields are included.

The tabular file data from ActivitySim are written to CSV files, which can easily be input into R, Python, and many other software packages. Excel is generally not recommended, however, since the files tend to be larger than what Excel allows. Excel allows 1,048,576 rows, and a full ActivitySim run has several output tables that drastically exceed this limit. Example Python code to read the files into a Pandas dataframe and create some crosstabs is included in Section 7.3.

5.4 Model calibration and validation

This section describes advanced techniques for the calibration and validation of the Gen3 Model. These techniques are different than those for model estimation[44] because estimation is conducted

[43] Pandas, API reference, 2025. https://pandas.pydata.org/docs/reference/api/pandas.Series.reindex.html
[44] RSG and Metropolitan Washington Council of Governments. "GEN3 Model Phase 2 Model Estimation," Technical Report, Prepared for Metropolitan Washington Council of Governments, March 2, 2023.

using ActivitySim only, whereas calibration and validation are conducted in the main model process to allow for highway and transit assignments and potential changes to the skims. Additionally, these techniques will allow for more rapid testing and adjusting of ActivitySim.

The procedures in this chapter are for advanced users only. These procedures involve editing the Windows command line batch files, editing ActivitySim files (many of which are text-based), and running scripts to use the outputs and adjust the model configuration files.

It is also important to note that both the ActivitySim Consortium and ActivitySim end users, such as COG/TPB, are working to streamline and automate the model calibration and validation processes. The procedures described below represent only one possible way of how calibration/validation can be conducted in the Gen3 Model. A recent memo developed by COG/TPB staff discussed the general calibration methodologies and various calibration tools that have been developed for the Gen3 Model so far.[45]

5.4.1 Calibration preparation

Prior to starting model calibration, comment out the script that moves unimportant model outputs into a temp_files folder. This is near the end of run_modelsteps.bat, and should look like this once commented out:

```
:: Move unimportant files to a temp folder
::call %BATCH_DIR%\move_temp_files_gen3.bat %1
```

The reason for this is because that script will move the ActivitySim pipeline files that are used to re-start ActivitySim from partway through the run.

Next, run the entire model. This ensures that the skims used in ActivitySim are the final skims after the feedback loop iterations. Alternatively, if a full model run, which was completed on the same computer the calibration will be performed on, is available and the temp files (in {scenario}\temp_files) have not been deleted, simply move all of the pipeline folders (pipeline.parquetpipeline, mp_households_*n*-pipeline.parquetpipeline, and mp_households_*n*-pipeline, where n is 0 - number of processors) back to the ActivitySim output folder ({scenario}\outputs\activitysim). Please also ensure that the breadcrumbs.yaml file is present in the ActivitySim output folder.

Next, turn off steps that do not need to be run. This is best completed by editing the run_modelsteps.bat script. This script is broken up into sections - the first section from the beginning of the file to `ECHO ====== Pump Prime Iteration` sets several environment variables needed for the script (as does run_model.bat, which calls this script). The next sections are for the speed feedback loop iterations, and all have the same set of calls to scripts - transit skims, ActivitySim, auxiliary trips, highway assignment, and highway skimming. For calibration and validation of ActivitySim only (and when using an existing model run), all steps can be commented out except the Iteration 4 RunABM script. Do NOT comment out any `set` statements, as those are necessary for the model to run. Alternatively, for those who are familiar with Windows command line batch scripting, a GOTO statement can be added before the Pump Prime Iteration marker sending execution to immediately follow the Iteration 4 marker; Then the call to transit skims needs to be commented out and a GOTO MSEND command needs to be added after the runABM call.

[45] Feng Xie, Glenn Lang and Bahar Shahverdi to Mark Moran, "ActivitySim Model Calibration for the Gen3 Travel Model: Methodologies, Past Efforts, and Current Tools", COG/TPB Memorandum, November 17, 2025.

The next step is to edit RunABM.bat (in source\scripts\batch). For calibration, it may not be necessary to run ActivitySim for all households, so the household sample size for iteration 4 (SAMPLE_HH4) can be changed from 0 (run all) to another number (this needs to be a specific number of households, as unfortunately ActivitySim does not yet support percentages.).[46] In the case of the Gen3 Model, since the TPB modeled region has around 3 million households, a sample of one million households would probably be sufficient for model calibration. In the case that skims are not changing, the scripts to convert Voyager scripts to OMX (look `START ..\source\scripts\batch\skm_to_omx.bat %1 [AM|MD|PM|NT] [1|2|3|4]`) and the script to fix the index in the OMX files (look for `%PYTHON% %PYTHON_SCRIPTS_DIR%\cube_to_omx.py %_iter_% %_prev_% %SCEN_DIRECTORY%`) and the script to consolidate the skims OMX file (look for `%PYTHON% %PYTHON_SCRIPTS_DIR%\build_omx.py %_iter_% %_prev_% %SCEN_DIRECTORY%)` can be commented out (using either `REM` or `::` at the beginning of the command lines). Additionally, the second skim builder (look for `%PYTHON% %PYTHON_SCRIPTS_DIR%\make_county_omx.py`) can also be commented out since the files created by this process will have already been created (there are two instances of this call, both can be commented out).

In the case of repeated ActivitySim runs, the skim cache can be turned on. This is done by editing the network_los.yaml file in source\configs\activitysim\configs. The options `read_skim_cache` and `write_skim_cache` can be set to True (case sensitive!) to direct ActivitySim to cache the skims file in the output cache folder. Be advised that this option will result in a 55-80GB cache file being written tothe output cache folder. Once calibration is complete, this option needs to be set to False and the skim cache file can be deleted.

To reduce ActivitySim runtime, ActivitySim can be set to start at a specific model by opening the settings file in source\configs\activitysim\configs_mp\settings_source.yaml and uncommenting and setting `resume_after:` to the step before the step in calibration. To avoid running several additional models, comment out (using a hash symbol, #) the steps following the step in calibration up until the `### mp_summarize step` comment. This will start ActivitySim at the step to calibrate , run the uncommented steps, and then run the outputs (writing output tables and matrices). After the

[46] The sample size for model calibration depends on the model being calibrated, the number of decision-makers that the model is applied to, and the alternative-specific-constants (ASCs) that are being calibrated. Unfortunately, there is no magic formula for this, so we use judgement when setting the sample rate when calibrating models. For example, calibration of ASCs for each auto ownership alternative can be completed with a relatively small sample, like 5% of all households. That's because the model is applied to all households, there are a small number of alternatives, and the choice set is available to all decision-makers. Now consider at-work subtour destination choice. In this case the sample needs to be larger, because the model only applies to work tours and there are a large number of alternatives. In that case we might want to run a 20% sample. Mode choice is somewhere in between for overall calibration, but might require a bigger sample to get specific transit markets dialed in. One thing to note though is that sometimes we want to assign trips to the network as part of the calibration process, e.g. calibration of constants for rail modes in mode choice. In that case sometimes we calibrate using 100% of the population. Similarly, the transit subsidy model applies to the entire population and has a small number of alternatives, so a small sample rate of 10% should be adequate. Generally, for those who are not familiar with the ActivitySim models, one million households (which translates into a roughly 33% sample as there are around three million households in this region) would be adequate for the calibration of most models (Personal communication, 4/11/25, Joel Freedman to Feng Xie and Ali Etezady).

`### mp_summarize step` comment, comment out the `write_data_dictionary` step, and further down, under `multiprocess_steps`, where - `name: mp_summarize` is, change the value of begin from `write_data_dictionary` to `write_tables` Note that the model output tables (final_households.csv, final_persons.csv, etc.) are only written if the appropriate steps are run - for example, the final_trips.csv file is not written unless the stop frequency model is run. Additionally, the output tables will not include data and fields from steps that are not run - for example, if the at-work subtour models are not run, the final_tours.csv file will not include the parent_tour_id field and will not include any at-work tour data at all. For most situations, running just the step to calibrate is fine. For tour mode choice, however, the group of models of tour_mode_choice_simulate, atwork_subtour_frequency, atwork_subtour_destination, atwork_subtour_scheduling, and atwork_subtour_mode_choice need to be run.

If any annotation files before the step being run are changed (e.g. annotate_households, annotate_persons, annotate_land_use), the model will need to be run from the start to reflect those changes. If the annotate choosers file for the step being run is changed, then just that step needs to be run.

At this point, the model should run and output updated results for only the steps selected. The outputs can be read into another software package to estimate alternative specific constants.

5.4.2 Model calibration

Model calibration refers to the adjustment of model constants to match observed data. This observed data should be different than the data used to estimate models (if applicable). For example, if the work-from-home model was estimated using household survey data, the calibration data should come from Census American Community Survey data.

An example model calibration script is supplied in Section 7.3. Additionally, the ActivitySim Visualizer displays many comparisons of the current model run to the survey by default (note that this can be changed by following the steps in Section 2.3 Step 2).

In-depth concepts surrounding model calibration/validation are included The Model Validation and Reasonableness Checking Manual[47] prepared for the Federal Highway Administration Transportation Model Improvement Program.

5.4.3 Model validation

Validation of the model refers to the comparison of model output results against observed data. If applicable, the observed data should be independent from the data used for model estimation or calibration). Prior to model validation, observed data and model outputs must be gathered and formatted in a way that they can be compared with each other.

[47] Cambridge Systematics. Travel Model Validation and Reasonableness Checking Manual, Second Edition, prepared for Travel Model Improvement Program, Federal Highway Administration, Washington, D.C. https://www.fhwa.dot.gov/planning/tmip/publications/other_reports/validation_and_reasonableness_2010/fhwahep10042.pdf.

Validation of ActivitySim outputs is best completed through the ActivitySim Visualizer ({scenario}\outputs\SURVEY_vs_Gen3.html). This visualizer is already set up to aggregate data as appropriate and handles much of the joining and linking needed to show charts.

Two major items of validation are not addressed by the visualizer: highway and transit assignment validation. RSG has developed two Jupyter Notebooks to prepare data for these two items. COG staff has converted the Jupyter Notebooks to a Python-based post-processing program that conducts both highway and transit validation for the Gen3 Travel Model.

5.4.3.1 Highway validation

The highway validation Jupyter Notebook reads the output highway network ({scenario}\outputs\hwy_net\i4_Assign_Output.net), converts it to a DBF, and then processes the data to populate the highway validation spreadsheet. This notebook then compares highway counts and simulated volumes from highway assignment on a segment (with counts) basis and aggregates the data by area type, facility type, jurisdiction, and screenline. An example spreadsheet table from this script is presented in Figure 6.

Table 4: Highway Validation - Estimated VMT vs. Observed VMT based on Links with Traffic Count - By Area and Facility Type

Model Run: **Gen3 Model (01_29_2026)**

Estimate_2018

	Facility Type						
Area Type	Freeway	Major Arterial	Minor Arterial	Collector	Expressway	Ramp	TOTAL
1	510,884	749,346	320,206	96,229	338,834	0	2,015,500
2	5,243,644	3,222,063	1,674,398	270,026	606,284	0	11,016,415
3	8,746,122	2,840,366	2,149,447	586,949	1,713,432	0	16,036,315
4	5,133,058	2,065,871	1,927,239	420,345	707,547	0	10,254,060
5	9,280,783	3,832,517	2,324,853	573,015	1,436,154	28,543	17,475,865
6	6,850,756	5,191,868	6,159,046	574,525	659,691	0	19,435,887
TOTAL	35,765,248	17,902,031	14,555,189	2,521,089	5,461,942	28,543	76,234,042

Observed_2018

	Facility Type						
Area Type	Freeway	Major Arterial	Minor Arterial	Collector	Expressway	Ramp	TOTAL
1	554,106	700,612	363,478	93,258	412,572	0	2,124,026
2	5,466,431	3,419,008	1,769,923	362,173	888,596	0	11,906,131
3	8,878,084	2,964,965	2,343,673	785,763	1,927,039	0	16,899,524
4	4,872,291	2,042,296	2,122,981	535,084	787,959	0	10,360,611
5	9,472,552	3,391,423	2,144,360	719,425	1,297,292	36,284	17,061,336
6	6,119,478	4,381,387	4,234,617	657,833	667,139	0	16,060,454
TOTAL	35,362,942	16,899,691	12,979,032	3,153,536	5,980,597	36,284	74,412,082

Estimate/Observed Ratio

	Facility Type						
Area Type	Freeway	Major Arterial	Minor Arterial	Collector	Expressway	Ramp	TOTAL
1	0.92	1.07	0.88	1.03	0.82	-	0.95
2	0.96	0.94	0.95	0.75	0.68	-	0.93
3	0.99	0.96	0.92	0.75	0.89	-	0.95
4	1.05	1.01	0.91	0.79	0.90	-	0.99
5	0.98	1.13	1.08	0.80	1.11	0.79	1.02
6	1.12	1.18	1.45	0.87	0.99	-	1.21
TOTAL	1.01	1.06	1.12	0.80	0.91	0.79	1.02

Figure 6: Example Highway Validation Spreadsheet Output

5.4.3.2 Transit validation

The transit validation Jupyter Notebook reads all of the transit route loading files and station-to-station output files and summarizes them in comparison to the ridership data. The spreadsheet shows an overview of transit loadings by mode, Metrorail loadings by station group, Metrorail station from-to comparison, and commuter rail loadings by station. An example of this spreadsheet is included as Table 47.

Table 47: Example transit Validation Spreadsheet Table

	2018 Daily Boardings and Alightings Estimates (Gen3 Model)														
Mode Name	All Bus: Sum of ONA	All Bus: Sum of OFFB	Bus & Metrorail: Sum of ONA	Bus & Metrorail: Sum of OFFB	Metrorail: Sum of ONA	Metrorail: Sum of OFFB	Metrorail: Boardings without Transfers	Commuter Rail: Sum of ONA	Commuter Rail: Sum of OFFB	**Combined: Sum of ONA**	Combined: Sum of OFFB	**2018 Observed Station ENTRIES or BOARDINGS**	2018 Ratio E/O for Boardings vs. Entries	**Notes**	**Gen 2.4 2014 E/O**
Local Metrobus	272,402	272,403	61,227	61,226	0	0		4,526	4,526	338,155	338,156		n/a		
Express Metrobus	20,864	20,863	13,623	13,623	0	0		266	266	34,753	34,753		n/a		
Metrorail	0	0	169,952	169,951	678,011	678,011	659,167	28,563	28,563	876,525	876,526	641,227	1.03	see Note 1	1.01
Commuter Rail	0	0	0	0	0	0		53,284	53,284	53,284	53,284	56,580	0.94	see Note 2	0.76
Other Local Bus in the WMATA Area	127,226	127,226	43,660	43,660	0	0		2,454	2,454	173,339	173,339		n/a		
Other Express Bus in the WMATA Area	2,414	2,414	2,556	2,556	0	0		183	183	5,152	5,153		n/a		
Other Local Bus beyond the WMATA Area	19,856	19,856	2,983	2,983	0	0		1,216	1,216	24,055	24,055		n/a		
Other Express Bus beyond the WMATA Area	26,475	26,475	14,256	14,257	0	0		1,961	1,961	42,693	42,693		n/a		
Bus Rapid Transit and Street Car	123	123	3,155	3,155	0	0		390	390	3,668	3,668		n/a		
All Bus	469,236	469,238	138,304	138,304	0	0		10,607	10,607	618,147	618,149	575,642	1.07	see Note 3	1.09
Metrobus Total	293,266	293,267	74,849	74,849	0	0	0	4,793	4,793	372,908	372,908	360,000	1.04	see Note 4	
Other Bus in WMATA Area	129,639	129,639	46,215	46,216	0	0	0	2,637	2,637	178,492	178,492	141,390	1.26	see Note 5	
Other Bus not in WMATA Area	46,331	46,332	17,239	17,239	0	0	0	3,177	3,177	66,748	66,749	74,252	0.90	see Note 6	

6 MORE DETAILED DESCRIPTION OF MODEL SYSTEM

6.1 Highway skimming and assignment

6.1.1 Overview

Highway skimming begins with best-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. Consistent with traffic assignment, highway skimming in the Gen3 Model is set up to develop skims for four time-of-day periods: AM peak period (AM), midday (MD), PM peak period (PM), and nighttime (NT). Highway skims in the Gen3 Model are generated after the traffic assignment step in each speed feedback loop iteration.

Highway skims are also generated by highway travel mode (SOV, HOV 2- occupant, HOV 3+occupant). In addition, truck and commercial vehicle (CV) 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 highway skimming process in the Gen3 Model develops zone-to-zone (3722 x 3722) skim matrices for the entire model region for use in ActivitySim and the auxiliary models. The entire highway skimming process is applied with the scripts named *Highway_Skims_4TOD.s Remove_PP_Speed.s and Prepare_Non_Motorized_Skims.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 48 and Table 49, respectively.

Table 48: Highway Skimming Program Inputs

DESCRIPTION	FILE NAME	FILE FORMAT
Network File	<iter>_HWY.NET	Cube Binary Network
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
PM toll factors by Vehicle Type	Inputs\PM_Tfac.dbf	DBF
NT toll factors by Vehicle Type	Inputs\NT_Tfac.dbf	DBF

Table 49: Highway Skimming Output Files

DESCRIPTION	FILE NAME	FILE FORMAT
Total highway skims	<ITER>_SKIMTOT.TXT	Text
Truck skims	<ITER>_MD_TRK.SKM	TP+ Binary Skim
SOV skims	<ITER>_<Prd>_SOV.SKM	TP+ Binary Skim

HOV2 skims	<ITER>_<Prd>_HOV2.SKM	TP+ Binary Skim
HOV3+ skims	<ITER>_<Prd>_HOV3.SKM	TP+ Binary Skim
AM highway skims	<ITER>_HWY_AM.SKM	TP+ Binary Skim
Off peak highway skims	<ITER>_HWY_OP.SKM	TP+ Binary Skim
Network with added station centroid connectors	<ITER>_HWYMOD.NET	Cube Binary Network
Walk access links	WalkAcc_Links.dbf	DBF
Highway network with PP speeds removed	ZoneHWY.NET	Cube Binary Network

6.1.2 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 are initially populated by values from look-up tables based on time period, facility type, and area type. After the PP iteration is completed, highway skimming is accomplished using traffic assignment-based (congested) link speeds. The generalized impedance for which paths are developed for highway skimming is defined as follows:

Equation 2: Assignment Generalized Cost

$$Impedance_v = (T)_v + (Toll_v * Tfac_v * Vfac_{vf})$$

Where:
Impedance = Total impedance for vehicle type v
T = Capacity-restrained time for vehicle type v
Toll = Toll (2018 dollars) for vehicle type v
Tfac = Toll factor for vehicle type v
VFac = Vehicle factor for vehicle-type v on facility f
Note: Vehicle classes are: SOVs, HOV2-occs, HOV3+occs, Commercial Vehicles, Trucks, and airport passenger vehicles.

	AM	MD	PM	NT
SOV	2.1	2.5	2.5	2.5
HOV 2	1.2	3.3	1.7	3.3
HOV 3+	0.8	3.3	0.8	3.3
Commercial Vehicles	1.7	1.7	1.7	1.7
Trucks	1.7	1.7	1.7	1.7
Air passengers	1.7	1.7	1.7	1.7

The assumed value-of-time rates that convert monetary travel cost (such as tolls) in dollars to equivalent travel time in minutes are provided by vehicle class and time period in toll_minutes.txt, which is located in the Support folder. The values shown are derived from average household income levels and information from the 2017/18 RTS. The values should not be altered.

The equivalent toll minutes for 2018 were increased from the 2007/2008 values using the US Consumer Price Index inflation factor[48] of 1.21 from May 2007 to May 2018.

The toll factors are provided by time period in the input files AM_Tfac.dbf and MD_Tfac.dbf. The file is available to allow for the ability to reflect a facility-specific toll policy differential between vehicle classes. 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.

Information about the "toll setting" process that is used to estimate reasonable toll values can be found in Section 6.2.

The Remove_PP_Speed.s script is used to remove the "PP" iteration speed attributes from the highway network. This is necessary at the end of the initial (PP) iteration as the initial speeds from the lookup table are to be replaced by traffic assignment speeds in subsequent speed feedback loop iterations.

6.2 Toll searching

In support of the regional travel demand forecasting, COG/TPB staff developed a heuristic toll searching algorithm that simulates tolls on variably priced tolling facilities in an iterative process for the Gen2 Travel Model. In 2015, COG/TPB staff implemented a toll setting procedure that streamlined the intermediate steps and eliminated manual intervention in between these steps. Since then, this toll setting procedure has been incorporated into COG/TPB's travel demand modeling processes for the Gen2 Travel Model and has by and large remained the same.[49]

In applications, model users usually use the toll input files that are provided by COG/TPB in the model transmittal package. In cases where the toll inputs need to be altered, model users need to follow a **two-run** approach by which **a Pump Prime model run** performs travel demand forecasting with pre-specified, base toll rates on variably priced facilities and executes COG/TPB's toll searching procedure, and **a final model run** re-generates travel demand forecasts with the estimated toll rates from the toll searching process. However, this two-run process, including the toll setting scripts, is maintained separately from the Gen2 Travel Model and is available to external model users upon request only.

COG/TPB staff adopted the same two-run toll-setting process for the Gen3 Travel Model. In the Gen3/Ver. 1.0.1 Model, COG/TPB staff integrated the toll setting process into the model flow such that a model user can choose to execute toll setting as part of the travel demand forecasting process by switching on a toggle in the main batch file (explained in detail in Section 2.3.3). For the first time, TPB's toll setting scripts will become available to external model users as part of the model

[48] US Bureau of Labor Statistics, CPI Inflation Calculator. https://www.bls.gov/data/inflation_calculator.htm. Accessed 5/31/2023.

[49] In 2018, the toll searching algorithm was slightly modified for the latest Version 2.3.75 travel model. The modified algorithm can exit the iterative toll searching process much faster on special occasions where a cap toll rate is imposed. This modification, while being able to reduce the toll searching time on special occasions, didn't affect toll searching results at all.

transmittal package. The integrated process is detailed in a technical memorandum[50] and illustrated in Figure 7.

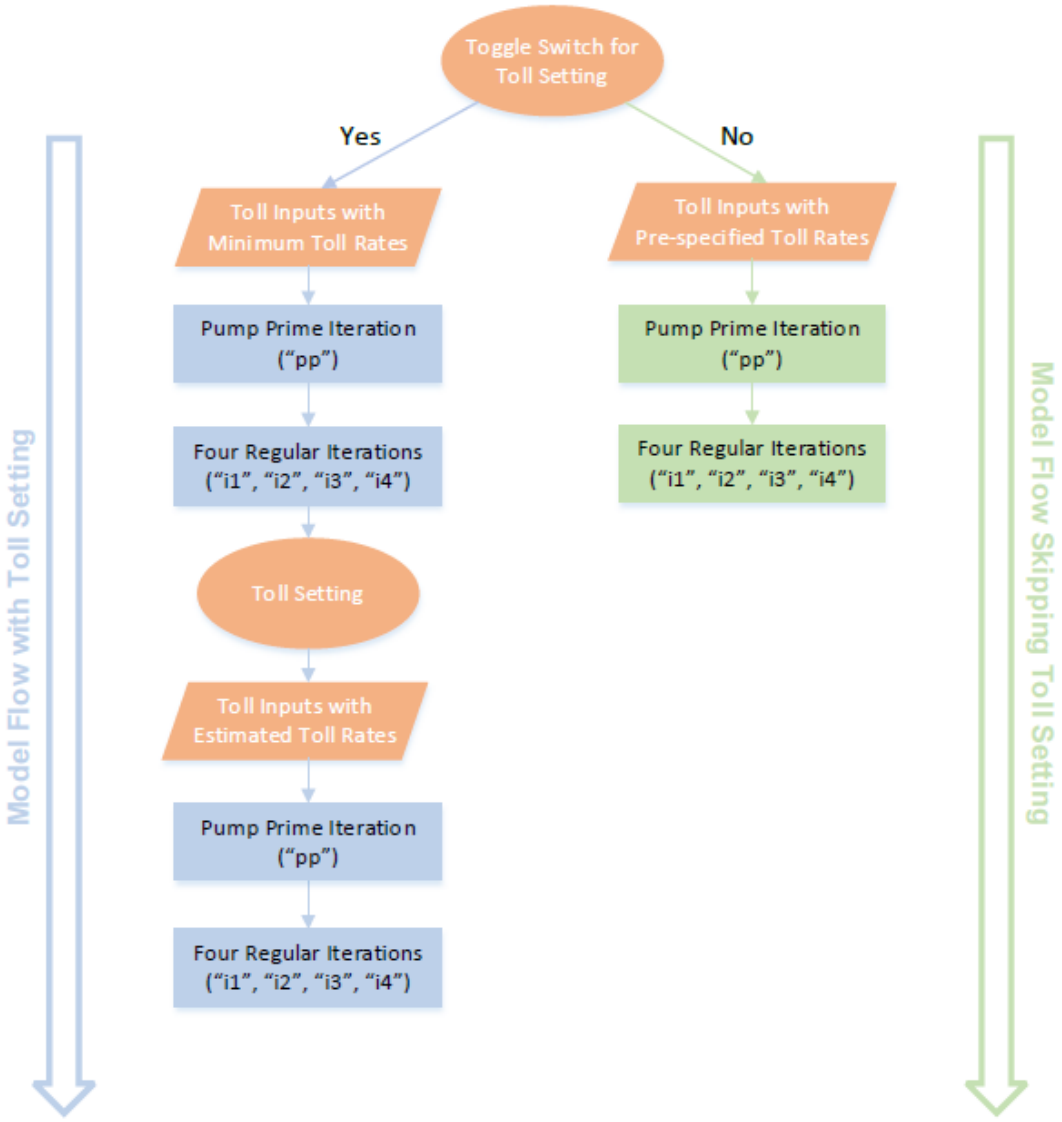


Figure 7: Gen3 Model Flow with an Integrated Toll Setting Process

During the model usability testing in Phase 3, COG/TPB staff found that variably priced tolls estimated from the Gen3 Model were significantly lower than those from the Gen2 Model, which were already low compared to the field data. As such, staff explored the possibility of increasing the tolls in the Gen3 Model by tinkering with the threshold volume-to-capacity ratio (V/C) in the toll searching algorithm. Based on the testing results, however, staff recommended keeping the current toll searching algorithm intact despite the lower toll estimates. Staff also made recommendations in

[50] Feng Xie and Bahar Shahverdi to Mark Moran, "Integrating Toll Setting in Gen3 Model Flow: Testing and Recommendations", COG/TPB Technical Memorandum, March 15, 2024.

terms of possible future endeavors to improve the realism of the toll representation in the Gen3 Model. These recommendations are documented in a technical memorandum.[51]

As noted in the memo, the toll setting process included in the Gen3 Travel model is primarily designed to provide "planning-grade" estimates of variably priced tolls where higher tolls could serve as an indicator of bottlenecks in the freeway system. Similarly, lower tolls could simply be an indication that the trip distribution or mode choice models underestimate the SOV demand in some of the HOT-lane corridors. The COG/TPB toll setting process has never been calibrated or validated to observed data, and thus, it should not be viewed as a realistic representation of the tolling system on the ground. **TPB staff always recommend against the practice of directly using the outputs from the COG/TPB toll setting process in any type of operation-level studies.**

6.3 Transit skimming and assignment

6.3.1 Background

While the Gen3, Phase 1, Model adopts the Bentley Citilabs Cube Public Transport (PT) Best-Pathing algorithm, which identifies a single best transit path between an origin zone (O) and a destination zone (D), and loads all O/D demand onto it, COG and RSG decided to switch to the PT Multipathing algorithm in the Gen3, Phase 2, Model, which builds multiple transit paths and finds the respective probabilities of using them. This decision was made largely due to the theoretical appeal of the multi-routing algorithm (i.e., PT is designed as a multi-path path builder to provide a more realistic representation of an urbanized environment with multiple transit options) and the significant runtime reduction associated with it.

Subsequently, COG/TPB staff implemented the PT Multipathing algorithm in the Gen3 Model in May 2022 based on an example implementation from the PT Arlington Model, which was developed by Bentley, the developer of the Cube software, as part of the Arlington County Travel Model.[52] Some of the specifications were further modified based on the RSG suggestions.

When examining the transit skims resulting from a test model run, however, COG/TPB staff noticed a hyperpath issue in the proposed PT Multipathing implementation.[53] Specifically, staff found that the multiple "hyperpaths" created by the PT multi-path path builder for commuter rail and "Bus + Metrorail" sub-modes did not always use the transit mode(s) of interest. Joel Freedman and Bill Woodford, the modeling experts at RSG, commented that the hyperpath issue is common in strategic transit path-finding algorithms and that multipath algorithms are probably not fully compatible with

[51] Feng Xie and Bahar Shahverdi, "Improving the Toll Setting Process in the Gen3 Travel Model: Testing and Recommendations", COG/TPB Memorandum, May 15, 2025.
[52] See, for example, Christine Sherman Baker and Filippo Contiero, "Modeling Public Transport in the Arlington Co. Tour-Based Travel Model," with William G. Allen Jr. (Bill), COG/TPB Travel Forecasting Subcommittee, held at the Metropolitan Washington Council of Governments, Washington, D.C., January 28, 2022, https://www.mwcog.org/events/2022/1/28/travel-forecasting-subcommittee/.
[53] See, Feng Xie and Meseret Seifu to Files, "Plotting Commuter Rail In-Vehicle Time (IVT) Skims based on the Gen3 Model with Proposed Public Transport (PT) Implementation", MWCOG/TPB Memorandum, June 7, 2022. Note that the term "hyperpath" is defined by Nguyen, Sang, Stefano Pallottino, and Michel Gendreau. "Implicit Enumeration of Hyperpaths in a Logit Model for Transit Networks." Transportation Science 32, no. 1 (February 1, 1998): 54–64. https://doi.org/10.1287/trsc.32.1.54

nested transit choice models. After reviewing the COG implementation files and testing different methodologies to address the hyperpath issue, Filippo Contiero, a PT expert at Bentley, acknowledged that there is no perfect solution in the current PT algorithm.

COG/TPB staff proposed a workaround to address the hyperpath issue in the Gen3 Model. The implementation and testing results of this workaround are detailed in a technical memo dated August 16, 2022.[54] While the proposed workaround resolved the hyperpath issue for commuter rail, it did not fully address the issue for "Bus + Metrorail" due to a discrepancy in the proposed methodology. Following Joel's suggestion, COG/TPB staff decided to adopt the workaround partially and move forward with the PT multipathing implementation that includes this partial fix to the hyperpath issue. Filippo concurred with this decision.

This section documents the implementation of the PT multi-pathing algorithm as well as the partial fix to the hyperpath issue that was implemented in the Gen3 Model.

6.3.2 Implementation

Switching from PT best-pathing to PT multipathing did not involve modifying any script file in the base Gen3 Model (i.e., the Gen3, Phase 1, Model received from RSG in January 2022). Instead, PT multipathing was implemented by modifying the PT factor files (.FAC) that are included as part of the transit inputs files (\inputs\trn). Specifically, the following changes were made to the factor files:

- Route enumeration parameters (e.g., ***AONMAXFERS, MAXFERS, EXTRAXFERS1, EXTRAXFERS2, SPREADFACT, REWAITMAX, RECOSTMAX***, etc.) and route evaluation parameters (e.g., ***ALPHA, LAMBDAW, LAMBDAA, CHOICECUT, SERVICEMODEL***, etc.) were either added or modified to enable multi-routing in PT. The specifications for those parameters were borrowed from those in the PT Arlington Model.[55]
- Mode-to-mode transfer penalties (***XFERPEN, XFERCONST*** and ***XFERFACTOR***) were added. Penalty values were specified per suggestion from RSG.
- Value of time (***VALUEOFTIME***) by mode, which was originally uniform across all transit modes, was updated per suggestion from RSG to reflect the higher value of time for Metrorail and commuter rail riders.
- On-board transit runtime factors (***RUNFACTOR***) by mode were kept the same as those in the base model, except for light rail (Mode 5), whose runtime factor was updated to be consistent with Bus Rapid Transit (Mode 10).
- Boarding penalties (***BRDPEN***) by mode were kept unchanged, except that the boarding penalty for Metrorail (Mode 4) was changed from 4 minutes to 5 minutes for off-peak periods. In the Gen3, Phase 1, Model, the boarding penalty for Metrorail was the only difference between the peak-period factor files (AM_TRN.FAC|PM_TRN.FAC) and off-peak factor files (MD_TRN.FAC|NT_TRN.FAC). This change made PT factor files identical across all time periods.
- Different initial and transfer wait curve definitions (***IWAITCURVE, XWAITCURVE***) specified in the PT factor files and system file (TSYSD.PTS) were tested per suggestion from RSG but were not adopted due to problematic pathtracing results. Instead, the original wait curve

[54] See, Feng Xie to Files, "Implementing a Workaround to Address the Hyperpath Issue in the Public Transport (PT) Multipathing Implementation for the Gen3 Travel Model", MWCOG/TPB Memorandum, August 16, 2022.
[55] The PT Arlington Model includes multiple user classes, and the route enumeration/evaluation parameters slightly differ by user class. The Gen3 Model, which includes a single user class in PT, adopts the parameters for the most representative user class (i.e., User Class 1) in the PT Arlington Model.

definitions from the TPB's developmental Gen2/Ver. 2.5 Model were used. The testing resulted in a different time stamp for the system file, but its content did not change relative to the base model.

- PT fare specifications (***FARESYSTEM, OPERATOR***), which were proposed by COG/TPB staff in a developmental Gen2/Ver. 2.3 Model[56] and were later incorporated into the Gen3 Model, are kept unchanged.
- Specifications for the ***MUSTUSEMODE*** keyword[57] by transit submode were kept unchanged during the implementation of PT multipathing. However, they were modified later as part of the partial fix to the hyperpath issue, which will be discussed in the next section.

6.3.3 Implementing a partial fix to the hyperpath issue

In the August 16 memo, COG/TPB staff proposed a workaround to address the hyperpath issue for both commuter rail and "Bus + Metrorail". As part of the workaround implementation, COG/TPB staff created two additional sets of PT factor files (AM|MD|PM|NT_TRN_CR.FAC and AM|MD|PM|NT_TRN_BM.FAC) for commuter rail and "Bus + Metrorail", respectively.

The fix for commuter rail was straightforward. COG/TPB staff simply changed the specifications for ***MUSTUSEMODE*** to "***MUSTUSEMODE***=4" in the commuter rail factor files, which enforces the use of at least one commuter rail (Mode 4) link on every route enumerated or evaluated for commuter rail and thus eliminates the hyperpath issue for the commuter rail sub-mode.

However, the methodology proposed to address the hyperpath issue for "Bus + Metrorail" was much more complicated because there is currently no perfect solution in the PT algorithm. In a nutshell, the proposed methodology involved the following steps:

I. The specifications for ***MUSTUSEMODE*** were changed to "***MUSTUSEMODE=3,5***" in the "Bus + Metrorail" factor files, which enforces the use of at least one Metrorail (Mode 3) or light rail (Mode 5) link[58] in the "Bus + Metrorail" path building.
II. An additional transit skim matrix (herein referred to as "MW[x]") was computed, which indicates the probability of taking the routes that use both bus and Metrorail/light rail links (herein referred to as "B/M" routes), as opposed to the routes that use Metrorail/light rail modes only (herein referred to as "M/O" routes).
III. Transit skims for "Bus + Metrorail" were re-factored based on the value of MW[x]:
    a. If MW[x] equals 0, there are no B/M routes being built, and thus final transit skims for the "Bus + Metrorail" sub-mode are zeroed out.
    b. If MW[x] equals 1, all the enumerated or evaluated routes are B/M routes, and thus the final transit skims are kept unchanged.
    c. If MW[x] is between 0 and 1, the routes built for the "Bus + Metrorail" submode include both M/O routes and B/M routes. A convoluted refactoring method is used to derive the transit skims just for B/M routes.

After reviewing the August 16 memo, Joel (RSG) recommended a partial solution that would include Step I to Step III.b of the proposed workaround but skip Step III.c. COG/TPB staff agreed with the RSG recommendation on the following grounds:

---

[56] See, Feng Xie to Mark Moran, "Proposed Cube Public Transport (PT) Fare Systems in the TPB Ver. 2.3 Travel Model", MWCOG/TPB Memorandum, September 2, 2020.

[57] ***MUSTUSEMODE*** specifies required transit modes in a route for enumeration or evaluation,

[58] Note that light rail is treated as Metrorail in transit path building in the Gen3 Model.

- The refactoring method in Step III.c was developed based on the assumption that the M/O routes built for the "Bus + Metrorail" sub-mode are identical to the M/O routes built for the "Metrorail Only" sub-mode. While both Bentley and COG/TPB staff suspected that the assumption is valid for the majority of cases,[59] there is a small chance that the two sets of routes are slightly different, as the minimum-cost paths are developed on different transit networks (one includes bus links, but the other not). A detailed discussion of this discrepancy and its implications on the resulting transit skims can be found in the August 16 memo.
- COG/TPB staff found that the Origin/Destination (O/D) pairs falling into the categories of Step III.a and Step III.b accounted for about 97% of all O/Ds. On the other hand, the O/D pairs that needed to be addressed by the imperfect refactoring method in Step III.c accounted for only about 3% of all cases. Joel suggested that, by skipping Step III.c, the effects of using the incorrect skims for some 3% of the OD pairs on the subsequent mode choice and transit assignment models would be very small.
- Furthermore, Joel pointed out that even if the refactoring method in Step III.c is perfectly correct, it would fix only the transit skims for the "Bus + Metrorail" submode, but not the route set and the associated "trip leakage" issue in transit assignment. Thus, he was concerned that "calculating corrected skims leads to an inconsistency between the path attributes fed to mode choice versus the paths to which the resulting trips are assigned, which can make debugging the results difficult".

For the reasons stated above, COG and RSG decided to move forward with a PT Multipathing implementation with the partial fix to the hyperpath issue. In addition to the PT factor files that were specifically created for commuter rail and "Bus + Metrorail" submodes, the implementation of the partial fix also involved updating five Cube scripts in the Gen3 Model. Specifically,

- The "Transit_Skims_PT_AB.s" script and "Transit_Skims_PT_MR.s" script include minor, aesthetic changes that don't affect modeling results.
- The "Transit_Skims_PT_CR.s" script was modified to address the hyperpath issue found in commuter rail path building.
- The "Transit_Skims_PT_BM.s" script was modified to partially address the hyperpath issue found in "Bus + Metrorail" path building.
- The "PT_asgn_CR.s" script was modified to address an issue arising from the above-mentioned change to the "***MUSTUSEMODE***" specification for commuter rail. Due to the change, PT can no longer build any path for a small number of external commuter rail trips (fewer than 800) and thus reports a fatal error during transit assignment. After consulting with RSG and BMG, the sub-contractor who developed the external commuter rail trip table based on the transit on-board surveys (TOBS), revisiting the TOBS data and recoding the transit mode for these trips would take a significant amount of time and resources, and COG/TPB staff provided a quick fix to this issue by removing the few unassigned external commuter rail trips from transit assignment.

COG/TPB staff transmitted the implementation files to RSG on September 13 through Box.[60] RSG staff subsequently incorporated them into the Gen3, Phase 2, Model code and used them for the Gen3, Phase 2, Model development work.

[59] Based on the pathtrace summaries for a limited number of origin/destination (O/D) pairs, the two sets of M/O routes were indeed identical for the majority of O/D pairs, but Bentley and COG staff could not find a way to quantify the percentage of all 3722 by 3722 O/D pairs where this assumption held.

[60] Box Link: https://app.box.com/s/ntxh6l5j87ly889wvxuvxztrbk8p3xbs

6.4 PopulationSim inputs and process

PopulationSim is a Population Synthesizer that is designed to develop a representative population for the region. Setting up and running the Population Synthesizer is discussed in the COG Population Synthesizer Final Report[61]. This section includes only a discussion of the outputs, data preparation and application, scenario applications, controls and settings, and validation notes. Validation statistics of the population synthesizer are included in the final report and in the COG Validation Summary[62]. It should be noted that the synthetic population outputs and validation results discussed in this section were generated based on the Round 9.1a Cooperative Land Use Forecasts when the COG Population Synthesizer software was developed. The software has since been updated to work with more recent rounds of Cooperative Land Use Forecasts (e.g., Round 10.0). Recently, PopulationSim has also been upgraded to Version 0.10.0 in the COG Population Synthesizer.

6.4.1 Population Synthesis outputs

As the scripts run, data will be downloaded, processed, and stored in additional folders within the root directory.

PopulationSim input data (seed sample data, marginal controls, and raw downloads from the US Census Bureau) are stored in the data/ directory. The following shows the appearance of the directory after a base year run (2018). The size of this directory following the downloading and processing of the data is approximately **2 GB**.

```
data/
├── Census/               Raw data from US Census Bureau
├── PUMS/                 Raw data from ACS PUMS
├── TAZ/                  TAZ shapefile from MWCOG
├── land_use/             Round 9.1a Cooperative Forecast DBF files
├── xwalk/                Geographic crosswalks needed for pre-processing
├── control_taz_2018.csv  Marginal controls at TAZ level
├── geo_cross_walk.csv    TAZ to PUMA crosswalk required by PopulationSim
├── seed_hh.csv           Residential seed sample household data
├── seed_hh_gq.csv        Group quarter seed sample household data
├── seed_per.csv          Residential seed sample person data
└── seed_per_gq.csv       Group quarter seed sample person data
```

PopulationSim output data appears in the output/ directory. This includes the raw output from PopulationSim, zip archives of the output, and the combined residential/group quarters synthetic population data. The following shows the appearance of the directory after a base year run (2018).

```
output/
├── output_gq_2018/                                group quarters output +
validation
├── output_res_2018/                               residential output + validation
├── combined_synthetic_hh_2018.csv            synthetic data
├── combined_synthetic_per_2018.csv            synthetic data
```

[61] RSG. MWCOG Population Synthesizer. Final Report. Metropolitan Washington Council of Governments, National Capital Region Transportation Planning Board, August 4, 2021.
[62] RSG. MWGOC Gen3 Model Calibration and Validation Report. November 20, 2023.

```
└── popsim_output_2018_20201001_1916.zip  archive of PopulationSim output
```

After one run of PopulationSim, the size of this directory is approximately 2 GB. To use the synthetic population files generated by PopulationSim (i.e., combined_synthetic_hh|per_2018.csv in the above example) in a Gen3 Model run, the user needs to rename them as "combined_synthetic_hh|per.csv" and copy the renamed files to the {scenario_dir}\inputs\popsyn folder.

6.4.2 Data preparation and application

There are two main data processing steps in the COG Population Synthesizer: 1) the preparation of inputs for the PopulationSim software ("pre-processing") and 2) the processing of the PopulationSim outputs ("post-processing"), such as combining Group Quarters and residential population outputs and preparation of validation charts and summaries. The following sub-sections present the details of data preparation and application of the COG Population Synthesizer.

6.4.2.1 Input data preparation

The main data inputs to PopulationSim are:

- A disaggregate population sample (seed sample)
- Marginal control distributions (control variables)[63]

PopulationSim can work with both household-level and person-level controls. The controls can also be specified at multiple geographic levels. The geographic resolution of the seed sample is referred to as the "seed" geography. The marginal controls can be specified at the level of seed geography or any number of sub-seed geographies. The marginal controls can also be specified at a "meta" geographic level which is above the seed geography. For the COG implementation, the Public Use Microdata Sample (PUMS) dataset was used as the seed sample, which is available at the Public Use Microdata Area (PUMA) level. The marginal controls are generated at the TAZ level from Census data (ACS and Decennial Census) and multi-year land use forecasts. As a result, the hierarchical geographic structure for the COG implementation is defined as follows:

- Region
- PUMA
- TAZ

A geographic crosswalk file defines the hierarchical structure of the various geographies. The crosswalk between TAZ and PUMA is created by one of the pre-processing scripts, 04_create_crosswalk.py, and formatted for PopulationSim in 06_create_controls.py.

6.4.2.2 Seed sample

The seed sample data are generated by the script 05_create_seed_sample.py. The main requirement for the seed sample is that it should be representative of the modeling region. The seed sample must contain the appropriate fields needed to specify various marginal controls. Also, it must contain variables that are needed for the ABM but that are not specified as controls. The ACS PUMS data satisfies these requirements and was used as the seed sample. The 2014-2018 5-year ACS

[63] In the context of List Balancing or Iterative Proportional Fitting, marginal controls refer to the row and column totals of the seed table.

PUMS data is the most recent vintage of the PUMS sample and the closest to the selected base year (2018). PUMS data for District of Columbia, Maryland, Virginia, and West Virginia were used. The pre-processing scripts filter the PUMS sample to include only those records belonging to the PUMAs overlapping the modeling region.

6.4.2.3 Marginal controls

The control data are generated by the script 06_create_controls.py. The residential population marginal controls are produced from the 2018 ACS 5-year dataset downloaded at the tract level, based on specifications in configs/census_variables_needed.csv (see also 03_get_census.py). The tract-level data are aggregated to TAZ level based on the area fraction of each Census tract that overlaps each TAZ. The group quarters data are not available at the sub-state level in the ACS 2018 sample, so the 2010 Summary File 1 Census data were used at the block level. The block-level data were aggregated to TAZ by summing blocks within each TAZ. Finally, the Census/ACS counts were scaled to Round 9.1a Cooperative Forecasts for each simulation year and each TAZ by calculating the ratio between the forecast data total and the Census/ACS data total within TAZ and multiplying each Census/ACS count by this adjustment factor.

Table 50 presents the list of control variables specified in the COG Population Synthesizer.

Table 50. Data Sources for Seed Sample and Marginal Controls

VARIABLE	CATEGORIES	PUMS FIELD	CONTROL SOURCE
Total HH		WGTP	Round 9.1a Forecasts
HH Size	1, 2, 3, 4+	NP	2018 ACS 5-year. Census Tract [Table S2501]
HH Income	0-$25K, $25K-$50K, $50K-$100K, $100k-$150K, $150k-$200K, $200K+	HINCP	2018 ACS 5-year. Census Tract [Table B19001]
Number of Workers	0, 1, 2, 3+	ESR	2018 ACS 5-year. Census Tract [Table B08202]
Presence of Children	0, 1	HUPAC	2018 ACS 5-year. Census Tract [Table S1101]
Person Age	0-4, 5-19, 20-34, 35-64, 65+	AGEP	2018 ACS 5-year. Census Tract [Table S0101]
Person Race	White, Hispanic, Black, Asian, Other	HISP, RAC1P	2018 ACS 5-year. Census Tract [Table DP05]
Total non-institutional GQ units		TYPE	Round 9.1a Forecasts, adjusted to exclude institutional GQ units
GQ Type	University, Military, Other Non-Institutional (group homes, missions, shelters, etc.)	SCHG, MIL, TYPE	2010 Census SF1 [Table P042]

6.4.2.4 HH size control adjustments

The household size controls were further adjusted to resolve the inconsistencies between the TAZ-level population inferred from the Census/ACS household size distributions and the TAZ-level household and population estimates from the Round 9.1a Cooperative Forecasts. A lookup table between average household size and household size distribution (1,2,3, 4+ categories) was computed from the Tract level Census/ACS data. The household size controls were recomputed for TAZs with problematic household size controls. To identify problematic TAZs, minimum and average implied populations were computed for each TAZ using the existing household size controls. The minimum implied population is computed by counting only 4 persons for the 4-plus household size category. For the average implied population, the average number of persons in a 4-plus person household is used. Problematic TAZs are the ones for which either of the following conditions is true.

1. Round 9.1a Cooperative Forecasts population < minimum implied population
2. Round 9.1a Cooperative Forecast population > 1.5 * average implied population

Using the above rules, about 600 TAZs were tagged. The household size controls were recomputed for these TAZs using the household size distribution lookup table. This fixed the inconsistencies for more than 90% of the problematic TAZs. Table 51 shows some example TAZs that failed one of the above checks and how their household size controls were adjusted.

Table 51: Example Household Size Control Adjustments

	ROUND 9.1A CF			POPULATIONSIM CONTROLS						DATA CHECKS		
TAZ	Total HH	HH Persons	Avg HH Size	hh_1	hh_2	hh_3	hh_4p	Min implied persons	Avg implied persons	Check 1	Check 2	Version
159	1,211	4,098	3	502	420	87	202	2,411	2,546	✓	✗	Initial
159	1,211	4,098	3	171	335	244	461	3,417	3,726	✓	✓	Adjusted
411	660	1,396	2	190	155	140	175	1,620	1,737	✗	✓	Initial
411	660	1,396	2	283	227	80	70	1,257	1,304	✓	✓	Adjusted
1500	1,578	2,402	2	855	539	139	45	2,530	2,560	✗	✓	Initial
1500	1,578	2,402	2	1,018	459	69	32	2,271	2,292	✓	✓	Adjusted
3669	2,242	6,065	3	371	761	252	858	6,081	6,656	✗	✓	Initial
3669	2,242	6,065	3	586	745	383	528	5,337	5,691	✓	✓	Adjusted

6.4.3 Scenario applications

The marginal controls for PopulationSim need to be updated for modeling scenarios involving a change in demographics. To prepare the synthetic population for such scenarios, the user must create appropriate marginal control data. For example, when modeling an aging population scenario, the person-age controls must be adjusted to represent an aging population. For such a scenario, however, just updating the age controls will not be sufficient. The aging population will likely impact other distributions such as household income and auto ownership. The user must evaluate such effects and update all marginal distributions to represent demographic distributions under an aging population scenario. The modified controls can be specified at a different geographic level compared to the base year. This depends on the availability of the control data at that geographic level and the accuracy of the data. Typically, the seed data remains the same.

6.4.4 Controls and settings

This section presents the final settings for the COG Population Synthesizer. Different settings and configurations were tried during the initial testing of PopulationSim, such as varying the maximum expansion factor, combining controls, and altering importance factors on controls. The settings and configuration that resulted in the best overall validation performance were retained as the final version. The final version of the COG PopulationSim uses a maximum expansion factor of 30. This is the default maximum expansion factor used by other agencies such as the Metropolitan Transportation Council (MTC), Oregon Department of Transportation, Portland Metro, Fresno Council of Governments, and Metropolitan Council (Minneapolis) in their PopulationSim implementation. Table 52 (residential) and Table 53 (group quarters) present the final set of controls and importance factors.

Table 52: COG PopulationSim Marginal controls, Residential

Target	Geography	Seed Table	Importance
hh_total	TAZ	households	1,000,000,000
hh_size_1	TAZ	households	5000
hh_size_2	TAZ	households	5000
hh_size_3	TAZ	households	5000
hh_size_4_plus	TAZ	households	5000
hh_inc_0_25	TAZ	households	1000
hh_inc_25_50	TAZ	households	1000
hh_inc_50_100	TAZ	households	1000
hh_inc_100_150	TAZ	households	1000
hh_inc_150_200	TAZ	households	1000
hh_inc_200_plus	TAZ	households	1000
hh_worker_0	TAZ	households	5000
hh_worker_1	TAZ	households	5000
hh_worker_2	TAZ	households	5000
hh_worker_3plus	TAZ	households	5000
hh_w_kid	TAZ	households	1000
hh_wo_kid	TAZ	households	1000
per_age_0_4	TAZ	persons	1000
per_age_5_19	TAZ	persons	1000
per_age_20_34	TAZ	persons	1000
per_age_35_64	TAZ	persons	1000
per_age_65plus	TAZ	persons	1000
per_race_anyhispanic	TAZ	persons	5000
per_race_white	TAZ	persons	5000
per_race_black	TAZ	persons	5000
per_race_asian	TAZ	persons	5000
per_race_other	TAZ	persons	5000

Table 53: COG PopulationSim Marginal Controls, Group Quarters

Target	Geography	Seed Table	Importance
gq_noninst	TAZ	households	1,000,000,000
gq_univ	TAZ	persons	1000
gq_mil	TAZ	persons	1000
gq_other	TAZ	persons	1000

6.4.5 Validation

One of the most critical steps in population synthesis is validating the final synthetic population. Validation can give clues about inconsistencies among controls, data processing errors, or misspecification of any settings. This section describes the validation procedures and then presents the validation results from the final base-year (2018) run and future-year (2045) run.

6.4.5.1 Validation procedures

PopulationSim reports the difference between the synthesized totals and the control totals for all the controls at each geographic level. The Python validation script generates advanced summary statistics and validation plots. These are described briefly below.

6.4.5.2 Validation summary statistics

The validation script reports the following information for each control: the total number of records (household/person) desired by the control, the total number of records synthesized, the difference between the synthesized total and the control total, and the percentage difference. Statistics that inform us of convergence at a more disaggregate level are also computed – please note that these statistics are computed for the geography at which the controls are specified i.e. TAZ or Region as the case might be. The following three statistics are computed as a part of this exercise:

1. the average percentage difference between the control totals and the synthesized totals,

$$Average\ Percentage\ Difference\ (APD) = \frac{\sum_{i=1}^{N}(PercDiff_i)}{N}$$

   Where, $PercDiff_i$ is the percentage difference between the control totals and the synthesized totals at each geography and N is the total number of geographies.

2. the standard deviation (STDEV) of the percentage difference – this measure informs us of how much dispersion from the average exists,

$$STDEV = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(PercDiff_i - APD)^2}$$

3. the percentage root mean square error (PRMSE) - an indicator of the proximity of synthesized and control totals.

$$PRMSE = \left(\frac{\sqrt{\frac{\sum_{i=1}^{N}(Diff_i)^2}{N}}}{\sum_{i=1}^{N} Control_i}\right) * N * 100$$

   Where, Diffi is the difference between the control totals and the synthesized totals at each geography and N is the total number of geographies.

4. The number of geographies for which the control is non-zero (N) is also reported.

Traditionally, the performance of population synthesis is assessed at the regional level. These disaggregate statistics provide a tool to investigate data inconsistencies and misspecification errors. In the absence of any data inconsistencies and errors, the average percentage differences are expected to be close to zero across all controls. However, the observed data usually has some inconsistencies. Therefore, an average percentage difference in the range of -5% to +5% is considered acceptable. Please note that the control on the total number of households, in any case, should match perfectly. The sum of average percentage differences across all control categories should also be close to zero. For example, the average percentage difference on the 1-person household control may not be close to zero but the sum across all household size categories should be close to zero. A systematic pattern across all control categories may indicate issues in the control data or a specification error. The STDEV and PRMSE should ideally be in the -20% to +20% range. However, these statistics are very sensitive to the actual control values. For example, a control on a minority segment of the population may have a very low control value for each geography. A difference of 5 on a control value of 100 results in a percentage difference of 5% while the same difference on a control value of 20 results in a percentage difference of 20%. A higher percentage difference is generally acceptable for zones with a low control value, but the high percentage differences increase the STDEV and PRMSE. Therefore, a higher STDEV or PRMSE may be acceptable for controls with low control values that are hard to match exactly.

6.4.5.3 Validation chart

The validation chart is a visualization of the disaggregate summary statistics – mean percentage difference, STDEV, and PRMSE of percentage differences. A form of dot and whisker plot is generated for each control where the dots are the mean percentage differences and horizontal bars are twice the STDEV or PRMSE centered around zero.

6.4.5.4 Frequency distribution plots[64]

These are simply frequency distribution plots of differences between control and synthesized values across the geography at which the controls were specified.

6.4.5.5 Expansion factor distribution[65]

While a synthetic population may match the controls well, it is important to know how uniform the household weights are, and how different they are from the initial weights. The closer the final weights are to the initial PUMS weight, the higher is the chance of matching the distribution of uncontrolled variables. An expansion factor is computed for each record in the PUMS data as total final weight/initial weight. A distribution plot of these expansion factors is created for each PUMA. A good synthetic population would have most of these expansion factors as close to one as possible.

6.5 ActivitySim resident travel model

ActivitySim is an activity-based travel model that simulates transportation decisions throughout a person's day. This is done by reading a representative population of households and persons in the model region and modeling decisions that are made throughout the day. This model begins with long-term choices - decisions that are not made on a daily basis, such as work and school locations.

---

[64] Frequency distribution plot example: https://activitysim.github.io/populationsim/validation.html#frequency-distribution-plots

[65] Expansion factor distribution example: https://activitysim.github.io/populationsim/validation.html#expansion-factor-distributions

This includes whether a person works from home full-time or out-of-home, and then if they work out-of-home, the frequency that they telecommute. The second set of choices are daily activity pattern and mandatory tour frequency pattern, which dictate the makeup of tours in a day. The third set of choices are tour-level choices, which determine tour mode, tour scheduling, and sometimes the tour destination (mandatory tours - work and school tours - have their primary tour destination set by the long-term work and school location choice models). Finally, the fourth set of choices are trip-level choices, which determine the individual trip choices for an entire tour.

6.5.1 Long-term models

These steps model the long-term choices that are not easily changed by households or individuals and choices that are made largely based on the outcomes of those long-term choices. This group of models includes the following models, in order:

1. School Location
2. Work From Home
3. Workplace Location
4. Transit Fare Subsidy
5. AV Ownership
6. Auto Ownership
7. Vehicle Type Choice
8. Free Parking
9. Telecommute Frequency

6.5.1.1 School Location

The usual school location choice models assign a usual school location for the primary mandatory activity of each child and university student in the synthetic population. The models are composed of a set of accessibility-based parameters (including one-way distance between home and primary destination and the tour mode choice logsum - the expected maximum utility in the mode choice model which is given by the logarithm of the sum of exponentials in the denominator of the logit formula) and size terms, which describe the quantity of grade-school or university opportunities in each possible destination.

The school location model is made up of four steps:

1. Sampling - selects a sample of alternative school locations for the next model step. This selects 30 locations from the full set of model zones using a simple utility.

2. Logsums - starts with the table created above and calculates and adds the mode choice logsum expression for each alternative school location.

3. Simulate - starts with the table created above and chooses a final school location, this time with the mode choice logsum included.

4. Simulation constraint - compare modeled zonal destinations to target zonal size terms and re-simulate choices until convergence.

These steps are repeated until shadow pricing convergence criteria are satisfied or a max number of iterations is reached.

6.5.1.2 Work From Home

The work from home model determines if a worker will fully work from home and not have a usual out-of-home workplace. This situation is where a company may not have a local office, or the employee is a sole proprietor with no other office. This is different from telecommuting where a worker that telecommutes HAS a usual out-of-home workplace.

The work from home model is a binary logit model. It is run before the workplace location model and workers that are determined to work from home are filtered out and not simulated in the workplace location model, free parking model, or telecommute frequency model.

6.5.1.3 Workplace Location

The usual work location choice models assign a usual work location for the primary mandatory activity of each worker that works out-of-home in the synthetic population. The models are composed of a set of accessibility-based parameters (including one-way distance between home and primary destination and the tour mode choice logsum - the expected maximum utility in the mode choice model which is given by the logarithm of the sum of exponentials in the denominator of the logit formula) and size terms, which describe the quantity of work opportunities in each possible destination.

The work location model is made up of four steps:

1. Sample - selects a sample of alternative work locations for the next model step. This selects 30 locations from the full set of model zones using a simple utility.
2. Logsums - starts with the table created above and calculates and adds the mode choice logsum expression for each alternative work location.
3. Simulate - starts with the table created above and chooses a final work location, this time with the mode choice logsum included.
4. Simulation constraint - compare modeled zonal destinations to target zonal size terms and re-simulate choices until convergence.

These steps are repeated until shadow pricing convergence criteria are satisfied or a max number of iterations is reached.

6.5.1.4 Transit Fare Subsidy

The transit fare subsidy model estimates the amount of transit subsidy, if any, for workers in the model. This model uses a continuous distribution (listed in constants.yaml as the transit_subsidy array) and a random number generator to determine the subsidy amount.

6.5.1.5 Automated Vehicle Ownership

The Autonomous Vehicle (AV) ownership model predicts whether a household will own an AV. This is a binary logit model. For calibration purposes, this model uses a constant set to -999 to disable owning an AV. For application purposes requiring the AV model, that constant will need to be

disabled or set to 0 to allow the model to determine if a household will own an AV. This model is adapted from previous DaySim work[66,67].

6.5.1.6 Auto Ownership

The auto ownership model selects a number of autos for each household in the simulation. The primary model components are household demographics, zonal density, and accessibility. The model is a multinomial logit model with choices of 0 autos, 1 auto, 2 autos, 3 autos, or 4+ autos. In the case that a household owns an AV, there is a specification that disables the 0 auto and 4+ auto alternatives.

6.5.1.7 Vehicle Type Choice

The vehicle type choice model selects a vehicle type for each household vehicle. A vehicle type is a combination of the vehicle's body type, age, and fuel type. For example, a 13-year-old gas powered van would have a vehicle type of van_13_gas. This model is a multinomial logit model with simultaneous choice of body type, age, and fuel type. More information on this model can be found on the ActivitySim Documentation Website[68].

6.5.1.8 Vehicle Allocation

The vehicle allocation model selects which vehicle would be used for a tour of given occupancy. The alternatives for the vehicle allocation model consist of the vehicles owned by the household and an additional non-household vehicle option. (Zero-auto households would be assigned the non-household vehicle option since there are no owned vehicles in the household). A vehicle is selected for each occupancy level set by the user such that different tour modes that have different occupancies could see different operating characteristics. The output of the vehicle allocation model is appended to the tour table with column names vehicle_occup_{occupancy} and the values are the vehicle type selected.

6.5.1.9 Free Parking

The Free Parking Eligibility model predicts the availability of free parking at a person's workplace. It is applied for people who work in zones that have parking charges, which are generally located in the Central Business Districts. The purpose of the model is to adequately reflect the cost of driving to work in subsequent models, particularly in mode choice. This is a binary logit model.

6.5.1.10 Telecommute Frequency

Telecommuting is defined as workers who work from home instead of going to work. This model is only applied to workers with a regular workplace outside of home.
For all workers that work out of the home, the telecommute frequency choice model predicts the level of telecommuting. This is a multinomial logit model with alternatives of the frequency of telecommuting in days per week (0 days, 1 day, 2 to 3 days, 4+ days).

---

[66] Bradley, Mark. AVs and TNCs in Daysim. Presentation to SACOG. 1/17/2009.
https://www.sacog.org/sites/main/files/file-attachments/avs_and_tnc_in_daysim-sacsim-rsg_0.pdf?1548293104

[67] Ou, Yanmei and Griesenbeck, Bruce. Estimating the Potential Impacts of AVs and TNCs using ActivityBased Travel Demand Model in MTP/SCS Scenario Development. Presentation at 2018 Innovations in Travel Modeling Conference, Atlanta, GA. 2018.
https://onlinepubs.trb.org/onlinepubs/Conferences/2018/ITM/YOu.pdf

[68] https://activitysim.github.io/activitysim/v1.2.0/models.html#vehicle-type-choice

6.5.2 Daily models

There are two daily models in ActivitySim that bridge the gap between long-term models and tour models. These are the Coordinated Daily Activity Pattern and the mandatory tour frequency model, which are run sequentially.

6.5.2.1 Coordinated Daily Activity Pattern

The Coordinated Daily Activity Pattern (CDAP) model predicts the choice of daily activity pattern (DAP) for each member in the household, simultaneously. The DAP is categorized into three types as follows:

Mandatory (M): the person engages in travel to at least one out-of-home mandatory activity - work, university, or school on the simulation day. The mandatory pattern may also include non-mandatory activities such as separate home-based tours or intermediate stops on mandatory tours.

Non-mandatory (N): the person engages in only maintenance and discretionary tours, which, by definition, do not contain mandatory activities.

Home (H): the person does not travel outside the home on the simulation day.

The CDAP model is a sequence of vectorized table operations:

1. Create a person level table and rank each person in the household for inclusion in the CDAP model. Priority is given to full-time workers (up to two), then to part-time workers (up to two workers of any type), then to children (youngest to oldest, up to three). Additional members up to five are randomly included for the CDAP calculation.
2. Solve individual M/N/H utilities for each person
3. Take as input an interaction coefficients table and then programmatically produce and write out the expression files for household size 1, 2, 3, 4, and 5 models independent of one another
4. Select households of size 1, join all required person-attributes, and then read and solve the automatically generated expressions
5. Repeat for household size 2, 3, 4, and 5. Each model is independent of one another.

6.5.2.2 Mandatory Tour Frequency

The individual mandatory tour frequency model predicts the number of work and school tours taken by each person with a mandatory DAP. The primary drivers of mandatory tour frequency are demographics, accessibility-based parameters such as driving time to work and household automobile ownership. It also creates mandatory tours in the data pipeline. This is a multinomial logit model with alternatives of 1 work, 2 work, 1 school, 2 school, and work+school. Not all alternatives are allowed for all person types, and that is set by using constants in the model specification to disable irrelevant alternatives (e.g. school pre-driving age children have a constant of -999 for the 1 work, 2 work, and work+school alternatives since it is very uncommon and/or against laws for children of this age to work).

6.5.3 Tour models

There are sixteen total tour models. These models determine when, how, and sometimes why a person leaves their house.

The first tour model is mandatory tour scheduling, which determines when a person travels to and returns from work or school, if applicable. The next group of five models are joint tour models, and the following three are non-mandatory tour models. The next two models allocate vehicles onto tours and simulate tour mode choices. Following that is a group of four at-work subtour models and finally, the stop frequency model.

6.5.3.1 Mandatory Tour Scheduling

The mandatory tour scheduling model selects a tour departure period and tour duration (and therefore a tour arrival period as well) for each mandatory tour. The primary drivers in the model are accessibility-based parameters such as the mode choice logsum for the departure/arrival hour combination, demographics, and time pattern characteristics such as the time windows available from previously scheduled tours. This model uses Person Time Windows, which are adjacent time periods that are available for travel. Time windows are stored in a timetable table, and each row is a person and each time period.

6.5.3.2 Joint Tour Models

This group of models estimates fully-joint tours, which are tours where multiple people from a household travel together for the entire tour. An example of this would be a family eating out, where all members of the household stay together for the entire tour.

This model includes its own frequency model that determines the number of joint tours, a tour composition model and a tour participation model that determines the household members to include on the tour, a destination choice model, and finally a scheduling model.

6.5.3.2.1 Joint Tour Frequency

The joint tour generation models are divided into three sub-models: the joint tour frequency model, the party composition model, and the person participation model. In the joint tour frequency model, the household chooses the purposes and number (up to two) of its fully joint travel tours on a typical day. It also creates joints tours in the data pipeline.

6.5.3.2.2 Joint Tour Composition

In the joint tour party composition model, the makeup of the travel party (adults, children, or mixed - adults and children) is determined for each joint tour. The party composition determines the general makeup of the party of participants in each joint tour to allow the micro-simulation to faithfully represent the prevalence of adult-only, children-only, and mixed joint travel tours for each purpose while permitting simplicity in the subsequent person participation model.

6.5.3.2.3 Joint Tour Participation

In the joint tour person participation model, each eligible person sequentially makes a choice to participate or not participate in each joint tour. Since the party composition model determines what types of people are eligible to join a given tour, the person participation model can operate in an iterative fashion, with each household member choosing to join or not to join a travel party independent of the decisions of other household members. In the event that the constraints posed

by the result of the party composition model are not met, the person participation model cycles through the household members multiple times until the required types of people have joined the travel party.

6.5.3.2.4 Joint Tour Destination

The joint tour destination choice model operates similarly to the usual work and school location choice model, selecting the primary destination for travel tours. The only procedural difference between the models is that the usual work and school location choice models select the usual location of a mandatory activity whether the activity is undertaken during the travel day, while the joint tour destination choice model selects the location for an activity which has already been generated.

The primary destination of a tour is the location of the activity that is assumed to provide the greatest impetus for engaging in the travel tour. In the household survey, the primary destination was not asked but rather inferred from the pattern of stops in a closed loop in the respondents' travel diaries. The inference was made by weighing multiple criteria including a defined hierarchy of purposes, the duration of activities, and the distance from the tour origin. The model operates in the reverse direction, designating the primary purpose and destination and then adding intermediate stops based on spatial, temporal, and modal characteristics of the inbound and outbound journeys to the primary destination.

The joint tour destination choice model is made up of three model steps:

1. Sample - selects a sample of alternative locations for the next model step. This selects 30 locations from the full set of model zones using a simple utility.
2. Logsums - starts with the table created above and calculates and adds the mode choice logsum expression for each alternative location.
3. Simulate - starts with the table created above and chooses a final location, this time with the mode choice logsum included.

6.5.3.2.5 Joint Tour Scheduling

The joint tour scheduling model selects a tour departure period and tour duration (and therefore a tour arrival period as well) for each joint tour. This model uses Person Time Windows. The primary drivers in the models are accessibility-based parameters such as the auto travel time for the departure/arrival hour combination, demographics, and time pattern characteristics such as the time windows available from previously scheduled tours. The joint tour scheduling model does not use mode choice logsums.

6.5.3.3 Non-Mandatory Models

6.5.3.3.1 Non-Mandatory Tour Frequency

The non-mandatory tour frequency model selects the number of non-mandatory tours made by each person on the simulation day. It also adds non-mandatory tours to the tours in the data pipeline. The individual non-mandatory tour frequency model operates in two stages:

1. A choice is made using a random utility model between combinations of tours containing zero, one, and two or more escort tours, and between zero and one or more tours of each other purpose.
2. Up to two additional tours of each purpose are added according to fixed extension probabilities.

6.5.3.3.2 Non-Mandatory Tour Destination

The non-mandatory tour destination choice model chooses a destination zone for non-mandatory tours. The three-step (sample, logsums, final choice) process used for mandatory tour destination choice is also used for non-mandatory tour destination choice.

6.5.3.3.3 Non-Mandatory Tour Scheduling

The non-mandatory tour scheduling model selects a tour departure period and tour duration (and therefore a tour arrival period as well) for each non-mandatory tour. This model uses Person Time Windows.

6.5.3.4 Tour Mode Choice Simulation

The mandatory, non-mandatory, and joint tour mode choice model assigns to each tour the "primary" mode that is used to get from the origin to the primary destination. The tour-based modeling approach requires a reconsideration of the conventional mode choice structure. Instead of a single mode choice model used in a four-step structure, there are two different levels where the mode choice decision is modeled: (a) the tour mode level (upper-level choice); and (b) the trip mode level (lower-level choice conditional upon the upper-level choice).

The mandatory, non-mandatory, and joint tour mode level represents the decisions that apply to the entire tour, and that will affect the alternatives available for each individual trip or joint trip. These decisions include the choice to use a private car versus using public transit, walking, or biking; whether carpooling will be considered; and whether transit will be accessed by car or by foot. Trip-level decisions correspond to details of the exact mode used for each trip, which may or may not change over the trips in the tour.

The mandatory, non-mandatory, and joint tour mode choice structure is a nested logit model which separates similar modes into different nests to more accurately model the cross-elasticities between the alternatives. Eighteen modes are incorporated into the nesting structure specified in the model settings file. The first level of nesting represents the use of a private car, non-motorized means, or transit. In the second level of nesting, the auto nest is divided into vehicle occupancy categories, the non-motorized nest is divided into walk and bike modes, and transit is divided into walk access and drive access nests. The final level splits the transit nests into the specific line-haul modes.

The primary variables are in-vehicle time, other travel times, cost (the influence of which is derived from the automobile in-vehicle time coefficient and modeled value of time for individual travelers), characteristics of the destination zone, demographics, and the household's level of auto ownership.

6.5.3.5 Vehicle Allocation Model

The vehicle allocation model selects which vehicle would be used for a tour of given occupancy. The alternatives for the vehicle allocation model consist of the vehicles owned by the household and an additional non household vehicle option. (Zero-auto households would be assigned the non-household vehicle option since there are no owned vehicles in the household). A vehicle is selected for each occupancy level set by the model user such that different tour modes that have different occupancies can see different operating characteristics.

The model has three occupancy levels: 1, 2, and 3.5. The auto operating cost for occupancy level 1 is used in the drive-alone mode and drive access transit modes. The auto operating costs for Occupancy levels 2 and 3.5 are used for shared ride 2 and shared ride 3+ modes, respectively. Auto

operating costs are selected in the mode choice pre-processors by selecting the allocated vehicle type data from the vehicles table. If the allocated vehicle type was the non-household vehicle, the auto operating costs uses the previous default value from the constants.yaml file. All trips and at-work subtours use the auto operating cost of the parent tour.

6.5.3.6 At-work Subtour Models

The at-work subtour model is applied to work tours to determine if a worker leaves and returns to work during the day. An example would be a lunch sub-tour, where a worker leaves the office location for lunch and returns to the office prior to returning home.

The models in this group include a frequency model to determine if and how many subtours are made, and a destination choice model, a tour scheduling model, and a mode choice model for the subtours that are made.

6.5.3.6.1 At-work Subtour Frequency

The at-work subtour frequency model selects the number/purpose of at-work subtours made for each work tour. It also creates at-work subtours by adding them to the tours table in the data pipeline. These at-work sub-tours are travel tours taken during the workday with their origin at the work location, rather than from home. Explanatory variables include employment status, income, auto ownership, the frequency of other tours, characteristics of the parent work tour, and characteristics of the workplace zone.

Choosers: work tours
Alternatives:

- none
- 1 eating-out tour
- 1 business tour
- 1 maintenance tour
- 2 business tours
- 1 eating-out tour + 1 business tour

6.5.3.6.2 At-work Subtour Destination

The at-work subtours destination choice model is made up of three model steps:

1. Sample - selects a sample of alternative locations for the next model step. This selects X locations from the full set of model zones using a simple utility.
2. Logsums - starts with the table created above and calculates and adds the mode choice logsum expression for each alternative location.
3. Simulate - starts with the table created above and chooses a final location, this time with the mode choice logsum included.

6.5.3.6.3 At-work Subtour Scheduling

The at-work subtours scheduling model selects a tour departure and duration period (and therefore a start and end period as well) for each at-work subtour. This model uses Person Time Windows. This model is the same as the mandatory tour scheduling model except it operates on the at-work tours and constrains the alternative set to available Person Time Windows. The at-work subtour scheduling model does not use mode choice logsums. The at-work subtour frequency model can choose multiple tours so this model must process all first tours and then second tours since isFirstAtWorkTour is an explanatory variable.

Choosers: at-work tours
Alternatives: alternative departure time and arrival-back-at-origin time pairs WITHIN the work-tour timeframe between the first-arrival-at-work time and last-departure-from-work time (i.e., the duration of work activity) AND the person time window. If no time window is available for the tour, the model will make the first and last time periods within the work tour available, make the choice, and log the number of times this occurs. Dependent tables: skims, person, land use, work tour
Outputs: at-work tour departure time and arrival back at origin time, updated person time windows

6.5.3.6.4 At-work Subtour Mode Choice

The at-work subtour mode choice model assigns a travel mode to each at-work subtour using the Tour Mode Choice model.

6.5.3.7 Stop Frequency

The stop frequency model assigns to each tour the number of intermediate destinations a person will travel to on each leg of the tour from the origin to tour primary destination and back. The model incorporates the ability to make more than one stop in each direction, up to a maximum of 3, for a total of 8 trips per tour (four on each tour leg).

Intermediate stops are not modeled for drive-transit tours because doing so can have unintended consequences because of the difficulty of tracking the location of the vehicle.

The stop frequency model's output is ultimately the trip list that moves the model from the tour level to the trip level.

6.5.4 Trip models

There are five trip models. These models include a trip purpose model that determines the actual purpose of a trip (which may not be the same as the tour), a trip destination model that determines the destination of the trip, a purpose and destination model that is used to clean up trips that are unable to be assigned a destination in the trip destination model, a scheduling model, and finally a mode choice model.

6.5.4.1 Trip Purpose

For trips other than the last trip outbound or inbound, assign a purpose based on an observed frequency distribution. The distribution is segmented by tour purpose, tour direction and person type. Work tours are also segmented by departure or arrival time period.

6.5.4.2 Trip Destination

The trip destination (or stop location) choice model predicts the destinations of trips (or locations of stops) along the tour other than the primary destination. The stop-location model is structured as a multinomial logit model using a zone attraction size variable and route deviation measure as impedance. The alternatives are sampled from the full set of zones, subject to availability of a zonal attraction size term. The sampling mechanism is also based on accessibility between tour origin and primary destination and is subject to certain rules based on tour mode.

All destinations are available for auto tour modes, so long as there is a positive size term for the zone. Intermediate stops on walk tours must be within 3 miles of both the tour origin and primary destination zones. Intermediate stops on bike tours must be within 8 miles of both the tour origin

and primary destination zones. Intermediate stops on walk-transit tours must either be within 3 miles walking distance of both the tour origin and primary destination or have transit access to both the tour origin and primary destination. Additionally, only zones with short or long walk access are available destinations on walk-transit tours.

The intermediate stop location choice model works by cycling through stops on tours. The level-of-service (LOS) variables (including mode choice logsums) are calculated as the additional utility between the last location and the next known location on the tour. For example, the LOS variable for the first stop on the outbound direction of the tour is based on additional impedance between the tour origin and the tour primary destination. The LOS variable for the next outbound stop is based on the additional impedance between the previous stop and the tour primary destination. Stops on return tour legs work similarly, except that the location of the first stop is a function of the additional impedance between the tour primary destination and the tour origin. The next stop location is based on the additional impedance between the first stop on the return leg and the tour origin, and so on.

6.5.4.3 Trip Purpose and Destination

After running trip purpose and trip destination choice models separately, the two models can be run together in an iterative fashion on the remaining failed trips (i.e., trips that cannot be assigned a destination). Each iteration uses new random numbers.

6.5.4.4 Trip Scheduling

For each trip, assign a departure half-hour based on an input lookup table of percents by tour purpose, direction (inbound/outbound), tour half-hour, and trip index.

The tour half-hour is the tour start half-hour for outbound trips and the tour end half-hour for inbound trips. The trip index is the trip sequence on the tour, with up to four trips per half tour.

For outbound trips, the trip depart half-hour must be greater than or equal to the previously selected trip depart half-hour.

For inbound trips, trips are handled in reverse order from the next-to-last trip in the leg back to the first. The tour end half-hour serves as the anchor time point from which to start assigning trip time periods.

Outbound trips on at-work subtours are assigned the tour depart half-hour and inbound trips on at-work subtours are assigned the tour end half-hour.

The assignment of trip depart time is run iteratively up to a max number of iterations since it is possible that the time period selected for an earlier trip in a half-tour makes selection of a later trip time period impossible (or very low probability). Thus, the sampling is re-run until a feasible set of trip time periods is found. If a trip can't be scheduled after the max number of iterations, then the trip is assigned the previous trip's choice (i.e., assumed to happen right after the previous trip) or dropped, as configured by the user. The trip scheduling model does not use mode choice logsums.
Alternatives: Available time periods in the tour window (i.e., tour start and end period). When processing stops on work tours, the available time periods are constrained by the at-work subtour start and end period as well.

6.5.4.5 Trip Mode Choice

The trip mode choice model assigns a travel mode for each trip on a given tour. It operates similarly to the tour mode choice model, but only certain trip modes are available for each tour mode. The correspondence rules are defined according to the following principles:

- Pay trip modes are only available for pay tour modes (for example, drive-alone pay is only available at the trip mode level if drive-alone pay is selected as a tour mode).
- The auto occupancy of the tour mode is determined by the maximum occupancy across all auto trips that make up the tour. Therefore, the auto occupancy for the tour mode is the maximum auto occupancy for any trip on the tour.
- Transit tours can include auto shared-ride trips for particular legs. Therefore, casual carpool', wherein travelers share a ride to work and take transit back to the tour origin, is explicitly allowed in the tour/trip mode choice model structure.
- Walking is allowed for any trip less than 3 miles.
- The availability of transit line-haul sub-modes on transit tours depends on the skimming and tour mode choice hierarchy. Free shared-ride modes are also available in walk-transit tours, albeit with a low probability. Paid shared-ride modes are not allowed on transit tours because no stated preference data is available on the sensitivity of transit riders to automobile value tolls, and no observed data is available to verify the number of people shifting into paid shared-ride trips on transit tours.

The trip mode choice model's explanatory variables include household and person variables, level-of-service between the trip origin and destination according to the time period for the tour leg, urban form variables, and alternative-specific constants segmented by tour mode.

6.6 Truck Model

The definition of medium and heavy trucks in the regional travel model is based strictly on the vehicle characteristics. Medium trucks are those vehicles with 2 axles and 6 tires. Heavy trucks are vehicles with 3 or more axles. However, the definition of commercial vehicles, for the purposes of this modeling effort are based on both objective and subjective considerations. Commercial vehicles are distinguished as vehicles with two axles and four tires, and visually, as vehicles bearing equipment and/or signage that would indicate some relation to a service or business enterprise. [69]

In terms of the mapping to the FHWA 13 vehicle classes, Medium Trucks in our models strictly correspond to Class 5, while Heavy Trucks strictly correspond to Class 6-13. Commercial Vehicles in our models, however, correspond to Class 3 and a small portion of Class 2 vehicles.[70]

The truck trip generation model in the Gen3 Model uses the same truck trip rates as in the Gen2 Model. The origin/destination truck trip generation rates are based on area type and land activity variables as shown in Table 54. The truck trip generation model includes provisions to remove external trucks generated because external truck travel is accounted for exogenously. The truck trip

[69] Ronald Milone to Commercial Vehicle Model Files, "Data Collection for the Commercial Vehicle Model", Memorandum, April 19, 2007.

[70] The definition of vehicle classification can be found on the FHWA website: https://www.fhwa.dot.gov/policyinformation/tmguide/tmg_2013/vehicle-types.cfm.

generation process also includes network checks provisions to ascertain whether or not truck access from each TAZ to the highway network is valid. There are some zonal centroids in the regional network that have a single connection to a parkway where trucks are prohibited. In these types of cases, truck trip generation is suppressed. Finally, the truck model also considers a limited number of special-generator TAZs, or locations where truck traffic generation is known to be more intensive. Global trip generation adjustments are applied to the special generator TAZs - The medium truck generation is factored by 2.70 while heavy trucks are factored by 5.3.

Table 54: Truck Trip Generation Rates as a Function of Truck Type, Area Type, and Land Use Category

		Land Use Category				
Vehicle Type	Area Type	Office	Retail	Industrial	Other	HH
Medium Truck (Single Unit 6+ Tires)	1 (CBD)	0.004	0.088	0.088	0.014	0.070
	2 - 4	0.005	0.125	0.125	0.020	0.100
	5	0.006	0.150	0.150	0.024	0.120
	6	0.006	0.150	0.150	0.024	0.120
Heavy Truck (All Combination Vehicles)	1 (CBD)	0.001	0.027	0.055	0.002	0.011
	2 - 4	0.002	0.039	0.078	0.003	0.015
	5	0.002	0.043	0.086	0.003	0.017
	6	0.002	0.043	0.086	0.003	0.017

Ref: I:\ateam\docum\FY09\Version2.3_modelDoc_2008-07\tgcheck.xls

6.7 Commercial vehicle model

Consistent with the Gen2 Model, the trip generation of zonal commercial vehicle trips is developed with Equation 3.[71]

Equation 3: Commercial Vehicle Productions

$$\boldsymbol{CommVehP_i = (0.056 * INDEMP_i + 0.168 * OFFEMP_i + 0.494 * RETEMP_i + 0.082 * OTHEMP_i + 0.13 * HH_i) * ATFAC_a}$$

Where:
indemp = industrial employment
offemp = office employment
retemp = retail employment
othemp = other employment
HH = households
ATFAC = area type adjustment factor:

Area type	Factor
1	1.05
2	0.90
6	1.15

Note: no factor is applied to area types 3-5.

[71] Allen, Development of a Model for Commercial Vehicle Trips, 46.

6.8 Auxiliary model and input trip tables

These models, which COG/TPB staff generally refer to as exogenous travel, miscellaneous travel, or residual travel models,[72] reflect the auxiliary automobile travel in the region that ActivitySim does not account for, such as

- Airport Passengers
- Visitors
- Visitor Taxi Trips
- Auto-Person External-External (EE) Trips
- Trucks and Commercial Vehicles
- Auto-Person Internal-External (IE), External-Internal (EI) Trips

Figure 8 shows the auxiliary trip models used in the Gen3 model and how they fit together as components of the final traffic assignment.

The auto-person external-internal trips are trips made by a non-resident of the COG region with one end in the region and the other end at an external station. Similarly, the auto-person internal-external trips are trips made by a resident of the COG region with one end in the region and the other end at an external station. In both cases, these are routine or semi-routine trips, not visitor trips.

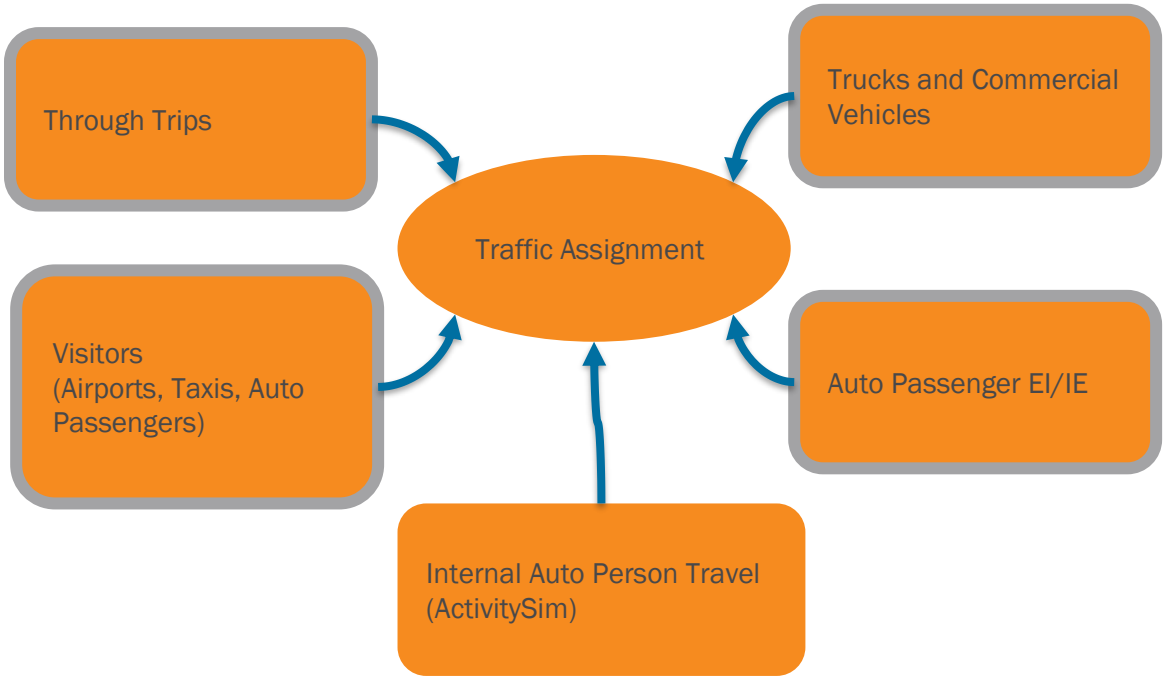


Figure 8: Traffic Assignment Components and Auxiliary Models

[72] See, for example, Ray Ngo, "Exogenous Demand Inputs to the TPB Travel Demand Model: Update for Round 9.2 Cooperative Forecasts," Memorandum to DTP Technical Staff et al., June 21, 2021.

The auxiliary travel model process used in the Gen3, Phase 1, Model is shown in Figure 8. This process used static trip matrices as inputs for the auxiliary traffic sources. The updated process in the Gen3, Phase 2, Model is shown in Figure 9. In the cases of airport passengers, visitors, and external-external auto trips, existing procedures are used to prepare these static exogenous trip tables that are inputs for the regional travel demand model. Truck and commercial vehicles and auto passenger internal-external and external-internal trip tables are also created as part of this process. The supplementary school trip table was removed due to the possibility of double counting with school trips from ActivitySim.

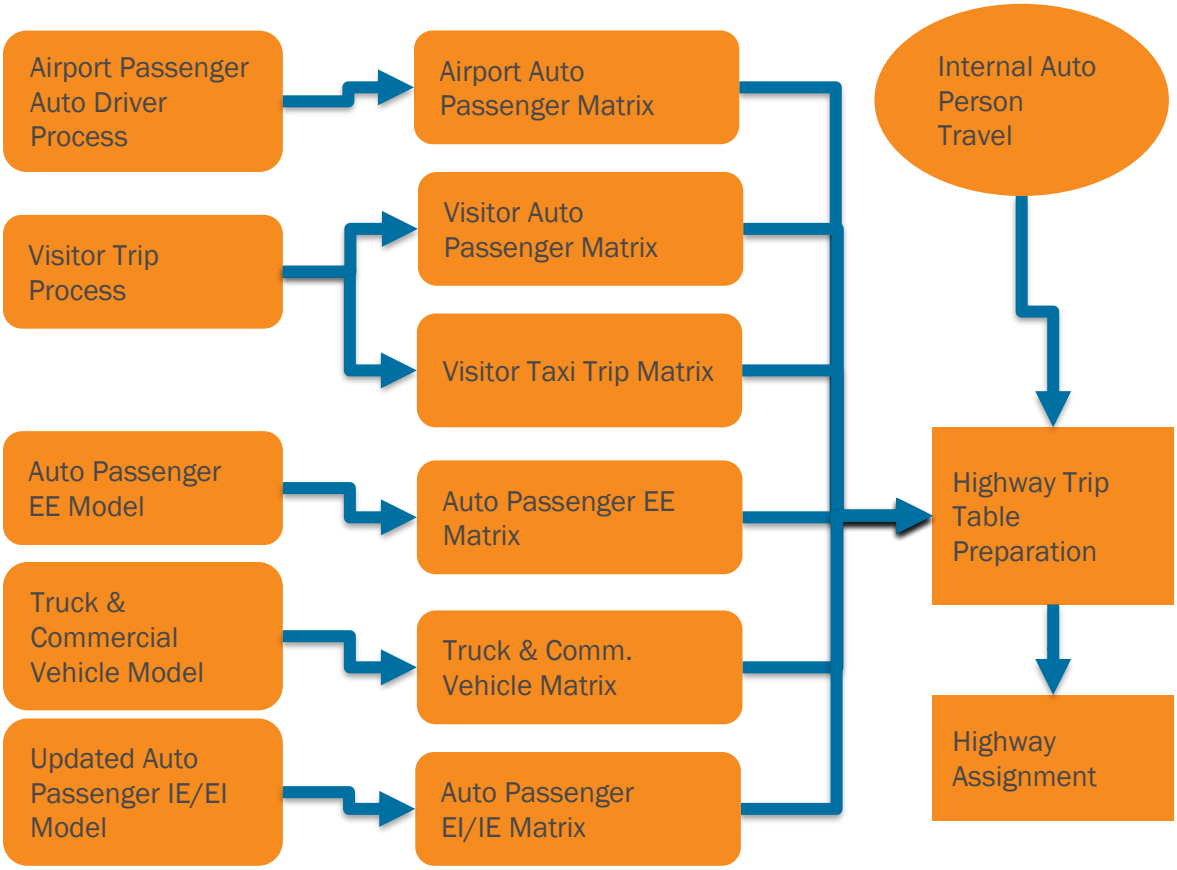


Figure 9: Auxiliary Highway Model Process

6.8.1 Airport passengers, visitors, taxi trips, and auto-person External-External Trips

The same airport passengers, visitors, taxi trips, and auto-person external-external trips that are used as inputs in the Gen2 Model are used as input trip tables for the Gen3 Model as well. These were prepared by COG/TPB staff in a separate process for both the base year and future years. Note that, in the case of taxi trips, the input trip tables represent special-generator markets that are not well represented by the household travel survey data, and do not significantly overlap with the resident ride hailing travel outputs from ActivitySim (which includes taxi and TNC).

6.8.2 External-Internal (EI) and Internal-External (IE) auto trips model

Big Data, (specifically, passively collected origin-destination data from AirSage that was used in the Gen2 Model have been used to develop separate EI and IE models. The process is shown in Figure 10.

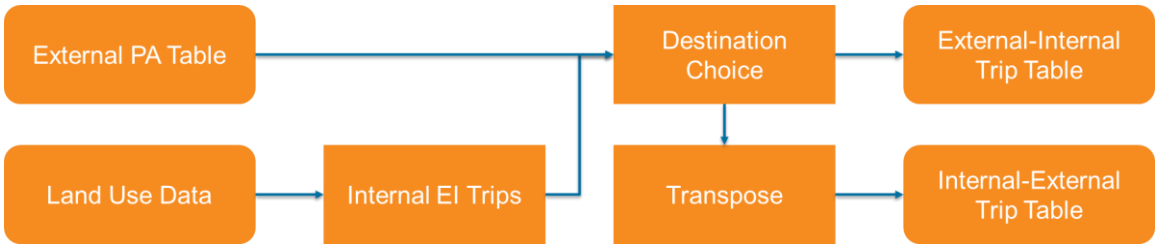


Figure 10: Auto Person External-Internal and Internal-External Trip Flowchart

In this model, internal-external trips are made by residents of the metropolitan Washington region from the modeled area to external stations and external-internal trips are made by non-residents (e.g., out-of-region commuters) who travel into the metropolitan Washington region. Inputs to this model included:

- Forecasts of volumes at external stations and a percentage of those trips that are IE at each external station, by trip purpose
- Land Use data
    - For home-based IE trips, total households by TAZ
    - For home-based EI trips, employment by type by TAZ
    - For NHB trips (both directions), total households and employment by type by TAZ
- Drive-alone midday time, distance, and toll cost skims, which are used to calculated a generalized cost matrix: time + y * ((distance * auto operating cost)+tolls) where y is a term that converts cost to minutes

The inputs to the calibration of the auto passenger internal-external and external-internal models were the passively collected origin-destination data that COG procured from AirSage in 2014. Trip purposes in the AirSage data include Home-based Work (HBW), Home-based Shopping (HBS) and Home-based Other (HBO), and Non-home-based (NHB), and directions are IE and EI, referring to internal-external and external-internal, respectively. In all cases, there is a definite pattern showing higher numbers of external trips in the north and west areas to the TPB's modeling area (i.e., Baltimore and Virginia), which was considered in both the trip generation and attraction functions as well as the destination choice impedance functions. For this short-term approach, the EI model operated with autos only (and may continue as such for the long term), and the IE model used a factor to convert person trips (AirSage) to auto trips (based on external PA traffic counts).

Once the trip generation and attraction models were calibrated, a destination choice model was calibrated to the AirSage EI and IE trip tables. The model forms are shown in Equation 4 and Equation 5. The destination choice utility functions were based on similar factors as the trip generation data (zonal data and generalized cost), and it was adjusted during model calibration.

Equation 4: Internal-External DEstination Choice Model

$$IETrips_{ij} = Attractions_i * \frac{\exp(\beta_{time} * time_{ij} + \beta_{cost} * CostPerMile * Distance_{ij})}{\sum \exp(\beta_{time} * time_{ij} + \beta_{cost} * CostPerMile * Distance_{ij})}$$

Equation 5: External-Internal Destination CHoice Model

$$EITrips_{ij} = Productions_i * \frac{\exp(\beta_{time} * time_{ij} + \beta_{cost} * CostPerMile * Distance_{ij})}{\sum \exp(\beta_{time} * time_{ij} + \beta_{cost} * CostPerMile * Distance_{ij})}$$

The external-internal models are singly-constrained, meaning that one iteration is run with no attraction balancing since the rate of internal attractions of external day trips to COG is unknown for any given zone. The internal-external models are also singly-constrained to ensure that the external trip ends matched the external PA counts. The resulting trip table was transposed to prepare them for assignment.

The external-internal and internal-external models were calibrated to trip generation and attraction by zone, the trip length frequency distribution from the AirSage data, and the external PA counts. The implementation of the auxiliary travel model in the Gen3 Model is documented in a technical memorandum[73].

6.9 External and Visitor transit travel model

The external transit model is input to the Gen3 Model as static trip tables. These trip tables primarily represent the transit entering the region via Metrorail, VRE, and MARC commuter rail with a smaller portion using Amtrak Rail. These tables were developed using survey data from the WMATA Metrorail Survey, MTA on-board survey, and VRE on-board survey.

The survey data was processed to determine the home location of riders and then determined using GIS if respondents were external users that reported a home location within 75 miles of the model region or visitors, who reported a home location outside of 75 miles and did not transfer from VRE, MARC or Amtrak. Additional information can be found in the Gen3 Model Phase 1 Development Report[74].

The trip tables are input from the transit input folder. These tables are unchanged via the model processes. If growth is expected for external and visitor transit, the tables would need to be updated via a separate process.

[73] RSG, Inc. Gen3 Auxiliary Travel Model Implementation in the Phase 2 Development. Memorandum. August 18, 2022. https://app.box.com/s/epegylkxr9vvyvay0h3cjbqn53ydypvr.
[74] RSG, Inc. and BMG. Gen3 Data Development. 12/29/21. https://app.box.com/s/xe5vb28daox1aqtw895iy2r5ocy584w8.

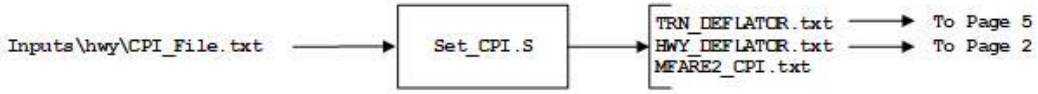
7 APPENDIX

7.1 Detailed model flowcharts

TITLE:	Flowchart for the TPB activity-based, regional travel demand forecasting model
COMPANY: COG/TPB	Authors COG/TPB Staff
DATE: 3/17/2026	PG: 1 OF 14
FILENAME:	Gen3_Ver1.1.0_Flowchart.vsd

***Note: While the current Gen3 Model is primarily designed for use with Cube 6.5.1, it is also compatible with Cube 2025. As shown in this flowchart, the model includes two alternative versions of certain script files (e.g., Highway_Assignment_Parallel.bat) to support execution with either Cube 6.5.1 or Cube 2025.

Set_CPI.bat

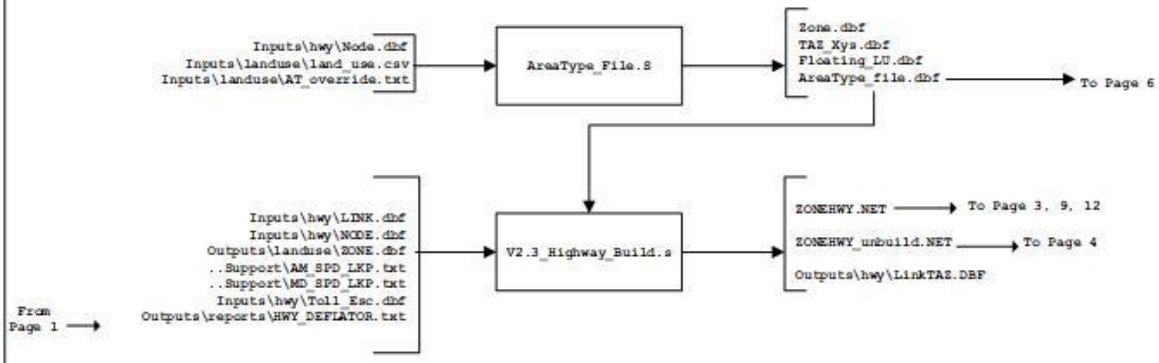


Report Files Generated by Set_CPI.bat:

Set_CPI.rpt
Set_Factors.rpt

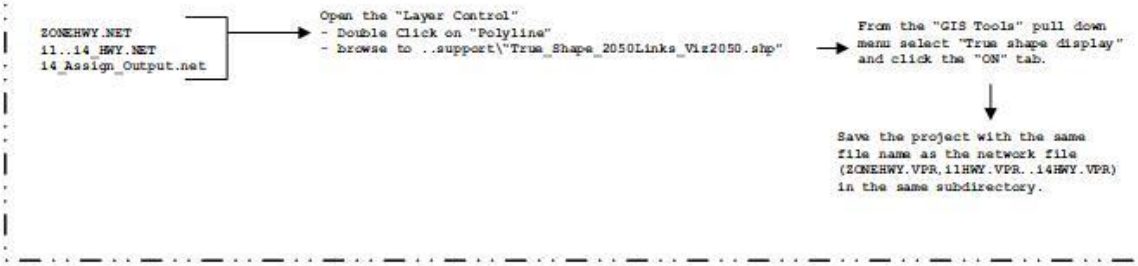
TITLE: Flowchart for the TPB activity-based, regional travel demand forecasting model			
COMPANY: COG/TPB	Authors	COG/TPB Staff	
DATE: 3/17/2026	PG: 2	OF 14	
FILENAME: Gen3_Ver1.1.0_Flowchart.vsd			

PP_Highway_Build.bat



Optional

True shape display



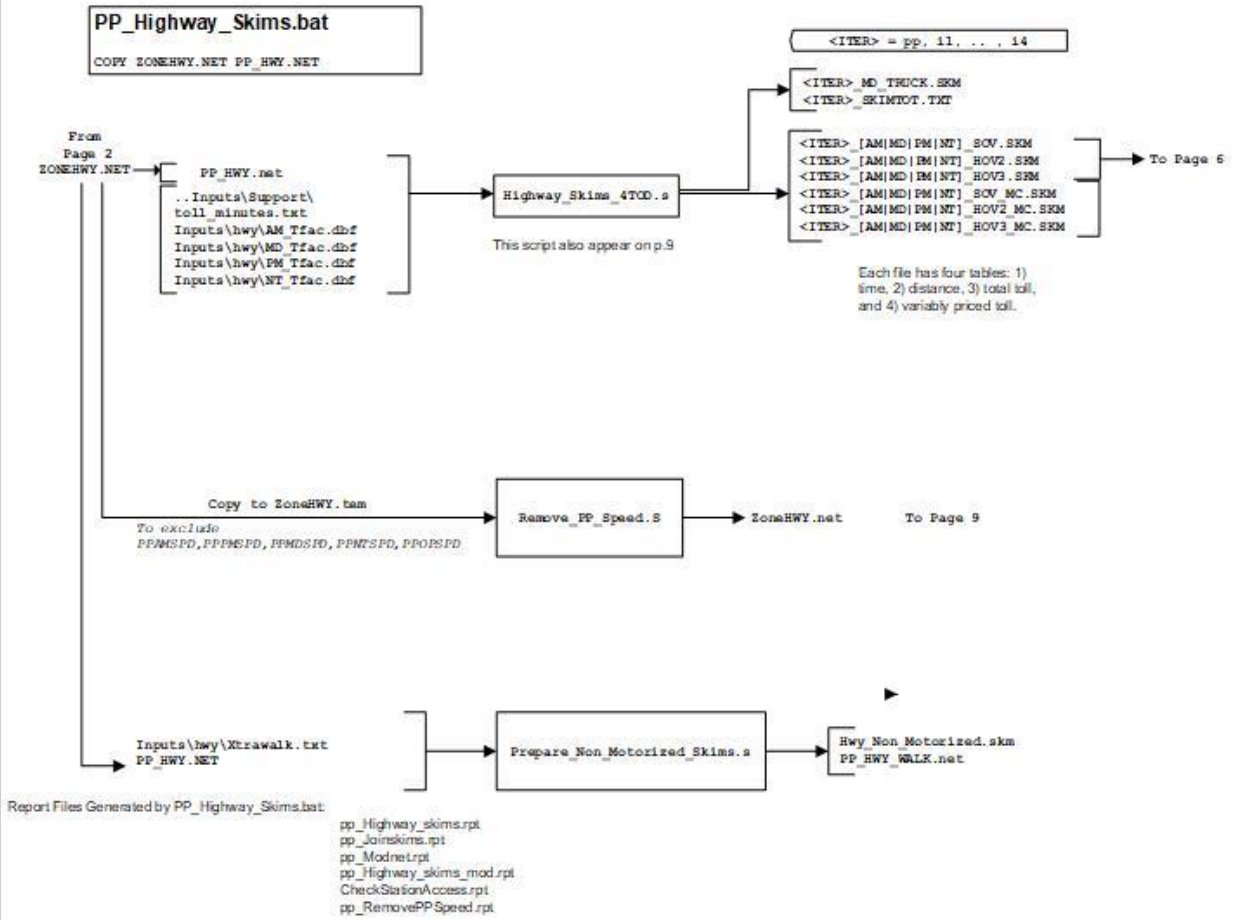
Report Files Generated by PP_Highway_Build.bat:

AreaType_File.rpt
V2.3_highway_build.rpt

TITLE:	Flowchart for the TPB activity-based, regional travel demand forecasting model
COMPANY: COG/TPB	Authors COG/TPB Staff
DATE: 3/17/2026	PG: 3 OF 14
FILENAME:	Gen3_Ver1.1.0_Flowchart.vsd

PP_Highway_Skims.bat

(See also Page 5 for highway skimming process used in speed feedback iterations 1-4)



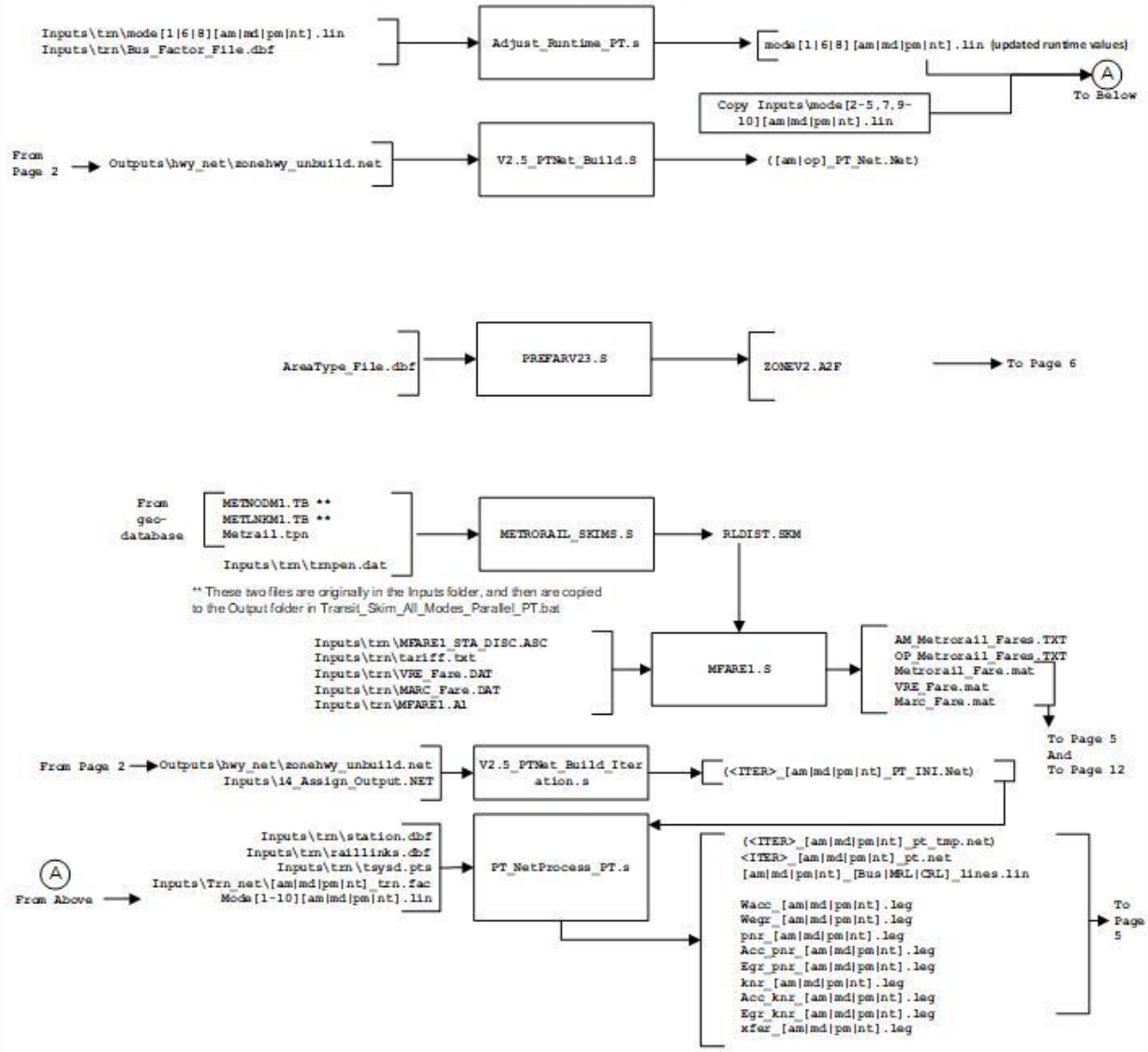
TITLE: Flowchart for the TPB activity-based, regional travel demand forecasting model

COMPANY: COG/TPB Authors COG/TPB Staff

DATE: 3/17/2026 PG: 4 OF 14

FILENAME: Gen3_Ver1.1.0_Flowchart.vsd

Transit_Fare_PT.bat
Transit_Skims_All_Modes_Parallel_PT.bat



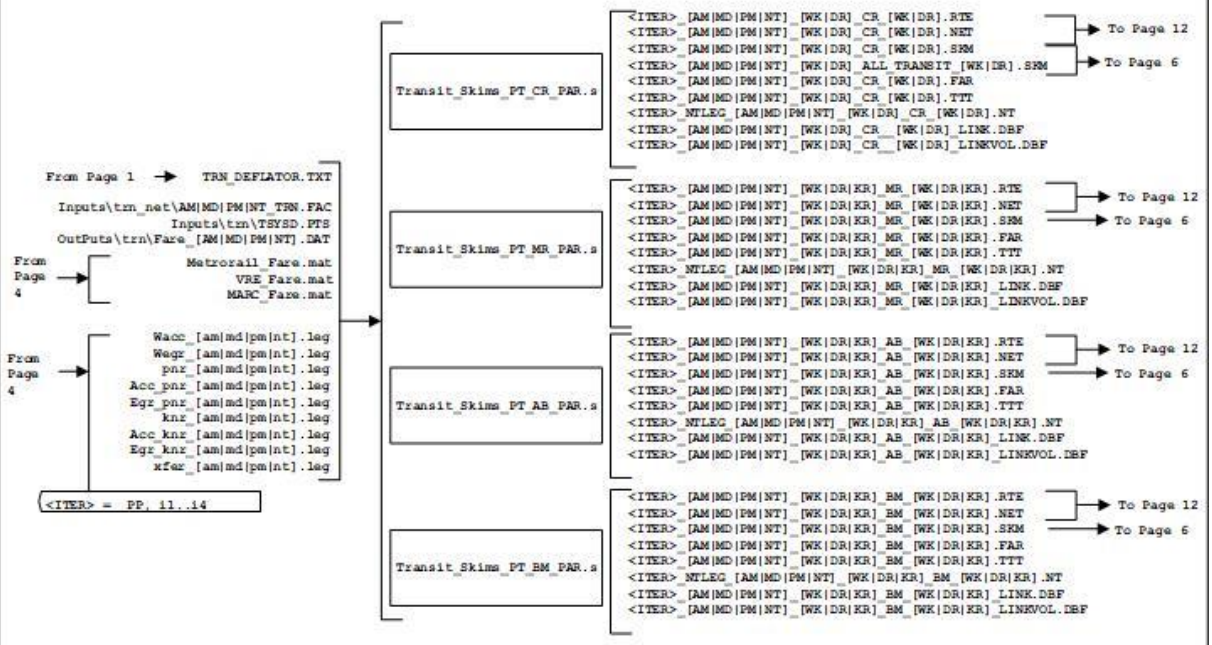
Report Files Generated by Transit_Fare_PT.bat and Transit_Skims_All_Modes_Parallel_PT.bat:

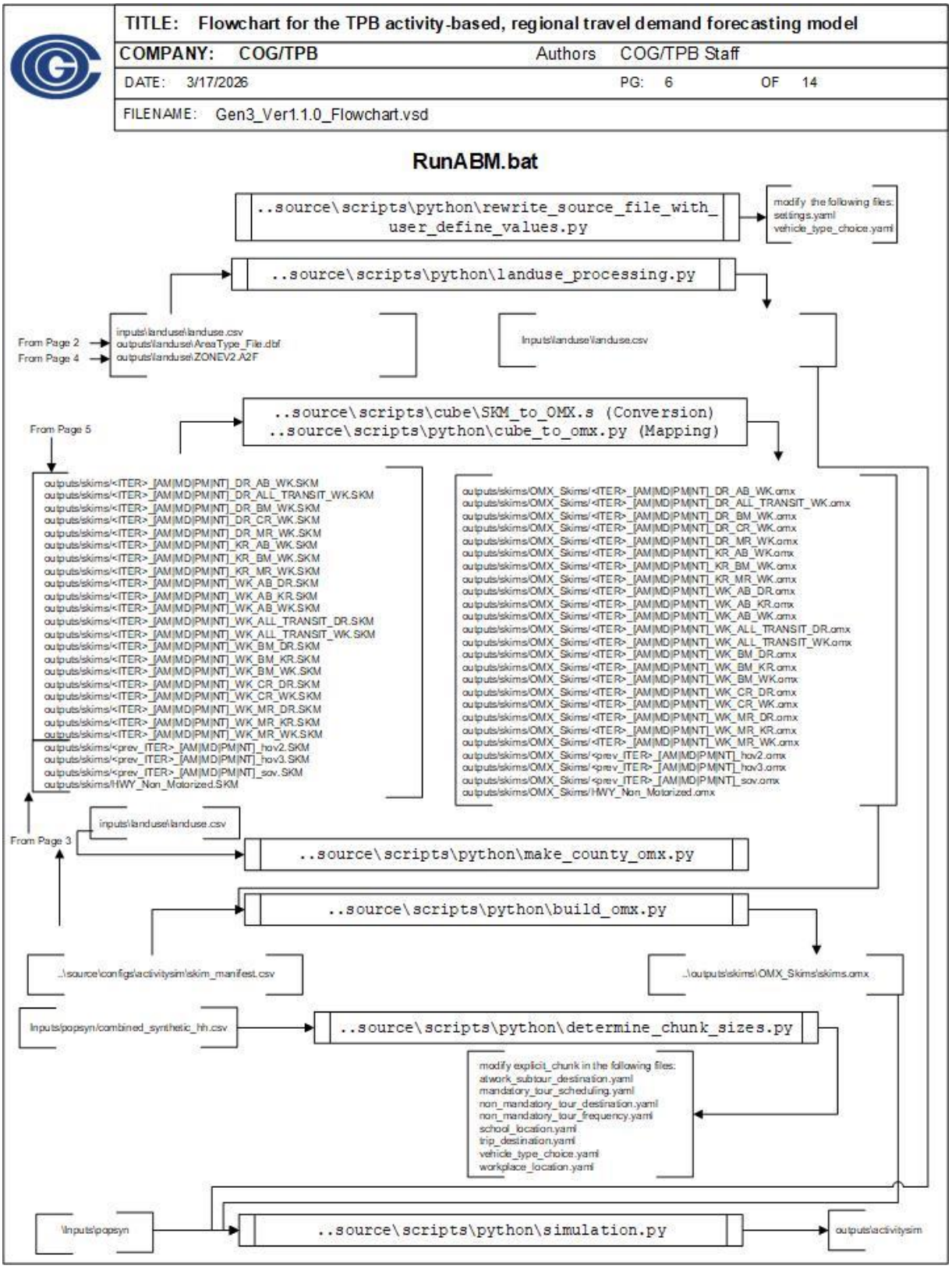
walkacc.rpt
<iter>_prefarV23.rpt
<iter>_Metrorail_skims.rpt
<iter>_MFARE1.rpt
pp_RemovePPSpeed.rpt
Prepare_MC_ZFile.rpt
PT_NetProcess_PT.rpt
PT_skim_PT.rpt
MFare2_PT.rpt
Post_Skim_PT.rpt
<iter>_TRANSIT_Accessibility.RPT

TITLE:	Flowchart for the TPB activity-based, regional travel demand forecasting model
COMPANY: COG/TPB	Authors COG/TPB Staff
DATE: 3/17/2026	PG: 5 OF 14
FILENAME:	Gen3_Ver1.1.0_Flowchart.vsd

Transit_Skim_LineHaul_Parallel_PT.bat

Transit_Skim_TOD_Parallel_PT.bat



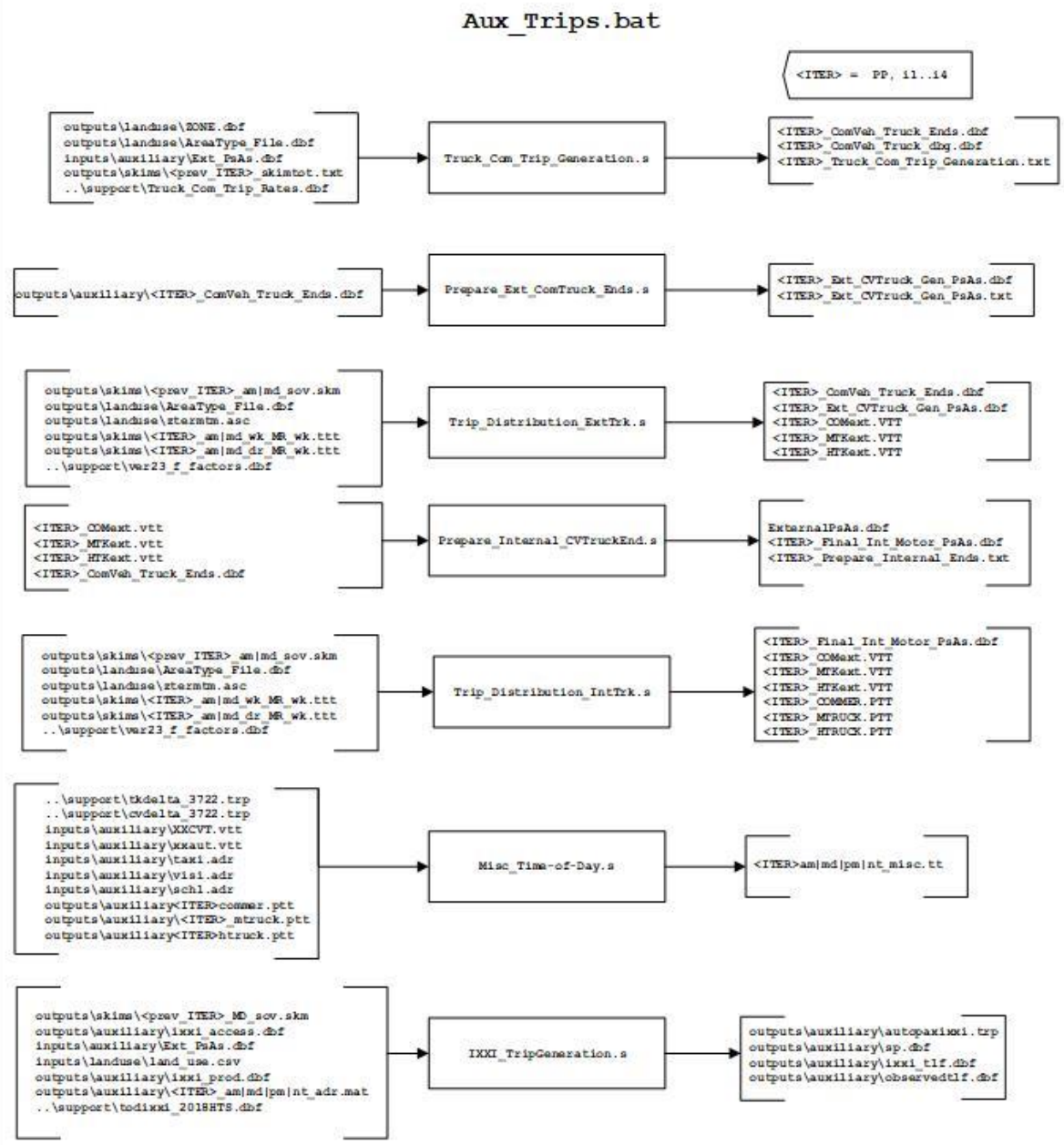

TITLE: Flowchart for the TPB activity-based, regional travel demand forecasting model
COMPANY: COG/TPB
Authors COG/TPB Staff
DATE: 3/17/2026
PG: 6 OF 14
FILENAME: Gen3_Ver1.1.0_Flowchart.vsd
RunABM.bat
..source\scripts\python\rewrite_source_file_with_user_define_values.py
modify the following files:
settings.yaml
vehicle_type_choice.yaml
..source\scripts\python\landuse_processing.py
From Page 2
From Page 4
inputs\landuse\landuse.csv
outputs\landuse\AreaType_File.dbf
outputs\landuse\ZONEV2.A2F
Inputs\landuse\landuse.csv
..source\scripts\cube\SKM_to_OMX.s (Conversion)
..source\scripts\python\cube_to_omx.py (Mapping)
From Page 5
outputs/skims/<ITER>_[AM|MD|PM|NT]_DR_AB_WK.SKM
outputs/skims/<ITER>_[AM|MD|PM|NT]_DR_ALL_TRANSIT_WK.SKM
outputs/skims/<ITER>_[AM|MD|PM|NT]_DR_BM_WK.SKM
outputs/skims/<ITER>_[AM|MD|PM|NT]_DR_CR_WK.SKM
outputs/skims/<ITER>_[AM|MD|PM|NT]_DR_MR_WK.SKM
outputs/skims/<ITER>_[AM|MD|PM|NT]_KR_AB_WK.SKM
outputs/skims/<ITER>_[AM|MD|PM|NT]_KR_BM_WK.SKM
outputs/skims/<ITER>_[AM|MD|PM|NT]_KR_MR_WK.SKM
outputs/skims/<ITER>_[AM|MD|PM|NT]_WK_AB_DR.SKM
outputs/skims/<ITER>_[AM|MD|PM|NT]_WK_AB_KR.SKM
outputs/skims/<ITER>_[AM|MD|PM|NT]_WK_AB_WK.SKM
outputs/skims/<ITER>_[AM|MD|PM|NT]_WK_ALL_TRANSIT_DR.SKM
outputs/skims/<ITER>_[AM|MD|PM|NT]_WK_ALL_TRANSIT_WK.SKM
outputs/skims/<ITER>_[AM|MD|PM|NT]_WK_BM_DR.SKM
outputs/skims/<ITER>_[AM|MD|PM|NT]_WK_BM_KR.SKM
outputs/skims/<ITER>_[AM|MD|PM|NT]_WK_BM_WK.SKM
outputs/skims/<ITER>_[AM|MD|PM|NT]_WK_CR_DR.SKM
outputs/skims/<ITER>_[AM|MD|PM|NT]_WK_CR_WK.SKM
outputs/skims/<ITER>_[AM|MD|PM|NT]_WK_MR_DR.SKM
outputs/skims/<ITER>_[AM|MD|PM|NT]_WK_MR_KR.SKM
outputs/skims/<ITER>_[AM|MD|PM|NT]_WK_MR_WK.SKM
outputs/skims/<prev_ITER>_[AM|MD|PM|NT]_hov2.SKM
outputs/skims/<prev_ITER>_[AM|MD|PM|NT]_hov3.SKM
outputs/skims/<prev_ITER>_[AM|MD|PM|NT]_sov.SKM
outputs/skims/HWY_Non_Motorized.SKM
outputs/skims/OMX_Skims/<ITER>_[AM|MD|PM|NT]_DR_AB_WK.omx
outputs/skims/OMX_Skims/<ITER>_[AM|MD|PM|NT]_DR_ALL_TRANSIT_WK.omx
outputs/skims/OMX_Skims/<ITER>_[AM|MD|PM|NT]_DR_BM_WK.omx
outputs/skims/OMX_Skims/<ITER>_[AM|MD|PM|NT]_DR_CR_WK.omx
outputs/skims/OMX_Skims/<ITER>_[AM|MD|PM|NT]_DR_MR_WK.omx
outputs/skims/OMX_Skims/<ITER>_[AM|MD|PM|NT]_KR_AB_WK.omx
outputs/skims/OMX_Skims/<ITER>_[AM|MD|PM|NT]_KR_BM_WK.omx
outputs/skims/OMX_Skims/<ITER>_[AM|MD|PM|NT]_KR_MR_WK.omx
outputs/skims/OMX_Skims/<ITER>_[AM|MD|PM|NT]_WK_AB_DR.omx
outputs/skims/OMX_Skims/<ITER>_[AM|MD|PM|NT]_WK_AB_KR.omx
outputs/skims/OMX_Skims/<ITER>_[AM|MD|PM|NT]_WK_AB_WK.omx
outputs/skims/OMX_Skims/<ITER>_[AM|MD|PM|NT]_WK_ALL_TRANSIT_DR.omx
outputs/skims/OMX_Skims/<ITER>_[AM|MD|PM|NT]_WK_ALL_TRANSIT_WK.omx
outputs/skims/OMX_Skims/<ITER>_[AM|MD|PM|NT]_WK_BM_DR.omx
outputs/skims/OMX_Skims/<ITER>_[AM|MD|PM|NT]_WK_BM_KR.omx
outputs/skims/OMX_Skims/<ITER>_[AM|MD|PM|NT]_WK_BM_WK.omx
outputs/skims/OMX_Skims/<ITER>_[AM|MD|PM|NT]_WK_CR_DR.omx
outputs/skims/OMX_Skims/<ITER>_[AM|MD|PM|NT]_WK_CR_WK.omx
outputs/skims/OMX_Skims/<ITER>_[AM|MD|PM|NT]_WK_MR_DR.omx
outputs/skims/OMX_Skims/<ITER>_[AM|MD|PM|NT]_WK_MR_KR.omx
outputs/skims/OMX_Skims/<ITER>_[AM|MD|PM|NT]_WK_MR_WK.omx
outputs/skims/OMX_Skims/<prev_ITER>_[AM|MD|PM|NT]_hov2.omx
outputs/skims/OMX_Skims/<prev_ITER>_[AM|MD|PM|NT]_hov3.omx
outputs/skims/OMX_Skims/<prev_ITER>_[AM|MD|PM|NT]_sov.omx
outputs/skims/OMX_Skims/HWY_Non_Motorized.omx
From Page 3
inputs\landuse\landuse.csv
..source\scripts\python\make_county_omx.py
..source\scripts\python\build_omx.py
..\source\configs\activitysim\skim_manifest.csv
..\outputs\skims\OMX_Skims\skims.omx
Inputs/popsyn/combined_synthetic_hh.csv
..source\scripts\python\determine_chunk_sizes.py
modify explicit_chunk in the following files:
atwork_subtour_destination.yaml
mandatory_tour_scheduling.yaml
non_mandatory_tour_destination.yaml
non_mandatory_tour_frequency.yaml
school_location.yaml
trip_destination.yaml
vehicle_type_choice.yaml
workplace_location.yaml
\Inputs\popsyn
..source\scripts\python\simulation.py
outputs\activitysim

TITLE: Flowchart for the TPB activity-based, regional travel demand forecasting model

COMPANY: COG/TPB Authors COG/TPB Staff

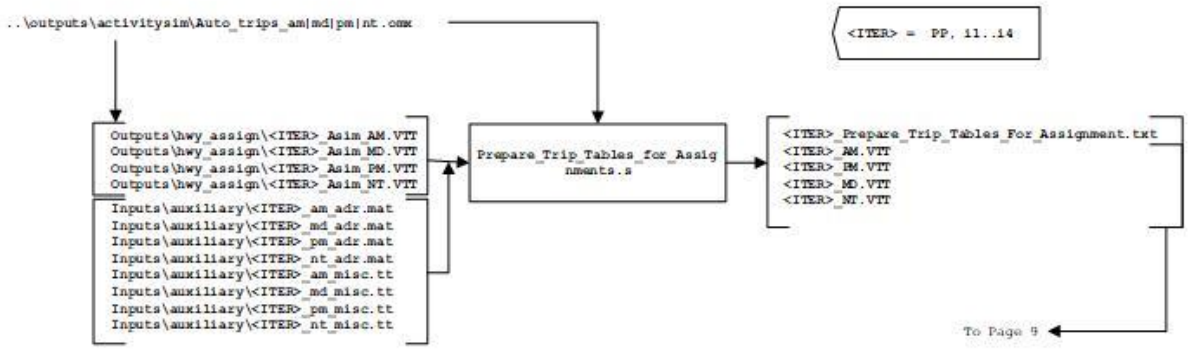
DATE: 3/17/2026 PG: 7 OF 14

FILENAME: Gen3_Ver1.1.0_Flowchart.vsd



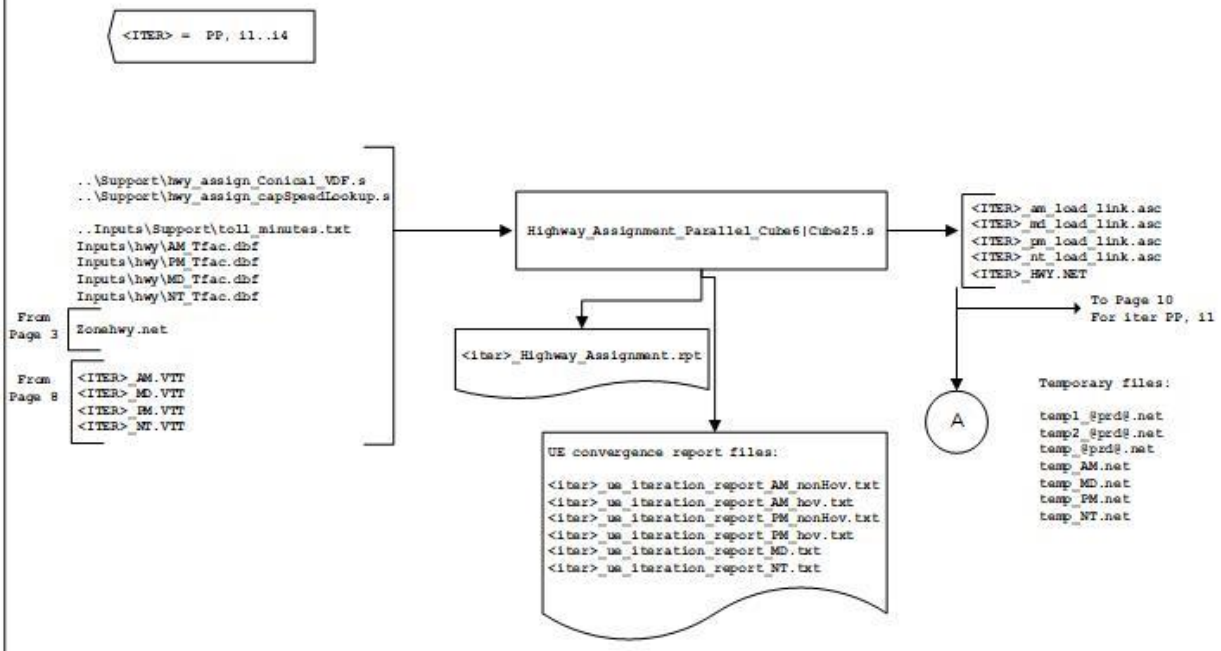
TITLE:	Flowchart for the TPB activity-based, regional travel demand forecasting model
COMPANY: COG/TPB	Authors COG/TPB Staff
DATE: 3/17/2026	PG: 8 OF 14
FILENAME:	Gen3_Ver1.1.0_Flowchart.vsd

Highway_Assignment_Prep.bat

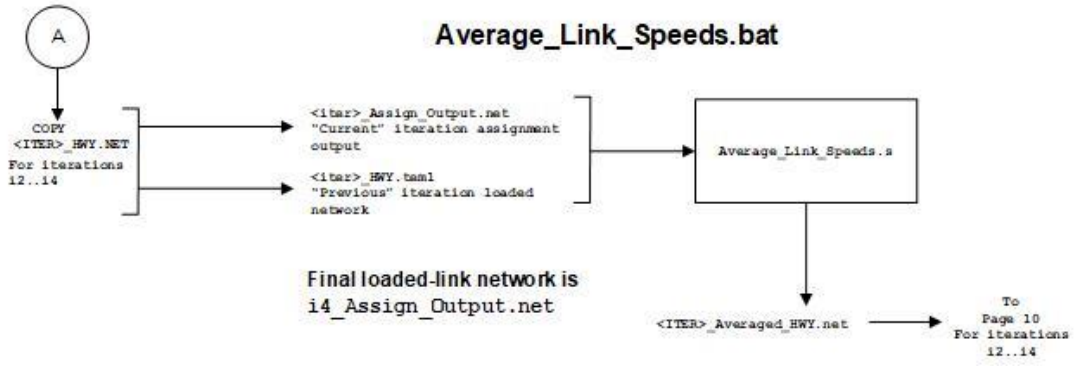


TITLE:	Flowchart for the TPB activity-based, regional travel demand forecasting model
COMPANY: COG/TPB	Authors COG/TPB Staff
DATE: 3/17/2026	PG: 9 OF 14
FILENAME:	Gen3_Ver1.1.0_Flowchart.vsd

Highway_Assignment_parallel_Cube6|Cube25.bat



Average_Link_Speeds.bat



Report Files Generated by Highway_Assignment.bat:

<ITER>_Highway_assignment.rpt
Average_Link_Speeds.rpt

TITLE:	Flowchart for the TPB activity-based, regional travel demand forecasting model
COMPANY: COG/TPB	Authors COG/TPB Staff
DATE: 3/17/2026	PG: 10 OF 14
FILENAME:	Gen3_Ver1.1.0_Flowchart.vsd

Highway_Skims.bat

(See also page 3 for highway skimming process used in the "pump prime" speed feedback iteration)

Report Files Generated by Highway_Skims.bat:

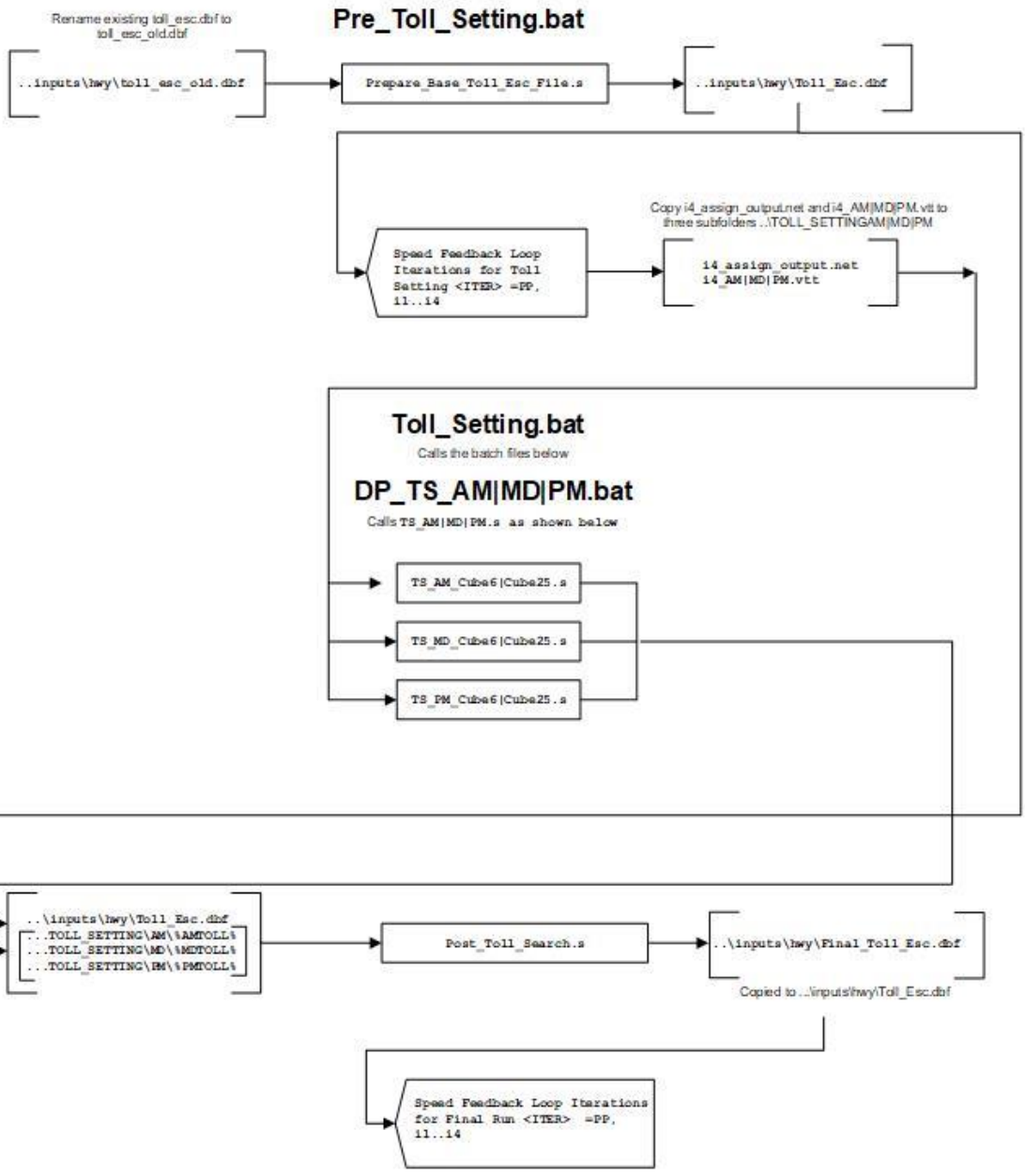
<ITER>_Highway_skims.rpt
<ITER>_Joinskims.rpt
<ITER>_Modnet.rpt
<ITER>_Highway_skims_mod.rpt

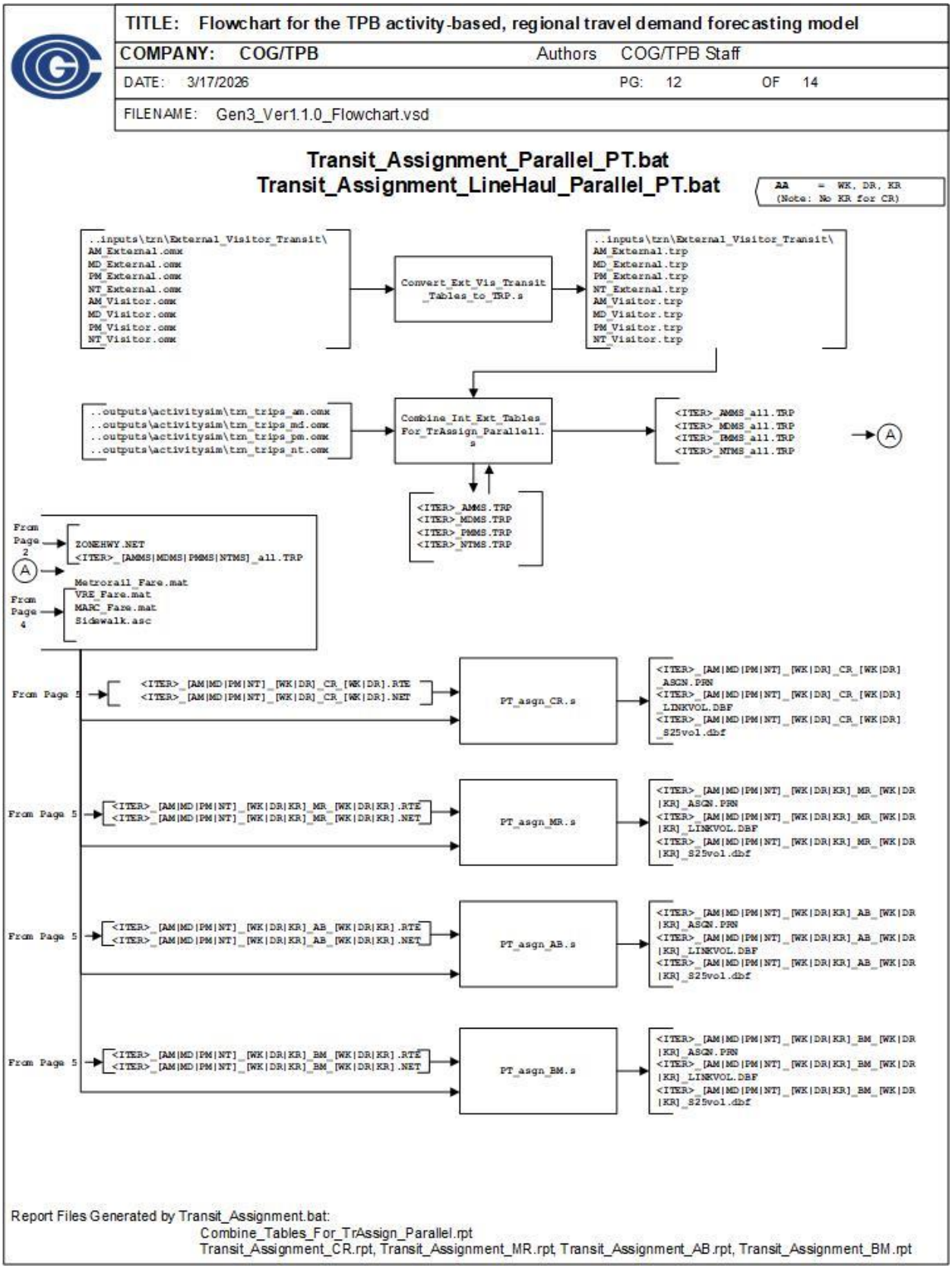
TITLE: Flowchart for the TPB activity-based, regional travel demand forecasting model

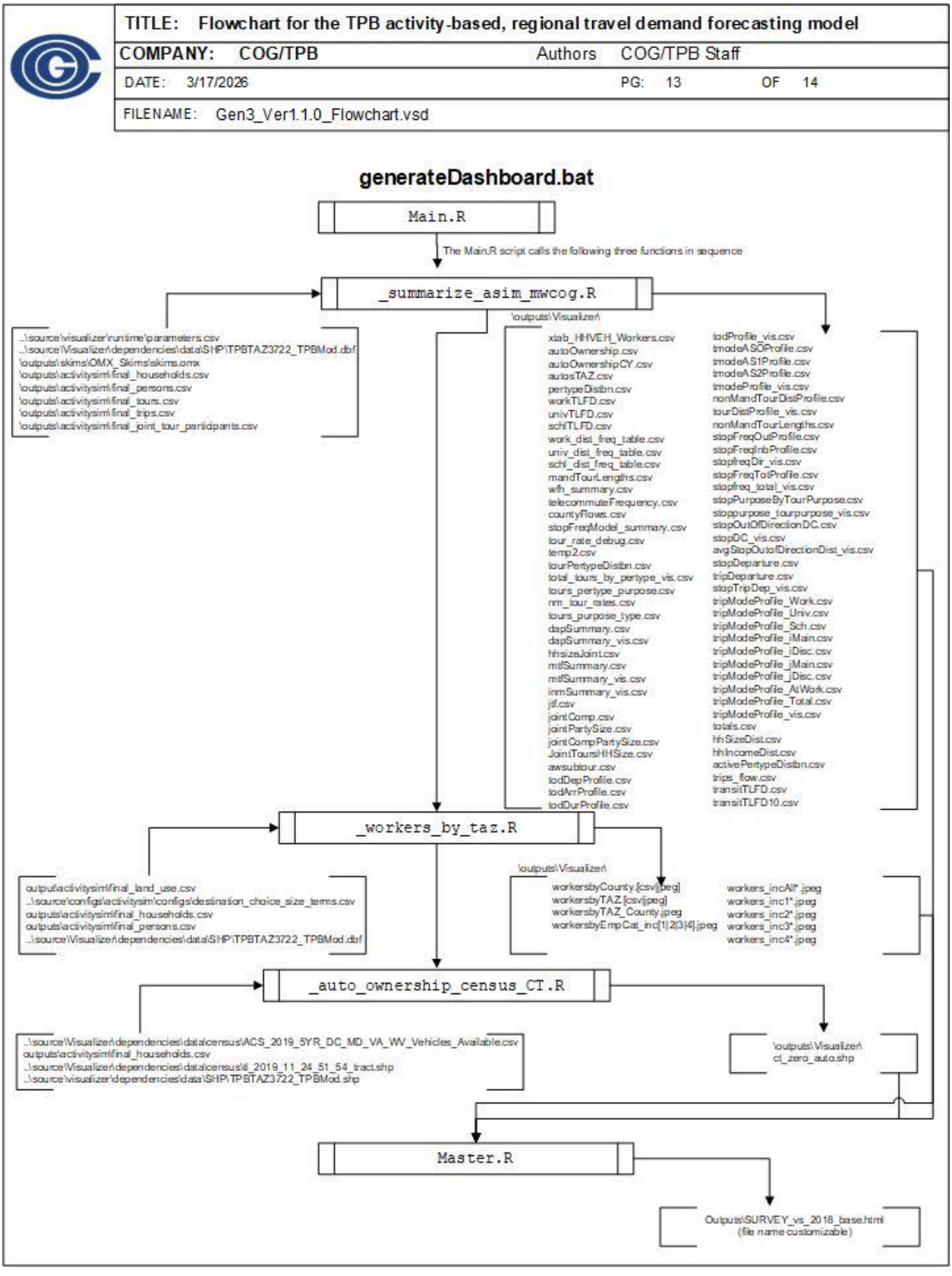
COMPANY: COG/TPB Authors COG/TPB Staff

DATE: 3/17/2026 PG: 11 OF 14

FILENAME: Gen3_Ver1.1.0_Flowchart.vsd

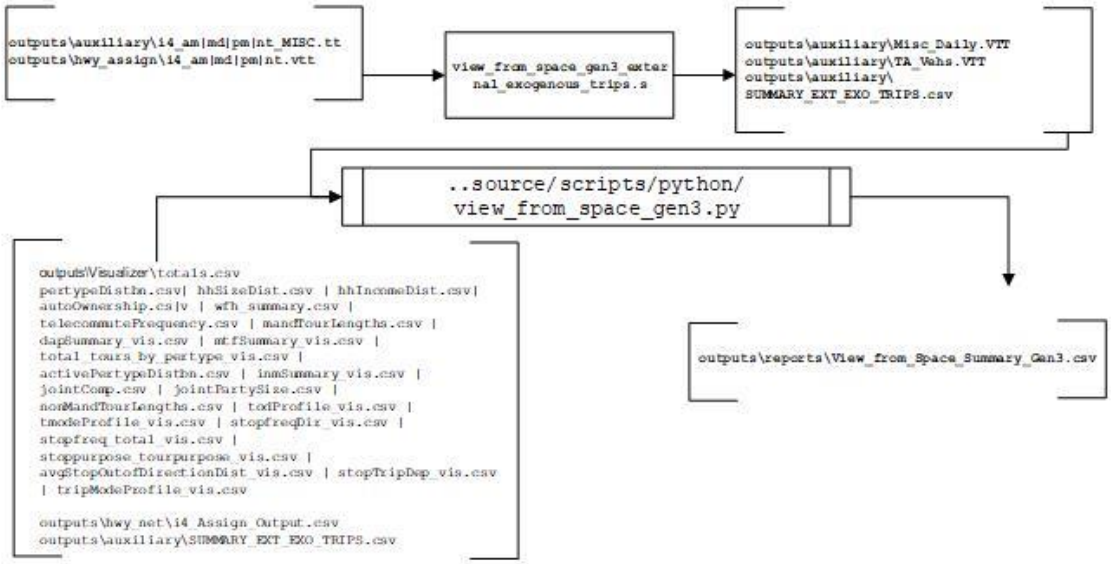



TITLE: Flowchart for the TPB activity-based, regional travel demand forecasting model
COMPANY: COG/TPB
Authors COG/TPB Staff
DATE: 3/17/2026
PG: 12 OF 14
FILENAME: Gen3_Ver1.1.0_Flowchart.vsd
Transit_Assignment_Parallel_PT.bat
Transit_Assignment_LineHaul_Parallel_PT.bat
AA = WK, DR, KR
(Note: No KR for CR)
..inputs\trn\External_Visitor_Transit\
AM_External.omx
MD_External.omx
PM_External.omx
NT_External.omx
AM_Visitor.omx
MD_Visitor.omx
PM_Visitor.omx
NT_Visitor.omx
Convert_Ext_Vis_Transit_Tables_to_TRP.s
..inputs\trn\External_Visitor_Transit\
AM_External.trp
MD_External.trp
PM_External.trp
NT_External.trp
AM_Visitor.trp
MD_Visitor.trp
PM_Visitor.trp
NT_Visitor.trp
..outputs\activitysim\trn_trips_am.omx
..outputs\activitysim\trn_trips_md.omx
..outputs\activitysim\trn_trips_pm.omx
..outputs\activitysim\trn_trips_nt.omx
Combine_Int_Ext_Tables_For_TrAssign_Parallel1.s
<ITER>_AMMS_all.TRP
<ITER>_MDMS_all.TRP
<ITER>_PMMS_all.TRP
<ITER>_NTMS_all.TRP
A
<ITER>_AMMS.TRP
<ITER>_MDMS.TRP
<ITER>_PMMS.TRP
<ITER>_NTMS.TRP
From Page 2
A
ZONEHWY.NET
<ITER>_[AMMS|MDMS|PMMS|NTMS]_all.TRP
From Page 4
Metrorail_Fare.mat
VRE_Fare.mat
MARC_Fare.mat
Sidewalk.asc
From Page 5
<ITER>_[AM|MD|PM|NT]_[WK|DR]_CR_[WK|DR].RTE
<ITER>_[AM|MD|PM|NT]_[WK|DR]_CR_[WK|DR].NET
PT_asgn_CR.s
<ITER>_[AM|MD|PM|NT]_[WK|DR]_CR_[WK|DR]_ASGN.PRN
<ITER>_[AM|MD|PM|NT]_[WK|DR]_CR_[WK|DR]_LINKVOL.DBF
<ITER>_[AM|MD|PM|NT]_[WK|DR]_CR_[WK|DR]_S25vol.dbf
From Page 5
<ITER>_[AM|MD|PM|NT]_[WK|DR|KR]_MR_[WK|DR|KR].RTE
<ITER>_[AM|MD|PM|NT]_[WK|DR|KR]_MR_[WK|DR|KR].NET
PT_asgn_MR.s
<ITER>_[AM|MD|PM|NT]_[WK|DR|KR]_MR_[WK|DR|KR]_ASGN.PRN
<ITER>_[AM|MD|PM|NT]_[WK|DR|KR]_MR_[WK|DR|KR]_LINKVOL.DBF
<ITER>_[AM|MD|PM|NT]_[WK|DR|KR]_MR_[WK|DR|KR]_S25vol.dbf
From Page 5
<ITER>_[AM|MD|PM|NT]_[WK|DR|KR]_AB_[WK|DR|KR].RTE
<ITER>_[AM|MD|PM|NT]_[WK|DR|KR]_AB_[WK|DR|KR].NET
PT_asgn_AB.s
<ITER>_[AM|MD|PM|NT]_[WK|DR|KR]_AB_[WK|DR|KR]_ASGN.PRN
<ITER>_[AM|MD|PM|NT]_[WK|DR|KR]_AB_[WK|DR|KR]_LINKVOL.DBF
<ITER>_[AM|MD|PM|NT]_[WK|DR|KR]_AB_[WK|DR|KR]_S25vol.dbf
From Page 5
<ITER>_[AM|MD|PM|NT]_[WK|DR|KR]_BM_[WK|DR|KR].RTE
<ITER>_[AM|MD|PM|NT]_[WK|DR|KR]_BM_[WK|DR|KR].NET
PT_asgn_BM.s
<ITER>_[AM|MD|PM|NT]_[WK|DR|KR]_BM_[WK|DR|KR]_ASGN.PRN
<ITER>_[AM|MD|PM|NT]_[WK|DR|KR]_BM_[WK|DR|KR]_LINKVOL.DBF
<ITER>_[AM|MD|PM|NT]_[WK|DR|KR]_BM_[WK|DR|KR]_S25vol.dbf
Report Files Generated by Transit_Assignment.bat:
Combine_Tables_For_TrAssign_Parallel.rpt
Transit_Assignment_CR.rpt, Transit_Assignment_MR.rpt, Transit_Assignment_AB.rpt, Transit_Assignment_BM.rpt

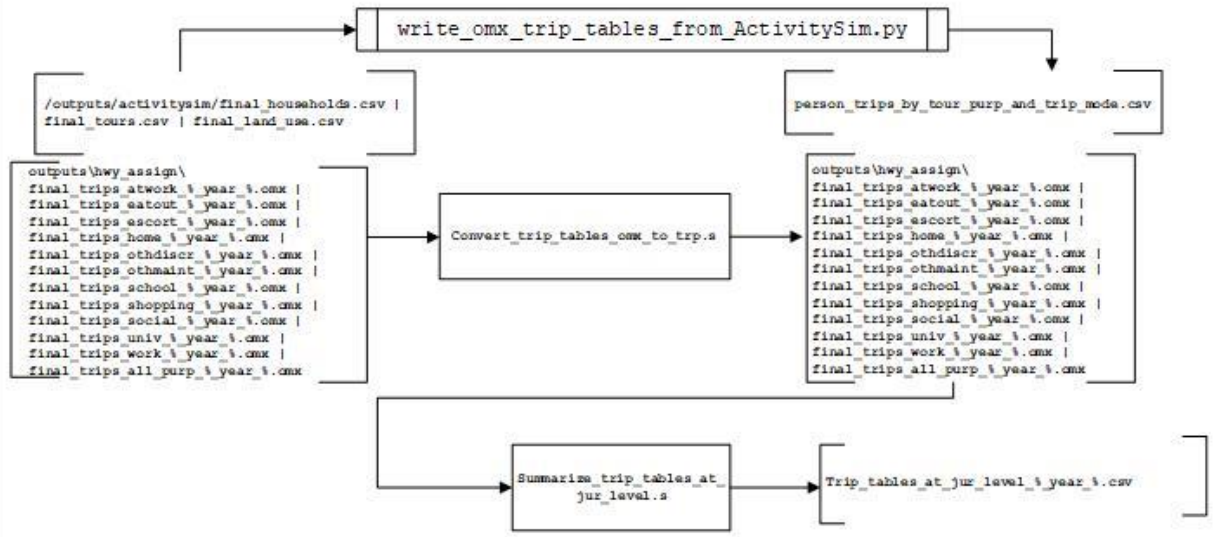

TITLE: Flowchart for the TPB activity-based, regional travel demand forecasting model
COMPANY: COG/TPB
Authors COG/TPB Staff
DATE: 3/17/2026
PG: 13 OF 14
FILENAME: Gen3_Ver1.1.0_Flowchart.vsd
generateDashboard.bat
Main.R
The Main.R script calls the following three functions in sequence
_summarize_asim_mwcog.R
..\source\visualizer\runtime\parameters.csv
..\source\Visualizer\dependencies\data\SHP\TPBTAZ3722_TPBMod.dbf
\outputs\skims\OMX_Skims\skims.omx
\outputs\activitysim\final_households.csv
\outputs\activitysim\final_persons.csv
\outputs\activitysim\final_tours.csv
\outputs\activitysim\final_trips.csv
\outputs\activitysim\final_joint_tour_participants.csv
\outputs\Visualizer\
xtab_HHVEH_Workers.csv
autoOwnership.csv
autoOwnershipCY.csv
autosTAZ.csv
pertypeDistbn.csv
workTLFD.csv
univTLFD.csv
schlTLFD.csv
work_dist_freq_table.csv
univ_dist_freq_table.csv
schl_dist_freq_table.csv
mandTourLengths.csv
wfh_summary.csv
telecommuteFrequency.csv
countyFlows.csv
stopFreqModel_summary.csv
tour_rate_debug.csv
temp2.csv
tourPertypeDistbn.csv
total_tours_by_pertype_vis.csv
tours_pertype_purpose.csv
nm_tour_rates.csv
tours_purpose_type.csv
dapSummary.csv
dapSummary_vis.csv
hhsizeJoint.csv
mtfSummary.csv
mtfSummary_vis.csv
nmSummary_vis.csv
jtf.csv
jointComp.csv
jointPartySize.csv
jointCompPartySize.csv
JointTourHHSize.csv
awsubtour.csv
todDepProfile.csv
todArrProfile.csv
todDurProfile.csv
todProfile_vis.csv
tmodeASOProfile.csv
tmodeAS1Profile.csv
tmodeAS2Profile.csv
tmodeProfile_vis.csv
nonMandTourDistProfile.csv
tourDistProfile_vis.csv
nonMandTourLengths.csv
stopFreqOutProfile.csv
stopFreqInbProfile.csv
stopfreqDir_vis.csv
stopFreqTotProfile.csv
stopfreq_total_vis.csv
stopPurposeByTourPurpose.csv
stoppurpose_tourpurpose_vis.csv
stopOutOfDirectionDC.csv
stopDC_vis.csv
avgStopOutofDirectionDist_vis.csv
stopDeparture.csv
tripDeparture.csv
stopTripDep_vis.csv
tripModeProfile_Work.csv
tripModeProfile_Univ.csv
tripModeProfile_Sch.csv
tripModeProfile_iMain.csv
tripModeProfile_iDisc.csv
tripModeProfile_jMain.csv
tripModeProfile_jDisc.csv
tripModeProfile_AtWork.csv
tripModeProfile_Total.csv
tripModeProfile_vis.csv
totals.csv
hhSizeDist.csv
hhIncomeDist.csv
activePertypeDistbn.csv
trips_flow.csv
transitTLFD.csv
transitTLFD10.csv
_workers_by_taz.R
output\activitysim\final_land_use.csv
..\source\configs\activitysim\configs\destination_choice_size_terms.csv
outputs\activitysim\final_households.csv
outputs\activitysim\final_persons.csv
..\source\Visualizer\dependencies\data\SHP\TPBTAZ3722_TPBMod.dbf
\outputs\Visualizer\
workersbyCounty.[csv|jpeg]
workersbyTAZ.[csv|jpeg]
workersbyTAZ_County.jpeg
workersbyEmpCat_inc[1|2|3|4].jpeg
workers_incAll*.jpeg
workers_inc1*.jpeg
workers_inc2*.jpeg
workers_inc3*.jpeg
workers_inc4*.jpeg
_auto_ownership_census_CT.R
..\source\Visualizer\dependencies\data\census\ACS_2019_5YR_DC_MD_VA_WV_Vehicles_Available.csv
outputs\activitysim\final_households.csv
..\source\Visualizer\dependencies\data\census\tl_2019_11_24_51_54_tract.shp
..\source\visualizer\dependencies\data\SHP\TPBTAZ3722_TPBMod.shp
\outputs\Visualizer\
ct_zero_auto.shp
Master.R
Outputs\SURVEY_vs_2018_base.html
(file name customizable)

TITLE:	Flowchart for the TPB activity-based, regional travel demand forecasting model
COMPANY: COG/TPB	Authors COG/TPB Staff
DATE: 3/17/2026	PG: 14 OF 14
FILENAME:	Gen3_Ver1.1.0_Flowchart.vsd

view_from_space_gen3.bat



Summarize_Person_Trips_from_ActivitySim.bat



*** Note: At the end of the model run, move_temp_files_gen3.bat moves temporary files to a temp folder, and shutdown_60s.bat shuts down the server with a 60-second delay, whether the run completes successfully or with an error.

7.2 Using the Gen3 Model for AV ownership

This section is a supplement to section 5.4 and is aimed at adjusting the model for automated vehicle scenario studies. There are two options in this section - one is for simple studies, and the other is for more detailed studies. A simple study is where there is a regional target percentage of AV-owning households. A more detailed study is where there are detailed targets - whether by income, location, or any other variable used in ActivitySim. In both cases, there is a group of mode choice and vehicle selection updates that should be adjusted.

7.2.1 Simple studies

A simple study can be undertaken by changing two files, av_ownership.yaml file and av_ownership_coeffs.csv.
In av_ownership.yaml, set AV_OWNERSHIP_TARGET_PERCENT to a the target percentage of households in the region that should own AVs. Then, set the AV_OWNERSHIP_TARGET_PERCENT_TOLERANCE to an allowable tolerance. For example, if the target is 10% and the tolerance is 1%, the model will iterate until the regional share is between 9% and 11%. The number of iterations is controlled by AV_OWNERSHIP_ITERATIONS. Finally, check in av_ownership_coeffs.csv to ensure that the coefficient named in the YAML file as AV_OWNERSHIP_COEFFICIENT_CONSTANT is not set to -999 (if it is, 0 is an appropriate value to start with).

7.2.2 Complex studies

For a complex study, the individual constants will be adjusted. It is expected that the adjustments will happen in outside software, so this section is an outline of how things should be updated in the ActivitySim model configuration files. At least three of the four AV ownership model files will need to be updated for a complex study.

The first update is av_ownership.yaml. In this file, AV_OWNERSHIP_TARGET_PERCENT, AV_OWNERSHIP_TARGET_PERCENT_TOLERANCE, AV_OWNERSHIP_ITERATIONS, and AV_OWNERSHIP_COEFFICIENT_CONSTANT will all need to be commented out (using a hashtag). This will keep ActivitySim from attempting to match a regional target.

The second update is in av_ownership.csv, the model spec. This will define how the segmentation is defined. This file has four fields, an identifier, a description, an expression, and a constant to add to the utility for being an AV-owning household, and a 0 for the utility of a household not owning an AV. The expression must result in the value of 1 or True and needs to follow the expression format used in Pandas. An example is below, which uses income segments (setup in annotate_households.csv) to determine if a constant is applied to a household. In the case of multiple items, expressions must be in parentheses and joined with either an ampersand (&) for ‘and’ or a pipe (|) for ‘or’. In the case of complex expressions, they can be added to av_ownership_preprocessor.csv (see section 5.4 for more details).

```
util_inc1,Utility for income group 1,income_segment == 1,coef_av_inc1,0
util_inc2,Utility for income group 2,income_segment == 2,coef_av_inc2,0
util_inc3,Utility for income group 3,income_segment == 3,coef_av_inc3,0
util_inc4,Utility for income group 4,income_segment == 4,coef_av_inc4,0
```

The third change is the coefficients file. The coefficients listed in the model specification will need to be added to the file. This file is formatted with the coefficient name (which must match the model specification), the value, and an F or T for if the coefficient should NOT be adjusted[75]. An example is below.

```
coef_av_inc1,0.8779621242338954,F
coef_av_inc2,-0.13572485598037048,F
coef_av_inc3,-0.06581044921537069,F
coef_av_inc4,3.099819356052244,F
```

7.2.3 AV mode choice updates

The tour and trip mode choice models will reflect a reduced burden on the in-vehicle time, parking cost, and terminal time for all motorized modes (single-occupant vehicle, 2-person shared-ride, and 3 or more person shared-ride). These adjustments are listed in Table 55. Additionally, if an AV is available for the tour and trip, the minimum age to drive is reduced, using a default of 13 years old.

All four of these adjustments are included in the Gen3_Model\source\configs\activitysim\configs\constants.yaml ActivitySim configuration file.

Table 55: AV Mode Choice Adjustment Defaults

ITEM	CONSTANT NAME (USED IN CONSTANTS.YAML)	ADJUSTMENT DEFAULT
In-Vehicle Time Multiplier	autoIVTFactorAV	0.75
Parking Cost Multiplier	autoParkingCostFactorAV	0.5
Terminal Time Multiplier	autoTerminalTimeFactorAV	0.65
Minimum Driving Age (years old)	minAgeDriveAloneAV	13

ActivitySim 1.2.1 is required for this model to operate correctly. The main benefit of this update is that ActivitySim 1.2.1 allows users to relax the requirement that vehicle type data exist for every possible alternative, including those that may not exist in the future[76].

7.2.4 Vehicle type data

The vehicle type data included in the model includes some basic AV assumptions. In the case that the assumptions need to be updated, a Jupyter Notebook is included in the Gen3 Model to assist in creating vehicle type data at Gen3_Model\source\scripts\python\construct_veh_type_data.ipynb. This notebook allows users to set up a new vehicle allocation file for future years reflecting AVs. This notebook uses information setup in Gen3_Model\source\configs\activitysim\configs\av_extrapolation.yaml to determine the variables, such as body types and fuel types, to extrapolate. For AV body types, the body type names must end with "-AV" (case sensitive). In addition, the NumMakes, NumModels, MPG, Range, NewPrice, and auto_operating_cost must include sections for AV body types. For example, if only Hybrid (PEV) and Battery-only (BEV) fuel types will be available for only car body types that can be AVs, the body_types section needs to be updated to include Car-AV, and NumMakes, NumModels, MPG, Range,

[75] This is observed by Larch for estimating models. Otherwise it must be built in to scripts that are used to adjust constants.

[76] See https://github.com/ActivitySim/activitysim/issues/587 for a detailed explanation.

NewPrice, and auto_operating_cost need to be updated to include entries for Car-AV_PEV and Car-AV_BEV. Some suggested values are included in av_extrapolation.yaml, but they should be reviewed prior to use, particularly as AV adoption increases. The av_extrapolation.yaml file included on the COG's Gen3 Model GitHub repository includes AV body types for car, van, SUV, and pickup truck. The av_extrapolation.yaml file includes setup variables for the minimum fit year (min_fit_year, the first year to output, currently set to 1998) and the maximum year to extrapolate out to (max_extension, currently set to 2050). The next three array variables are for the variables to extrapolate (variables_to_extrapolate), which are set to include the number of makes, number of models, fuel efficiency in miles per gallon, range, new price, and auto operating cost (indicated as NumMakes, NumModels, MPG, Range, NewPrice, and auto_operating_cost, respectively). The next array is body types (body_types), and lists the body types that will be used in the extrapolation. When using AVs, the body type MUST end in '-AV' (case sensitive). By default, this array includes Car, Car-AV, Van, Van-AV, SUV, SUV-AV, Pickup, Pickup-AV, and Motorcycle. The final array is for fuel types (fuel_types), and it includes gas, diesel, hybrid, PEV, and BEV. In the current implementation (as of 6/28/2023), hybrid refers to non-plug-in hybrid, PEV is plugin hybrid, and BEV is battery plug-in EV (not hybrid/no internal combustion engine).

The next six array groups in the file are named according to the variables to extrapolate, and each value (e.g. NumMakes) is split into a variable named with the model, an underscore, and a fuel type (e.g. Car_Gas, Car-AV_Gas, etc.). Each model+fuel type has an interpolation method, years (which can be an array of multiple, such as [2030, 2050], or a single value), and values (if the years is an array, the values need to be an array of the same size). The interpolation methods are listed in Table 56.
By default, this file is output to vehicle_type_data_extended.csv, and that will need to be reflected in the vehicle type choice model setup.

A series of plots are generated from running the construct_veh_type_data.ipynb file showing the predicted variables (NumMakes, NumModels, MPG, Range, NewPrice, and auto_operating_cost) through 2050. Some example plots can be found in Figure 11.
.

Table 56: AV interpolation Methods

METHOD NAME	METHOD DESCRIPTION
assert_constant	Asserts a constant value through the future year
assert_values	Asserts the future-year value and linearly extrapolates to that value
average	Uses the average value from the input vehicle table through the future year
linear	Linear extrapolation through the future year using observed data from the input vehicle table through the figure year
percentage_change	Percent change extrapolation (Vf = Vs * (1 + pct_change)^y)

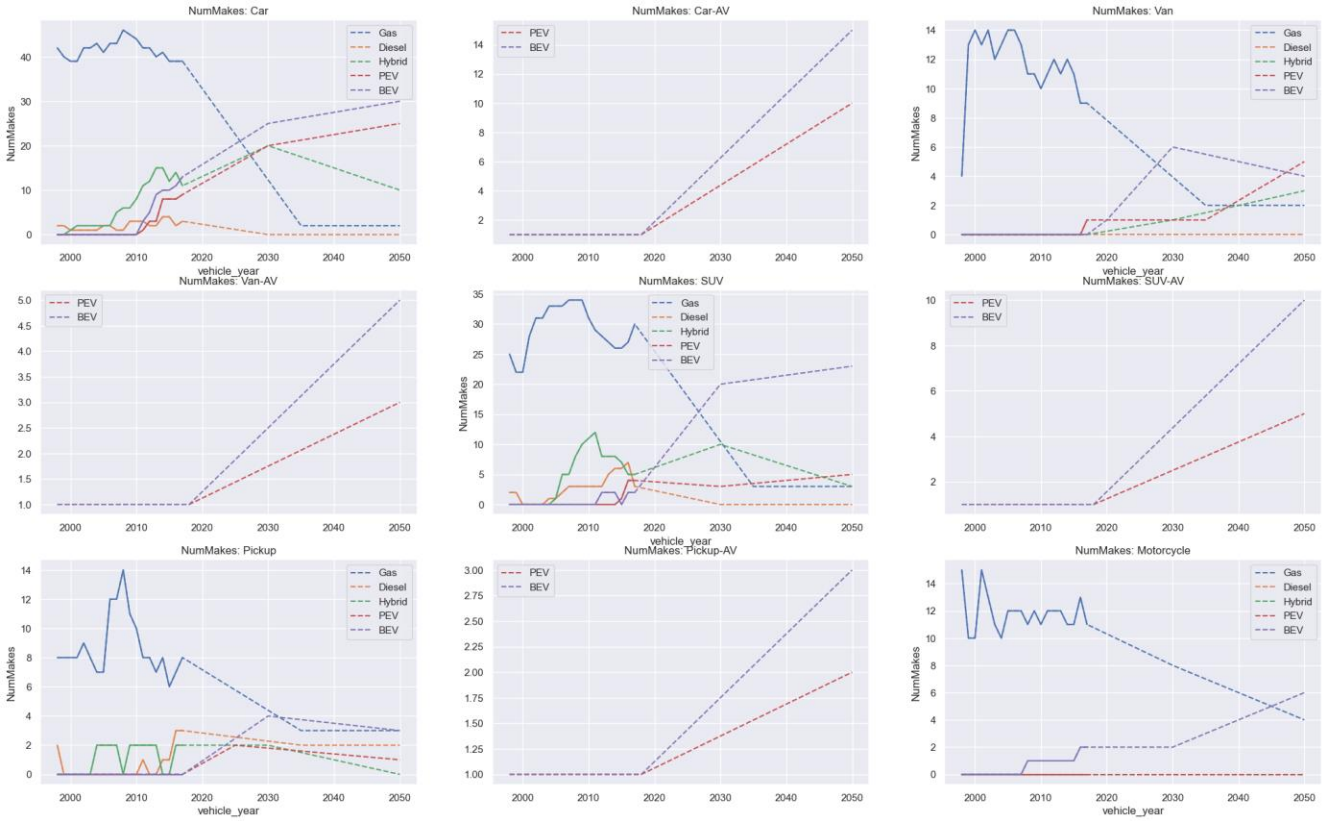


Figure 11: Example Plot From construct_veh_type_data.ipynb

7.2.5 Vehicle type choice setup

The vehicle type choice setup file, vehicle_type_choice.yaml also needs to be updated if the vehicle type assumptions have been updated. This requires updating the VEHICLE_TYPE_DATA_FILE as indicated below:

```
VEHICLE_TYPE_DATA_FILE: vehicle_type_data_extended.csv
```

In the current Gen3 Model, the vehicle_type_data.csv file used in the vehicle type choice model has already been extended to include vehicle type data between 1998 and 2050. Thus, there is no need to further extend it unless the vehicle type assumptions need to be updated.

7.3 Example scripts

7.3.1 Model crosstab script

This script imports the person and trips tables from ActivitySim and returns a table of trips by person type. The first two lines import the pandas library and the os library. The next line sets the MODEL_FOLDER variable, which is used to simplify the code. The following two lines read the person and trips CSV files. The next line (trips_ptype = …) sets an index on the persons table to person id and joins the trip table that is grouped by person id and aggregated to output the count of trips. Then, this is grouped by ptype and aggregated to sum the trips. An example output follows the code.

```
import pandas as pd
import os

MODEL_FOLDER = r"E:\Gen3_Model\2018_base\outputs\activitysim"

person = pd.read_csv(os.path.join(MODEL_FOLDER, "final_persons.csv"))
trips = pd.read_csv(os.path.join(MODEL_FOLDER, "final_trips.csv"))

trips_ptype = person.set_index("person_id").join(trips.groupby("person_id").agg(trips = ('trip_id',
'count'))).groupby('ptype').agg(trips = ('trips', 'sum'))
trips_ptype
```

Table 57: Example Script Output - Trips by Person Type Crosstab

PTYPE	TRIPS
1	2,329,534
2	290,270
3	187,006
4	446,242
5	369,753
6	110,169
7	512,477
8	200,079

7.3.2 Model calibration scripts

The following set of scripts is similar to many scripts used to calibrate ActivitySim model steps. This is written for the AV model but can be applied to many of the binary and multinomial logit models in ActivitySim.

7.3.2.1 Imports and constants

The libraries used in most of the calibration code are listed in Table 58 along with what they are used for.

Table 58: Python Libraries used in Calibration

LIBRARY NAME	LIBRARY DESCRIPTION

Pandas[77]	An open-source data analysis and manipulation tool
Os[78]	Provides many operating system functions
Numpy[79]	An open-source package for scientific computing
Matplotlib[80]	A package to produce visualizations
Seaborn[81]	A Python data visualization library based on Matplotlib

```
import pandas as pd
import os
import numpy as np
from matplotlib.pyplot import plot, show, draw, figure, cm
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib.colors import rgb2hex
```

In addition to the imports, the calibration scripts need some constants for the file location of the ActivitySim outputs, the sample rate, and the damping factor. Frequently, these are used in code as all-uppercase variables, such as below:

```
MODEL_FOLDER = "C:\\projects\\Gen3_Model\\2018_base\\outputs\\activitysim"
SAMPLE_RATE = 1.0
DAMPING = 0.75
```

7.3.2.2 File reading and manipulation

```
The next task in calibration is to read the validation data and ActivitySim o
utput into memory.
census_data = pd.read_csv("ACS_2019_5YR_DC_MD_VA_WV_Vehicles_Available.csv")
model_households = pd.read_csv(os.path.join(MODEL_OUTPUT, "final_households.c
sv"))
```

In the case above, the files being read in are CSV files. Pandas can read Excel files, DBF files can be read by using the SimpleDBF Python library, and other file types can usually be read with an additional library if not with Pandas.
Since ActivitySim Output is disaggregate and most validation data is aggregate, ActivitySim output will need to be grouped. In some cases, validation data may also need to be grouped.

7.3.2.3 Example scripts

These scripts are expected to be run in a Jupyter Notebook and are broken up into groups. The first group imports the libraries needed for the Notebook to work. This group also sets up paths, constants, and reads in the existing coefficients and model outputs.

```
import pandas as pd
import os
import numpy as np
from matplotlib.pyplot import plot, show, draw, figure, cm
import matplotlib.pyplot as plt
```

[77] https://pandas.pydata.org/
[78] https://docs.python.org/3/library/os.html
[79] https://numpy.org/
[80] https://matplotlib.org/
[81] https://seaborn.pydata.org/

```
import seaborn as sns
from matplotlib.colors import rgb2hex

MODEL_OUTPUT = r"C:\PATH"
MODEL_CONFIGS = r" C:\PATH "

TARGET_AV_PCT = {1: 0.05, 2: 0.1, 3: 0.25, 4: 0.85} # Income group : percent
COEF_DICT = {1: 'coef_av_inc1', 2: 'coef_av_inc2', 3: 'coef_av_inc3', 4: 'coe
f_av_inc4'}
# NOTE: This expects to see coef_av_inc1, coef_av_inc2, coef_av_inc3, and coe
f_av_inc4 in the coefficients file.
DAMPING = 0.75
av_coeffs = pd.read_csv(os.path.join(MODEL_CONFIGS, 'av_ownership_coeffs.csv'
))
model_hh = pd.read_csv(os.path.join(MODEL_OUTPUT, 'final_households.csv'))
```

This second group summarizes the household file by av ownership and compares it to the target percentages listed in the first code group. An example output table follows the

```
model_av_summary = model_hh[model_hh['av_ownership'] == True].groupby(['income_segment',
'av_ownership']).agg(owns_av = ('av_ownership', 'count'))
model_av_summary['model'] = model_av_summary[['owns_av']] /
model_hh.groupby('income_segment').agg(owns_av = ('income_segment', 'count'))
model_av_summary['target'] =
model_av_summary.index.get_level_values('income_segment').map(TARGET_AV_PCT)
model_av_summary
```

Table 59: Example Script Output - AV Comparison

INCOME_SEGMENT	AV_OWNERSHIP	OWNS_AV	MODEL	TARGET
1	True	632	0.044753	0.05
2	True	555	0.101910	0.10
3	True	1020	0.252538	0.25
4	True	4804	0.754753	0.85

The third code block shows the data as a chart, suitable for saving or copying and pasting into a document. This is optional. Should this code not be used, the matplotlib and seaborn imports in the first code block can be omitted. Following the code is an example of the resulting chart.

```
plot_data = model_av_summary.reset_index().melt(id_vars = 'income_segment', value_vars =
['model', 'target'], value_name = 'percent')
fig = plt.figure(figsize=(20, 6))
plot_idx = 111
plt.subplot(plot_idx)
```

```
sns.barplot(data = plot_data, x = 'income_segment', y = 'percent', hue = 'variable')
plt.title(f"AV Ownership", fontsize=18)
plt.xticks(fontsize=16, rotation = 90)
plt.yticks(fontsize=16)
plt.ylabel('Percent', fontsize=16)
plt.xlabel('Income Segment', fontsize=16)
```

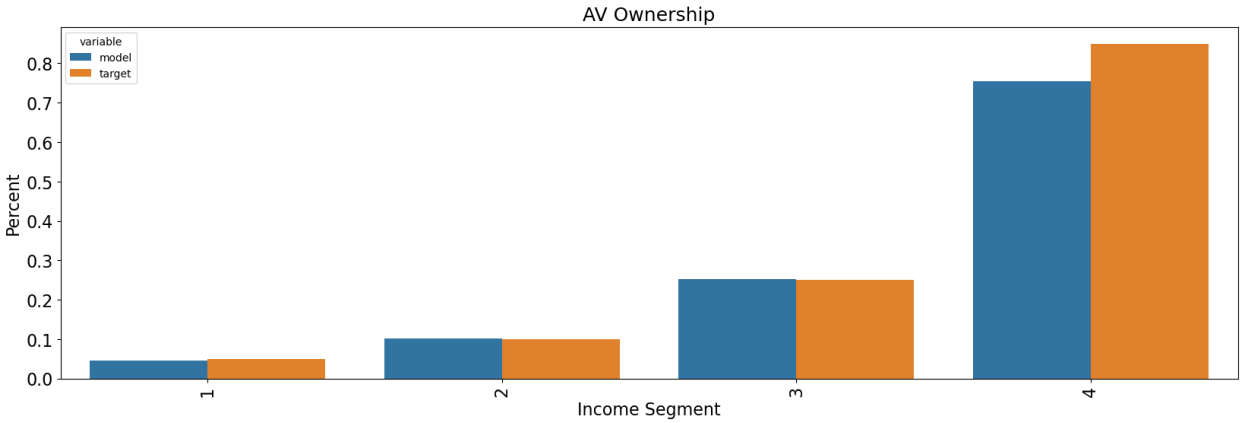


Figure 12: Example Chart Output - AV Comparison

This fourth code block computes an adjustment to the coefficients. This uses a simple method to determine the new coefficient as shown in Equation 6. In this equation, the %Target and %Model can be replaced by actual sample numbers, keeping in mind that the model must be able to attain the sample number (for example, having a target number of households that exceeds the number of households in the model will cause the constant to be continually increased).

Equation 6: Constant Calibration Equation

$$K_{new} = K_{old} + damping * ln\left(\frac{\%Target}{\%Model}\right)$$

Where:

K_{new} is the new constant
K_{old} is the old constant
damping is the damping factor
%Target is the percent of households being targeted for that class
%Model is the percent of households the model is currently predicting for that class

```
model_av_summary['income_group'] =
model_av_summary.index.get_level_values('income_segment')
model_av_summary['coefficient_name'] =
model_av_summary.index.get_level_values('income_segment').map(COEF_DICT)
model_av_summary2 = model_av_summary.merge(av_coeffs.rename(columns = {'value':
'input_coef'}), how = 'left', on = 'coefficient_name')
```

```
model_av_summary2['coef_adjust'] = np.log(model_av_summary2['target'] /
model_av_summary2['model'])
model_av_summary2['value'] = model_av_summary2['input_coef']
model_av_summary2.loc[model_av_summary2['constrain'] == 'F', 'value'] =
model_av_summary2['input_coef'] + DAMPING * model_av_summary2['coef_adjust']
model_av_summary2[['income_group', 'owns_av', 'model', 'target', 'coefficient_name', 'input_coef',
'constrain', 'coef_adjust', 'value']]
```

This final code block will update the coefficient file and output the new coefficient file. This will overwrite the existing file without prompting the user, so there is a comment on the very first line to alert the user as to what they are doing.

```
# ⚠ THIS WILL CHANGE THE COEFFICIENT FILE!!!
av_coeffs.set_index('coefficient_name', inplace = True)
av_coeffs.update(model_av_summary2[['coefficient_name', 'value']].set_index('coefficient_name'))
av_coeffs.to_csv(os.path.join(MODEL_CONFIGS, 'av_ownership_coeffs.csv'))
```

One caveat with this process is that it expects that the user will run the model after writing the new coefficients. There is no check to see if the coefficients file is newer than the output file and running these script blocks multiple times without running the model after the final group will cause this script to re-adjust the coefficients.

7.4 Miscellaneous

The items in this section are important to the functionality of the model, but not to the results of the model.

7.4.1 Changing the disk space threshold

To ensure that the model has sufficient disk space to ensure the model can complete, the model script checks disk space and will not proceed if the server has less than 500 GB disk space. This is changed by opening Gen3_Model\source\scripts\python\check_free_space.py and changing the DISK_SPACE_THRESHOLD variable near the top of the file.