|  |  |
| --- | --- |
|  | |
|  |  |
|  | |

|  |  |
| --- | --- |
| 1515 SW 5th Ave Suite 1030 Portland, Oregon 97201 **www.rsginc.com** | Image by Little Mountain 5 - CC BY-SA 3.0, via Wikimedia Commons |
| **prepared for:** |
| Oregon Department Of Transportation |
| **Submitted By:** |
| rsg |
|  |
| **In cooperation with:** |
| Parsons Brinckerhoff |

[1.0 overview 1](#_Toc492570149)

[1.1 | Overview 1](#_Toc492570150)

[1.2 | Model Components 3](#_Toc492570151)

[2.0 HARDWARE AND SOFTWARE SETUP 4](#_Toc492570152)

[2.1 | Hardware Requirements 4](#_Toc492570153)

[2.2 | Software Requirements 4](#_Toc492570154)

[2.3 | Template Model Setup 4](#_Toc492570155)

[2.4 | Running the Model 5](#_Toc492570156)

[3.0 Model Steps 6](#_Toc492570157)

[3.1 | Model Flow 6](#_Toc492570158)

[3.2 | STEP 0 – Set Properties 6](#_Toc492570159)

[3.3 | STEP 1 – Build Networks for Skimming 7](#_Toc492570160)

[TAZ Network 7](#_Toc492570161)

[TAP Network 7](#_Toc492570162)

[MAZ Network 8](#_Toc492570163)

[3.4 | STEP 2 – Create Skims using TomTom Speeds 9](#_Toc492570164)

[TAZ Skims 10](#_Toc492570165)

[MAZ Skims 11](#_Toc492570166)

[TAP Skims 11](#_Toc492570167)

[3.5 | STEP 3 – Run the Zone Checker 13](#_Toc492570168)

[3.6 | STEP 4 – Run the Commercial Vehicle Model 13](#_Toc492570169)

[3.7 | STEP 5 – Run the External Model 13](#_Toc492570170)

[3.8 | START Feedback Loop 14](#_Toc492570171)

[3.9 | STEP 6 – Run OR-RAMP ABM 14](#_Toc492570172)

[3.10 | STEP 7 – Build Trip Demand Matrices 19](#_Toc492570173)

[3.11 | STEP 8 – Create Skims and Assignments Using Congested Speeds 20](#_Toc492570174)

[TAZ Skims 20](#_Toc492570175)

[TAP Skims 21](#_Toc492570176)

[3.12 | STEP 9 – Check for Completion 21](#_Toc492570177)

[4.0 Inputs 22](#_Toc492570178)

[4.1 | Networks and Zones 22](#_Toc492570179)

[4.2 | Population Synthesizer 32](#_Toc492570180)

[4.3 | CVM and External Model 35](#_Toc492570181)

[4.4 | OR-RAMP Properties File 36](#_Toc492570182)

[4.5 | UEC Files 69](#_Toc492570183)

[4.6 | Major UniversIty Model Parking Lots (Optional) 74](#_Toc492570184)

[4.7 | Costs 74](#_Toc492570185)

[5.0 Outputs 76](#_Toc492570186)

[5.1 | Network and Zone Data 76](#_Toc492570187)

[5.2 | CVM and External Model 80](#_Toc492570188)

[5.3 | Skims 81](#_Toc492570189)

[5.4 | Trip Lists 83](#_Toc492570190)

[5.5 | Demand Matrices 91](#_Toc492570191)

[5.6 | Assignments 92](#_Toc492570192)

[6.0 Creating and Running a New Scenario 95](#_Toc492570193)

[7.0 Model Logging/Trace Results 96](#_Toc492570194)

[8.0 Methodology for developing TAZ boundaries 98](#_Toc492570195)

[8.1 | Recognizing boundaries 98](#_Toc492570196)

[Census Boundaries 98](#_Toc492570197)

[Physical Boundaries 98](#_Toc492570198)

[Jurisdictional/Political/Planning Boundaries 98](#_Toc492570199)

[Historical Zonal Boundary 98](#_Toc492570200)

[8.2 | Transportation Network 98](#_Toc492570201)

[8.3 | Homogenous Land-use 99](#_Toc492570202)

[8.4 | Uniqueness 99](#_Toc492570203)

[8.5 | Centroids 99](#_Toc492570204)

[8.6 | Anticipating Developments 99](#_Toc492570205)

[9.0 Methodology for developing MAZ boundaries 100](#_Toc492570206)

[9.1 | Procedure to create MAZs 100](#_Toc492570207)

[10.0 Methodology for developing TAPs 103](#_Toc492570208)

[10.1 | Procedure to create TAPs 103](#_Toc492570209)

[11.0 SOURCE OF LAND-USE INputs 105](#_Toc492570210)

[12.0 Network Revisions and Maintenance 110](#_Toc492570211)

[12.1 | Non-Motorized Network 110](#_Toc492570212)

[12.1 | Issues to Be Aware of When Editing the Network 110](#_Toc492570213)

**List of Figures**

[Figure 1 - TAZs, MAZS, and TAPS 2](#_Toc492570214)

[Figure 2 - Transit Virtual Path Building 3](#_Toc492570215)

[Figure 3 - Template Model Directory 4](#_Toc492570216)

[Figure 4 - TomTom Data Integration 10](#_Toc492570217)

[Figure 5 - SWIM Flows ENTERING SOUTHERN OREGON at STATION 1 (I-5 North) 13](#_Toc492570218)

[Figure 6 - OR-RAMP MoDel Flow 15](file:///C:\projects\development\soabm\SOABM_UserGuide.DOCX#_Toc492570219)

[Figure 7 – OR-RAMP Travel Demand Model Software Components 18](#_Toc492570220)

[Figure 8 - TomTom Speed Links 23](#_Toc492570221)

[Figure 9 - VISUM Highway NeTWork 24](#_Toc492570222)

[Figure 10 - Medford Highway Network 24](#_Toc492570223)

[Figure 11 – Grants PAss Highway Network 25](#_Toc492570224)

[Figure 12 - VISUM Stop Model 25](#_Toc492570225)

[Figure 13 – Grants Pass TRANSIT Network 26](#_Toc492570226)

[Figure 14 – MeDford TRANSIT Network 26](#_Toc492570227)

[Figure 15 - PopSyn Households File EXample 32](#_Toc492570228)

[Figure 16 - PopSYn Persons File Example 34](#_Toc492570229)

[Figure 17 - UEC Example 70](#_Toc492570230)

[Figure 18 - InDivTour Data Example 83](#_Toc492570231)

[Figure 19 - INDIVTrip Data Example 85](#_Toc492570232)

[Figure 20 - JointTour DATA Example 88](#_Toc492570233)

[Figure 21 - JOINTTRIP DATA EXAMPLE 89](#_Toc492570234)

[Figure 22 - Demand MAtrices in OMX Format 91](#_Toc492570235)

[Figure 23 - TAZ Assignment REsults 92](#_Toc492570236)

[Figure 24 - TAP Assignment Results 93](#_Toc492570237)

[Figure 25 - RunModel.Bat 95](#_Toc492570238)

[Figure 26 - Household Trace Results 97](#_Toc492570239)

[Figure 27 – Network Shortest PATH Search CheCk 111](#_Toc492570240)

[Figure 28 - Line Route Run and Dwell Times 112](#_Toc492570241)

**List of Tables**

[Table 4.1 Link Attributes 27](#_Toc492570242)

[Table 4.2 MAZ (MainZone) Attributes 28](#_Toc492570243)

[Table 4.3 LiNERoute Attributes 31](#_Toc492570244)

[Table 4.4 TAZ (Zone) Attributes 31](#_Toc492570245)

[Table 4.5 TAP (STOPAREA) Attributes 31](#_Toc492570246)

[Table 4.6 Time-oF-Day periods and Link Capacities 32](#_Toc492570247)

[Table 4.7 PopSYN Households 32](#_Toc492570248)

[Table 4.8 PopSYN Persons 34](#_Toc492570249)

[Table 4.9 CVM Specific Inputs 35](#_Toc492570250)

[Table 4.10 SWIm External MODEL SpeciFIC Inputs 36](#_Toc492570251)

[Table 4.11 or-ramp pROPERTIES fiLE 36](#_Toc492570252)

[Table 4.12 uec fiLES 72](#_Toc492570253)

[Table 4.13 Parking Lots For Major Universities 74](#_Toc492570254)

[Table 4.14 Costs 75](#_Toc492570255)

[Table 5.1 MAZ Data File Export from VISUM – Inputs\maz\_data\_export.csv 76](#_Toc492570256)

[Table 5.2 TAZ Assignment Speeds for Transit ASsignment - taz\_skim\_<TOD> \_speed\_linkspeeds.csv 79](#_Toc492570257)

[Table 5.3 TAPS Input File FOR OR-RAMP - tap\_data.csv 79](#_Toc492570258)

[Table 5.4 Tap Lines CSV File for OR-RAMP - tapLines.csv 79](#_Toc492570259)

[Table 5.5 park Locations Alternatives for OR-RAMP - ParkLocationAlts.csv 80](#_Toc492570260)

[Table 5.6 Destination ChOICE Alternatives for OR-RAMP - DestinationChoiceAlternatives.csv 80](#_Toc492570261)

[Table 5.7 Sample of Alternatives TAZ Distances Alternatives for OR-RAMP - SoaTazDistAlternatives.csv 80](#_Toc492570262)

[Table 5.8 park Location sampling Alternatives for OR-RAMP - ParkLocationSampleAlts.csv 80](#_Toc492570263)

[Table 5.9 CVM Outputs 80](#_Toc492570264)

[Table 5.10 SWIm External MODEL Outputs 81](#_Toc492570265)

[Table 5.11 TAP Skim Matrices - tap\_skim\_<TOD> \_set<SETID>.omx 81](#_Toc492570266)

[Table 5.12 MAZ to MAZ Distances (Bike and WALK) - maz2maz\_<Bike|WALK>.csv 81](#_Toc492570267)

[Table 5.13 TAP to MAZ Distances (Bike and WALK) - tap2maz\_<BIKE|Walk>.csv 81](#_Toc492570268)

[Table 5.14 TAZ to NEAR TAPS ImpedanceS for DRIVE TRANSIT - drive\_taz\_tap.csv 82](#_Toc492570269)

[Table 5.15 TAZ Skim Matrices - taz\_skim\_<DSEG> \_<TOD>.omx 82](#_Toc492570270)

[Table 5.16 TAZ Skim Definitions 82](#_Toc492570271)

[Table 5.17 Individual TouRS File - indivTourData\_<iteration>.csv 83](#_Toc492570272)

[Table 5.18 InDvidUal Trips File - indivTripData\_<iteration>.csv 85](#_Toc492570273)

[Table 5.19 Joint TourS File - jointTourData\_<iteration>.csv 88](#_Toc492570274)

[Table 5.20 JoiNT Trips File - jointTripData\_<iteration>.csv 89](#_Toc492570275)

[Table 5.21 Tour and TRIP Modes COdes 90](#_Toc492570276)

[Table 5.22 Demand Matrices 91](#_Toc492570277)

[Table 5.23 Common ASSignMENT OutputS - taz\_skim\_<tod>\_speed.ver, tap\_skim\_<tod>\_speed\_set<setid>.ver 93](#_Toc492570278)

[Table 9.1 Thresholds used for creating MAZ system for MORPC 101](#_Toc492570279)

[Table 10.1 Thresholds used for creating MAZ system for MORPC 104](#_Toc492570280)

[Table 11.1 MAZ Inputs and Their Source 105](#_Toc492570281)

# overview

## Overview

The Oregon Department of Transportation (ODOT) Southern Oregon Activity-Based Travel Demand Model (ABM) is a new travel demand model for the Middle Rogue and Rogue Valley MPOs. The new modeling system is based on the CT-RAMP family of ABMs and includes the 50,000+ person Grants Pass and 175,000+ person Medford urban areas.

Compared to traditional trip-based models, the model system has more detailed and accurate representation of space, time, travel patterns, and significantly more person and context-based explanatory variables. Currently the new model better models non-motorized travel, time-of-day, ride sharing, non-home-based travel, accessibility effects, and provides a flexible household travel survey-like database for custom summaries. The new modeling system was also developed as the eventual framework for exploring new policy issues: new vehicle types and emissions, parking and different pricing scenarios, connected and automated vehicles, vehicle ownership moving to service, light-weight vehicle infrastructure, telecommuting, and others. ODOT sees modeling and information needs headed in this direction, and the new microsimulation-based SOABM is the best tool available.

The new person travel demand model is based on the CT-RAMP (Coordinated Travel Regional Activity-Based Modeling Platform) family of ABMs. The CT-RAMP framework, which is fully described in the model specification, adheres to the following basic principles:

* The CT-RAMP design attempt to model individual travel choices with maximum behavioral realism. In particular, it addresses both household-level and person-level travel choices including intra-household interactions between household members.
* Operates at a detailed temporal (half-hourly) level, and considers congestion and pricing effects on time-of-day and peak spreading.
* Reflects and responds to detailed demographic information, including household structure, aging, changes in wealth, and other key attributes.
* Is implemented in the Common Modeling Framework (CMF), an open-source library specifically for implementing advanced travel demand models.
* Offers sensitivity to demographic and socio-economic changes observed or expected in the dynamic Southern Oregon metropolitan region. This is ensured by the enhanced and flexible population synthesis procedures as well as by the fine level of model segmentation. In particular, the model incorporates different household, family, and housing types including a detail analysis of different household compositions in their relation to activity-travel patterns.

The model integrates with the ODOT Statewide Integrated Model (SWIM) for external flows, includes a commercial vehicle model, uses open matrix (OMX) for data interchange, Google transit for transit routes, TomTom for observed speed data, and depends on VISUM for all network level-of-service information and zonal data management.

Activity-based models can exploit more explicit geographic and location information, but the advantages of additional spatial detail must be balanced against the additional efforts required to develop zone and associated network information at this level of detail, as well as against the increases in model runtime associated primarily with path-building and assignment. The use of a spatially disaggregate zone system helps ensure appropriate model sensitivity. Use of large zones may produce aggregation biases, especially in destination choice, where the use of aggregate data can lead to illogical parameter estimates due to reduced variation in estimation data, and in mode choice, where modal access may be distorted. Smaller zones help minimize these effects, and can also support more detailed network assignments. Strategies to address the modal access limitations of large zones through the use of transit sub-zonal procedures are discussed in the transit network section of this document.

The new model has three zone systems – microzones (MAZs) for modeling non-motorized travel such as walk, bike, and transit access/egress, Transportation Analysis Zones (TAZs) for auto travel, and transit access points (TAPs), also called ‘stop areas’, for modeling transit trips from stop to stop, as shown in Figure 1 . All transit access and egress calculations and ultimate transit paths from origin MAZ to destination MAZ through boarding and alighting TAPs are computed within OR-RAMP, and rely upon detailed geographic information regarding MAZ-TAP distances and accessibilities. A graphical depiction of the MAZ – TAP transit calculations is given in Figure 2.

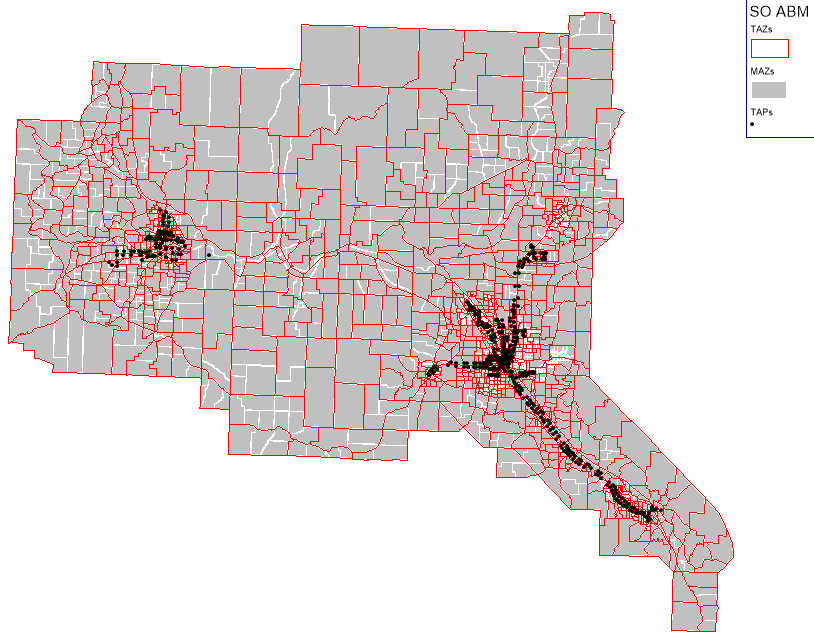


Figure - TAZs, MAZS, and TAPS

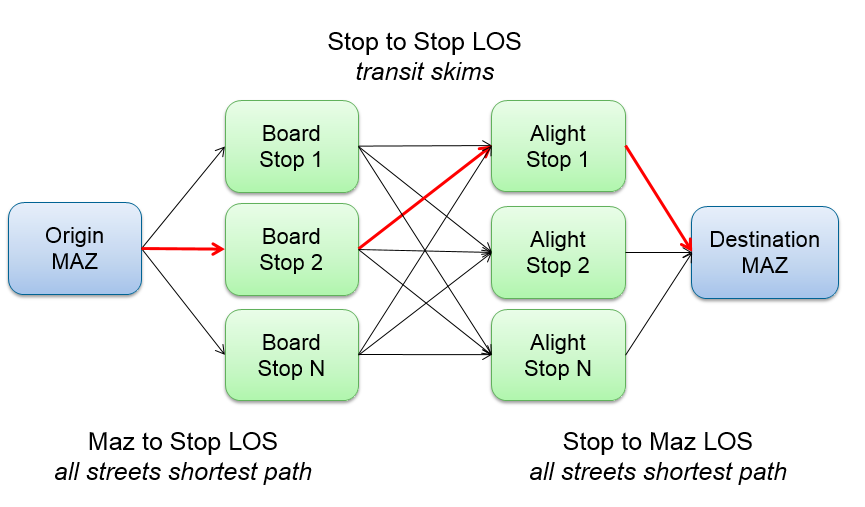


Figure - Transit Virtual Path Building

This document describes how to setup and run the model, key inputs, and key outputs. For more information about the model components, including their design and validation/calibration, refer to other technical memos developed as part of this effort.

## Model Components

The SOABM is a collection of travel modeling components. The core components of the system are:

* OR-RAMP – ODOT’s version of the CT-RAMP family of ABMs for modeling resident travel.
* VISUM + Python – Zone and network data management, as well as network skimming and assignment procedures.
* Commercial vehicle model – ODOT’s trip-based commercial vehicle model implemented in R.
* External model – ODOT’s external travel model based on select link analysis flows from SWIM at each SOABM external station location.
* RunModel – A DOS batch program for running the overall model system.

# HARDWARE AND SOFTWARE SETUP

## Hardware Requirements

The SOABM can be run on a typical Windows 64-bit machine, with the following settings:

* At least 24GB of RAM is required to run the base year version of the model
* At least 4 processors – additional cores will reduce runtime
* At least 10GB of free storage – a single model run currently takes around 7GB

## Software Requirements

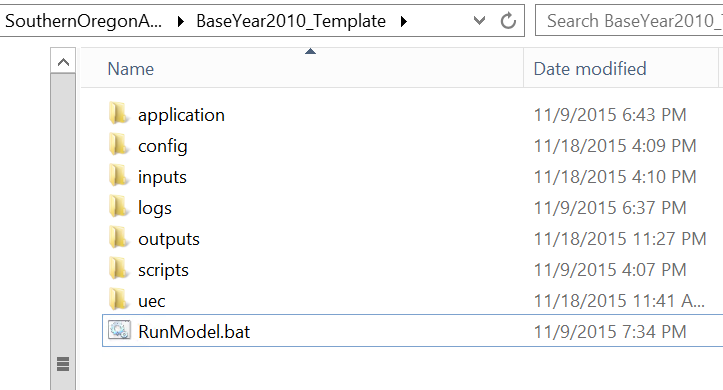
The model is configured to run on Windows, and uses VISUM, Python, Java, R, and SQL Server. To ease installation and setup, complete installations of Python, Java, and R are included in the model setup and are used when running the model. Python is VISUM’s preferred scripting tool, Java is used for CT-RAMP, and R is used for the commercial vehicle model and external model. The installations include the third-party libraries required as well.

VISUM is a commercial transport modeling software package developed by PTV. 64bit VISUM 16 is required and the latest installer is available here: <http://cgi.ptvgroup.com/cgi-bin/en/traffic/visum_download.pl>. The free Microsoft SQL Server Express is required for the population synthesizer. Refer to the Population Synthesizer technical memo for detailed installation instruction. When installing SQL Server Express, make sure to enable TCP/IP via Start + All Programs + SQL Server Configuration Manager, then SQL Server Network Configuration and then “Protocols” and then “TCP/IP” and set to enable. This will ensure that the population synthesizer Java program can find the database.

## Template Model Setup

The template model setup is maintained here: <https://github.com/RSGInc/SOABM> and contains seven subfolders, as shown in Figure 3. In addition, the template model directory contains the RunModel.bat file that is used to launch a full model run.

Figure - Template Model Directory



To setup a new scenario, check out the repository to your local machine using Git. Instructions for downloading and setting up the template scenario are included with the download. On initial checkout, expand the dependencies.zip to a folder called dependencies. This includes installs of Java, Python, and R. Every scenario is contained within its own folder, with a unique name. The folder name is the same as the scenario name. The dependencies folder must be next to the scenario folder since it is referenced in the RunModel.bat via ..\dependencies.

For privacy reasons, the repository does not include the input files. Make sure to copy a set of inputs to the inputs folder before running the model.

If the amount of RAM or the number of processing threads for CT-RAMP needs to be updated (because the population increased or the run machine changed) then the following settings should be reviewed:

* application\runHhMgr.cmd – near the bottom of the script, the START … -Xmx10000m setting specifies how much RAM is allocated to the household data manager
* application\runMtxMgr.cmd – near the bottom of the script, the START … the -Xmx10g setting specifies how much RAM is allocated to the matrix data manager
* application\runORRAMP.cmd – near the bottom of the script, the java … the –Xmx40g setting specifies how much RAM is allocated to the main process
* config\jppf-clientLocal.properties – at the bottom of the file, the jppf.local.execution.threads = 10 setting specifies how many threads are allocated to CT-RAMP

## Running the Model

To run the model, open a DOS command window in the scenario folder and run RunModel.bat. The DOS bat file now determines the machine IP address and knows the location of the included Java, Python, and R installs in order to automatically configure the model setup.

# Model Steps

## Model Flow

The SOABM is run via the RunModel.bat DOS batch file. The model system has been developed as a ‘turnkey’ system; the user has only to code the model inputs and run the model through the batch file. It will run all model components, iterate the system, and write all outputs. The batch file performs the following sequential steps:

1. Set properties
2. Build networks for skimming in VISUM + Python
   1. TAZ
   2. MAZ
   3. TAP
3. Create skims using TomTom speeds in VISUM + Python
   1. TAZ
   2. MAZ
   3. TAP
4. Run the CVM in R
5. Run the external model in R
6. Start a feedback loop up to max iterations
   1. Run OR-RAMP ABM in Java
   2. Build TAZ and TAP trip demand matrices for assignment in Python
   3. Create skims and assignments using congested speeds in VISUM + Python
      1. TAZ
      2. TAP
   4. Check for completion and repeat feedback loop if needed

The subsequent sections describe these steps in detail.

## STEP 0 – Set Properties

Before running any modeling procedures, software paths and scenario properties are set for the model run. These settings are automatically determined in the latest version of the script and are logged to the console. This allows a model setup (i.e. scenario) to be easily copied from one scenario to another, and from one machine to another.

## STEP 1 – Build Networks for Skimming

The first step in the model process is to build the TAZ, TAP, and MAZ networks for skimming based on the input master VISUM version – SOABM.ver.

### TAZ Network

The first network created is the TAZ network, which is used for auto skims. This network is created using the *taz\_initial* entry point in the SOABM.py Python script. The taz\_initial call of the script does the following:

1. Starts VISUM via the Python win32 COM server
2. Loads the inputs/SOABM.ver master version file
3. Codes TAZs connectors. Coding TAZ connectors starts by finding all candidate nodes, which are network nodes that serve arterial and collector link facility types (see Table 4.1). Next, the procedure creates a TAZ connector for up to the four closest candidate nodes and calculates the travel time at 25 mph. The final step is to go back through the set of connectors generated and remove any that are too close to one another. Connectors are deemed too close if their network nodes are within 500 feet of one another. This procedure generated a reasonable set of TAZ connectors that ODOT further edited. As a result of the manual revisions to the connectors, the codeTAZConnectors() function that implements this step is currently commented out in the script.
4. Save the version file for TAZ skimming to outputs\taz\_skim\_initial.ver
5. Closes VISUM

### TAP Network

The second network created is the TAP network, which is used for transit skims from boarding TAP through any potential transfer TAPs, to alighting TAP. This network is created using the *tap\_initial* entry point in the SOABM.py Python script. The tap\_initial call of the script does the following:

1. Starts VISUM via the Python win32 COM server
2. Loads the inputs/SOABM.ver master version file
3. Assigns VISUM stops areas (TAPs) to transit access nodes so trips can get from TAPs to the network. The results are essentially transit access connectors. The procedure selects all candidate network nodes (all links except for freeways and ramps) and then assigns each stop area to the closest candidate node by setting the StopArea’s NodeNo attribute.
4. Switches the VISUM Zone objects from TAZs to TAPs. The initial SOABM version file has Zones coded as TAZs and transit stop areas coded as TAPs. In order to generate TAP to TAP skims, the procedure deletes the TAZ zones and creates TAP zones. Each stop area X and Y is used to create a new zone X and Y centroid. Then a zone connector is created to the previously defined transit access node. The travel time is calculated at 3mph.
5. Save the version file for TAP skimming to outputs\tap\_skim\_initial.ver
6. Creates the TAP lines file – outputs\tapLines.csv. The TAP lines file (Table 5.4) contains a list of the line routes served at each TAP (stop area). This file is used by CT-RAMP to trim the available set of boarding and alighting stops to consider when doing transit virtual path building. Only stops that are further away and serve new lines are considered when building walk-transit paths.
7. Creates the fare matrix – outputs\fare.omx. Fares are modeled as TAP to TAP OD and assigned using the nearest fare district centroid. The fare district centroids were placed so all Grants Pass OD pairs were assigned the Grants Pass fare, while all Rogue Valley OD pairs were assigned the Rogue Valley fare. The fare district centroids are below:
   1. Grants Pass, 100 cents, x=548578, y=263859
   2. Rogue Valley, 200 cents, x=670480, y=226394
8. Closes VISUM

### MAZ Network

The third network created is the MAZ network, which is used to calculate non-motorized times and distances, including access and egress walking times between MAZs and boarding\alighting TAPs, and walk\bike times between close MAZs. Drive times to TAPs are calculated after TAZ skims are created in the next model step. This network is created using the *maz\_initial* entry point in the SOABM.py Python script. The maz\_initial call of the script does the following:

1. Starts VISUM via the Python win32 COM server
2. Loads the inputs/SOABM.ver master version file
3. Assigns VISUM stops areas (TAPs) to transit access nodes so trips can get from TAPs to the network. The results are essentially transit access connectors. The procedure selects all candidate network nodes (all links except for freeways and ramps) and then assigns each stop area to the closest candidate node by setting the StopArea’s NodeNo attribute. This is required for creating the MAZ to TAP walk access distance files.
4. Switches the VISUM Zone objects from TAZs to MAZs. The initial SOABM version file has Zones coded as TAZs and Mainzones coded as MAZs. In order to generate MAZ to MAZ/TAP distances, the procedure deletes the TAZ zones and creates MAZ zones. Each mainzone X and Y is used to create a new zone X and Y centroid. Then up to four zone connectors are created to the closest candidate nodes. Candidate nodes are only local and collector facility type links. Finally, all mainzone user-defined attributes are copied over to the zone attributes, including the zone polygons as well-known-text[[1]](#footnote-1) (WKT).
5. Calculates CT-RAMP MAZ density measures as zone attributes. The attributes are:
   1. DUDEN – total MAZ households / total MAZ acres within a ½ mile of each MAZ centroid
   2. EMPDEN – total MAZ employment / total MAZ acres within a ½ mile of each MAZ centroid
   3. TOTINT – total intersections within a ½ mile of each MAZ centroid
   4. POPDEN – total MAZ population / total MAZ acres within a ½ mile of each MAZ centroid
   5. RETDEN – total MAZ retail employment / total MAZ acres within a ½ mile of each MAZ centroid
6. Save the version file for MAZ skimming to outputs\maz\_skim\_initial.ver
7. Creates the CT-RAMP sub-model destination alternative files in the output folder.
   1. ParkLocationAlts.csv – Alt, MAZ, ParkArea type (see Table 5.5)
   2. DestinationChoiceAlternatives.csv - Alt, MAZ, TAZ (see Table 5.6)
   3. SoaTazDistAlternatives.csv – Alt, TAZ (see Table 5.7)
   4. ParkLocationSampleAlts.csv – Alt, MAZ (see Table 5.8)
8. Writes the MAZ land use data output file to inputs\maz\_data\_export.csv. This is the key land use input file to CT-RAMP and contains the fields in Table 5.1.
9. Closes VISUM

## STEP 2 – Create Skims using TomTom Speeds

The next step in the model is to create skims using the TomTom link speed data. As shown in Figure 4 below, the VISUM master version file includes TomTom link speeds by time-of-day (TOD).

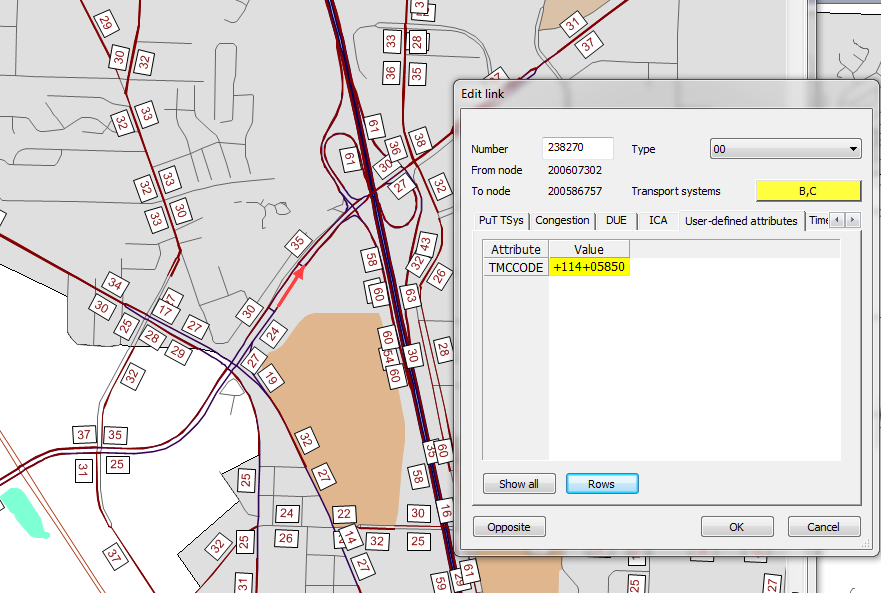


Figure - TomTom Data Integration

### TAZ Skims

The first set of skims created are the TAZ (auto) skims. The TAZ skims are created by calling the taz*\_skim\_speed* entry point in the SOABM.py Python script. The taz\_skim\_speed call of the script does the following by time-of-day:

1. Starts VISUM via the Python win32 COM server
2. Loads the outputs/taz\_skim\_initial.ver version file
3. Loads and run the TAZ skimming procedures stored in config/visum/taz\_skim\_<TOD>\_speed.par. The procedures do the following:
   1. Set time-of-day link number of lanes based on the generic number of lanes
   2. Set time-of-day link capacity based on number of lanes (LANES\_<TOD>, capacity per lane (LANECAP), and time-of-day (NETWORK\TOD\_FACTOR\_<TOD>)
   3. Set time-of-day link speed using <TOD>\_SPEED
   4. Calculates and output skims in OMX format, as defined in Table 5.15.
   5. The procedures define the link generalized cost by demand segment and volume-delay functions under Calculate + Procedures + General Procedure Settings + PrT Settings + Impedance & Volume-delay functions as:

GenCost = 1 \* Tcur\_PrTSys(<DSEG>) + (3600/<VOT>) \* Toll\_PrTSys(<DSEG>)

VOT = Value-of-time, currently $25/hour; The value is coded as 144 since 3600/25 = 144.

VDF = BPR(0.2, 10); connectors have fixed travel times

1. Save the version file, which includes the skims, to outputs\taz\_skim\_<TOD> \_speed.ver
2. Creates the TAZ to TAP drive access time information for drive transit impedance. The procedure gets the SOV time, distance, and toll skims, as well as the list of TAPs with CANPNR=1 (true). It then assigns each TAP centroid to the nearest TAZ centroid. This information is written to the outputs/tap\_data.csv file required by CT-RAMP. The final step is to loop through all TAZs and TAPs with CANPNR=1 and write data for the interchanges that are within the max distance. The procedure has a different max distance by transit sub-mode, which is currently set to 4 miles for local bus. The results are written to outputs/drive\_taz\_tap.csv. See Table 5.3 and Table 5.14 for more information on the output files.
3. Closes VISUM

### MAZ Skims

The second set of “skims” created are the MAZ (non-motorized) skims. The MAZ skims are created by calling the maz*\_skim* entry point in the SOABM.py Python script. The maz\_skim call of the script does the following for bike and then for walk:

1. Starts VISUM via the Python win32 COM server
2. Loads the outputs/maz\_skim\_initial.ver version file
3. Creates the MAZ to TAP output files – outputs/tap2maz\_<walk|bike>.csv (see Table 5.13). The procedure loops by TAP node and runs the VISUM Isochrone procedure to code the travel distance from the TAP node to all network elements (including MAZ centroids). The setting by mode are:
   1. Walk - 3 mph and 2 miles max
   2. Bike – 10 mph and 5 miles max
4. Loads and run the MAZ to MAZ skimming procedures stored in config/visum/maz\_skim\_<walk|bike>.par. The output distance skim is stored in the version file. The walk network is all links with walk in the TSysSet. The bike network is all links with bike in the TSysSet.
5. Creates the nearby MAZ file by looping through cells in the MAZ to MAZ skim matrix and writing to the output CSV file all OD pairs less than the maximum distances. The output file is outputs/maz2maz\_<walk|bike>.csv (as defined in Table 5.12).
6. Save the version file, which includes the swalkkims, to outputs\maz\_skim\_<walk|bike>.ver
7. Closes VISUM

### TAP Skims

The third set of skims created are the TAP (transit stop-to-stop) skims. The TAP skims are created by calling the *tap\_skim\_speed* entry point in the SOABM.py Python script. The tap\_skim\_speed call of the script does the following by time-of-day and skim set:

1. Starts VISUM via the Python win32 COM server
2. Loads the outputs/taz\_skim\_<TOD>\_speed.ver version file and writes the link speeds to a CSV file for updating the transit run times - outputs/taz\_skim\_<TOD>\_speed\_linkspeeds.csv. The link speed used in the VCur\_PrTSys(HOV3Toll) attribute (see Table 5.2).
3. Loads the outputs/tap\_skim\_initial.ver version file and loads the link speeds written out earlier into the link V0PrT attribute in order to set the transit runtimes as a function of the highway congested link times.
4. Loads and runs the config/visum/tap\_skim\_speed\_<TOD>.par procedure file to calculate the transit run and dwell times and skims. The transit run and dwell time functions, plus the path-finding settings are below:
   1. Link runtime for freeways = tCur / 0.9
   2. Link runtime for other facilities = tCur / 0.75
   3. Node dwell time = 120 seconds per mile traveled
   4. Path finding choice using ‘optimal strategies’
   5. Path finding weights for set 1 = IVT=1, WLK=2, OWT=2, TWT=2, NTR=10min
   6. Path finding weights for set 2 = IVT=1, WLK=2, OWT=2, TWT=2, NTR=20min
   7. Path finding weights for set 3 = IVT=1, WLK=2, OWT=2, TWT=2, NTR=20min (same as set2 for now)
   8. Headway attribute = LineRoute\<TOD>\_HEADWAY (seconds)
   9. Transit walk links are permitted and max walk time is set to 30 minutes
   10. Uses 0.0001 for one OD pair in order to trick VISUM to run assignment and skimming
5. Writes the skims to OMX files using the config/visum/tap\_skim\_<TOD>\_set<SETID>.par file. The skims are defined in Table 5.11.
6. Save the version file, which includes the skims, to outputs/tap\_skim\_<TOD>\_speed\_set<SETID>.ver
7. Updates the fare matrix created in the skimming procedure (which is 0) with the custom fare matrix created earlier – outputs\fare.omx
8. Revises skims across skim sets with duplicate levels-of-service in order to avoid CT-RAMP seeing duplicate service. This is done by setting all skims to 0 if the total time in an OD pair (IVT+OWT+TWT+WKT) is the same across skim sets. This is done only in the OMX files and not in the skims in the VISUM version files.
9. Closes VISUM

## STEP 3 – Run the Zone Checker

The next step in the model is to run the zone checker and updater. This script is run via Python and is called by running the scripts\zoneChecker.py script. The script renumbers the synthetic households’ MAZ to a sequential MAZ based on the MAZ data input file output by VISUM and also checks to ensure that all households have a home MAZ that is also in the MAZ data file.

## STEP 4 – Run the Commercial Vehicle Model

The next step in the model is to run the commercial vehicle model (CVM). The CVM model is run via RSCRIPT and is called by running the scripts\cvm.R script. The CVM model input and output files are described in Table 5.9. The CVM is a trip-based truck model that produces trips by car, single-unit truck, and multi-unit truck by time-of-day. For more information on the CVM, refer to the CVM model documentation.

## STEP 5 – Run the External Model

The next step in the model is to run the SWIM external model. The SWIM external model is run via RSCRIPT and is called by running the scripts\externalModel\_SWIM.R script. The SWIM external model input and output files are described in Table 5.10. The SWIM external model, based on the Oregon statewide integrated model runs select link analysis in the SWIM assignment at each SOABM external station in order to build traversal matrices by mode and time-of-day for the SOABM external flows (as shown for example in Figure 5). For more information on the SWIM external model, refer to the SWIM external model documentation.

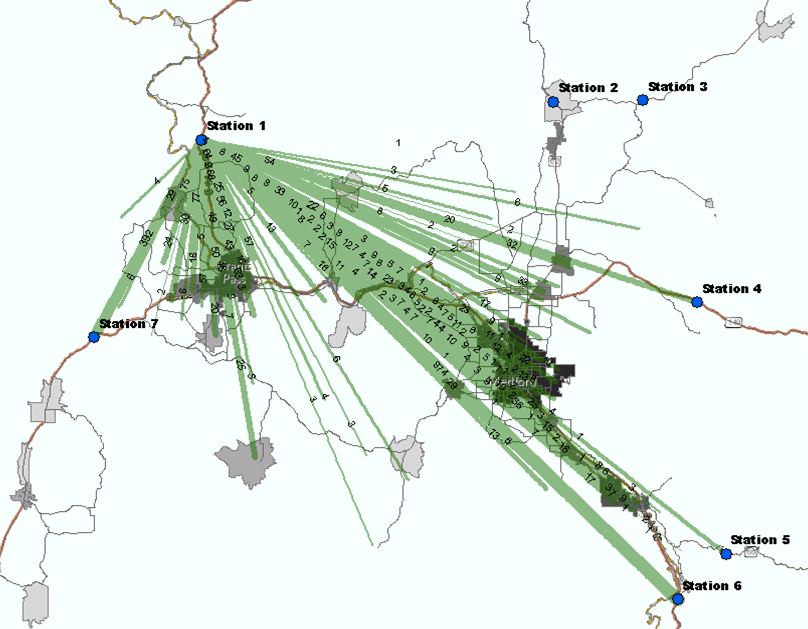


Figure - SWIM Flows ENTERING SOUTHERN OREGON at STATION 1 (I-5 North)

## START Feedback Loop

This step starts the overall model feedback loop which runs up to iterations equal to %MAX\_ITER%, set in Step 0. The feedback loop is used to equilibrate the person travel demand models and the network capacity models. First, the model generates estimates of person trips. The person trips and auxiliary trips (commercial trips and external trips) are then loaded onto the network. As a result, congestion occurs on select links and new OD travel times are calculated using the link volume-delay functions. After generating new travel time matrices, the person travel models are re-run and new destinations, modes, etc. are chosen and the trips re-loaded onto the network. This process is repeated until an acceptable level of convergence is achieved.

The model makes use of assignment warm starting in order to smooth convergence. The VISUM highway assignment procedure files - config/visum/taz\_skim\_<TOD>.par - are set to use the existing paths in the version file if available as the starting point for assignment.

## STEP 6 – Run OR-RAMP ABM

OR-RAMP, the Oregon DOT version of CT-RAMP, is run in this model step. The procedure starts by instantiating three Java processes:

1. Household data server to store all households and their persons, tours, and trips
2. Matrix data server to store all the TAZ and TAP matrices required for the model
3. The main model process that starts the model run, reads in the households, runs each household through the synthetic travel diary creation process, and then writes out tour and trip files for each household.

For SOABM, all three processes are run on the same machine. OR-RAMP can be setup to run on multiple machines if required (i.e. faster runtimes and/or more RAM is required).

The OR-RAMP model flow chart is shown in Figure 6. The model starts with long-term and mobility choice models such as auto ownership and usual work and school location choice. Next, the model simulates a typical day of travel for each individual in the region. It starts with day pattern – whether a person stays at home on the simulation day, makes mandatory travel (work and/or school), and/or non-mandatory travel. For each type of travel, it then generates tours and assigns all the required attributes – purpose, destination, time-of-day, and mode. After generating tours, each tour is processed to assign the number of stops along the tour, their purpose, location, and time-of-day. These stops are then processed as trips to assign mode and parking location. Finally, trips are aggregated into demand matrices for assignment at the TAZ and TAP level in VISUM

Each model component is described briefly below.



Figure - OR-RAMP MoDel Flow

.

1. Population Synthesis

A synthetic population is created that samples households and persons from Census Public Use Microdata Sample (PUMS) or American Community Survey (ACS) data.

1. Long-Term Models
   1. Usual Workplace/School Location Choice: This model component chooses the usual workplace for each worker in the synthetic population, and the school location for each student.
2. Medium-Term Mobility Models

3.1 Free Parking Eligibility: This model component determines the free parking eligibility status of each worker who works in a parking-constrained area (a zone with daily parking costs). The choices are fully reimbursed, partially reimbursed, or not reimbursed. Workers who are partially reimbursed draw the percent of daily parking reimbursement from a distribution.

3.2 Car Ownership: This model component chooses the number of autos owned by each household.

1. Daily Models
   1. Coordinated Daily Activity Pattern Model: This model chooses the main activity\travel pattern for each household member. The alternatives are Mandatory (at least one work\school tour), Non-mandatory (no mandatory tours but at least one non-mandatory tour) or Stay-at-Home. The model also predicts the presence or absence of fully-joint tours.
   2. Mandatory Tour Models
      1. Mandatory Tour Frequency Model: This model selects the number (1 or 2) and purpose of mandatory tours for each worker and student assigned a mandatory activity pattern from model 4.1.
      2. Mandatory Tour Time-of-Day Choice Model: This model schedules work and school tours into half-hourly time periods.
      3. Mandatory Tour Escort Model: This model determines whether school children are escorted to/from school, and if so, which adult household member escorts the child. It generates escort tours for non-working adult chauffeurs and inserts a stop on tours for working adult chauffeurs.
      4. Mandatory Tour Mode Choice Model: This model selects the tour mode for each work, university and school tour.
   3. Joint Tours
      1. Joint Tour Frequency Model: This model determines the exact number (1 or 2) of joint tours, the purpose of each joint tour, and the composition (adults, children, mixed) of each joint tour.
      2. Joint Tour Person Participation Model: This model determines whether each household member participates in each joint tour.
      3. Joint Tour Destination Choice Model: This model selects the primary destination zone for each joint tour.
      4. Joint Tour Time-of-Day Choice Model: This model selects the outbound and return time period for each joint tour.
      5. Joint Tour Mode Choice Model: This model selects the tour mode for each joint tour.
   4. Individual Non-Mandatory Tours
      1. Individual Non-mandatory Tour Frequency: This model predicts the number (0-3+) and purpose of non-mandatory tours for each individual with either a mandatory or non-mandatory daily activity pattern.
      2. Individual Non-Mandatory Tour Destination Choice Model: This model selects the primary destination zone for each individual non-mandatory tour.
      3. Individual Non-Mandatory Tour Time-of-Day Choice Model: This model selects the outbound and return time period for each individual non-mandatory tour.
      4. Individual Non-Mandatory Tour Mode Choice Model: This model selects the tour mode for each individual non-mandatory tour.
   5. At-Work Sub-Tour Models
      1. At-Work Sub-Tour Frequency Model: This model selects the number (0-2) and purpose of at-work sub-tours (tours that begin and end at the workplace) for each work tour generated in model 4.2.1.
      2. At-Work Sub-Tour Time-of-Day Choice Model: This model selects the outbound and return time period for each at-work sub-tour.
      3. At-Work Sub-Tour Mode Choice Model: This model selects the tour mode for each at-work sub-tour.
      4. At-Work Sub-Tour Destination Choice Model: This model selects the primary destination zone for each at-work sub-tour. The primary destination is the ‘main’ destination of the tour.
2. Tour level:
   1. Stop Frequency Model: Predicts number of outbound (0-4) and inbound (0-4) stops on each tour
   2. Stop Purpose Model: Predicts purpose of each intermediate stop on tour. Intermediate stops are stops that are made on the tour between the tour origin (home or work) and the tour primary destination.
   3. Stop Location Choice Model: Predicts zone for each secondary stop on tour
   4. Stop Departure Time Choice Model: Predicts half-hour departure time period for each intermediate stop
3. Trip level:
   1. Trip Mode Choice Model: Predicts mode of each trip on each tour
   2. Trip assignment: Assignment of trips to network (VISUM)
4. Emissions Calculator (not implemented):
   1. VMT Allocation Model: Predicts vehicle-miles of travel on each household vehicle

As shown in Figure 7, the CT-RAMP software for the microsimulation components of the model relies on the Common Modeling Framework (CMF), a collection of Java libraries specifically designed for the implementation of disaggregate travel demand models. The OR-RAMP ABM utilizes the CT-RAMP Java package, which contains model logic, choice model structure, and model flow, while utility equations and model inputs and outputs specific to OR-RAMP are contained in Utility Expression Calculator (UEC) files. These Excel-based files open up the models so the model parameters (in particular, all coefficients of choice utilities for each variable), input filenames with all population, employment, land-use, and level-of-service data, etc. can be easily accessed which helps prevent errors and makes the model equations more accessible. The CT-RAMP package is in the ABM setup.

Common Modeling Framework:

* Matrix Classes
* LogitModel
* Utility Expression Calculator

CT-RAMP:

* Model Flow
* Model Components
* Inputs/Outputs

OR-RAMP

* UEC spreadsheets
* Market definitions

Figure – OR-RAMP Travel Demand Model Software Components

Table 4.11 presents the properties in the OR-RAMP properties file. The following are the key inputs to model:

* MAZ data – inputs/maz\_data\_export.csv
* Population synthesizer households – inputs/households.csv
* Population synthesizer persons – inputs/persons\_sorted\_uni.csv
* TAZ skims by mode and period – outputs/taz\_skim\_<MODE>\_<TOD>.omx
* TAP skims by period and set ID – outputs/tap\_skim\_<TOD>\_set<SETID>.omx
* MAZ to MAZ distance by non-motorized mode – outputs/maz2maz\_<walk|bike>.csv
* TAP to MAZ distance by non-motorized mode – outputs/tap2maz\_<walk|bike>.csv
* Drive TAZ to TAP impedances for transit auto access/egress – outputs/drive\_taz\_tap.csv
* Model parameters/coefficients – uec/\*.\*

The key outputs of OR-RAMP are:

* Individual tour file (one record per individual tour) - Outputs/indivTourData\_<iteration>.csv
* Individual trip file (one record per individual trip) - Outputs/indivTripData\_<iteration>.csv
* Joint tour file (one record per joint tour, including all participants)- Outputs/jointTourData\_<iteration>.csv
* Joint trip file (one record per joint trip, including all participants)- - Outputs/jointTripData\_<iteration>.csv

## STEP 7 – Build Trip Demand Matrices

After running the microsimulation person travel models, the trips output need to be aggregated to demand matrices for assignment in VISUM. This is done with the *build\_trip\_matrices* entry point in the SOABM.py Python script. The procedure has two steps:

1. build CT-RAMP TAZ and TAP matrices
2. load the CT-RAMP matrices + the CVM and external matrices into VISUM

The following trip matrices by mode and time period are created from the CT-RAMP trips:

* Outputs\ctrampTazTrips.omx
  + SOV\_<TOD>
  + SOVToll\_<TOD>
  + HOV2\_<TOD>
  + HOV2Toll\_<TOD>
  + HOV3\_<TOD>
  + HOV3Toll\_<TOD>
* Outputs\ctrampTapTrips.omx
  + Set1\_<TOD>
  + Set2\_<TOD>
  + Set3\_<TOD>

The person trips are converted to vehicle trips using an assumed occupancy of 2 for HOV2 and 3.33 for HOV3+. If a trip selected a different parking zone from its destination zone (in the constrained parking areas only), then trip destination is changed to the parking destination in the trip matrix. For escort trips, any case where the driver person number is equal to the escort person number, the trip is converted to one vehicle trip with an occupancy equal to the occupancy for the mode. For park and ride and kiss and ride trips, an auto trip from the MAZ to the TAP is added into the auto trip matrices as well (SOV for park and ride and HOV2 for kiss and ride). The matrices are scaled by the household sample rate at this point in the model to ensure the number of trips is consistent with the link capacities when calculating congested travel times. For example, if a 50% sample of households is run, then each trip generated by CT-RAMP is multiplied by 2.

When loading the matrices into VISUM for assignment, the demand matrices are aggregated as follows:

* SOV (matrix 100) = cvm\_car + ext\_hbw + ext\_nhbnw + ext\_hbo + ext\_hbcoll + ext\_hbr + ext\_hbs + ext\_hbsch + ext\_nhbw + ct\_sov
* HOV2hov2 (matrix 101) = ct\_hov2
* HOV3 (matrix 102) = ct\_hov3
* Truck (matrix 103) = cvm\_su + cvm\_mu + ext\_truck
* SOVToll (matrix 104) = ct\_sovtoll
* HOV2Toll (matrix 105) = ct\_hov2toll
* HOV3Toll (matrix 106) = ct\_hov3toll
* Transit (matrix 107) = ct\_transit #each transit set is in a different version file

*where*

*ct = CT-RAMP*

*cvm = commercial vehicle model*

*ext = external model*

## STEP 8 – Create Skims and Assignments Using Congested Speeds

The next step in the model is to assign the trip demand matrices to the network and to re-generate the TAZ and TAP skims as a function of congestion. The resident, commercial, and external trips generated in previous step are already in VISUM at this point, so this step essentially loads the previously created VISUM skimming version files and run different skimming and assignment procedures that use congested travel times instead of travel times calculated from TomTom speeds. The MAZ level network impedances remained unchanged because they are a function of shortest path network distances. This step is split into two entry points in the SOABM.py Python script: *taz\_skim* and *tap\_skim*.

### TAZ Skims

The TAZ skims are created by calling the *taz\_skim* entry point in the SOABM.py Python script. The taz\_skim call of the script does the following by time-of-day:

1. Starts VISUM via the Python win32 COM server
2. Loads the outputs/taz\_skim\_<TOD>\_speed.ver version file created earlier
3. Loads and run the TAZ skimming procedures stored in config/visum/taz\_skim\_<TOD>.par. The procedures do the following:
   1. Set time-of-day link number of lanes based on the generic number of lanes
   2. Set time-of-day link capacity based on number of lanes (LANES\_<TOD>, capacity per lane (LANECAP), and time-of-day (NETWORK\TOD\_FACTOR\_<TOD>)
   3. Set link speed using SPEED user-defined attribute (i.e. free flow speed)
   4. Runs equilibrium assignment with a relative gap of 10-4, 100 iterations, and warm starting
   5. Calculates and output skims in OMX format, as defined in Table 5.15.
   6. The settings are the same as the previous skimming procedures
4. Save the version file, which includes the skims and assignment paths, to outputs\taz\_skim\_<TOD> \_speed.ver
5. Creates the same TAZ to TAP drive access time information file as before
6. Closes VISUM

### TAP Skims

The TAP skims are created by calling the *tap\_skim* entry point in the SOABM.py Python script. The tap\_skim call of the script does the following by time-of-day and skim set:

1. Starts VISUM via the Python win32 COM server
2. Loads the outputs/taz\_skim\_<TOD>\_speed.ver version file and writes the link speeds to a CSV file for updating the transit run times - outputs/taz\_skim\_<TOD>\_speed\_linkspeeds.csv. The link speed used in the VCur\_PrTSys(HOV3Toll) attribute (see Table 5.2).
3. Loads the outputs/tap\_skim\_initial\_speed\_set<SETID>.ver version file created earlier and loads the link speeds written out earlier into the link V0PrT attribute in order to set the transit runtimes as a function of the highway congested link times.
4. Loads and runs the config/visum/tap\_skim\_<TOD>.par procedure file to calculate the transit run and dwell times and skims using the same function as before. This time it assigns trips as opposed to using a very small number of trips for one OD pair.
5. Writes the skims to OMX files using the config/visum/tap\_skim\_<TOD>\_set<SETID>.par file. The skims are defined in Table 5.11.
6. Save the version file, which includes the skims, to outputs/tap\_skim\_<TOD>\_speed\_set<SETID>.ver
7. Revises skims across skim sets that have duplicate levels-of-service like before.
8. Closes VISUM

## STEP 9 – Check for Completion

The final step is to check for model completion. If the max number of overall model iterations (loops) has not been completed, then the model returns to the start of the feedback loop, runs CT-RAMP with the new set of skim matrices and then runs assignment and skimming as well. Once the max number of iterations has been completed, then model run is finished.

# Inputs

All inputs to the model are stored in the inputs sub-directory under the project folder.

## Networks and Zones

The primary input for a scenario is the SOABM.ver VISUM master version file. This file contains the multiple zone systems, all zonal attributes, the highway network, and the transit network. The version file uses the following VISUM network objects:

1. Nodes – network nodes, with no required attributes.
2. Links – network links, with the attributes required in Table 5.1.  VISUM’s CapPrT attribute is set during the skimming and assignment procedures based on the user-defined attributes for per lane capacity, number of lanes, and time-of-day.
3. Turns – network turns, with no required attributes except for TSysSet=empty for prohibited turns.
4. Zones – TAZS, with the attributes required in Table 4.4. TAZs in SOABM are only for use auto skims and assignment and therefore do not contain land use data. External TAZs are numbered 1-99 and internal TAZs are 100+.
5. Connectors – TAZ connectors, with TSySet and t0 (initial travel time) set. Only TAZ (auto) connectors are coded in the input file. The VISUM Python procedures code the MAZ and TAP connectors. Make sure to code nodes along walk and bike paths so MAZ connectors can be built.
6. Main Zones – MAZs, with the attributes required in Table 4.2. SOABM generates all trips to and from MAZs, which requires all land use data to be coded at the MAZ level as well. The MainZone SEQMAZ field is export to OR-RAMP as the MAZ ID. The MainZone TAZID field is export to OR-RAMP as the MAZ’s TAZ. The SEQMAZs must be numbered from 1 to X, with X equal to the number of MAZs. There are no external MAZs. The MAZ polygon needs to be drawn since the VISUM calculated attribute AreaMi2 is used for the MAZ area for the density measures.
7. Stop Points – Physical transit stops, with no required attributes.
8. Stop Areas – TAPs, with the attribute required in Table 4.5. TAPs are numbered from 1 to X, with X equal to the number of TAPs. The Stop area ID is used as the TAP ID in OR-RAMP. There can be multiple stop points per stop area in VISUM, but this is currently not the case since the network was imported from Google transit.
9. Stops – Groups of stop areas. There can be multiple stop areas per stop in VISUM, but this is currently not the case since the network was imported from Google transit.
10. Lines – Transit lines, with no required attributes. The line name is currently set to the Google transit operator and route number. The available Transport Systems for transit lines are Bus, Commuter Rail, Express Bus, and Light Rail.
11. Line Routes – Transit lines, with a headway attribute for each time period (see Table 5.3).  The headway needs to be coded in seconds.  If the line is not available in a time period, then set the headway to a very large number such as 99999.  Line routes code the sequence of links and stop points for each line.  The lines are currently coded as round trip line routes due to the Google transit import.  Line routes can be coded by direction if desired.
12. Time Profiles – Transit line link run times and stop dwell times. The transit line run and dwell times are updated after highway assignment using the VISUM procedure – Set Run and Dwell Times. Even though Google transit schedules were loaded into the version file in order to calculate headways, they are not used in the model.
13. Network – Each version file has a master Network object that all network objects inherit from. The SOABM Network has a link capacity time-of-day factor for calculating time period specific link capacities from the input link attribute one-hour capacity.

The highway network is shown in Figure 9, with a detailed view of Medford in Figure 10 and Grants Pass in Figure 11. Table 4.1 lists the input link attributes. In addition to standard VISUM link attributes such as TsysSet, Length, TypeNo (for VDF lookup), V0PrT (free flow speed), and Toll\_PrTSys(Dseg), the network includes some additional attributes calculated by ODOT’s network development script. These additional attributes include hourly capacity, number of lanes, facility type, and defaults congested speeds by time-of-day based on the TomTom data. The percent of links with TomTom speed data is summarized in Figure 8 below. This data was used to develop the general method for coding congested speeds in the script. Please refer to the script when coding the additional required link attributes.



Figure - TomTom Speed Links

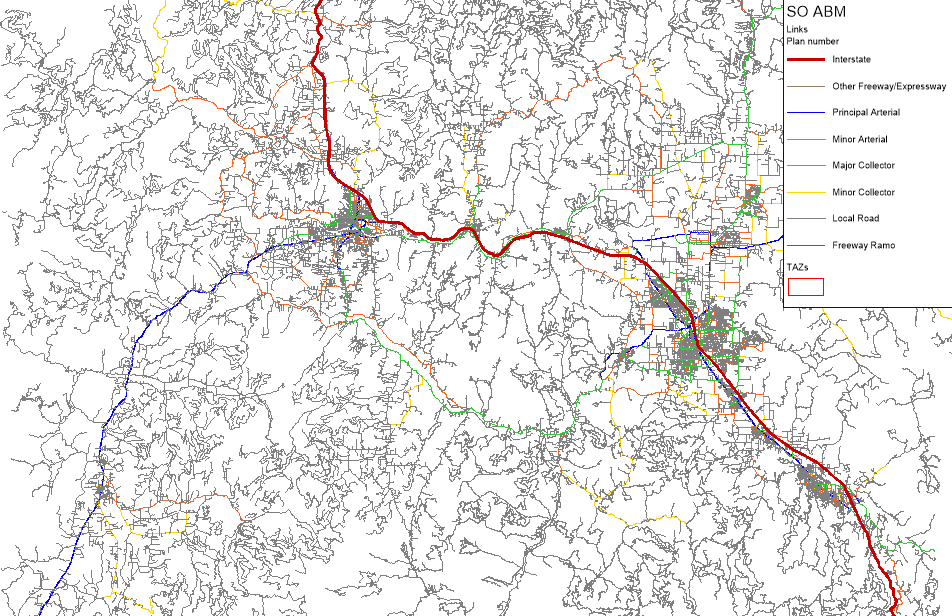


Figure - VISUM Highway NeTWork

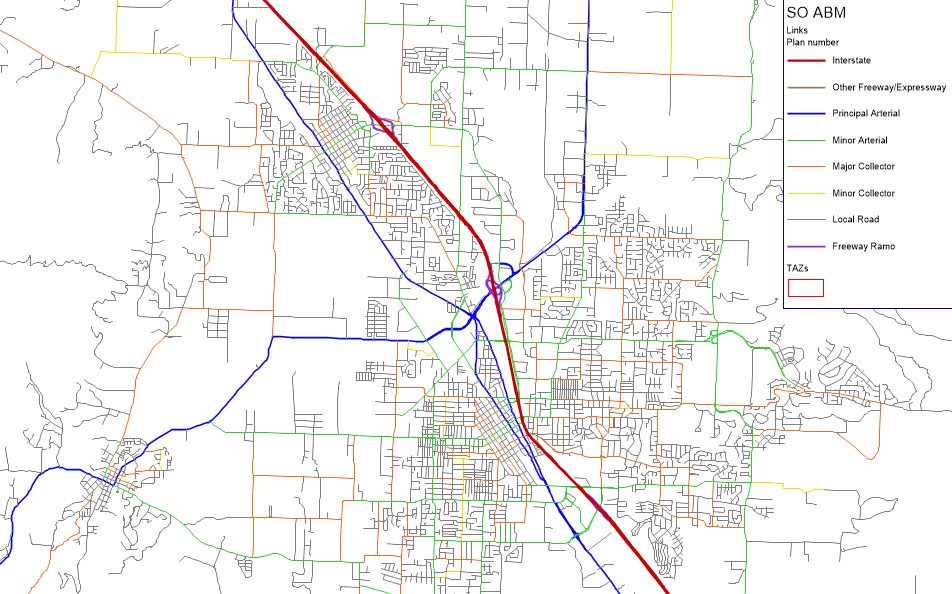


Figure - Medford Highway Network

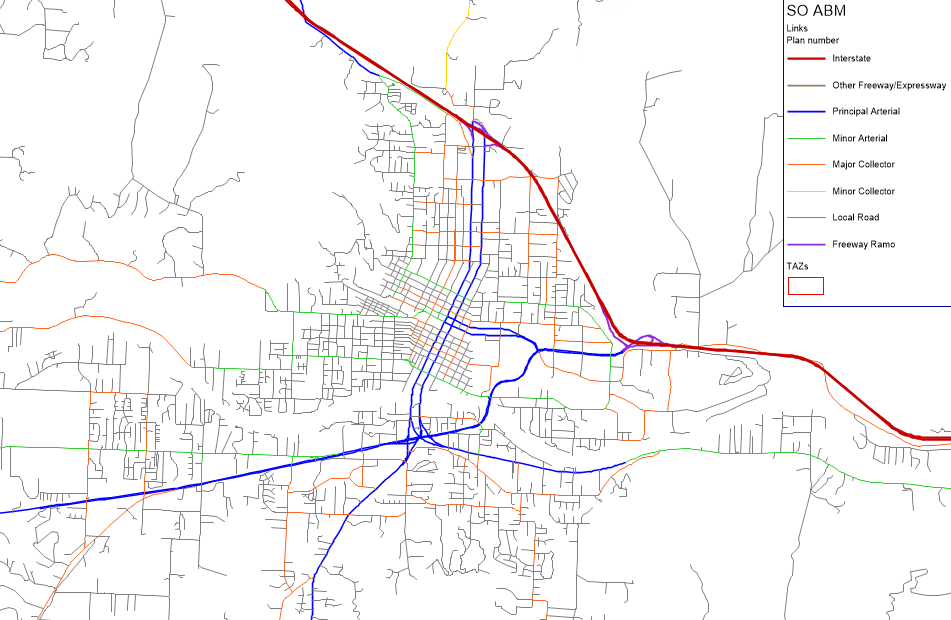


Figure – Grants PAss Highway Network

The version file also contains the transit networks. Each transit line route has a headway by time-of-day and each line route is currently coded roundtrip. Line routes could be coded by direction if desired. The VISUM stop model (shown in Figure 12) is used as follows: 1) stop points – transit stop locations, 2) stop areas – TAPs, and 3) stops – groups of stop areas for virtual transferring. Transit walk transferring is also allowed by the assignment and skimming procedures if desired. Since Stop Areas are converted to VISUM zones for TAP skimming and assignment, a stop area to nearest network node connector is built on-the-fly (as described earlier). The Grants Pass transit network is shown in Figure 13, whereas the Medford transit network is shown in Figure 14.

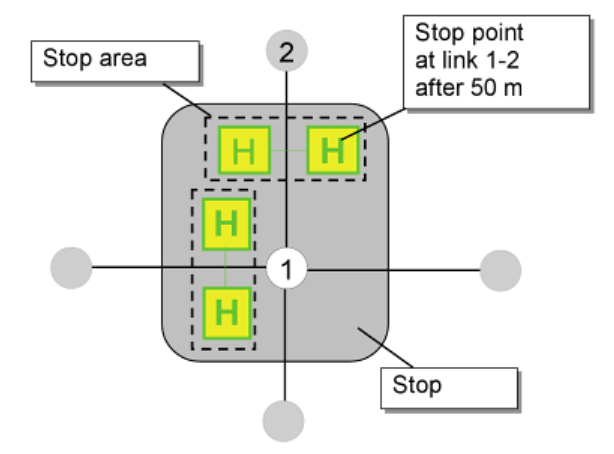


Figure - VISUM Stop Model

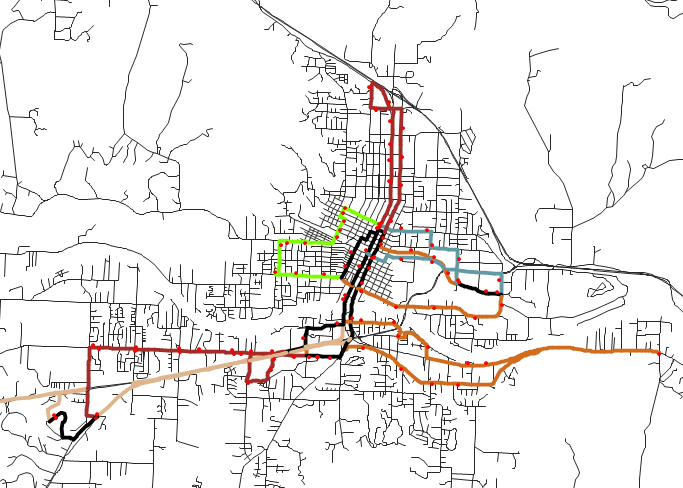


Figure – Grants Pass TRANSIT Network

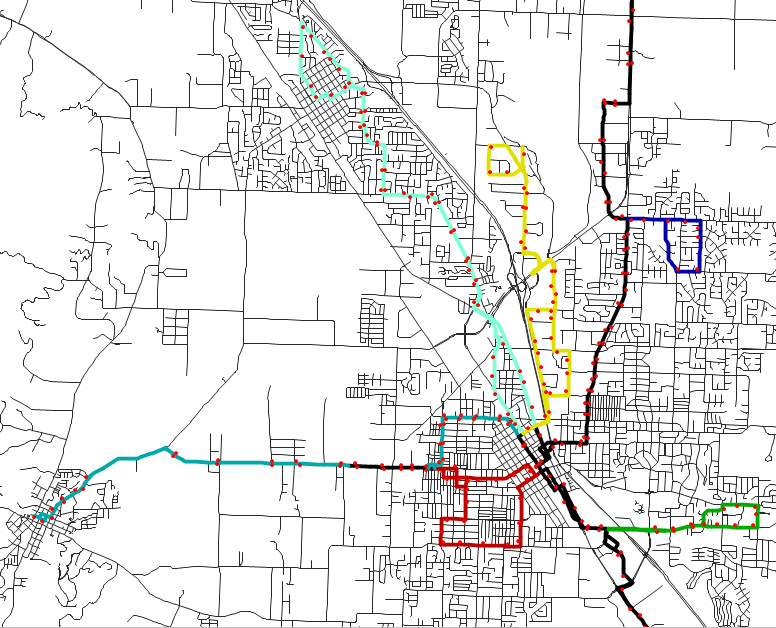


Figure – MeDford TRANSIT Network

Table . Link Attributes

|  |  |
| --- | --- |
| **Field** | **Description** |
| FromNodeNo | From node |
| ToNodeNo | To node |
| TypeNo | Link type (used for the VDF function lookup) |
| TSysSet | Allowed transport systems (i.e. modes):  SOV, SOVToll  HOV2, HOV2Toll  HOV3, HOV3Toll  Truck, TruckToll  Bike  Walk |
| V0PrT | VISUM’s built-in attribute for free flow speed |
| Length | Link length in miles |
| Toll\_PrTSys(DSEG) | Toll for each demand segment |
| Numlanes | Number of lanes. This attribute is edited by the user. |
| LaneCap | Per lane capacity, which is used to calculate the CapPrT attribute used in the skimming and assignment procedures. |
| Speed | Free flow speed |
| <TOD>\_Speed | TomTom speed by time-of-day |
| Lanes\_<TOD> | Lanes by time-of-day, which is set equal to NUMLANES by the time-of-day TAZ skimming and assignment procedures. This attribute is an intermediate attribute and does not need to be edited by the user. |
| Sidewalk | 0 = No sidewalk, 1 = Sidewalk present. Note that this attribute is not currently used for assignment or skimming since presence of a sidewalk is attributed based on TSys. |
| BikeFac | Bike facility. Note that this attribute is not currently used for assignment or skimming since presence of a sidewalk is attributed based on TSys.  1 = Class I Path  2 = Class II Lane  3 = Class III Route |
| PlanNo | Facility type  1=Interstate  3=Principal Arterial  4=Minor Arterial  5=Major Collector  6=Minor Collector  7=Local Road  30=Ramp |
| AddVal1,2,3 | Are used for internal calculations and should not be used by the user |

Table . MAZ (MainZone) Attributes

|  |  |
| --- | --- |
| **Name** | **Description** |
| SEQMAZ | MAZ number for CT-RAMP. From 1 to N (number of MAZs) |
| NO | VISUM zone number |
| TAZID | TAZ number for CT-RAMP |
| DISTNAME | District name, which is read by CT-RAMP |
| COUNTY | County, which is read by CT-RAMP |
| HH | Number of households |
| SF\_HH | Number of single family households. This field is used for population synthesis and not CT-RAMP. |
| DUPLEX\_HH | Number of duplex households. This field is used for population synthesis and not CT-RAMP. |
| MF\_HH | Number of multifamily households. This field is used for population synthesis and not CT-RAMP. |
| MH\_HH | Number of mobile home households. This field is used for population synthesis and not CT-RAMP. |
| EMP\_CONSTR | Construction employment |
| EMP\_WHOLE | Wholesale Trade employment |
| EMP\_RETAIL | Retail Trade employment |
| EMP\_SPORT | Sporting Goods, Hobby, Musical Instruments and Book stores employment |
| EMP\_ACCFD | Accommodation and Food Services employment |
| EMP\_AGR | Agriculture, Forestry, Fishing and Hunting employment |
| EMP\_MIN | Mining, Quarrying, and Oil and Gas Extraction employment |
| EMP\_UTIL | Utilities employment |
| EMP\_FOOD | Food Manufacturing employment |
| EMP\_WOOD | Wood Product Manufacturing employment |
| EMP\_METAL | Primary Metal Manufacturing employment |
| EMP\_TRANS | Transportation employment |
| EMP\_POSTAL | Postal Service employment |
| EMP\_INFO | Information employment |
| EMP\_FINANC | Finance and Insurance employment |
| EMP\_REALES | Real Estate and Rental and Leasing employment |
| EMP\_PROF | Professional, Scientific, and Technical Services employment |
| EMP\_MGMT | Management of Companies and Enterprises employment |
| EMP\_ADMIN | Administrative and Support and Waste Management employment |
| EMP\_EDUC | Educational Services employment |
| EMP\_HEALTH | Health Care and Social Assistance employment |
| EMP\_ARTS | Arts, Entertainment, and Recreation employment |
| EMP\_OTHER | Other Services (except Public Administration) Religious employment |
| EMP\_PUBADM | Public Administration employment |
| EMP\_TOTAL | Total employment |
| ENROLLK\_8 | Grade School K-8 enrollment |
| ENROLL9\_12 | Grade School 9-12 enrollment |
| ENROLLCOLL | Major College enrollment |
| ENROLLCOOT | Other College enrollment |
| ENROLLADSC | Adult School enrollment |
| universitySqFtClass | Major university square footage classrooms, which is export for use by the major university student tour destination choice model |
| universitySqFtOffice | Major university square footage office, which is export for use by the major university student tour destination choice model |
| universitySqFtRec | Major university square footage recreation, which is export for use by the major university student tour destination choice model |
| ECH\_DIST | Elementary school district |
| HCH\_DIST | High school district |
| HOTELRMTOT | Total number of hotel rooms |
| PARKAREA | Parking model code:  1 - Trips with destinations in this MAZ may choose to park in a different MAZ, parking charges apply (downtown)  2 - Trips with destinations in parkarea 1 may choose to park in this MAZ, parking charges might apply (1/4 mile buffer around downtown)  3 - Only trips with destinations in this MAZ may park here, parking charges apply (outside downtown paid parking, only show cost no capacity issue)  4 - Only trips with destinations in this MAZ may park here, parking charges do not apply (outside downtown, free parking) |
| HSTALLSOTH | Number of stalls allowing hourly parking for trips with destinations in other MAZs |
| HSTALLSSAM | Number of stalls allowing hourly parking for trips with destinations in the same MAZ |
| HPARKCOST | Average cost of parking for one hour in hourly stalls in this MAZ, dollars |
| NUMFREEHRS | Number of hours of free parking allowed before parking charges begin in hourly stalls |
| DSTALLSOTH | Stalls allowing daily parking for trips with destinations in other MAZs |
| DSTALLSSAM | Stalls allowing daily parking for trips with destinations in the same MAZ |
| DPARKCOST | Average cost of parking for one day in daily stalls, dollars |
| MSTALLSOTH | Stalls allowing monthly parking for trips with destinations in other MAZs |
| MSTALLSSAM | Stalls allowing monthly parking for trips with destinations in the same MAZ |
| MPARKCOST | Average cost of parking for one day in monthly stalls, amortized over 22 workdays, dollars |
| HHS | Average household size in the MAZ. This field is used for population synthesis and not CT-RAMP. |
| POP | Total population. This field is used for calculating the zone buffer measures. |
| HHP | Total household population (exclude GQ pop) This field is used in CT-RAMP. |
| HH\_GQ\_CIV | GQ civilian population (half-way houses and other similar GQ not associated with military housing, on-campus housing, or fraternities/sororities) |
| HH\_GQ\_MIL | GQ military population (includes military group quarters housing) |
| HH\_GQ\_UNIV | GQ University students (on-campus or off-campus dormitories, fraternities, and sororities) |
| WRK\_EXT\_PR | Probability of working outside the region |
| SCHDIST\_NA | School district name |
| SCHDIST | School district code |
| HH\_1 | Number of 1 person household. This field is used for population synthesis and not CT-RAMP. |
| HH\_2 | Number of 2 persons household. This field is used for population synthesis and not CT-RAMP. |
| HH\_3 | Number of 3 persons household. This field is used for population synthesis and not CT-RAMP. |
| HH\_4 | Number of 4+ persons household. This field is used for population synthesis and not CT-RAMP. |
| TERMTIME | Terminal time |
| PARKACRES | Park acres |

Table . LiNERoute Attributes

|  |  |
| --- | --- |
| **Field** | **Description** |
| EA\_Headway | EA period headway in seconds |
| AM\_Headway | AM period headway in seconds |
| MD\_Headway | MD period headway in seconds |
| PM\_Headway | PM period headway in seconds |
| EV\_Headway | EV period headway in seconds |

Table . TAZ (Zone) Attributes

|  |  |
| --- | --- |
| **Field** | **Description** |
| No | 1-7 external; 100+ internal |

Table . TAP (STOPAREA) Attributes

|  |  |
| --- | --- |
| **Field** | **Description** |
| No | TAP number. From 1 to N (number of TAPs) |
| CanPnr | 1 for park and ride available; 0 if not |

Table . Time-oF-Day periods and Link Capacities

|  |  |  |  |
| --- | --- | --- | --- |
| **Time-of-Day** | **Description** | **Hourly Factor** | **VISUM Network Object** |
| EA | 3:00-7:00 | 4 | Network\TOD\_FACTOR\_EA |
| AM | 7:00-8:30 | 1.5 | Network\TOD\_FACTOR\_AM |
| MD | 8:30-16:30 | 8 | Network\TOD\_FACTOR\_MD |
| PM | 16:30-18:30 | 2 | Network\TOD\_FACTOR\_PM |
| EV | 18:30-27:00 | 8.5 | Network\TOD\_FACTOR\_EV |

## Population Synthesizer

The primary input for a scenario is the SOABM.ver VISUM master version file. This file contains the multiple zone systems, all zonal attributes, the highway network, and the transit network. The population synthesizer outputs the synthetic household and person files that are required for running the SOABM. These tables are below. The definitions for other PUMS variables can be found in the 2007-2011 ACS PUMS data dictionary[[2]](#footnote-2)

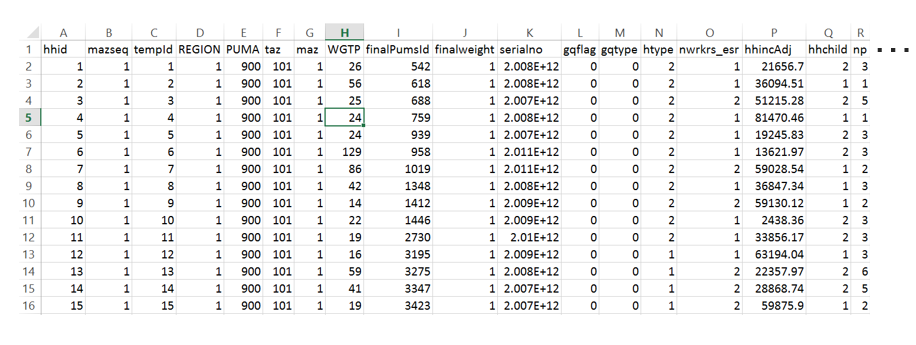


Figure - PopSyn Households File EXample

Table . PopSYN Households

| **Field** | **Description** | **Key** |
| --- | --- | --- |
| tempId | Unexpanded household ID |  |
| REGION | META Geography code in which HH is located |  |
| PUMA | PUMA Geography code in which HH is located |  |
| taz | HH TAZ |  |
| maz | HH MAZ; this field will be updated by the zoneChecker to be the sequential MAZ number and the initial field will be copied to maz\_initial |  |
| finalPumsId | HH ID generated during population synthesis |  |
| finalweight | HH weight generated during population synthesis |  |
| serialno | Unique housing PUMS record identifier |  |
|  |  |  |
| gqflag | Dummy variable for gq population | 0: general population 1: non-institutional gq population |
| gqtype | Type of GQ record | 0: general population 1: university gq record (placeholder) 2: military gq record (placeholder) 4: other civilian non-institutional gq record |
| htype | Housing type | 1: Single-family 2: Multi-family 3: Mobile-home 4: Duplex 999: GQ - Not applicable |
| nwrkrs\_esr | Number of workers in the household |  |
| hhincAdj | Household income adjusted to 2010$s |  |
| hhchild |  | 1: No children 2: One or more children 999: GQ - Not applicable |
| n | Index of record within an expanded household set. Ranges from 1 to finalweight |  |
| hhid | Unique household ID |  |
| \* | Other PUMS HH fields |  |

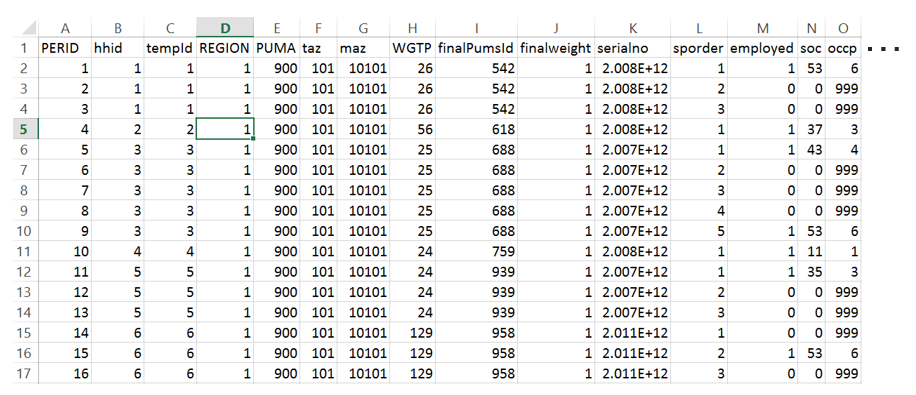


Figure - PopSYn Persons File Example

Table . PopSYN Persons

| **Field** | **Description** | **Key** |
| --- | --- | --- |
| tempId | Unexpanded household ID |  |
| REGION | META Geography code in which HH is located |  |
| PUMA | PUMA Geography code in which HH is located |  |
| taz | HH TAZ |  |
| maz | HH MAZ |  |
| WGTP | PUMS housing weight |  |
| finalPumsId | Person ID generated during population synthesis |  |
| finalweight | HH weight generated during population synthesis |  |
| serialno | Unique housing PUMS record identifier |  |
| sporder | Person number within HH |  |
| employed | Is person employed (based on ESR) | 0: not employed 1: employed |
| soc | Unified 2 digit soc codes (derived from multi-year pums SOC codes) |  |
| occp | Occupation code for this person | 1: Management, Business, Science, and Arts 2: White Collar Service Occupations 3: Blue Collar Service Occupations 4: Sales and Office Support 5: Natural Resources, Construction, and Maintenance 6: Production, Transportation, and Material Moving 999: Not in labor force |
| gqflag | Dummy variable for gq population | 0: general population 1: non-institutional gq population |
| gqtype | Type of GQ record | 0: general population 1: university gq record (placeholder) 2: military gq record (placeholder) 4: other civilian non-institutional gq record |
| n | Index of record within an expanded household set. Ranges from 1 to finalweight |  |
| perid | Unique person ID |  |
| hhid | Unique household ID |  |
| \* | Other PUMS Person fields |  |
| majoruni | Major university student required for major university model | 0=no; 1=yes |

## CVM and External Model

The CVM and SWIM external model require a number of additional unique inputs, as described in the tables below.

Table . CVM Specific Inputs

|  |  |
| --- | --- |
| **File** | **Description** |
| nonWorkAttr.csv | Non-work trip attraction rates |
| nonWorkFriction.csv | Non-work distribution friction factors |
| nonWorkProd\_IntraTrips.csv | Non-work trip production rates |
| TOD\_Car.csv | Time-of-day factors for car |
| TOD\_MUTruck.csv | Time-of-day factors for multi-unit trucks |
| TOD\_Periods.csv | Time-of-day factors for periods for demand matrices |
| TOD\_SUTruck.csv | Time-of-day factors for single unit trucks |
| workAttr.csv | Work trip attraction rates |
| workFriction.csv | Work distribution friction factors |
| workProd\_IntraTrips.csv | Work trip production rates |

Table . SWIm External MODEL SpeciFIC Inputs

|  |  |
| --- | --- |
| **File** | **Description** |
| swimControls.RData | R data object of various SWIM external model settings |
| SWIM\_JEMnR\_TAZ\_CW.csv | SWIM zone to SOABM zone crosswalk |
| select\_link\_1\_2010\_outputs.zip | SWIM assignment select link results at each SOABM external station |
| selectLinks.csv | Select link locations |

## OR-RAMP Properties File

OR-RAMP is controlled by a Java properties file. The table below identifies, describes, and provides an example for each of the variables expected to be in the properties file. After initially configuring the travel model, only a handful of these properties will be modified for a typical scenario analysis. The primary use for many of the properties is to facilitate software execution when calibrating the travel model and/or to locate and fix errors. Comments preceded with a pound (#) sign are ignored when reading in the properties file.

Table . or-ramp pROPERTIES fiLE

| Module | Property | Data Type | Example | Purpose |
| --- | --- | --- | --- | --- |
| Cluster properties: HH manager | RunModel.HouseholdServerAddress | String | 156.75.49.78 | household server address; set automatically by RunModel.bat |
| Cluster properties: HH manager | RunModel.HouseholdServerPort | Integer | 1117 | household server port number |
| Cluster properties: Matrix server | RunModel.MatrixServerAddress | String | 156.75.49.78 | matrix server address; set automatically by RunModel.bat |
| Cluster properties: Matrix server | RunModel.MatrixServerPort | Integer | 1191 | matrix server port number |
|  |  |  |  |  |
| Logging and debugging | Trace | Boolean | FALSE | True or False whether to trace zones |
| Logging and debugging | Trace.otaz | Integer | 1638 | Specify which origin taz to trace |
| Logging and debugging | Trace.dtaz | Integer | 2447 | Specify which destination taz to trace |
| Logging and debugging | Seek | Boolean | FALSE | True or False whether to seek households |
| Logging and debugging | Process.Debug.HHs.Only | Boolean | FALSE | True of False whether to debug households |
| Logging and debugging | Debug.Trace.HouseholdIdList | String | 566425 | Specify which household IDs to trace |
| Logging and debugging | run.this.household.only | String | 566425 | Specify that this household ID only will be run through the model |
| Logging and debugging: Tour mode choice (4.2.4, 4.5.3, 4.3.5, 4.4.4) | TourModeChoice.Save.UtilsAndProbs | Boolean | TRUE | Save utilities and probabilities in tour mode choice output files |
|  |  |  |  |  |
| Path properties | Project.Directory | String | %project.folder% | File locations specified in the properties file as well as the UEC DataSheet pages are expressed as relative to this location. |
| Path properties | generic.path | String | %project.folder% | Generic path location |
| Path properties | scenario.path | String | %project.folder% | Scenario location |
| Path properties | skims.path | String | %project.folder%/outputs | Location of skims |
| Path properties | uec.path | String | %project.folder%/uec/ | Location of UECs |
|  |  |  |  |  |
| Scenario properties | mgra.socec.file | String | /inputs/maz\_data\_export.csv | location of MAZ land use file |
| Scenario properties: University Model | MajorUniversity.referenceMaz | Integer | 813 | Southern Oregon University MAZ |
| Scenario properties: Trip table creation | occ3plus.purpose.Work | Float | 3.5 | Specify 3+ occupancy multiplier for trip table creation for Work Purpose |
| Scenario properties: Trip table creation | occ3plus.purpose.University | Float | 3.5 | Specify 3+ occupancy multiplier for trip table creation for University Purpose |
| Scenario properties: Trip table creation | occ3plus.purpose.School | Float | 3.5 | Specify 3+ occupancy multiplier for trip table creation for School Purpose |
| Scenario properties: Trip table creation | occ3plus.purpose.Escort | Float | 3.5 | Specify 3+ occupancy multiplier for trip table creation for Escort Purpose |
| Scenario properties: Trip table creation | occ3plus.purpose.Shop | Float | 3.5 | Specify 3+ occupancy multiplier for trip table creation for Shop Purpose |
| Scenario properties: Trip table creation | occ3plus.purpose.Maintenance | Float | 3.5 | Specify 3+ occupancy multiplier for trip table creation for Maintenance Purpose |
| Scenario properties: Trip table creation | occ3plus.purpose.EatingOut | Float | 3.5 | Specify 3+ occupancy multiplier for trip table creation for Eating Out Purpose |
| Scenario properties: Trip table creation | occ3plus.purpose.Visiting | Float | 3.5 | Specify 3+ occupancy multiplier for trip table creation for Visiting Purpose |
| Scenario properties: Trip table creation | occ3plus.purpose.Discretionary | Float | 3.5 | Specify 3+ occupancy multiplier for trip table creation for Discretionary Purpose |
| Scenario properties: Trip table creation | occ3plus.purpose.WorkBased | Float | 3.5 | Specify 3+ occupancy multiplier for trip table creation for At-Work Sub Tour Purpose |
| Scenario properties | aoc.fuel | Double | 13.5 | Fuel cost per mile |
| Scenario properties | aoc.maintenance | Double | 6.3 | Maintenance cost per mile |
| Scenario Properties: HH Manager | HouseholdManager.MinValueOfTime | Float | 1 | Distributed person VOT settings |
| Scenario Properties: HH Manager | HouseholdManager.MaxValueOfTime | Float | 50 | Distributed person VOT settings |
| Scenario Properties: HH Manager | HouseholdManager.MeanValueOfTime.Values | String | 6.01, 8.81, 10.44, 12.86 | Distributed person VOT settings |
| Scenario Properties: HH Manager | HouseholdManager.MeanValueOfTime.Income.Limits | String | 30000, 60000, 100000 | Distributed person VOT settings |
| Scenario Properties: HH Manager | HouseholdManager.Mean.ValueOfTime.Multiplier.Mu | Float | 0.684 | Distributed person VOT settings |
| Scenario Properties: HH Manager | HouseholdManager.ValueOfTime.Lognormal.Sigma | Float | 0.87 | Distributed person VOT settings |
| Scenario Properties: HH Manager | HouseholdManager.HH.ValueOfTime.Multiplier.Under18 | Float | 0.66667 | Distributed person VOT settings |
| Scenario Properties: HH Manager | HouseholdManager.MinValueOfTime | Float | 1 | Distributed person VOT settings |
| Scenario properties | mgra.useExternals | Boolean | FALSE | Use externals for MGRA |
|  |  |  |  |  |
| Core model run properties | Model.Random.Seed | Integer | 1 | Starting value for model random seed number (added to household IDs to create unique random number for each household) |
| Core model run: Accessibility | acc.read.input.file | Boolean | FALSE | Read the accessibilities as input instead of running the module |
| Core model run: JPPF | distributed.task.packet.size | Integer | 200 | JPPF distributed task packet size |
| Core model run: HH Manager | RunModel.RestartWithHhServer | String | none | model can be restarted with certain files already generated…..values include 'none' (run whole model), 'uwsl', 'ao', 'stf' |
| Core model run: Pre AO (2.1) | RunModel.PreAutoOwnership | Boolean | TRUE | True or False whether to run this model component |
| Core model run: Work/school location (2.2) | RunModel.UsualWorkAndSchoolLocationChoice | Boolean | TRUE | True or False whether to run this model component |
| Core model run: Final AO (3.2) | RunModel.AutoOwnership | Boolean | TRUE | True or False whether to run this model component |
| Core model run: Transponder Choice | RunModel.TransponderChoice | Boolean | TRUE | True or False whether to run this model component |
| Core model run: Free parking | RunModel.FreeParking | Boolean | TRUE | True or False whether to run this model component |
| Core model run: CDAP (4.1) | RunModel.CoordinatedDailyActivityPattern | Boolean | TRUE | True or False whether to run this model component |
| Core model run: MTF (4.2.1) | RunModel.IndividualMandatoryTourFrequency | Boolean | TRUE | True or False whether to run this model component |
| Core model run: Mandatory tourmode (4.2.4) | RunModel.MandatoryTourModeChoice | Boolean | TRUE | True or False whether to run this model component |
| Core model run: Mandatory TOD (4.2.2) | RunModel.MandatoryTourDepartureTimeAndDuration | Boolean | TRUE | True or False whether to run this model component |
| Core model run: JTF (4.3.1) | RunModel.JointTourFrequency | Boolean | TRUE | True or False whether to run this model component |
| Core model run: Joint destination (4.3.3) | RunModel.JointTourLocationChoice | Boolean | TRUE | True or False whether to run this model component |
| Core model run: Joint TOD (4.3.4) | RunModel.JointTourDepartureTimeAndDuration | Boolean | TRUE | True or False whether to run this model component |
| Core model run: Joint mode (4.3.5) | RunModel.JointTourModeChoice | Boolean | TRUE | True or False whether to run this model component |
| Core model run: Indi non-mand freq (4.4.1) | RunModel.IndividualNonMandatoryTourFrequency | Boolean | TRUE | True or False whether to run this model component |
| Core model run: Indi non-mand destination (4.4.2) | RunModel.IndividualNonMandatoryTourLocationChoice | Boolean | TRUE | True or False whether to run this model component |
| Core model run: Indi non-mand TOD (4.4.3) | RunModel.IndividualNonMandatoryTourDepartureTimeAndDuration | Boolean | TRUE | True or False whether to run this model component |
| Core model run: Indi non-mand mode (4.4.4) | RunModel.IndividualNonMandatoryTourModeChoice | Boolean | TRUE | True or False whether to run this model component |
| Core model run: Atwork freq (4.5.1) | RunModel.AtWorkSubTourFrequency | Boolean | TRUE | True or False whether to run this model component |
| Core model run: Atwork destination (4.5.4) | RunModel.AtWorkSubTourLocationChoice | Boolean | TRUE | True or False whether to run this model component |
| Core model run: Atwork TOD (4.5.2) | RunModel.AtWorkSubTourDepartureTimeAndDuration | Boolean | TRUE | True or False whether to run this model component |
| Core model run: Atwork mode (4.5.3) | RunModel.AtWorkSubTourModeChoice | Boolean | TRUE | True or False whether to run this model component |
| Core model run: Stop freq (5.1) | RunModel.StopFrequency | Boolean | TRUE | True or False whether to run this model component |
| Core model run: Stop freq (5.3) | RunModel.StopLocation | Boolean | TRUE | True or False whether to run this model component |
|  |  |  |  |  |
| Core model run: Usual workplace/school (2.2) | uwsl.use.new.soa | Boolean | FALSE | true or false whether to use new soa for the work/school DC model |
| Core model run: Non-mandatory destination (4.3.3, 4.4.2) | nmdc.use.new.soa | Boolean | FALSE | true or false whether to use new soa for the non-mandatory DC models |
| Core model run: Stop location (5.3) | slc.use.new.soa | Boolean | FALSE | true or false whether to use new soa for the stop location chocie models |
| Core model run: Usual workplace/school (2.2) | uwsl.run.workLocChoice | Boolean | TRUE | True or False whether to run this model component |
| Core model run: Usual workplace/school (2.2) | uwsl.run.schoolLocChoice | Boolean | TRUE | True or False whether to run this model component |
| Core model run: Usual workplace/school (2.2) | uwsl.write.results | Boolean | TRUE | True of False whether to write out usual work and school location results |
| Core model run: Usual workplace/school (2.2) | uwsl.ShadowPricing.Work.MaximumIterations | Integer | 1 | maximum number of iterations for work shadow price |
| Core model run: Usual workplace/school (2.2) | uwsl.ShadowPricing.School.MaximumIterations | Integer | 1 | maximum number of iterations for school shadow price |
| Core model run: Usual workplace/school (2.2) | uwsl.ShadowPricing.OutputFile | String | output/ShadowPricingOutput.csv | output file name for shadow price |
| Core model run: Matrix manager | RunModel.Clear.MatrixMgr.At.Start | Boolean | FALSE | Clear matrix manager at the start of run? |
|  |  |  |  |  |
| Input properties | maz.tap.tapLines | String | outputs/tapLines.csv | TAP lines file |
| Input properties | maz.tap.distance.file | String | outputs/tap2maz\_Walk.csv | TAP to MAZ distances file |
| Input properties | maz.maz.distance.file | String | outputs/maz2maz\_Walk.csv | MAZ to MAZ walk distances file |
| Input properties | maz.maz.bike.distance.file | String | outputs/maz2maz\_Bike.csv | MAZ to MAZ bike distances file |
| Input properties: Population synthesis (1) | PopulationSynthesizer.InputToCTRAMP.HouseholdFile | String | input/households.csv | location of popsyn households file |
| Input properties: Population synthesis (1) | PopulationSynthesizer.InputToCTRAMP.PersonFile | String | input/persons.csv | location of popsyn persons file |
|  |  |  |  |  |
| Output properties | Results.WriteDataToFiles | Boolean | TRUE | write data to files? |
| Output properties | Results.HouseholdDataFile | String | output/householdData.csv | output name of household data file |
| Output properties | Results.PersonDataFile | String | output/personData.csv | output name of person data file |
| Output properties | Results.IndivTourDataFile | String | output/indivTourData.csv | output name of individual tour data file |
| Output properties | Results.JointTourDataFile | String | output/jointTourData.csv | output name of joint tour data file |
| Output properties | Results.IndivTripDataFile | String | output/indivTripData.csv | output name of individual trip data file |
| Output properties | Results.JointTripDataFile | String | output/jointTripData.csv | output name of joint trip data file |
| Output properties | Results.CBDFile | String | output/cbdParking.csv | output name of CBD parking file |
| Output properties | Results.PNRFile | String | output/pnrParking.csv | output name of PNR parking file |
| Output properties | Results.WriteDataToDatabase | Boolean | FALSE | write data to a database (does not currently work) |
| Output properties | Results.HouseholdTable | String | household\_data | output name of household data file in database |
| Output properties | Results.PersonTable | String | person\_data | output name of person data file in database |
| Output properties | Results.IndivTourTable | String | indiv\_tour\_data | output name of individual tour data file in database |
| Output properties | Results.JointTourTable | String | joint\_tour\_data | output name of joint tour data file in database |
| Output properties | Results.IndivTripTable | String | indiv\_trip\_data | output name of individual trip data file in database |
| Output properties | Results.JointTripTable | String | joint\_trip\_data | output name of joint trip data file in database |
| Output properties | Results.AutoTripMatrix | String | output/autoTrips | output name of auto trip matrix |
| Output properties | Results.TranTripMatrix | String | output/tranTrips | output name of transit trip matrix |
| Output properties | Results.NMotTripMatrix | String | output/nmotTrips | output name of non-motorized trip matrix |
| Output properties | Results.OthrTripMatrix | String | output/othrTrips | output name of other modes trip matrix |
| Output properties | Results.PNRFile | String | output/PNRByTAP\_Vehicles.csv | output name of PNR by TAP Vehicle Trip file |
| Output properties | Results.CBDFile | String | output/CBDByMGRA\_Vehicles.csv | output name of CBD by MGRA Vehicle Trip file |
| Output properties: AO (3.2) | Results.AutoOwnership | String | output/aoResults.csv | auto ownership output file name and location |
| Output properties: AO (3.2) | read.pre.ao.results | Boolean | FALSE | read in the old pre-auto ownership results file |
| Output properties: AO (3.2) | read.pre.ao.filename | String | output/aoResults\_pre.csv | pre auto ownership output file nAme and location |
| Output properties: Work/school location (2.2) | Results.UsualWorkAndSchoolLocationChoice | String | output/wsLocResults.csv | usual work and school location output file name and location |
| Output properties: Work/school location (2.2) | read.uwsl.results | Boolean | FALSE | Read in the old uwsl results? |
| Output properties: Work/school location (2.2) | read.uwsl.filename | String | output/wsLocResults\_1.csv | old uwsl result file name and location to read in |
| Output properties: Work/school location (2.2) | workSchoolSegments.definitions | String | workSchoolSegments.definitions | Work school segment definitions file |
|  |  |  |  |  |
| Core model UECs: Accessibility | acc.uec.file | String | %project.folder%/uec/Accessibilities.xls | Accessibilities.xls location |
| Core model UECs: Accessibility | acc.data.page | Integer | 0 | Accessibilities data page |
| Core model UECs: Accessibility | acc.transit.offpeak.page | Integer | 1 | Accessibilities offpeak page |
| Core model UECs: Accessibility | acc.transit.peak.page | Integer | 2 | Accessibilities peak page |
| Core model UECs: Accessibility | acc.transit.walkaccess.page | Integer | 3 | Accessibilities walk access page |
| Core model UECs: Accessibility | acc.transit.driveaccess.page | Integer | 4 | Accessibilities drive access page |
| Core model UECs: Accessibility | acc.sov.offpeak.page | Integer | 5 | Accessibilities SOV offpeak |
| Core model UECs: Accessibility | acc.sov.peak.page | Integer | 6 | Accessibilities SOV peak |
| Core model UECs: Accessibility | acc.hov.offpeak.page | Integer | 7 | Accessibilities HOV offpeak |
| Core model UECs: Accessibility | acc.hov.peak.page | Integer | 8 | Accessibilities HOV peak |
| Core model UECs: Accessibility | acc.nonmotorized.page | Integer | 9 | Accessibilities non-motorized |
| Core model UECs: Accessibility | acc.constants.page | Integer | 10 | Accessibilities constants |
| Core model UECs: Accessibility | acc.sizeTerm.page | Integer | 11 | Accessibilities size terms |
| Core model UECs: Accessibility | acc.schoolSizeTerm.page | Integer | 12 | Accessibilities school size terms |
| Core model UECs: Accessibility | acc.workerSizeTerm.page | Integer | 13 | Accessibilities worker size terms |
| Core model UECs: Accessibility | acc.dcUtility.uec.file | String | %project.folder%/uec/Accessibilities\_DC.xls | Accessibilities\_DC.xls location |
| Core model UECs: Accessibility | acc.dcUtility.data.page | Integer | 0 | DC Accessibilities data page |
| Core model UECs: Accessibility | acc.dcUtility.page | Integer | 1 | DC Accessibilities utility page |
| Core model UECs: Accessibility | accessibility.alts.file | String | Acc\_alts.csv | Accessibilities alternatives |
| Core model UECs: Accessibility | acc.output.file | String | input/accessibilities.csv | Accessibilities.csv location |
| Core model UECs: Accessibility | acc.mandatory.uec.file | String | %project.folder%/uec/MandatoryAccess.xls | MandatoryAccess.xls location |
| Core model UECs: Accessibility | acc.mandatory.data.page | Integer | 0 | Mandatory Access data page |
| Core model UECs: Accessibility | acc.mandatory.auto.page | Integer | 1 | Mandatory Access auto page |
| Core model UECs: Accessibility | acc.mandatory.autoLogsum.page | Integer | 2 | Mandatory Access autoLogsum page |
| Core model UECs: Accessibility | acc.mandatory.bestWalkTransit.page | Integer | 3 | Mandatory Access best Walk Transit page |
| Core model UECs: Accessibility | acc.mandatory.bestDriveTransit.page | Integer | 4 | Mandatory Access best Drive Transit page |
| Core model UECs: Accessibility | acc.mandatory.transitLogsum.page | Integer | 5 | Mandatory Access transit logsum page |
| Core model UECs: Accessibility |  |  |  |  |
| Core model UECs: Accessibility | acc.jppf | Boolean | FALSE | Run Accessibilities calculation with JPPF |
| Core model UECs: Accessibility | acc.without.jppf.numThreads | Boolean | FALSE | Run Accessibilities calculation with JPPF |
| Core model UECs: Accessibility | acc.destination.sampleRate | Double | 1 | Sample destinations to improve runtimes |
| Core model UECs: Accessibility | acc.jppf.packet.size | Integer | 100 | Accessibilities calculation with JPPF packet size (zone) |
| Core model UECs: AO (3.2) | ao.uec.file | String | AutoOwnership.xls | File name of auto ownership UEC |
| Core model UECs: AO (3.2) | ao.data.page | Integer | 0 | Auto ownership UEC data page |
| Core model UECs: AO (3.2) | ao.model.page | Integer | 1 | Auto ownership UEC utility page |
| Core model UECs: Work/school location (2.2) | uwsl.dc.uec.file | String | TourDestinationChoice.xls | File Name of Tour Destination Choice UEC |
| Core model UECs: Work/school location (2.2) | uwsl.dc2.uec.file | String | TourDestinationChoice2.xls | File Name of Tour Destination Choice 2 UEC |
| Core model UECs: Work/school location (2.2) | uwsl.soa.uec.file | String | DestinationChoiceAlternativeSample.xls | File Name of Destination Choice Alternative Sample UEC |
| Core model UECs: Work/school location (2.2) | uwsl.soa.alts.file | String | DestinationChoiceAlternatives.csv | File name of the alternatives (MGRAs) available to the destination choice models (part of the model design; this should not be changed) |
| Core model UECs: Work/school location (2.2) | uwsl.work.soa.SampleSize | Integer | 30 | Sample size of Work Destination Choice |
| Core model UECs: Work/school location (2.2) | uwsl.school.soa.SampleSize | Integer | 30 | Sample size of School Destination Choice |
| Core model UECs: Work/school location (2.2) | work.soa.uec.file | String | TourDcSoaDistance.xls | File Name of Tour Distance DC SOA UEC for Work Purpose, includes TAZ Size in the expressions |
| Core model UECs: Work/school location (2.2) | work.soa.uec.data | Integer | 0 | Work Tour Distance SOA UEC data page |
| Core model UECs: Work/school location (2.2) | work.soa.uec.model | Integer | 1 | Work Tour Distance SOA UEC utility page |
| Core model UECs: Work/school location (2.2) | univ.soa.uec.file | String | TourDcSoaDistanceNoSchoolSize.xls | File Name of Tour Distance DC SOA UEC for School Purpose; school purposes do not include TAZ Size in the expressions so that the utilities can be stored as exponentiated distance utility matrices for university, and then multiplied by the university segment size terms |
| Core model UECs: Work/school location (2.2) | univ.soa.uec.data | Integer | 0 | University Tour Distance SOA UEC data page |
| Core model UECs: Work/school location (2.2) | univ.soa.uec.model | Integer | 1 | University Tour Distance SOA UEC utility page |
| Core model UECs: Work/school location (2.2) | hs.soa.uec.file | String | TourDcSoaDistanceNoSchoolSize.xls | File Name of Tour Distance DC SOA UEC for School Purpose; school purposes do not include TAZ Size in the expressions so that the utilities can be stored as exponentiated distance utility matrices for high school, and then multiplied by the high school segment size terms |
| Core model UECs: Work/school location (2.2) | hs.soa.uec.data | Integer | 0 | High School Tour Distance SOA UEC data page |
| Core model UECs: Work/school location (2.2) | hs.soa.uec.model | Integer | 2 | High School Tour Distance SOA UEC utility page |
| Core model UECs: Work/school location (2.2) | gs.soa.uec.file | String | TourDcSoaDistanceNoSchoolSize.xls | File Name of Tour Distance DC SOA UEC for School Purpose; school purposes do not include TAZ Size in the expressions so that the utilities can be stored as exponentiated distance utility matrices for grade school, and then multiplied by the grade school segment size terms |
| Core model UECs: Work/school location (2.2) | gs.soa.uec.data | Integer | 0 | Grade School Tour Distance SOA UEC data page |
| Core model UECs: Work/school location (2.2) | gs.soa.uec.model | Integer | 3 | Grade School Tour Distance SOA UEC utility page |
| Core model UECs: Work/school location (2.2) | ps.soa.uec.file | String | TourDcSoaDistanceNoSchoolSize.xls | File Name of Tour Distance DC SOA UEC for School Purpose; school purposes do not include TAZ Size in the expressions so that the utilities can be stored as exponentiated distance utility matrices for preschool, and then multiplied by the preschool segment size terms |
| Core model UECs: Work/school location (2.2) | ps.soa.uec.data | Integer | 0 | Preschool Tour Distance SOA UEC data page |
| Core model UECs: Work/school location (2.2) | ps.soa.uec.model | Integer | 4 | Prescehool Tour Distance SOA UEC utility page |
|  |  |  |  |  |
| Core model UECs: Work/school location (2.2) | UsualWorkLocationChoice.ShadowPrice.Input.File | String | input/ShadowPricingOutput\_work\_39.csv | File Name Work Location shadow price |
| Core model UECs: Work/school location (2.2) | UsualSchoolLocationChoice.ShadowPrice.Input.File | String | input/ShadowPricingOutput\_school\_19.csv | File Name School Location shadow price |
|  |  |  |  |  |
| Core model UECs: Transponder choice | tc.choice.avgtts.file | String | input/tc\_avgtt.csv | File name of average travel times for transponder ownership |
| Core model UECs: Transponder choice | tc.uec.file | String | TransponderOwnership.xls | File name of transponder ownership UEC |
| Core model UECs: Transponder choice | tc.data.page | Integer | 0 | Transponder ownership UEC data page |
| Core model UECs: Transponder choice | tc.model.page | Integer | 1 | Transponder ownership UEC utility page |
| Core model UECs: Free parking choice |  |  |  |  |
| Core model UECs: Free parking choice | fp.uec.file | String | ParkingProvision.xls | File name of parking provision UEC |
| Core model UECs: Free parking choice | fp.data.page | Integer | 0 | Parking Provision UEC data page |
| Core model UECs: Free parking choice | fp.model.page | Integer | 1 | Parking Provision UEC utility page |
|  |  |  |  |  |
| Core model UECs: CDAP (4.1) | cdap.uec.file | String | CoordinatedDailyActivityPattern.xls | File name of CDAP UEC |
| Core model UECs: CDAP (4.1) | cdap.data.page | Integer | 0 | CDAP UEC data page |
| Core model UECs: CDAP (4.1) | cdap.one.person.page | Integer | 1 | CDAP UEC utility for one person page |
| Core model UECs: CDAP (4.1) | cdap.two.person.page | Integer | 2 | CDAP UEC utility for 2 persons page |
| Core model UECs: CDAP (4.1) | cdap.three.person.page | Integer | 3 | CDAP UEC utility for 3 persons page |
| Core model UECs: CDAP (4.1) | cdap.all.person.page | Integer | 4 | CDAP UEC utility for All member interation page |
| Core model UECs: CDAP (4.1) | cdap.joint.page | Integer | 5 | CDAP UEC utility for joint tours page |
| Core model UECs: MTF (4.2.1) | imtf.uec.file | String | MandatoryTourFrequency.xls | File name of Mandatory tour frequency UEC |
| Core model UECs: MTF (4.2.1) | imtf.data.page | Integer | 0 | Mandatory tour frequency UEC data page |
| Core model UECs: MTF (4.2.1) | imtf.model.page | Integer | 1 | mandatory tour frequency UEC utility page |
| Core model UECs: non-mand tour DC (4.3.3, 4.4.2, 4.5.4) | nonSchool.soa.uec.file | String | TourDcSoaDistance.xls | File Name of Tour Distance DC SOA UEC for Non Work/School Purposes, includes TAZ Size in the expressions |
| Core model UECs: non-mand tour DC (4.3.3, 4.4.2, 4.5.4) | escort.soa.uec.data | Integer | 0 | Escort Tour Distance SOA UEC data page |
| Core model UECs: non-mand tour DC sampling (4.3.3, 4.4.2, 4.5.4) | escort.soa.uec.model | Integer | 2 | Escort Tour Distance SOA UEC utility page |
| Core model UECs: non-mand tour DC sampling (4.3.3, 4.4.2, 4.5.4) | other.nonman.soa.uec.data | Integer | 0 | Other Non-mandatory Tour Distance SOA UEC data page |
| Core model UECs: non-mand tour DC sampling (4.3.3, 4.4.2, 4.5.4) | other.nonman.soa.uec.model | Integer | 3 | Other Non-mandatory Tour Distance SOA UEC utility page |
| Core model UECs: non-mand tour DC sampling (4.3.3, 4.4.2, 4.5.4) | atwork.soa.uec.data | Integer | 0 | At-Work Sub-Tour Distance SOA UEC data page |
| Core model UECs: non-mand tour DC sampling (4.3.3, 4.4.2, 4.5.4) | atwork.soa.uec.model | Integer | 4 | At-Work Sub-Tour Distance SOA UEC utility page |
| Core model UECs: non-mand tour DC sampling (4.3.3, 4.4.2, 4.5.4) | soa.taz.dist.alts.file | String | SoaTazDistAlts.csv | File name of Sample of Alternatives of TAZs |
| Core model UECs: non-mand tour DC sampling (4.3.3, 4.4.2, 4.5.4) |  |  |  |  |
| Core model UECs: non-mand tour DC sampling (4.3.3, 4.4.2, 4.5.4) | nmdc.dist.alts.file | String | NonMandatoryTlcAlternatives.csv | File name of non-mandatory tour alternatives |
| Core model UECs: non-mand tour DC sampling (4.3.3, 4.4.2, 4.5.4) | nmdc.soa.alts.file | String | DestinationChoiceAlternatives.csv | File name of the alternatives (MGRAs) available to the destination choice models (part of the model design; this should not be changed) |
| Core model UECs: non-mand tour DC sampling (4.3.3, 4.4.2, 4.5.4) | nmdc.soa.SampleSize | Integer | 30 | Sample size of non-mandatory Destination choice |
| Core model UECs: non-mand tour DC (4.3.3, 4.4.2, 4.5.4) | nmdc.uec.file2 | String | TourDestinationChoice2.xls | File Name of Tour Destination Choice 2 UEC |
| Core model UECs: non-mand tour DC (4.3.3, 4.4.2, 4.5.4) | nmdc.uec.file | String | TourDestinationChoice.xls | File Name of Tour Destination Choice UEC |
| Core model UECs: non-mand tour DC (4.3.3, 4.4.2, 4.5.4) | nmdc.data.page | Integer | 0 | Non-mandatory Tour DC UEC data page |
| Core model UECs: non-mand tour DC (4.3.3, 4.4.2, 4.5.4) | nmdc.escort.model.page | Integer | 7 | Escort Tour Destination Choice UEC utility page |
| Core model UECs: non-mand tour DC (4.3.3, 4.4.2, 4.5.4) | nmdc.shop.model.page | Integer | 8 | Shop Tour Destination Choice UEC utility page |
| Core model UECs: non-mand tour DC (4.3.3, 4.4.2, 4.5.4) | nmdc.maint.model.page | Integer | 9 | Maintenance Tour Destination Choice UEC utility page |
| Core model UECs: non-mand tour DC (4.3.3, 4.4.2, 4.5.4) | nmdc.eat.model.page | Integer | 10 | Eating Out Tour Destination Choice UEC utility page |
| Core model UECs: non-mand tour DC (4.3.3, 4.4.2, 4.5.4) | nmdc.visit.model.page | Integer | 11 | Visiting Tour Destination Choice UEC utility page |
| Core model UECs: non-mand tour DC (4.3.3, 4.4.2, 4.5.4) | nmdc.discr.model.page | Integer | 12 | Discretionary Tour Destination Choice UEC utility page |
| Core model UECs: non-mand tour DC (4.3.3, 4.4.2, 4.5.4) | nmdc.atwork.model.page | Integer | 13 | At-Work Sub-Tour Destination Choice UEC utility page |
| Core model UECs: non-mand tour DC (4.3.3, 4.4.2, 4.5.4) | nmdc.soa.uec.file | String | DestinationChoiceAlternativeSample.xls | File Name of Destination Choice Alternative Sample UEC |
| Core model UECs: non-mand tour DC (4.3.3, 4.4.2, 4.5.4) | nmdc.soa.data.page | Integer | 0 | Non-mandatory TOUR SOA UEC data page |
| Core model UECs: non-mand tour DC (4.3.3, 4.4.2, 4.5.4) | nmdc.soa.escort.model.page | Integer | 6 | Escort TOUR SOA UEC utility page |
| Core model UECs: non-mand tour DC (4.3.3, 4.4.2, 4.5.4) | nmdc.soa.shop.model.page | Integer | 7 | Shop TOUR SOA UEC utility page |
| Core model UECs: non-mand tour DC (4.3.3, 4.4.2, 4.5.4) | nmdc.soa.maint.model.page | Integer | 7 | Maintenance TOUR SOA UEC utility page |
| Core model UECs: non-mand tour DC (4.3.3, 4.4.2, 4.5.4) | nmdc.soa.eat.model.page | Integer | 7 | Eating Out TOUR SOA UEC utility page |
| Core model UECs: non-mand tour DC (4.3.3, 4.4.2, 4.5.4) | nmdc.soa.visit.model.page | Integer | 7 | Visiting TOUR SOA UEC utility page |
| Core model UECs: non-mand tour DC (4.3.3, 4.4.2, 4.5.4) | nmdc.soa.discr.model.page | Integer | 7 | Discretionary TOUR SOA UEC utility page |
| Core model UECs: non-mand tour DC (4.3.3, 4.4.2, 4.5.4) | nmdc.soa.atwork.model.page | Integer | 8 | At-Work Sub-Tour SOA UEC utility page |
|  |  |  |  |  |
| Core model UECs: Tour mode (4.2.4, 4.5.3, 4.3.5, 4.4.4) | tourModeChoice.uec.file | String | TourModeChoice.xls | File name of Tour Mode choice UEC |
| Core model UECs: Tour mode (4.2.4, 4.5.3, 4.3.5, 4.4.4) | tourModeChoice.maint.model.page | Integer | 4 | Maintenance Tour Mode Choice UEC utility page |
| Core model UECs: Tour mode (4.2.4, 4.5.3, 4.3.5, 4.4.4) | tourModeChoice.discr.model.page | Integer | 5 | Discretionary Tour Mode Choice UEC utility page |
| Core model UECs: Tour mode (4.2.4, 4.5.3, 4.3.5, 4.4.4) | tourModeChoice.atwork.model.page | Integer | 6 | At-Work Sub-Tour Mode Choice UEC utility page |
|  |  |  |  |  |
| Core model UECs: TOD (4.2.2, 4.5.2, 4.3.4, 4.4.3) | departTime.uec.file | String | TourDepartureAndDuration.xls | File name of Tour TOD Choice UEC |
| Core model UECs: TOD (4.2.2, 4.5.2, 4.3.4, 4.4.3) | departTime.data.page | Integer | 0 | Tour TOD Choice UEC data page |
| Core model UECs: TOD (4.2.2, 4.5.2, 4.3.4, 4.4.3) | departTime.work.page | Integer | 1 | Work Tour TOD Choice UEC utility page |
| Core model UECs: TOD (4.2.2, 4.5.2, 4.3.4, 4.4.3) | departTime.univ.page | Integer | 2 | University Tour TOD Choice UEC utility page |
| Core model UECs: TOD (4.2.2, 4.5.2, 4.3.4, 4.4.3) | departTime.school.page | Integer | 3 | School Tour TOD Choice UEC utility page |
| Core model UECs: TOD (4.2.2, 4.5.2, 4.3.4, 4.4.3) | departTime.escort.page | Integer | 4 | Escort Tour TOD Choice UEC utility page |
| Core model UECs: TOD (4.2.2, 4.5.2, 4.3.4, 4.4.3) | departTime.shop.page | Integer | 5 | Shop Tour TOD Choice UEC utility page |
| Core model UECs: TOD (4.2.2, 4.5.2, 4.3.4, 4.4.3) | departTime.maint.page | Integer | 6 | Maintenance Tour TOD Choice UEC utility page |
| Core model UECs: TOD (4.2.2, 4.5.2, 4.3.4, 4.4.3) | departTime.eat.page | Integer | 7 | Eating Out Tour TOD Choice UEC utility page |
| Core model UECs: TOD (4.2.2, 4.5.2, 4.3.4, 4.4.3) | departTime.visit.page | Integer | 8 | Visiting Tour TOD Choice UEC utility page |
| Core model UECs: TOD (4.2.2, 4.5.2, 4.3.4, 4.4.3) | departTime.discr.page | Integer | 9 | Discretionary Tour TOD Choice UEC utility page |
| Core model UECs: TOD (4.2.2, 4.5.2, 4.3.4, 4.4.3) | departTime.atwork.page | Integer | 10 | At-Work Sub-Tour TOD Choice UEC utility page |
| Core model UECs: TOD (4.2.2, 4.5.2, 4.3.4, 4.4.3) | departTime.alts.file | String | DepartureTimeAndDurationAlternatives.csv | File name of Departure time and duration alternatives |
| Core model UECs: JTF (4.3.1) | jtfcp.uec.file | String | JointTourFrequency.xls | File name of Joint Tour Frequency UEC |
| Core model UECs: JTF (4.3.1) | jtfcp.alternatives.file | String | JointAlternatives.csv | File name of joint tour alternatives by purpose and party composition combinations |
| Core model UECs: JTF (4.3.1) | jtfcp.data.page | Integer | 0 | Joint Tour Frequency UEC data page |
| Core model UECs: JTF (4.3.1) | jtfcp.freq.comp.page | Integer | 1 | Joint Tour Frequency UEC utility composition page |
| Core model UECs: JTF (4.3.1) | jtfcp.participate.page | Integer | 2 | Joint Tour Frequency UEC utility participation page |
| Core model UECs: non-mand freq (4.4.1) | inmtf.uec.file | String | NonMandatoryIndividualTourFrequency.xls | File name of Individual non-mandatory tour frequency UEC |
| Core model UECs: non-mand freq (4.4.1) | inmtf.FrequencyExtension.ProbabilityFile | String | IndividualNonMandatoryTourFrequencyExtensionProbabilities\_p1.csv | File name of Individual non-mandatory tour frequency extension probabilities |
| Core model UECs: non-mand freq (4.4.1) | IndividualNonMandatoryTourFrequency.AlternativesList.InputFile | String | IndividualNonMandatoryTourFrequencyAlternatives.csv | File name of individual non-mandatory tour frequency alternatives (combinations) |
| Core model UECs: non-mand freq (4.4.1) | inmtf.data.page | Integer | 0 | Individual Non-mandatory tour frequency UEC data page |
| Core model UECs: non-mand freq (4.4.1) | inmtf.perstype1.page | Integer | 1 | Individual Non-mandatory tour frequency UEC utility for Full time workers page |
| Core model UECs: non-mand freq (4.4.1) | inmtf.perstype2.page | Integer | 2 | Individual Non-mandatory tour frequency UEC utility for Part time workers page |
| Core model UECs: non-mand freq (4.4.1) | inmtf.perstype3.page | Integer | 3 | Individual Non-mandatory tour frequency UEC utility for University students page |
| Core model UECs: non-mand freq (4.4.1) | inmtf.perstype4.page | Integer | 4 | Individual Non-mandatory tour frequency UEC utility for Non-workers page |
| Core model UECs: non-mand freq (4.4.1) | inmtf.perstype5.page | Integer | 5 | Individual Non-mandatory tour frequency UEC utility for Retirees page |
| Core model UECs: non-mand freq (4.4.1) | inmtf.perstype6.page | Integer | 6 | Individual Non-mandatory tour frequency UEC utility for Driving students page |
| Core model UECs: non-mand freq (4.4.1) | inmtf.perstype7.page | Integer | 7 | Individual Non-mandatory tour frequency UEC utility for Pre-driving students page |
| Core model UECs: non-mand freq (4.4.1) | inmtf.perstype8.page | Integer | 8 | Individual Non-mandatory tour frequency UEC utility for Preschool students page |
| Core model UECs: at-work subtour freq (4.5.1) | awtf.uec.file |  | AtWorkSubtourFrequency.xls | File name of at-work sub-tour frequency UEC |
| Core model UECs: at-work subtour freq (4.5.1) | awtf.data.page | Integer | 0 | At-Work Sub-Tour Frequency UEC Data page |
| Core model UECs: at-work subtour freq (4.5.1) | awtf.model.page | Integer | 1 | At-Work Sub-Tour Frequency UEC Utility page |
| Core model UECs: Stop freq (5.1) | stf.uec.file | String | StopFrequency.xls | File name of Stop Frequency UEC |
| Core model UECs: Stop freq (5.1) | stf.purposeLookup.proportions | String | StopPurposeLookupProportions.csv | File name of Stop Purpose Lookup proportions |
| Core model UECs: Stop freq (5.1) | stf.data.page | Integer | 0 | Stop Frequency UEC data page |
| Core model UECs: Stop freq (5.1) | stf.work.page | Integer | 1 | Stop Frequency for Work Tour UEC utility page |
| Core model UECs: Stop freq (5.1) | stf.univ.page | Integer | 2 | Stop Frequency for University Tour UEC utility page |
| Core model UECs: Stop freq (5.1) | stf.school.page | Integer | 3 | Stop Frequency for School Tour UEC utility page |
| Core model UECs: Stop freq (5.1) | stf.escort.page | Integer | 4 | Stop Frequency for Escort Tour UEC utility page |
| Core model UECs: Stop freq (5.1) | stf.shop.page | Integer | 5 | Stop Frequency for Shop Tour UEC utility page |
| Core model UECs: Stop freq (5.1) | stf.maint.page | Integer | 6 | Stop Frequency for Maintenance Tour UEC utility page |
| Core model UECs: Stop freq (5.1) | stf.eat.page | Integer | 7 | Stop Frequency for Eating Out Tour UEC utility page |
| Core model UECs: Stop freq (5.1) | stf.visit.page | Integer | 8 | Stop Frequency for Visiting Tour UEC utility page |
| Core model UECs: Stop freq (5.1) | stf.discr.page | Integer | 9 | Stop Frequency for Discretionary Tour UEC utility page |
| Core model UECs: Stop freq (5.1) | stf.subtour.page | Integer | 10 | Stop Frequency for At-Work Sub-Tour UEC utility page |
| Core model UECs: Stop location (5.3) | slc.uec.file | String | StopLocationChoice.xls | File Name of Stop Location Choice UEC |
| Core model UECs: Stop location (5.3) | slc.uec.data.page | Integer | 0 | Stop Location Choice UEC data page |
| Core model UECs: Stop location (5.3) | slc.mandatory.uec.model.page | Integer | 1 | Stop Location Choice for Mandatory Tours UEC utility page |
| Core model UECs: Stop location (5.3) | slc.maintenance.uec.model.page | Integer | 2 | Stop Location Choice for Maintenance Tours UEC utility page |
| Core model UECs: Stop location (5.3) | slc.discretionary.uec.model.page | Integer | 3 | Stop Location Choice for Discretionary Tours UEC utility page |
| Core model UECs: Stop location (5.3) | slc.alts.file | String | SlcAlternatives.csv | File name of stop location choice alternatives |
| Core model UECs: Stop location (5.3) | slc.soa.alts.file | String | DestinationChoiceAlternatives.csv | File name of the alternatives (MGRAs) available to the destination choice models (part of the model design; this should not be changed) |
| Core model UECs: Stop location (5.3) | auto.slc.soa.distance.uec.file | String | SlcSoaDistanceUtility.xls | File name of Stop Location Sample of Alternatives Choice UEC for tourmodes other than walk or bike - for transit, availability of stop for transit is set in java code |
| Core model UECs: Stop location (5.3) | auto.slc.soa.distance.data.page | Integer | 0 | Stop Location SOA Choice UEC data page |
| Core model UECs: Stop location (5.3) | auto.slc.soa.distance.model.page | Integer | 1 | Stop Location SOA Choice UEC utility page |
| Core model UECs: Stop location (5.3) | slc.soa.size.uec.file | String | SlcSoaSize.xls | File Name of Stop Location Choice Size Terms UEC |
| Core model UECs: Stop location (5.3) | slc.soa.size.uec.data.page | Integer | 0 | Stop Location Choice Size terms UEC data page |
| Core model UECs: Stop location (5.3) | slc.soa.size.uec.model.page | Integer | 1 | Stop Location Choice Size terms UEC utility page |
| Core model UECs: Stop departure (5.4) | stop.depart.arrive.proportions | String | StopDepartArriveProportions.csv | File name of Stop Location Time of Day proportions |
| Core model UECs: Trip mode (6.1) | tripModeChoice.uec.file | String | TripModeChoice.xls | File name of Trip mode choice UEC |
| Core model UECs: Parking location choice | plc.uec.file | String | ParkLocationChoice.xls | File name of Parking Location Choice UEC |
| Core model UECs: Parking location choice | plc.uec.data.page | Integer | 0 | Parking Location Choice UEC data page |
| Core model UECs: Parking location choice | plc.uec.model.page | Integer | 1 | Parking Location Choice UEC utility page |
| Core model UECs: Parking location choice | plc.alts.corresp.file | String | ParkLocationAlts.csv | File name of parking location alternatives (MAZs) |
| Core model UECs: Parking location choice | plc.alts.file | String | ParkLocationSampleAlts.csv | File name of parking location sample of alternatives |
| Core model UECs: Parking location choice | mgra.avg.cost.output.file | String | mgraParkingCost.csv | File name of average parking costs by MAZ |
| Core model UECs: Parking location choice | mgra.avg.cost.trace.zone | Integer | 1814 | MAZ parking cost trace zone |
| Core model UECs: Parking location choice | mgra.max.parking.distance | Integer | 0.75 | Max parking walk distance |
| Core model UECs: Parking location choice | mgra.avg.cost.dist.coeff.work | Float | -8.6 | Parking location model coefficient for walking distance to destination for Work purpose |
| Core model UECs: Parking location choice | mgra.avg.cost.dist.coeff.other | Float | -4.9 | Parking location model coefficient for walking distance to destination for other purposes |
| Core model UECs: Parking location choice | park.cost.reimb.mean | Float | -0.05 | Parking location model mean parking cost reimbursement |
| Core model UECs: Parking location choice | park.cost.reimb.std.dev | Float | 0.54 | Parking location model standard deviation for parking cost reimbursement |
| Core model UECs: Transit path finder | utility.bestTransitPath.uec.file | String | BestTransitPathUtility.xls | File name of best transit path UEC |
| Core model UECs: Transit path finder | utility.bestTransitPath.data.page | Integer | 0 | Best Transit Path UEC data page |
| Core model UECs: Transit path finder | utility.bestTransitPath.tapToTap.page | Integer | 1 | Best Transit Path UEC for TAP to TAP utility page |
| Core model UECs: Transit path finder | utility.bestTransitPath.walkAccess.page | Integer | 2 | Best Transit Path UEC for Walk Access to Transit utility page |
| Core model UECs: Transit path finder | utility.bestTransitPath.driveAccess.page | Integer | 3 | Best Transit Path UEC for Drive Access to Transit utility page |
| Core model UECs: Transit path finder | utility.bestTransitPath.walkEgress.page | Integer | 4 | Best Transit Path UEC for Walk Egress to Transit utility page |
| Core model UECs: Transit path finder | utility.bestTransitPath.driveEgress.page | Integer | 5 | Best Transit Path UEC for Drive Egress to Transit utility page |
| Core model UECs: Auto skimming | skims.auto.uec.file | String | AutoSkims.xls | File name of Auto Skims UEC |
| Core model UECs: Auto skimming | skims.auto.data.page | Integer | 0 | Auto Skims data page |
| Core model UECs: Auto skimming | skims.auto.ea.page | Integer | 1 | Auto skims Early AM utility page |
| Core model UECs: Auto skimming | skims.auto.am.page | Integer | 2 | Auto skims AM utility page |
| Core model UECs: Auto skimming | skims.auto.md.page | Integer | 3 | Auto skims MD utility page |
| Core model UECs: Auto skimming | skims.auto.pm.page | Integer | 4 | Auto skims PM utility page |
| Core model UECs: Auto skimming | skims.auto.ev.page | Integer | 5 | Auto skims Evening utility page |
| Core model UECs: TAZ distances | taz.distance.uec.file | String | tazDistance.xls | File name of TAZ Distance UEC |
| Core model UECs: TAZ distances | taz.distance.data.page | Integer | 0 | TAZ Distance UEC data page |
| Core model UECs: TAZ distances | taz.od.distance.ea.page | Integer | 1 | TAZ Distance UEC Early AM utility page |
| Core model UECs: TAZ distances | taz.od.distance.am.page | Integer | 2 | TAZ Distance UEC AM utility page |
| Core model UECs: TAZ distances | taz.od.distance.md.page | Integer | 3 | TAZ Distance UEC MD utility page |
| Core model UECs: TAZ distances | taz.od.distance.pm.page | Integer | 4 | TAZ Distance UEC PM utility page |
| Core model UECs: TAZ distances | taz.od.distance.ev.page | Integer | 5 | TAZ Distance UEC Evening utility page |
| Core model UECs: Transit skimming | skim.walk.transit.walk.uec.file | String | WalkTransitWalkSkims.xls | File name of Walk Transit Walk Skims UEC |
| Core model UECs: Transit skimming | skim.walk.transit.walk.data.page | Integer | 0 | Walk Transit Walk Skims UEC data page |
| Core model UECs: Transit skimming | skim.walk.transit.walk.skim.page | Integer | 1 | Walk Transit Walk Skims UEC skims page |
| Core model UECs: Transit skimming | skim.walk.transit.walk.sets | Integer | 3 | Walk Transit Walk Skims sets |
| Core model UECs: Transit skimming | skim.walk.transit.walk.skims | Integer | 12 | Walk Transit Walk Skims number of skims |
| Core model UECs: Transit skimming | skim.walk.transit.drive.uec.file | String | WalkTransitDriveSkims.xls | File name of Walk Transit drive Skims UEC |
| Core model UECs: Transit skimming | skim.walk.transit.drive.data.page | Integer | 0 | Walk Transit drive Skims UEC data page |
| Core model UECs: Transit skimming | skim.walk.transit.drive.skim.page | Integer | 1 | Walk Transit drive Skims UEC skims page |
| Core model UECs: Transit skimming | skim.walk.transit.drive.sets | Integer | 3 | Walk Transit drive Skims sets |
| Core model UECs: Transit skimming | skim.walk.transit.drive.skims | Integer | 12 | Walk Transit drive Skims number of skims |
| Core model UECs: Transit skimming | skim.drive.transit.walk.uec.file | String | DriveTransitWalkSkims.xls | File name of Drive Transit walk Skims UEC |
| Core model UECs: Transit skimming | skim.drive.transit.walk.data.page | Integer | 0 | Drive Transit walk Skims UEC data page |
| Core model UECs: Transit skimming | skim.drive.transit.walk.skim.page | Integer | 1 | Drive Transit walk Skims UEC skims page |
| Core model UECs: Transit skimming | skim.drive.transit.walk.sets | Integer | 3 | Drive Transit walk Skims sets |
| Core model UECs: Transit skimming | skim.drive.transit.walk.skims | Integer | 12 | Walk Transit drive Skims number of skims |
| Core model UECs: University tour DC | majorUniversityStudent.tour.destination.uec.filename | String | MajorUniversityStudentTourDestinationChoice.xls | UEC |
| Core model UECs: University tour DC | majorUniversityStudent.tour.destination.data.sheet | Integer | 0 | Data page |
| Core model UECs: University tour DC | majorUniversityStudent.tour.destination.model.sheet | Integer | 1 | Mode page |
| Core model UECs: University tour DC | majorUniversityStudent.tour.destination.size.sheet | Integer | 2 | Size terms page |
| Core model UECs: University tour DC | majorUniversityStudent.tour.destination.RNG.offset | Integer | 234234 | Random number offset |
| Core model UECs: School escorting | school.escort.uec.filename | String | SchoolEscorting.xls | UEC |
| Core model UECs: School escorting | school.escort.alts.file | String | SchoolEscortingAlts.csv | UEC alts |
| Core model UECs: School escorting | school.escort.data.sheet | Integer | 0 | Data page |
| Core model UECs: School escorting | school.escort.outbound.model.sheet | Integer | 1 | Mode page |
| Core model UECs: School escorting | school.escort.inbound.conditonal.model.sheet | Integer | 2 | Mode page |
| Core model UECs: School escorting | school.escort.outbound.conditonal.model.sheet | Integer | 3 | Mode page |
| Core model UECs: School escorting | school.escort.RNG.offset | Integer | 384571483 | Random number offset |
| Core model UECs: University ParkingLot Choice | UniversityParkingLotChoiceModel.RunModel | Boolean | false | Run model |
| Core model UECs: University ParkingLot Choice | UniversityParkingLotChoiceModel.UnmetParkingDemand.file | String | %project.folder%/outputs/UnmetParkingDemand.csv | Unmet demand output file |
| Core model UECs: University ParkingLot Choice | UniversityParkingLotChoiceModel.RNG.offset | Integer | 219421 | RNG offset |
| Core model UECs: University ParkingLot Choice | UniversityParkingLotChoiceModel.ParkingLots.file | String | %project.folder%/inputs/Uni\_Parking\_Capacity.csv | Parkings lots input file |
| Core model UECs: University ParkingLot Choice | UniversityParkingLotChoiceModel.TransitConstant | Float | -0.1 | Transit constant |
| Core model UECs: University ParkingLot Choice | UniversityParkingLotChoiceModel.SimulationModel | Boolean | true | use simulation model |
| Core model UECs: University ParkingLot Choice | UniversityParkingLotChoiceModel.Util\_LD | Integer | 1 | utility term |
| Core model UECs: University ParkingLot Choice | UniversityParkingLotChoiceModel.IVTCoefficient | Float | -0.0424 | IVT coefficient |
| Core model UECs: University ParkingLot Choice | UniversityParkingLotChoiceModel.ParkingPriceDampingFactor | Integer | 1 | damping factor |
| Core model UECs: University ParkingLot Choice | UniversityParkingLotChoiceModel.InformalParkingConstant | Array | -1.1,-0.1,0 | informal parking constant faculty\staff, students, and visitors |
| Core model UECs: University ParkingLot Choice | UniversityParkingLotChoiceModel.SpaceTypesConstants.facultyStaff | Array | 0,-3.0,-5.0 | space type constants faculty\staff, students, and visitors |
| Core model UECs: University ParkingLot Choice | UniversityParkingLotChoiceModel.SpaceTypesConstants.student | Array | -999,0,-5.0 | space type constants faculty\staff, students, and visitors |
| Core model UECs: University ParkingLot Choice | UniversityParkingLotChoiceModel.SpaceTypesConstants.visitor | Array | -999,-999,0 | space type constants faculty\staff, students, and visitors |
| Core model UECs: University ParkingLot Choice | Results.UniversityParkingDataFile | String | /outputs/universityParkingData.csv | Output parking file location |
|  |  |  |  |  |
| Best transit path: Data files | tap.data.file | String | outputs/tap\_data.csv | TAP data file |
| Best transit path: Data files | tap.data.tap.column | String | tap | tap field |
| Best transit path: Data files | tap.data.taz.column | String | taz | taz field |
| Best transit path: Data files | tap.data.lotid.column | String | lotid | lotid field |
| Best transit path: Data files | tap.data.capacity.column | String | capacity | capacity field |
| Best transit path: Data files | taz.tap.access.file | String | /outputs/drive\_taz\_tap.csv | Drive times taz to tap file |
| Best transit path: Data files | taz.tap.access.ftaz.column | String | FTAZ | FTAZ field |
| Best transit path: Data files | taz.tap.access.mode.column | String | MODE | MODE field |
| Best transit path: Data files | taz.tap.access.period.column | String | PERIOD | PERIOD field |
| Best transit path: Data files | taz.tap.access.ttap.column | String | TTAP | TTAP field |
| Best transit path: Data files | taz.tap.access.tmaz.column | String | TMAZ | TMAZ field |
| Best transit path: Data files | taz.tap.access.ttaz.column | String | TTAZ | TTAZ field |
| Best transit path: Data files | taz.tap.access.dtime.column | String | DTIME | DTIME field |
| Best transit path: Data files | taz.tap.access.ddist.column | String | DDIST | DDIST field |
| Best transit path: Data files | taz.tap.access.dtoll.column | String | DTOLL | DTOLL field |
| Best transit path: Data files | taz.tap.access.wdist.column | String | WDIST | WDIST field |
|  |  |  |  |  |
| Summit properties | summit.output.directory | String | output/ | File location for Summit output |
| Summit properties | summit.purpose.Work | Integer | 1 | Specify code for Work Purpose |
| Summit properties | summit.purpose.University | Integer | 2 | Specify code for University Purpose |
| Summit properties | summit.purpose.School | Integer | 3 | Specify code for School Purpose |
| Summit properties | summit.purpose.Escort | Integer | 4 | Specify code for Escort Purpose |
| Summit properties | summit.purpose.Shop | Integer | 4 | Specify code for Shop Purpose |
| Summit properties | summit.purpose.Maintenance | Integer | 4 | Specify code for Maintenance Purpose |
| Summit properties | summit.purpose.EatingOut | Integer | 5 | Specify code for Eating out Purpose |
| Summit properties | summit.purpose.Visiting | Integer | 5 | Specify code for Visiting Purpose |
| Summit properties | summit.purpose.Discretionary | Integer | 5 | Specify code for Discretionary Purpose |
| Summit properties | summit.purpose.WorkBased | Integer | 6 | Specify code for At-Work Sub Tour Purpose |
| Summit properties | summit.filename.1 | String | Work | Specify file name for Work Purpose |
| Summit properties | summit.filename.2 | String | University | Specify file name for University Purpose |
| Summit properties | summit.filename.3 | String | School | Specify file name for School Purpose |
| Summit properties | summit.filename.4 | String | Maintenance | Specify file name for Maintenance Purpose |
| Summit properties | summit.filename.5 | String | Discretionary | Specify file name for Discretionary Purpose |
| Summit properties | summit.filename.6 | String | Workbased | Specify file name for At-Work Sub tour Purpose |
| Summit properties | summit.ivt.file.1 | Float | -0.016 | Specify in-vehicle time coefficient for Work Purpose |
| Summit properties | summit.ivt.file.2 | Float | -0.016 | Specify in-vehicle time coefficient for University Purpose |
| Summit properties | summit.ivt.file.3 | Float | -0.01 | Specify in-vehicle time coefficient for School Purpose |
| Summit properties | summit.ivt.file.4 | Float | -0.017 | Specify in-vehicle time coefficient for Maintenance Purpose |
| Summit properties | summit.ivt.file.5 | Float | -0.015 | Specify in-vehicle time coefficient for Discretionary Purpose |
| Summit properties | summit.ivt.file.6 | Float | -0.032 | Specify in-vehicle time coefficient for At-work Sub tour Purpose |
| Summit properties | summit.modes | Integer | 26 | Specify number of modes in the model |
| Summit properties | summit.mode.array | String | 0,0,0,0,0,0,0,0,0, 0, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0 | Specify mode array; 0 for auto modes/school bus mode, 1 for walk transit, 2 for drive transit modes |
| Summit properties | summit.upperEA | Integer | 3 | Specify upper limit code for Early AM time period |
| Summit properties | summit.upperAM | Integer | 9 | Specify upper limit code for AM time period |
| Summit properties | summit.upperMD | Integer | 22 | Specify upper limit code for Midday time period |
| Summit properties | summit.upperPM | Integer | 29 | Specify upper limit code for PM time period |
|  |  |  |  |  |
| Miscellaneous | taz.data.file | String | /inputs/taz\_data.csv | TAZ data input file |
| Miscellaneous | taz.data.taz.column | String | TAZ | TAZ column |
| Miscellaneous | taz.data.avgttd.column | String | AVGTTS | Average travel time to toll facility |
| Miscellaneous | taz.data.dist.column | String | DIST | Average distance to toll facility |
| Miscellaneous | taz.data.pctdetour.column | String | PCTDETOUR | Percent detour to toll facility |
| Miscellaneous | taz.data.terminal.column | String | TERMINALTIME | Terminal time to toll facility |

## UEC Files

The utility expression files (UEC) listed in Table 4.12 are required by CT-RAMP and are stored in the UEC folder. The open source Utility Expression Calculator (UEC) Java package originally written by Parsons Brinckerhoff locates input variables and specify utility equations that describe each discrete choice model. The input variables and specifications are defined and stored in a Microsoft Excel workbook. The use of Excel greatly enhances the flexibility and transparency of the model system -- utility coefficients, model structures, etc, can be edited via Excel (rather than via difficult to follow text files or source code). An example UEC is shown in Figure 17.

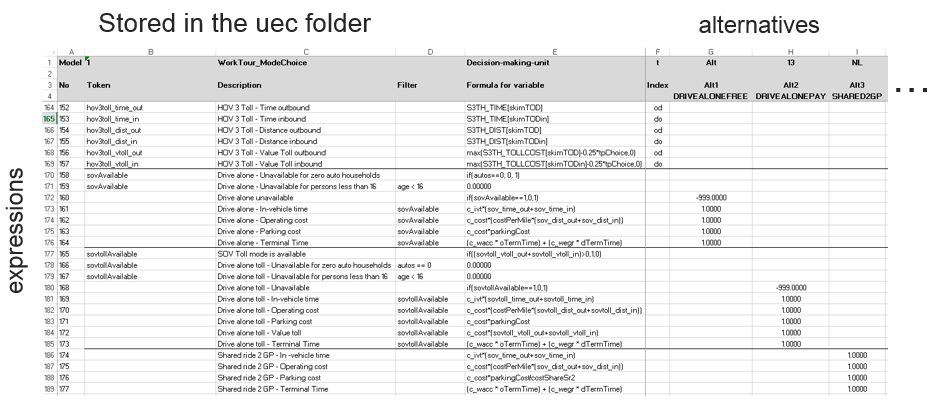


Figure - UEC Example

Each UEC workbook (i.e. Excel file) consists of at least two worksheets. One must be the UEC DataSheet, which defines the input files used in the utility expressions, including zonal (vector) data and level-of-service "skims" (matrix data). The second, third, fourth, etc. page specifies one or more multinomial or nested logit models via a unique UEC UtilitySheet. The UtilitySheet consists of three sections, as follows:

1. The first section specifies the nesting structure of the logit model -- if omitted, a multinomial structure is assumed;
2. Next, variable names, or tokens, are defined for use in subsequent (moving down rows) utility equations; and,
3. The final section defines the utility terms, typically a variable and a coefficient for each of the logit model's alternatives.

The CT-RAMP Java code controls model flow, handles output files (e.g., trip tables, tour records), facilitates debugging, and allows for the UECs to access variables stored in memory (e.g., the results from upstream logit models).

The basic steps required to use the UEC Java package are to instantiate the UEC object and to then solve the UEC object -- the solve method returns an array of utilities. An example UEC instantiation is as follows:

*// create a UEC for an example mode choice model*

*String controlExcelFile = "ExampleModeChoice.xls"*

*int dataSheetIndex = 0;*

*int utilitySheetIndex = 1;*

*HashMap rbHashMap = ResourceUtil.changeResourceBundleIntoHashMap(resourceBundle);*

*UtilityExpressionCalculator exampleUec = new UtilityExpressionCalculator(new File(controlExcelFile), utilitySheetIndex, dataSheetIndex, rbHashMap,ExampleModeChoiceDMU;*

The UEC constructor call will read all of the data defined in the DataSheet into working memory and check the syntax on the UtilitySheet for consistency. Errors will cause the program to terminate; error details will be written to the log file. The last argument in the Utility Expression Calculator constructor is a reference to a decision-making unit (DMU) class.

The DMU is the interface between the utility expressions specified in the UtilitySheet and variables expected to be stored in memory (i.e. because they were created from upstream model results or require complex calculations outside the easy functionality of the UEC). So-called DMU variables are specified in the UtilitySheet with the @ symbol. For example, "@autos" may refer to an automobile ownership market segment. Each @ variable specified in the UtilitySheet must have a corresponding getter method in the DMU class. To carry this example through, an @autos reference in the UtilitySheet of the ExampleModeChoice.xls control file requires a public getAutos() method to be a part of the ExampleModeChoiceDMU class.

The resource bundle Hash Map must identify variables that are defined in the UEC DataSheet with the % symbol. For example, if the DataSheet references the property "%Project.Directory%", the resource bundle must include a string defining the variable "Project.Directory". (This is a typical implementation of a Java properties or resource bundle file). The above code example demonstrates how a resource bundle file can be translated into a Hash Map for use in the Utility Expression Calculator object.

To solve the UEC for a given decision making unit (which may be a zone pair, a traveler, or a group of travelers), the UEC solve method can be implemented as follows:

*// use the UEC class to compute logit model utilities  
IndexValues indexValues = new IndexValues();  
indexValues.setOriginLocation(origin);  
indexValues.setDestinationLocation(destination);  
exampleModeChoiceDMU.setPersonDetails(person);  
double[] exampleModeChoiceUtilities = exampleUec.solve(indexValues, exampleModeChoiceDMU, availFlag);*

In this example, the origin and destination of the traveler are being set in the Index Values object, which contains the index values used in the zone- and matrix-file indexing described in the UtilitySheet page. The person attributes are being set in the DMU object, which is an instance of the ExampleModeChoiceDMU class described above. The last argument is a boolean array (availFlag) which lets the solve method know which of the alternatives should be solved for; the availability flags can be used to speed up calculations but are not strictly necessary. The availability flag array is dimensioned by the number of alternatives and can be set to all true as a default.

The UEC solve method returns an array of doubles dimensioned to the number of alternatives specified in the UtilitySheet. The array contains the sumproduct of each of the formulas and coefficients for each alternative, which is the utility for each alternative. This array can then be used with a logit model object to first compute alternative probabilities and then simulate choices.

For more a more thorough review of the UECs, refer to MTC’s excellent travel model wiki[[3]](#footnote-3)

Table . uec fiLES

| Module | File | Purpose |
| --- | --- | --- |
| Accessibility | Acc\_alts.csv | Accessibilities alternatives |
| Accessibility | Accessibilities.xls | Accessibilities model |
| Accessibility | Accessibilities\_DC.xls | Accessibilities destination choice model |
| At-work (4.5.1) | AtWorkSubtourFrequency.xls | At work sub tour frequencies model |
| AO (2.1, 3.2) | AutoOwnership.xls | Auto ownership model |
| Skimming (6.3) | AutoSkims.xls | Auto skims lookup UEC |
| Best Transit Path (6.3) | BestTransitPathUtility.xls | Best transit path utility model |
| CDAP (4.1) | CoordinatedDailyActivityPattern.xls | CDAP model |
| Tour TOD (4.2.2, 4.5.2, 4.3.4, 4.4.3) | DepartureTimeAndDurationAlternatives.csv | Departure time and duration alternatives |
| DC (4.5.4, 4.3.3, 4.4.2) | DestinationChoiceAlternativeSample.xls | Destination choice alternatives sample model |
| Skimming (6.3) | DriveTransitWalkSkims.xls | Drive transit walk skims lookup UEC |
| Non-mand freq (4.4.1) | IndividualNonMandatoryTourFrequencyAlternatives.csv | Individual non mandatory tour frequency alternatives |
| Non-mand freq (4.4.1) | IndividualNonMandatoryTourFrequencyExtensionProbabilities\_p1.csv | Individual non mandatory tour frequency extension probabilities lookup file |
| JTF (4.3.1) | JointAlternatives.csv | Joint tour alternatives |
| JTF (4.3.1) | JointTourFrequency.xls | Joint tour frequency model |
| University tour DC (4.2) | MajorUniversityStudentTourDestinationChoice.xls | Major university student tour destination choice model |
| Accessibility | MandatoryAccess.xls | Mandatory accessibilities model |
| MTF (4.2.1) | MandatoryTourFrequency.xls | Mandatory tour frequency model |
| Non-mand freq (4.4.1) | NonMandatoryIndividualTourFrequency.xls | Non mandatory individual tour frequency model |
| Non-mandatory destination (4.4.2) | NonMandatoryTlcAlternatives.csv | Non mandatory TLC alternatives |
| Free parking (3.1) | ParkingProvision.xls | Free parking model |
| Parking location (6.2) | ParkLocationAlts.csv | Parking location alternatives |
| Parking location (6.2) | ParkLocationChoice.xls | Parking location choice model |
| Parking location (6.2) | ParkLocationSampleAlts.csv | Parking location alternatives |
| School escorting | SchoolEscorting.xls | School escorting model |
| School escorting | SchoolEscortingAlts.csv | School escorting alternatives |
| Stop location (5.3) | SlcAlternatives.csv | Stop location choice alternatives |
| Stop location (5.3) | SlcSoaDistanceUtility.xls | Stop location choice sample of alternatives distance utility model |
| Stop location (5.3) | SlcSoaSize.xls | Stop location choice sample of alternatives size terms |
| Stop departure (5.4) | StopDepartArriveProportions.csv | Stop departure and arrival proportions lookup file |
| Stop frequency (5.4) | StopFrequency.xls | Stop frequency model |
| Stop location (5.3) | StopLocationChoice.xls | Stop location choice model |
| Stop purpose (5.2) | StopPurposeLookupProportions.csv | Stop purpose lookup proportions lookup file |
| TAZ distances | TazDistance.xls | TAZ distance lookup UEC |
| DC (4.5.4, 4.3.3, 4.4.2) | TourDcSoaDistance.xls | Tour destination choice sample of alternatives distance model |
| DC (4.5.4, 4.3.3, 4.4.2) | TourDcSoaDistanceNoSchoolSize.xls | Tour destination choice sample of alternatives distance for school purposes |
| Tour TOD (4.2.2, 4.5.2, 4.3.4, 4.4.3) | TourDepartureAndDuration.xls | Tour departure and duration model |
| DC (4.5.4, 4.3.3, 4.4.2) | TourDestinationChoice.xls | Tour destination choice model |
| Tour mode (4.2.4, 4.5.3, 4.3.5, 4.4.4) | TourModeChoice.xls | Tour mode choice model |
| Transponder | TransponderOwnership.xls | Transponder ownership model |
| Trip mode (6.1) | TripModeChoice.xls | Trip mode choice model |
| Skimming (6.3) | WalkTransitDriveSkims.xls | Walk transit drive skim lookup UEC |
| Skimming (6.3) | WalkTransitWalkSkims.xls | Walk transit walk skim lookup UEC |

## Major UniversIty Model Parking Lots (Optional)

If UniversityParkingLotChoiceModel.RunModel is set to true in the OR-RAMP properties, then the major university parking lot choice model will be run. The model only works if the MAZ data has square footage of major university space and the person file indicates which students are major university students. Note that in case this feature of the model is used, the enrollment should be zero for any MAZs that make up the major university, since the attraction of students to the university is controlled by the major university indicator field in the synthetic population person file rather than enrollment at the university. Also required is an input csv file containing number of spaces by MAZ and space type along with terminal times from the center of the lot to the edge. Informal lots are on-street spaces in non-university zones, to allow parking in a nearby neighborhood.

The model will choose a parking lot MAZ for each worker and student who travel to any of the university zones (the MAZs with university square footage) for work or university tours by auto tour mode. It will insert the parking lot as the first and last on-campus stop on their tour, which will be identified by a “Parking” purpose in the trip output file for the lot location. The model will also output a parking demand file (property Results.universityParkingDataFile) and an unmet demand file containing demand for parking that cannot be accommodated by available parking supply (UniversityParkingLotChoiceModel.UnmetParkingDemand.file).

Table . Parking Lots For Major Universities

| Field | Description | Example |
| --- | --- | --- |
| MAZ | Park MAZ | 813 |
| informalLot | Is informal lot | 0 or 1 |
| spaceType | 0=faculty\staff, 1=students, 2=visitors | 0, 1, or 2 |
| Spaces | Number of parking spaces | 100 |
| terminalTime | Terminal time | 1.5 minutes |

## Costs

The various costs input into the model are summarized in Table 4.14 below. All costs are for year 2010; model users can use a CPI adjustment for translating costs from other years to 2010 values for consistency.

Table . Costs

| Cost (Year 2010) | [File](http://mtcgis.mtc.ca.gov/foswiki/bin/view/Main/PropertiesFile?sortcol=3;table=1;up=0#sorted_table) | Current Settings |
| --- | --- | --- |
| Value-of-time for assignment | Assignment and skimming procedure file + General Procedure Settings + PrT Settings + Impedance + Impedance function by demand segment | 25 dollars per hour  144 seconds per dollar = 3600 / 25 |
| Transit fare matrix | VISUM Python script SOABM.py createTapFareMatrix() | 100, 200 cents (GP, RV) |
| Parking costs | VISUM master version file SOABM.ver MAZ attributes HPARKCOST, DPARKCOST, MPARKCOST | 0 dollars |
| Link toll | VISUM master version file SOABM.ver link attributes Toll\_PrTSys(Dseg) | 0 dollars |
| Auto operating cost | Orramp.properties file for reference in the OR-RAMP UECs | aoc.fuel =13.5 cents/mile  aoc.maintenance =6.3 cents/mile |

# Outputs

All outputs are stored in the directory “outputs/” under the project folder.

## Network and Zone Data

A number of initial output files are created before running OR-RAMP. These output files are illustrated in the following tables.

Table . MAZ Data File Export from VISUM – Inputs\maz\_data\_export.csv

|  |  |
| --- | --- |
| **Name** | **Description** |
| SEQMAZ | MAZ number for CT-RAMP. From 1 to N (number of MAZs) |
| NO | VISUM zone number |
| TAZID | TAZ number |
| DISTNAME | District name |
| COUNTY | County |
| HH | Number of households |
| SF\_HH | Number of single family households |
| DUPLEX\_HH | Number of duplex households |
| MF\_HH | Number of multifamily households |
| MH\_HH | Number of mobile home households |
| EMP\_CONSTR | Construction employment |
| EMP\_WHOLE | Wholesale Trade employment |
| EMP\_RETAIL | Retail Trade employment |
| EMP\_SPORT | Sporting Goods, Hobby, Musical Instruments and Book stores employment |
| EMP\_ACCFD | Accommodation and Food Services employment |
| EMP\_AGR | Agriculture, Forestry, Fishing and Hunting employment |
| EMP\_MIN | Mining, Quarrying, and Oil and Gas Extraction employment |
| EMP\_UTIL | Utilities employment |
| EMP\_FOOD | Food Manufacturing employment |
| EMP\_WOOD | Wood Product Manufacturing employment |
| EMP\_METAL | Primary Metal Manufacturing employment |
| EMP\_TRANS | Transportation employment |
| EMP\_POSTAL | Postal Service employment |
| EMP\_INFO | Information employment |
| EMP\_FINANC | Finance and Insurance employment |
| EMP\_REALES | Real Estate and Rental and Leasing employment |
| EMP\_PROF | Professional, Scientific, and Technical Services employment |
| EMP\_MGMT | Management of Companies and Enterprises employment |
| EMP\_ADMIN | Administrative and Support and Waste Management employment |
| EMP\_EDUC | Educational Services employment |
| EMP\_HEALTH | Health Care and Social Assistance employment |
| EMP\_ARTS | Arts, Entertainment, and Recreation employment |
| EMP\_OTHER | Other Services (except Public Administration) Religious employment |
| EMP\_PUBADM | Public Administration employment |
| EMP\_TOTAL | Total employment |
| ENROLLK\_8 | Grade School K-8 enrollment |
| ENROLL9\_12 | Grade School 9-12 enrollment |
| ENROLLCOLL | Major College enrollment |
| ENROLLCOOT | Other College enrollment |
| ENROLLADSC | Adult School enrollment |
| universitySqFtClass | University square footage classrooms |
| universitySqFtOffice | University square footage office |
| universitySqFtRec | University square footage recreation |
| ECH\_DIST | Elementary school district |
| HCH\_DIST | High school district |
| HOTELRMTOT | Total number of hotel rooms |
| PARKAREA | Parking model code:  1 - Trips with destinations in this MAZ may choose to park in a different MAZ, parking charges apply (downtown)  2 - Trips with destinations in parkarea 1 may choose to park in this MAZ, parking charges might apply (quarter mile buffer around downtown)  3 - Only trips with destinations in this MAZ may park here, parking charges apply (outside downtown paid parking, only show cost no capacity issue)  4 - Only trips with destinations in this MAZ may park here, parking charges do not apply (outside downtown, free parking) |
| HSTALLSOTH | Number of stalls allowing hourly parking for trips with destinations in other MAZs |
| HSTALLSSAM | Number of stalls allowing hourly parking for trips with destinations in the same MAZ |
| HPARKCOST | Average cost of parking for one hour in hourly stalls in this MAZ, dollars |
| NUMFREEHRS | Number of hours of free parking allowed before parking charges begin in hourly stalls |
| DSTALLSOTH | Stalls allowing daily parking for trips with destinations in other MAZs |
| DSTALLSSAM | Stalls allowing daily parking for trips with destinations in the same MAZ |
| DPARKCOST | Average cost of parking for one day in daily stalls, dollars |
| MSTALLSOTH | Stalls allowing monthly parking for trips with destinations in other MAZs |
| MSTALLSSAM | Stalls allowing monthly parking for trips with destinations in the same MAZ |
| MPARKCOST | Average cost of parking for one day in monthly stalls, amortized over 22 workdays, dollars |
| HHS | Average household size in the MAZ |
| POP | Total population |
| HHP | Total household population (exclude GQ pop) |
| HH\_GQ\_CIV | GQ civilian population (half-way houses and other similar group quarters not associated with military housing, on-campus housing, or fraternities/sororities) |
| HH\_GQ\_MIL | GQ military population (includes military group quarters housing) |
| HH\_GQ\_UNIV | GQ University students (on-campus or off-campus dormitories, fraternities, and sororities) |
| WRK\_EXT\_PR | Probability of working outside the region |
| SCHDIST\_NA | School district name |
| SCHDIST | School district code |
| HH\_1 | Number of 1 person household |
| HH\_2 | Number of 2 persons household |
| HH\_3 | Number of 3 persons household |
| HH\_4 | Number of 4+ persons household |
| DUDEN | Dwelling unit density within ½ mile |
| EMPDEN | Employment density within ½ mile |
| TOTINT | Intersection count within ½ mile |
| POPDEN | Population density |
| RETDEN | Retail employment density within ½ mile |
| TERMTIME | Terminal time |
| PARKACRES | Park acres |

Table . TAZ Assignment Speeds for Transit ASsignment - taz\_skim\_<TOD> \_speed\_linkspeeds.csv

|  |  |
| --- | --- |
| **Name** | **Units/Description** |
| FROMNODE | Link from node no |
| TONODE | Link to node no |
| V0PRT | Free flow speed for transit runtimes |

Table . TAPS Input File FOR OR-RAMP - tap\_data.csv

|  |  |
| --- | --- |
| **Field** | **Description** |
| TAP | TAP ID |
| TAP | TAP’s TAZ |
| LOTID | Parking lot ID, which is currently set to the TAP ID y tazsToTapsForDriveAccess() in SOABM.py |
| CAPACITY | Parking lot capacity, which is currently set to 1 by tazsToTapsForDriveAccess() in SOABM.py |

Table . Tap Lines CSV File for OR-RAMP - tapLines.csv

|  |  |
| --- | --- |
| **Name** | **Units/Description** |
| TAP | TAP number |
| LINES | Space delimited list of lines served |

Table . park Locations Alternatives for OR-RAMP - ParkLocationAlts.csv

|  |  |
| --- | --- |
| **Name** | **Units/Description** |
| A | Alternative number |
| MGRA | MAZ |
| PARKAREA | Park area type, which is set to 4 for all zones |

Table . Destination ChOICE Alternatives for OR-RAMP - DestinationChoiceAlternatives.csv

|  |  |
| --- | --- |
| **Name** | **Units/Description** |
| A | Alternative number for the CT-RAMP destination choice models. This is an index to the list of zones, i.e. from 1 to the max number of zones |
| MGRA | MAZ |
| DEST | TAZ |

Table . Sample of Alternatives TAZ Distances Alternatives for OR-RAMP - SoaTazDistAlternatives.csv

|  |  |
| --- | --- |
| **Name** | **Units/Description** |
| A | Alternative number |
| DEST | TAZ |

Table . park Location sampling Alternatives for OR-RAMP - ParkLocationSampleAlts.csv

|  |  |
| --- | --- |
| **Name** | **Units/Description** |
| A | Alternative number |
| MGRA | MAZ |

## CVM and External Model

The CVM and SWIM external model both output OMX trip matrices for import into VISUM for eventual assignment on the TAZ network. The output matrices are listed below.

Table . CVM Outputs

|  |  |  |
| --- | --- | --- |
| **File** | **Description** | **Input/Output** |
| outputs/cvmTrips.omx | Trip matrices by type and time period  CAR, SU, MU  EV1, EA, AM, MD, PM, EV2 | Output |

Table . SWIm External MODEL Outputs

|  |  |  |
| --- | --- | --- |
| **File** | **Description** | **Input/Output** |
| outputs/externalOD.omx | Trips matrices by purpose and time period  hbw, hbcoll, hbsch, hbo, hbr, hbs, nhbnw, nhbw, truck  EV1, EA, AM, MD, PM, EV2 | Output |

## Skims

SOABM produces multiple types of network LOS indicators, often called skim matrices. The following skims are output.

Table . TAP Skim Matrices - tap\_skim\_<TOD> \_set<SETID>.omx

|  |  |  |
| --- | --- | --- |
| **Skim** | **Description** | **Skim Number** |
| IVT | In-Vehicle Time | 1 |
| OWT | Origin Wait Time | 2 |
| TWT | Transfer Wait Time | 3 |
| WKT | Walk Time | 4 |
| NTR | Transfers (Boardings-1) | 5 |
| FAR | Fare | 6 |
| IVTT(Bus) | In-Vehicle Time Tsys Bus | 7 |
| IVTT(ComRail) | In-Vehicle Time Tsys ComRail | 8 |
| IVTT(ExpBus) | In-Vehicle Time Tsys ExpBus | 9 |
| IVTT(LightRail) | In-Vehicle Time Tsys LightRail | 10 |

Table . MAZ to MAZ Distances (Bike and WALK) - maz2maz\_<Bike|WALK>.csv

|  |  |
| --- | --- |
| **Name** | **Units/Description** |
| OMAZ | Origin MAZ |
| DMAZ | Destination MAZ |
| DISTMILES | Network distance in miles |

Table . TAP to MAZ Distances (Bike and WALK) - tap2maz\_<BIKE|Walk>.csv

|  |  |
| --- | --- |
| **Name** | **Units/Description** |
| TAP | TAP ID |
| MAZ | MAZ ID |
| DISTMILES | Network distance in miles |

Table . TAZ to NEAR TAPS ImpedanceS for DRIVE TRANSIT - drive\_taz\_tap.csv

|  |  |
| --- | --- |
| **Field** | **Description** |
| FTAZ | From TAZ |
| MODE | Transit mode |
| PERIOD | Time period |
| TTAP | To TAP |
| TMAZ | To TAP’s MAZ |
| TTAZ | To TAP’s TAZ |
| DTIME | Drive time |
| DDIST | Drive distance |
| DTOLL | Drive toll |
| WDIST | Walk distance |

Table . TAZ Skim Matrices - taz\_skim\_<DSEG> \_<TOD>.omx

|  |  |  |  |
| --- | --- | --- | --- |
| **Transport Systems** | **Demand Segment** | **Skims** | **Skim Numbers** |
| SOV | SOV | TTO,TTC,DIS | 1,2,3 |
| SOVToll | SOVToll | TTO,TTC,DIS,AD1,TOL | 4,5,6,7,8 |
| HOV2 | HOV2 | TTO,TTC,DIS,AD2 | 9,10,11,12 |
| HOV2Toll | HOV2Toll | TTO,TTC,DIS,AD1,AD2,TOL | 13,14,15,16,17,18 |
| HOV3 | HOV3 | TTO,TTC,DIS,AD3 | 19,20,21,22 |
| HOV3Toll | HOV3Toll | TTO,TTC,DIS,AD1,AD3,TOL | 23,24,25,26,27,28 |
| Truck | Truck | TTO,TTC,DIS | 29,30,31 |
| TruckToll | TruckToll | TTO,TTC,DIS,AD1,TOL | 32,33,34,35,36 |

Table . TAZ Skim Definitions

|  |  |  |
| --- | --- | --- |
| **Skim** | **Description** | **Intrazonal** |
| TT0 | Free flow travel time | 0.5 \* mean nearest 3 |
| TTC | Congested travel time | 0.5 \* mean nearest 3 |
| DIS | Trip distance | 0.5 \* mean nearest 3 |
| AD1 | Toll only distance\* | 0.5 \* mean nearest 3 |
| AD2 | HOV2 only distance\* | 0.5 \* mean nearest 3 |
| AD3 | HOV3 only distance\* | 0.5 \* mean nearest 3 |
| TOL | Toll | 0.5 \* mean nearest 3 |
| \*Before skimming each demand segment  AddVal1 = Length \* (Toll\_PrTSys(<DSEG>) > 0)  AddVal2 = Length \* (T0\_PrTSys(HOV2|HOV2Toll) < 99999 & T0\_PrTSys(SOV) >= 99999)  AddVal3 = Length \* (T0\_PrTSys(HOV3|HOV3Toll) < 99999 & T0\_PrTSys(SOV) >= 99999)  Note AddVal1,2,3 are used for internal calculations and should not be used by the user | | |

## Trip Lists

OR-RAMP produces the following tour and trip microsimulation trip lists for both individual and joint travel. The files output are listed below. The modes are defined in Table 5.21 below.

* Individual tours - Outputs/indivTourData\_<iteration>.csv
* Individual trips - Outputs/indivTripData\_<iteration>.csv
* Joint tours - Outputs/jointTourData\_<iteration>.csv
* Joint trips - Outputs/jointTripData\_<iteration>.csv

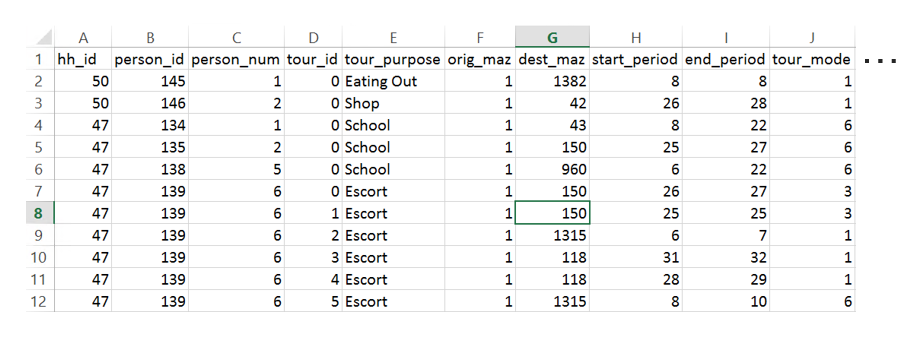


Figure - InDivTour Data Example

Table . Individual TouRS File - indivTourData\_<iteration>.csv

|  |  |
| --- | --- |
| **Field** | **Description** |
| hh\_id | Household ID |
| person\_id | Person ID |
| person\_num | Person number in HH |
| person\_type | Person type |
| tour\_id | Tour ID |
| tour\_category | Tour category:   INDIVIDUAL\_NON\_MANDATORY  MANDATORY  AT\_WORK |
| tour\_purpose | Tour purpose:  Discretionary  Eating Out  Escort  Maintenance  School  Shop  University  Visiting  Work  Work-Based |
| orig\_maz | Origin MAZ |
| dest\_maz | Destination MAZ |
| start\_period | Tour start period (30min periods starting at 4:30) |
| end\_period | Tour end period (30min periods starting at 4:30) |
| tour\_mode | Tour mode (see Table 5.21) |
| tour\_distance | Tour distance |
| tour\_time | Tour time |
| atWork\_freq | At work stop frequency |
| num\_ob\_stops | Number of outbound stops |
| num\_ib\_stops | Number of inbound stops |
| escort\_type\_out | Escort type outbound (0 = no escorting; 1 = ride-share, 2 = pure escort) |
| escort\_type\_in | Escort type inbound (0 = no escorting; 1 = ride-share, 2 = pure escort) |
| driver\_num\_out | Escort driver number outbound (person number of driver on outbound direction if escort tour) |
| driver\_num\_in | Escort driver number inbound (person number of driver on inbound direction if escort tour) |
| out\_btap | Outbound boarding tap |
| out\_atap | Outbound alighting tap |
| in\_btap | Inbound boarding tap |
| in\_atap | Inbound alighting tap |
| out\_set | Outbound transit set |
| in\_set | Inbound transit set |
| Util\_<1-14> | Utility for each tour mode (see Table 5.21) |
| Prob\_<1-14> | Probabilities for each tour mode (see Table 5.21) |

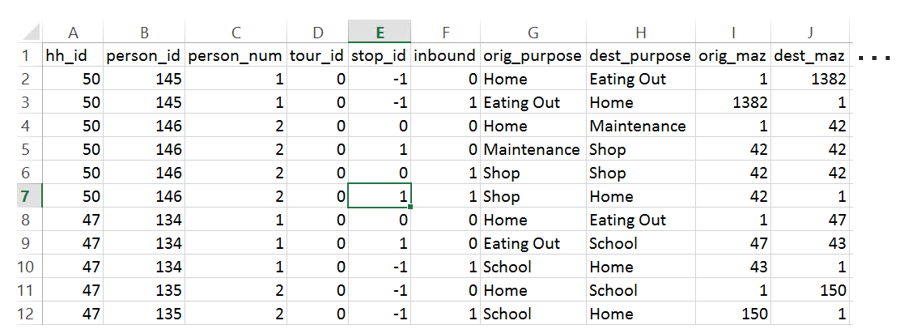


Figure - INDIVTrip Data Example

Table . InDvidUal Trips File - indivTripData\_<iteration>.csv

|  |  |
| --- | --- |
| **Field** | **Description** |
| hh\_id | Household ID |
| person\_id | Person ID |
| person\_num | Person number in HH |
| person\_type | Person type |
| tour\_id | Tour ID |
| Stop\_id | Trip ID |
| Inbound | Inbound or outbound trip |
| Tour\_purpose | Tour purpose:  Discretionary  Eating Out  Escort  Maintenance  School  Shop  University  Visiting  Work  Work-Based |
| orig\_purpose | Trip origin purpose:  Discretionary  Eating Out  Escort  Home  Maintenance  School  Shop  University  Visiting  Work  work related  Work-Based |
| dest\_purpose | Trip destination purpose:  Discretionary  Eating Out  Escort  Home  Maintenance  School  Shop  University  Visiting  Work  work related  Work-Based |
| orig\_maz | Origin MAZ |
| dest\_maz | Destination MAZ |
| universityParking | University parking (1 for true) |
| Parking\_maz | Parking MAZ |
| stop\_period | Trip period (30min periods starting at 4:30) |
| trip\_mode | Trip mode (see Table 5.21) |
| Trip\_board\_tap | Trip boarding tap |
| Trip\_alight\_tap | Trip alighting tap |
| Tour\_mode | Tour mode (see Table 5.21) |
| Driver\_pnum | The person\_num of the driver if the trip origin or destination is escort; else 0 if neither end of the trip is part of the escort model. |
| Orig\_escort\_stoptype | Origin escort trip stop type (0 = no escorting; 1 = dropped-off, 2 = picked-up) |
| orig\_escortee\_pnum | The person\_num of the child being picked-up or dropped off at the origin end of the trip |
| dest\_escort\_stoptype | Destination escort trip stop type (0 = no escorting; 1 = dropped-off, 2 = picked-up) |
| dest\_escortee\_pnum | The person\_num of the child being picked-up or dropped off at the destination end of the trip |
| Set | Transit set ID |

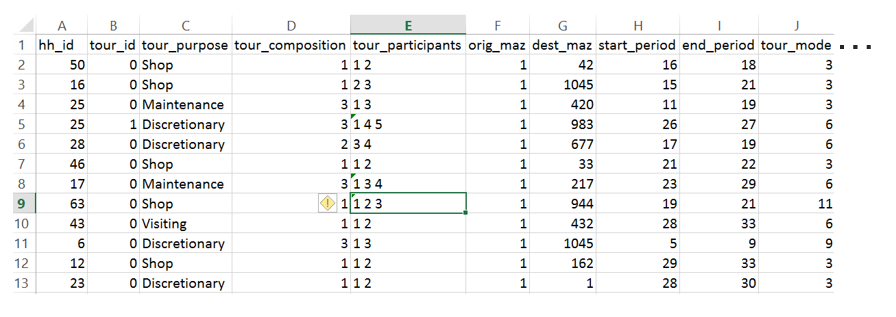


Figure - JointTour DATA Example

Table . Joint TourS File - jointTourData\_<iteration>.csv

|  |  |
| --- | --- |
| **Field** | **Description** |
| hh\_id | Household ID |
| tour\_id | Tour ID |
| tour\_category | Tour category |
| tour\_purpose | Tour purpose |
| tour\_composition | Tour composition code |
| tour\_participants | Household person numbers |
| orig\_maz | Origin MAZ |
| dest\_maz | Destination MAZ |
| start\_period | Tour start period (30min periods starting at 4:30) |
| end\_period | Tour end period (30min periods starting at 4:30) |
| tour\_mode | Tour mode (see Table 5.21) |
| tour\_distance | Tour distance |
| tour\_time | Tour time |
| num\_ob\_stops | Number of outbound stops |
| num\_ib\_stops | Number of inbound stops |
| out\_btap | Outbound boarding tap |
| out\_atap | Outbound alighting tap |
| in\_btap | Inbound boarding tap |
| in\_atap | Inbound alighting tap |
| out\_set | Outbound transit set |
| in\_set | Inbound transit set |
| Util\_<1-14> | Utility for each mode |
| Prob\_<1-14> | Probabilities for each mode |

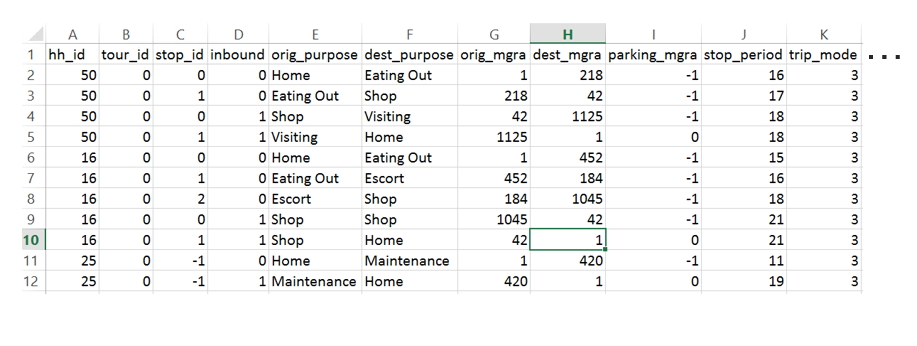


Figure - JOINTTRIP DATA EXAMPLE

Table . JoiNT Trips File - jointTripData\_<iteration>.csv

|  |  |
| --- | --- |
| **Field** | **Description** |
| hh\_id | Household ID |
| tour\_id | Tour ID |
| Stop\_id | Trip ID |
| Inbound | Inbound or outbound trip |
| tour\_purpose | Tour purpose |
| orig\_purpose | Trip origin purpose |
| dest\_purpose | Trip destination purpose |
| orig\_maz | Origin MAZ |
| dest\_maz | Destination MAZ |
| Parking\_maz | Parking MAZ |
| universityParking | University parking (1 for true) |
| stop\_period | Trip period (30min periods starting at 4:30) |
| trip\_mode | Trip mode (see Table 5.21) |
| Num\_participants | Number of participants |
| Trip\_board\_tap | Trip boarding tap |
| Trip\_alight\_tap | Trip alighting tap |
| Tour\_mode | Tour mode (see Table 5.21) |
| Set | Transit set ID |

Table . Tour and TRIP Modes COdes

|  |  |  |
| --- | --- | --- |
| **Code** | **Mode** | **Description** |
| 1 | DRIVEALONEFREE | Drive alone free (non-toll) |
| 2 | DRIVEALONEPAY | Drive alone pay (toll eligible) |
| 3 | SHARED2GP | Shared-2 general purpose lanes (non-toll, non-HOV) |
| 4 | SHARED2HOV | Shared-2 HOV-eligible (non-toll) |
| 5 | SHARED2PAY | Shared-2 pay (HOV and toll eligible) |
| 6 | SHARED3GP | Shared 3+ general purpose lanes (non-toll, non-HOV) |
| 7 | SHARED3HOV | Shared 3+ HOV-eligible (non-toll) |
| 8 | SHARED3PAY | Shared 3+ pay (HOV and toll eligible) |
| 9 | WALK | Walk |
| 10 | BIKE | Bike |
| 11 | WALK\_SET | Walk-transit tour, or walk-transit-walk trip |
| 12 | PNR\_SET | Park and Ride transit tour, or PNR-transit-walk if outbound trip, walk-transit-PNR if inbound trip |
| 13 | KNR\_SET | Kiss and Ride transit tour, or KNR-transit-walk if outbound trip, walk-transit-KNR if inbound trip |
| 14 | SCHBUS | School bus |

## Demand Matrices

The output demand matrices from a run are shown in Figure 22 and listed in Table 5.22 below.

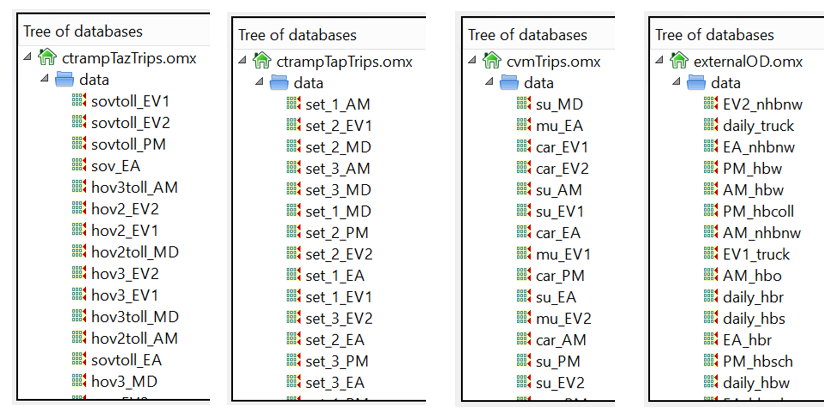


Figure - Demand MAtrices in OMX Format

Table . Demand Matrices

|  |  |  |
| --- | --- | --- |
| **File** | **Description** | **Matrices** |
| ctrampTazTrips.omx | TAP level matrices by mode and time-of-day | SOV\_<TOD>, SOVToll\_<TOD>, HOV2\_<TOD>, HOV2Toll\_<TOD>, HOV3\_<TOD>, HOV3Toll\_<TOD> |
| ctrampTapTrips.omx | TAP level matrices by mode and time-of-day | Set1\_<TOD>, Set2\_<TOD>, Set3\_<TOD> |
| cvmTrips.omx | CVM TAZ level matrices by mode and time-of-day | CAR\_<TOD>, SU\_<TOD>, MU\_<TOD> |
| externalOD.omx | External model TAZ level matrices by mode and time-of-day | hbw\_<TOD>, hbcoll\_<TOD>, hbsch\_<TOD>, hbo\_<TOD>, hbr\_<TOD>, hbs\_<TOD>, nhbnw\_<TOD>, nhbw\_<TOD>, truck\_<TOD> |

## Assignments

The TAZ and TAP level assignment results are stored in VISUM version files. The version files store the assignment paths so select-link, select-line, and other on-the-fly analyses can be performed. The assigned version files are:

* TAZ - taz\_skim\_<tod>\_speed.ver (see Figure 23)
* TAP - tap\_skim\_<tod>\_speed\_set<setid>.ver (see Figure 24)

Table 5.23 lists commonly used VISUM assignment attributes.

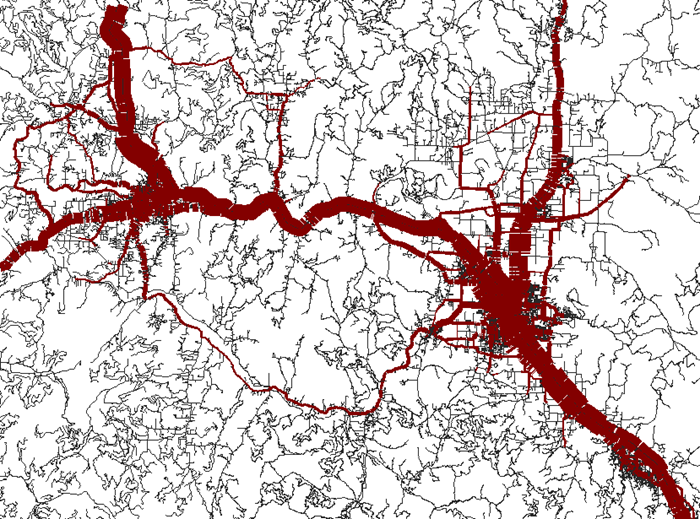


Figure - TAZ Assignment REsults

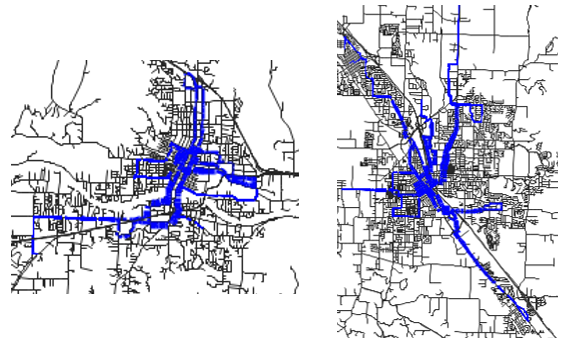


Figure - TAP Assignment Results

Table . Common ASSignMENT OutputS - taz\_skim\_<tod>\_speed.ver, tap\_skim\_<tod>\_speed\_set<setid>.ver

|  |  |
| --- | --- |
| **Field** | **Description** |
| Link\VolVehPrT | Total assigned link volume |
| Link\VCur-PrTSys(Tsys) | Link speed by mode |
| Link\VolVeh\_TSys(TSys) | Link volume by mode |
| Link\TCur\_PrTSys(TSys) | Link travel time by mode |
| Zone\OTraffic(DSeg) | Zone trips origins by demand segment |
| Zone\DTraffic(DSeg) | Zone trips destinations by demand segment |
| LineRoute\PTripsUnlinked | Line route unlinked trips |
| LineRoute\PTripsUnlinked\_Dseg(Dseg) | Line route unlinked trips by demand segment |
| StopPoint\PassBoard | Stop point total boardings |
| StopPoint\PassAlight | Stop point total alighting |
| StopPoint\PassOrigin | Stop point boardings at origin |
| StopPoint\PassDestination | Stop point alighting at destination |
| StopPoint\PassTransTotal | Stop point total transfers |
| StopPoint\PassThroughStop | Stop point through with stop |
| StopPoint\PassThroughNoStop | Stop point through without stop |
| StopPoint\PassBoard\_TSys(TSys) | Stop point total boardings by sub-mode |

# Creating and Running a New Scenario

To create and run a new model scenario (run), do the following:

1. Check out the repository to your local machine using Git. Instructions for downloading and setting up the template scenario are included with the download.
2. On initial checkout, expand the dependencies.zip to a folder called dependencies. This includes installs of Java, Python, and R. Every scenario is contained within its own folder, with a unique name. The folder name is the same as the scenario name. The dependencies folder must be next to the scenario folder since it is referenced in the RunModel.bat via ..\dependencies.
3. If running the population synthesizer, then create a new DB in SQL Server and run the procedure. It only needs to be run once for each land use scenario.
4. Ensure all required inputs are in the inputs folder.
5. Edit the necessary input files for the scenario:
   1. Network – If there are network revisions for the scenario, edit the SOABM.ver file in VISUM before running the scenario.
   2. TAZ land use data – If there are population, employment, or other TAZ related revisions, edit the zones or mainzones in the SOABM.ver file before running the scenario. VISUM zones are TAZs and VISUM mainzones are MAZs. TAPs are stops in VISUM.
   3. Model setup – If desired, set the number of feedback iterations and the OR-RAMP household sample rate in the RunModel.bat file.
6. Open a DOS command prompt in the scenario folder via Shift + Right click then Open Command Window Here
7. As shown in Figure 25, type RunModel.bat in the DOS window to run the model.

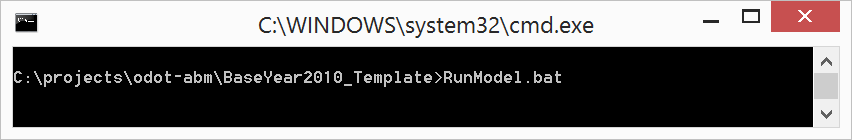


Figure - RunModel.Bat

1. The model runtime depends on the number of machine processors, the household sample rate, and the number of overall model feedback loops. The initial network build and skimming steps take about 1 hour, a 100% run of CT-RAMP takes about 45 minutes, and assignment/skimming takes about 30 minutes. As a result, a 3 feedback loop model run takes about 5 hours (1 hr + 45min + 30 min + 45min + 30 min + 45min + 30min).

# Model Logging/Trace Results

CT-RAMP writes a series of log files to the logs folder during a model run. These log files are extremely useful for understanding the model as well as for debugging a model run. The key log files in the logs folder are:

* event.log - the main overall process log file
* event\_hh.log - the household data manager
* event\_mtx.log - the matrix data manager
* event-nodeX.log - each remote node if applicable
* There are also model component specific log files such as event-ao.log for the auto ownership model.

In addition to writing log files, CT-RAMP can trace model calculations for a user specified household. To trace results for household 2949465 for example, set the following in the orramp.properties file:

Debug.Trace.HouseholdIdList = 2949465

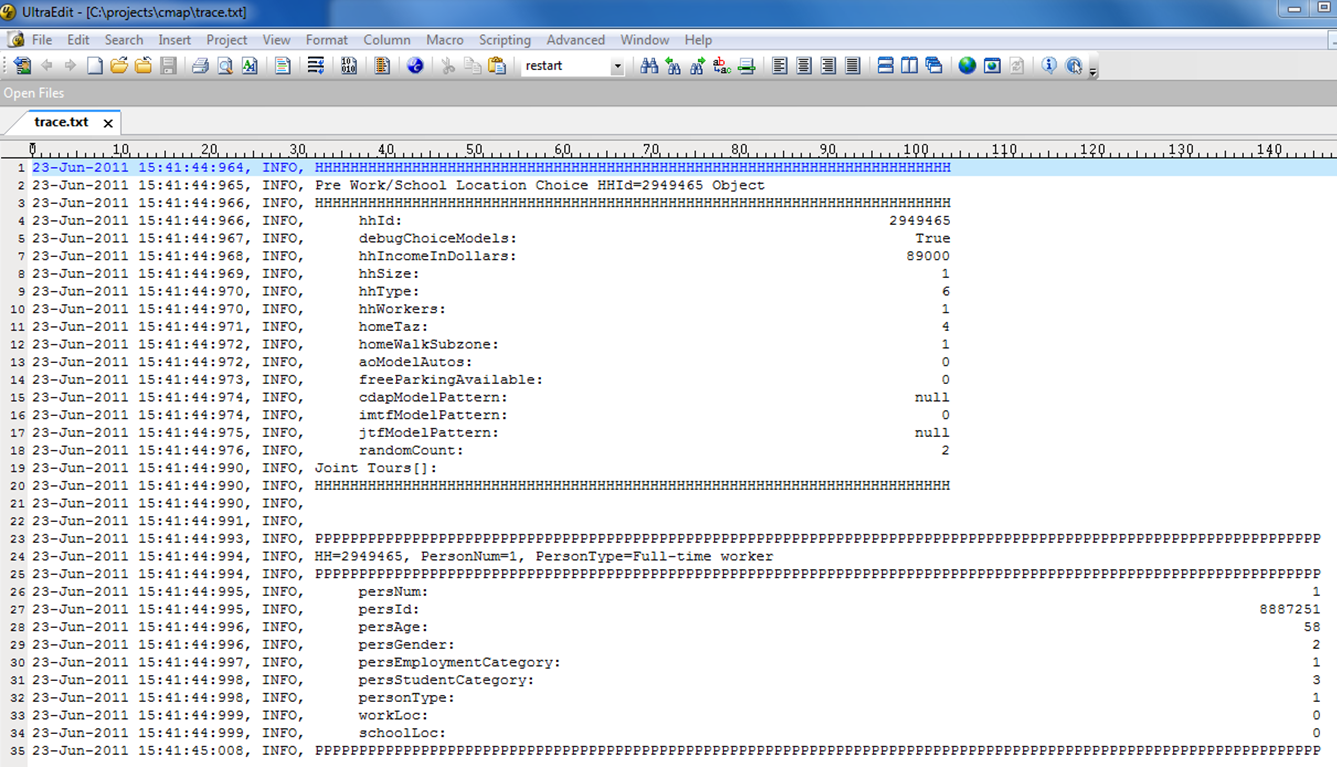
This tells CT-RAMP to write out all calculations for every person in household 2949465, including the results of the UEC calculations for each model.

Figure 26 below contains a sample of the household trace results. The first screenshot shows a sample of the trace results for the relevant chooser - person number 1 in the traced household. As is shown below in the first image, key attributes of the household and person are traced such as:

* hhIncomeInDollars - household income
* hhSize – household size
* hhWorkers - number of workers
* homeTaz - home TAZ
* persAge - person age

As is shown in the second image, which is the auto ownership model trace for household 2949465, the value of the coefficient for each alternative times the value of each expression is logged. The dimensions of this output are:

* The header contains the household ID traced - HHID\_2949465
* The 111 expression rows by five alternatives (0, 1, 2, 3, 4+ autos) correspond with the expressions and alternatives in the auto ownership UEC file
* Each row is an expression and each column is the coefficient times the value for the alternative



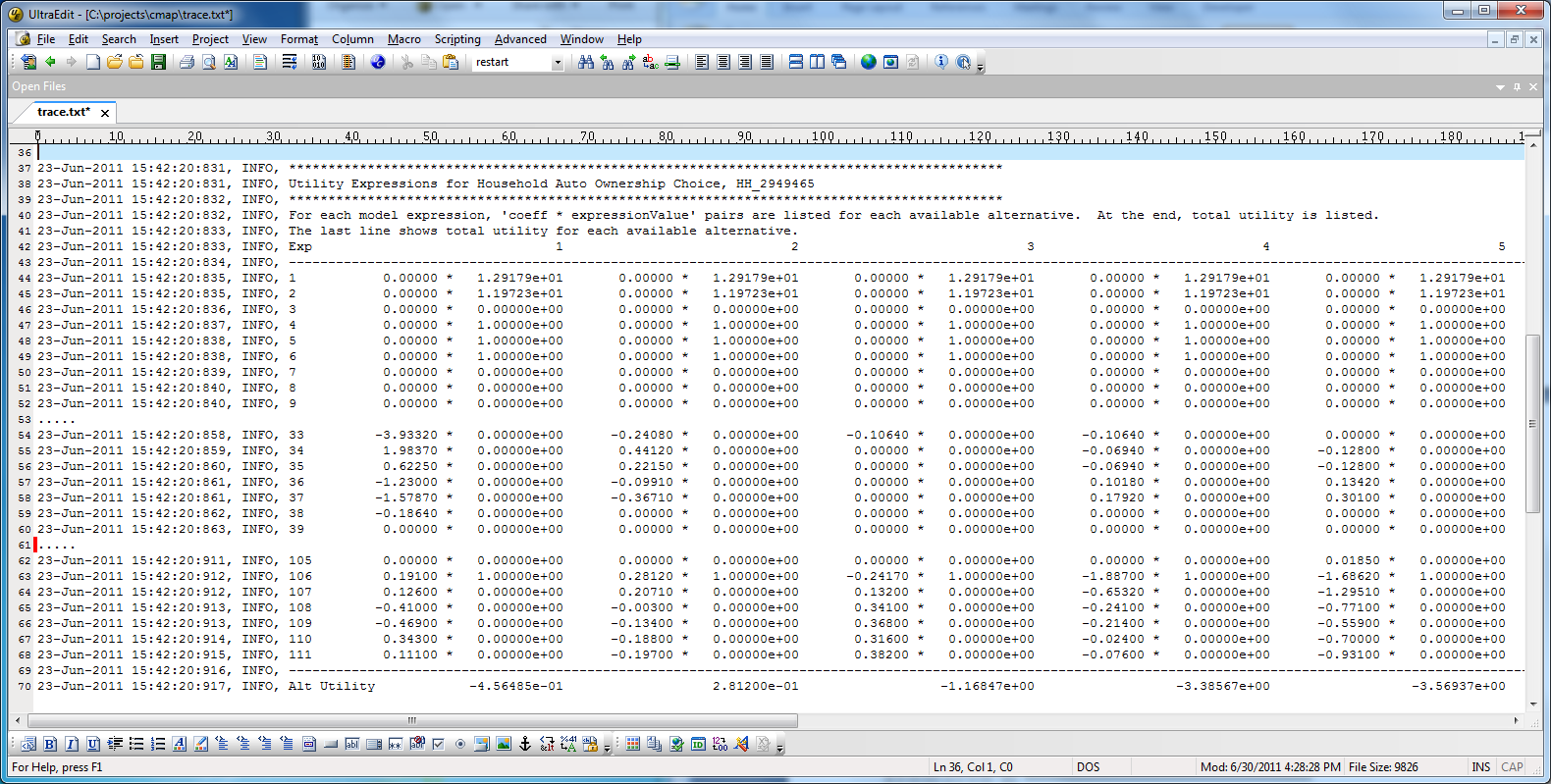


Figure - Household Trace Results

# Methodology for developing TAZ boundaries

A TAZ is one of the basic spatial unit of travel analysis in OR-RAMP. All of the model results from OR-RAMP can be analyzed at the TAZ level including commuting pattern and mode choice. TAZs are intended to model automobile travel. Therefore, they can be much larger than MAZs since automobile impedance is less sensitive to small differences in zone size/shape. The size of TAZs can vary with the model application and can be as small as a block or more than 10 square miles. A typical TAZ system should be specifically designed to fit local planning needs by MPOs. The analyst should be cognizant of the following guidelines when delineating TAZ boundaries:

## Recognizing boundaries

### Census Boundaries

Zonal boundaries should follow Census boundaries as faithfully as possible. This is essential as the Census Bureau directly provides socioeconomic data at a variety of geographies including Census Block and Census Tract which can be used as inputs to the ABM with little or no manipulation. If a TAZ were to split a Tract or a Block an allocation methodology to distribute the data from the split Census geography needs to be developed.

Also, Census provides a fully disaggregate dataset called Public Use Micro-data Sample at the level of a Public Use Micro-data Area (PUMA) – this dataset is used as seed data in the synthetic population. Defining TAZ boundary to be consistent with PUMA makes the population synthesizer allocation procedure more accurate.

### Physical Boundaries

Natural barriers such as rivers or mountains or man-made barriers such as railway tracks constitute physical boundaries. Physical boundaries restrict free movement and a centroid connector that passes over a barrier to access a road network would be unrealistic. This can be prevented by using the physical barrier as the zone boundary.

### Jurisdictional/Political/Planning Boundaries

TAZ boundaries should be consistent with political boundaries such as city, county, MPO and state boundaries. This will help in performing sub-area analysis such as city-city/county-county flows. If the region uses planning districts for performing similar analysis it would be useful to line the TAZs with the planning district boundaries as well.

### Historical Zonal Boundary

Consistency with previous zone boundaries should be maintained whenever possible. This would help preserve valuable socioeconomic data and the capability for comparisons.

## Transportation Network

Another important factor to consider when delineating TAZ boundary is the highway network. Similar to physical boundaries discussed above, facilities including freeways, expressways, and arterials act as access barriers and hence should not split a TAZ. Arterials are used as TAZ boundaries, because they are frequently used as Census boundaries. Linking centroid connectors to arterials results in better distribution of traffic on the network.

## Homogenous Land-use

TAZs should be created such that it has homogenous land-use and socioeconomic characteristic. For example, a major retail development should be placed in a different TAZ than a neighboring residential development. This segmentation helps the transportation planner systematically understand activity generation as well as it helps in minimizing intra-zonal travel. In areas of dense developments, such as the central business district, it is not always possible to satisfy this condition. However, MAZs nesting within these mixed use development TAZs can be used to address this issue to some extent.

## Uniqueness

Each TAZ should consist of one non-overlapping polygon and should be uniquely identifiable using an identification code. In general, the identification code should be continuous (no skipped IDs) for the set of TAZs in the region. Also, the TAZ system should be contiguous and its coverage should span the study region (MPO in this case).

## Centroids

TAZ centroids should be the center of activity for automobiles, i.e. where automobiles entering the zone want to go. They should accurately represent roadway access and should not cut across barriers. Also, sufficient number of centroid connectors needs to be coded so as to avoid concentrated loading of trips from a zone onto a particular link. It is also important not to connect a centroid to an intersection, ramps or any access restricted facility.

## Anticipating Developments

Areas in the region that could potentially undergo development in the future should be identified and the analyst should consider separate TAZs even if it falls within a bigger TAZ based on the rules stated above.

# Methodology for developing MAZ boundaries

MAZ is the smallest spatial unit of analysis in the ABM and is intended to model relatively short distance travel (i.e., walk and walk to/from transit). MAZs are generated by performing GIS operations on the Census Block and TAZ layers – the procedure will be explained shortly. Most of the rules that applied for TAZ boundaries can also be used for MAZ boundaries. The following general guidelines should be kept in mind when creating MAZs

* MAZs should nest within TAZs and they should be coded with the ID of the TAZ it nests in (the coding is achieved programmatically).
* As noted earlier, MAZs are used to characterize walk access – hence in areas where there is a significant amount of walk travel (dense developments such as CBD) MAZs should be small enough to walk across.
* As with TAZ, MAZs should contain homogenous land-use so that measures of density and land use mix are meaningful. For example, if one zone has two distinct areas, one with housing and one with employment, each distinct area should be a separate MAZ.
* MAZs of small irregular shapes should be reviewed and the possibility of merging it with adjacent MAZs should be explored.
* MAZs that only encompass a roadway element (e.g., a rectangular-shaped MAZ that covers a freeway and no open space, buildings, or households) should be combined with neighboring MAZs.
* MAZs should recognize physical boundaries. Ideally, if a TAZ is delineated properly there will not be any conflicts of MAZ boundaries with barriers and major facilities. But for a MAZ, care should be taken so as not to cut across streets, unless the street experiences very low volumes or it is an alley. If MAZs encompass land on either side of a higher-level street, they should be split.
* For areas in which development is anticipated, a relatively large MAZ that is generated from the procedure should be separated into smaller MAZs.
* MAZ centroids should be the center of activity for people, i.e. where people entering the zone want to go. Centroids generated based on geographic center should be reviewed to ensure this. Centroid connectors for MAZs connect to the all-streets navigation network and are generated on the fly.
* Another criteria for coding MAZs is that they should be small if there is a high density of TAPs, since we want to reflect the choice/walking time between MAZs and TAPs and if the MAZs are too big they won’t increase the accuracy of the walk times in the model.

## Procedure to create MAZs

Creation of MAZ is a partially automated and a partially manual process. For creating a MAZ layer, the Census Block layer and the TAZ layer for the region are required.

1. The Census block layer needs to be reviewed for small and irregular polygons. These should either be dissolved into adjacent blocks or when necessary the boundaries should be edited.
2. Clip the edited block layer by the TAZ layer.
3. Calculate “sliverness” and “roundness” (add two fields if it’s first time):
   * SLIVERNESS= [Shape\_Area]/[Shape\_Length]
   * ROUNDNESS= ([Shape\_Area]\*4\*3.14)/([Shape\_Length]^2)
4. Identify the sliver polygons. The thresholds (S and R) for identifying the sliver polygons should be arrived at based on experimentation by the analyst (an example set of thresholds is shown in Table 9.1 for the Mid-Ohio Regional Planning Commission (MORPC) model). For each pass of elimination the analyst should judge if the selection rule identifies sliver polygon to a reasonable degree.
   * SLIVERNESS<=S AND ROUNDNESS <=R

Table . Thresholds used for creating MAZ system for MORPC

|  |  |  |
| --- | --- | --- |
| **Iteration** | **S** | **R** |
| **1** | **<=30 feet** | **-** |
| **2** | **<=30 feet** | **-** |
| **3** | **<=120 feet** | **0.9-1.1** |
| **4** | **<=120 feet** | **<=0.1** |
| **5** | **<=50 feet** | **<=0.15** |
| **6** | **<=40 feet** | **-** |
| **7** | **<60 feet** | **<=0.40** |
| **8** | **<=42 feet** |  |
| **9** |  | **<=0.07** |

1. Apply “Eliminate” tool (Data Management/Generalization) to remove the sliver polygons (eliminating polygon by “LENGTH”).
2. Intersect the resulted polygon in Step 5 with the TAZ layer to generate the preliminary MAZ layer
3. Calculate “sliverness” and “roundness” for the preliminary MAZ layer polygons:
   * SLIVERNESS= [Shape\_Area]/[Shape\_Length]
   * ROUNDNESS= ([Shape\_Area]\*4\*3.14)/([Shape\_Length]^2)
4. Eliminate slivers TAZ by TAZ based on a selection criteria. (this can be automated using a script)
5. Determine how many slivers remain after Step 8 and repeat Step 8 if necessary.

The automated process just described should be accompanied by visual inspection of the resulting spatial system. The analyst should verify if the zone system makes intuitive sense and identify outliers (for example an “island zone” – a zone completely contained within another zone or zones that project into the ocean are some of the issues that were encountered in regions such as MORPC and MTC) that are missed in the cleanup process.

Please note that the approach just described assumes that the region has a finalized TAZ layer to work with. Another approach would be to start by creating a MAZ system from the Census block layer (applying steps 1, 3, 4 and 5) and then grouping MAZs to create TAZs.

# Methodology for developing TAPs

TAPs represents transit access points in the region – a TAP can be a single transit stop or an abstract location representing a collection of stops. TAPs serve as the origins and destinations of transit trips for network assignment purposes.

* TAPs should nest within MAZs and they should be coded with the ID of the TAZ it nests in (the coding is achieved programmatically).
* Stops served by express service should be coded as an individual TAP.
* When multiple stops are collapsed into a TAP, the location of the TAP should be closest to the stop that experiences maximum foot falls. This is done to accurately capture the relative competition of the stops within a TAP.
* Transfer points should be represented by a TAP as more often than not these also serve as major access points.
* Again, TAP to MAZ and TAP to stop connectors are generated on the fly.

## Procedure to create TAPs

The procedure selects stops to become TAPs through a series of sampling and buffering routines by mode. This procedure has been previously used to automate the TAP creation for MTC and Oregon Statewide network. For the SOABM model, all stops were used as TAPs since there were less than 500, which avoids runtime and data management issues. The procedure can be described in the following steps:

1. Calculate or retrieve the following stop attributes:
   1. Number
   2. Number of lines served
   3. X coordinate
   4. Y coordinate
   5. Modes served
   6. Is beginning or end of a line
   7. Number of stops within a user specified buffer distance (currently ½ mile)
   8. Best mode served (Express service if available; if not then bus)
2. Label each stop that is the beginning or ending of a line as a TAP.
3. Labels each stop that serves a premium (fixed-guideway transit such as bus rapid transit or light-rail) mode as a TAP.
4. By bus line, iterate through stops and calculate a dynamic buffer that is a function of the number of lines served and stops within the base buffer.



Table 10.1 shows the buffer thresholds used when processing the Oregon statewide network. These thresholds should be customized by the analyst to better represent TAP coverage in the region.

This buffer is used to identify candidate stops to become TAPs. Essentially it is used to sample stops along routes and is sensitive to stop density to ensure more TAPs are created in more stop dense areas.

Table . Thresholds used for creating MAZ system for MORPC

|  |  |  |
| --- | --- | --- |
| **Stops In Buffer** | **Base Buffer (miles)** | **Buffer range (from / to (miles))** |
| **1-5** | **0.50** | **0.224 to 0.500** |
| **6-15** | **0.33** | **0.085 to 0.135** |
| **16-30** | **0.25** | **0.046 to 0.063** |
| **31+** | **0.15** | **<0.027 to 0.027** |

1. After identifying candidate stops to label as TAPs, loop through the candidate stops and condense the candidate stops to TAPs using a restricted buffer set to a 0.25 miles distance.
2. All unassigned candidate stops by mode within the current TAP buffer are assigned to the current TAP.
3. Finally, all stops without a TAP are assigned the closest TAP by mode.

At the end of the procedure each stop is coded with its TAP stop. In practice, a new zone centroid will be created for each TAP node and stop to TAP connectors will be created programmatically to connect the transit network stops to the TAP nodes for assignment.

# SOURCE OF LAND-USE INputs

This section documents the source of land-use inputs to be prepared for new deployments of OR-RAMP and alternative land-use scenarios for the Southern Oregon model. The following sections lists out the variables in the input file and describes how to develop each variable.

Zonal data is specified for a number of purposes, including:

* Constraints on the population synthesizer
* To calculate of density, 4D variables, and accessibility variables
* Describe the cost of parking considered by mode choice
* Used as size (attractiveness) terms in destination choice models

The OR-RAMP model requires zonal data specified at the MAZ level. Table 11.1 shows the fields in the MAZ data file and the source of each field. Most of the data in the file is assembled using outputs from the socio-economic forecast and Census data.

Table . MAZ Inputs and Their Source

|  |  |  |
| --- | --- | --- |
| **Column** | **Description** | **Source** |
| MAZ | MAZ Number |  |
| TAZ | TAZ Number |  |
| HH | Total number of occupied housing unit | This set of data can be obtained at Census Block level from Census or at MAZ/TAZ level from socio-economic forecast. In case the data is obtained from the Census the households needs to be allocate to the MAZ based on point-in-polygon approach or proportionally distributed based on the overlapping area.  Group quarters control totals can be derived from Census non-institutionalized Group Quarters totals. However, they should be further broken out into military, university, and other group quarters by obtaining local land-use data.  The University GQ population can be estimated based on the number of beds in University dorms and fraternity/sorority houses, from local university plans or discussions with university facilities department staff.  Military group quarters can be obtained by discussions with local military base housing officials. It is only necessary for regions with significant military employment. |
| HH\_SF | Number of occupied single-family households |
| HH\_MF | Number of occupied multi-family households |
| HH\_MH | Number of occupied mobile-home households |
| HH\_DUPLEX | Number of occupied duplex households |
| HH\_GQ\_CIV | GQ civilian (half-way houses and other similar group quarters not associated with military housing, on-campus housing, or fraternities/sororities) |
| HH\_GQ\_MIL | GQ military (includes military group quarters housing) |
| HH\_GQ\_UNIV | GQ University students (on-campus or off-campus dormitories, fraternities, and sororities) |
| HHS | Average household size in the MAZ |
| POP | Total population |
| HHP | Total household population (exclude GQ pop) |
| EnrollGradeKto8 | Grade School K-8 enrollment | Enrollment data collected from schools/major University. |
| EnrollGrade9to12 | Grade School 9-12 enrollment |
| collegeEnroll | Major College enrollment |
| otherCollegeEnroll | Other College enrollment |
| AdultSchEnrl | Adult School enrollment |
| ECH\_DIST | Elementary school district | Point to polygon using MAZ centroid layer and school district. |
| HCH\_DIST | High school district |
| HotelRoomTotal | Total number of hotel rooms | Landuse/ Hotel inventory |
| WRK\_EXT\_PROB | Probability of working outside the region | Derived from SWIM model. A single value of probability for the entire region or a measure for district/county or any finer geography depending on availability. |
| universitySqFt | Number of square feet of major university active space | Land-use data |

Table 1: MAZ Inputs and Their Source [continued]

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Description** | **NAICS** | **Source** |
| EMP\_CONSTR | Construction | 23 | Number of employees working in the specified sector from SEC data or another employment data source. |
| EMP\_WHOLE | Wholesale Trade | 42 |
| EMP\_RETAIL | Retail Trade | 44 |
| EMP\_SPORT | Sporting Goods, Hobby, Musical Instruments and Book stores | 45 |
| EMP\_ACCFD | Accommodation and Food Services | 72 |
| EMP\_AGR | Agriculture, Forestry, Fishing and Hunting | 11 |
| EMP\_MIN | Mining, Quarrying, and Oil and Gas Extraction | 21 |
| EMP\_UTIL | Utilities | 22 |
| EMP\_FOOD | Food Manufacturing | 31 |
| EMP\_WOOD | Wood Product Manufacturing | 32 |
| EMP\_METAL | Primary Metal Manufacturing | 33 |
| EMP\_TRANS | Transportation | 48 |
| EMP\_POSTAL | Postal Service | 49 |
| EMP\_INFO | Information | 51 |
| EMP\_FINANCE | Finance and Insurance | 52 |
| EMP\_REALEST | Real Estate and Rental and Leasing | 53 |
| EMP\_PROF | Professional, Scientific, and Technical Services | 54 |
| EMP\_MGMT | Management of Companies and Enterprises | 55 |
| EMP\_ADMIN | Administrative and Support and Waste Management | 56 |
| EMP\_EDUC | Educational Services | 61 |
| EMP\_HEALTH | Health Care and Social Assistance | 62 |
| EMP\_ARTS | Arts, Entertainment, and Recreation | 71 |
| EMP\_OTHER | Other Services, Religious | 81 |
| EMP\_PUBADMIN | Public Administration | 92 |
| EMP\_TOTAL | Total employment |  |

Table 1: MAZ Inputs and Their Source [continued]

|  |  |  |
| --- | --- | --- |
| **Column** | **Description** | **Source** |
| parkarea | Category determining functionality of parking models –  parkarea field codes:   1. Trips with destinations in this MAZ may choose to park in a different MAZ, parking charges apply (downtown) 2. Trips with destinations in parkarea 1 may choose to park in this MAZ, parking charges might apply (quarter mile buffer around downtown) 3. Only trips with destinations in this MAZ may park here, parking charges apply (outside downtown paid parking, only show cost no capacity issue) 4. Only trips with destinations in this MAZ may park here, parking charges do not apply (outside downtown, free parking) | This data should be derived from a Parking lot inventory |
| hstallsoth | Number of stalls allowing hourly parking for trips with destinations in other MAZs |
| hstallssam | Number of stalls allowing hourly parking for trips with destinations in the same MAZ |
| hparkcost | Average cost of parking for one hour in hourly stalls in this MAZ, dollars |
| numfreehrs | Number of hours of free parking allowed before parking charges begin in hourly stalls |
| dstallsoth | Stalls allowing daily parking for trips with destinations in other MAZs |
| dstallssam | Stalls allowing daily parking for trips with destinations in the same MAZ |
| dparkcost | Average cost of parking for one day in daily stalls, dollars |
| mstallsoth | Stalls allowing monthly parking for trips with destinations in other MAZs |
| mstallssam | Stalls allowing monthly parking for trips with destinations in the same MAZ |
| mparkcost | Average cost of parking for one day in monthly stalls, amortized over 22 workdays, dollars |
| INTER\_HALFMI | Intersection count within ½ mile | Calculated internally by the OR-RAMP |
| DU\_DEN | Dwelling unit density within ½ mile |
| EmpDen | Employment density within ½ mile |
| PopDen | Population density |
| RetEmpDen | Retail employment density within ½ mile |
| TotIntBin | Intersection count bin |
| EmpDenBin | Employment density bin |
| DuDenBin | Dwelling unit density bin |

# Network Revisions and Maintenance

This section describes how to code the SOABM network for new scenarios. An overview of the usage of VISUM network objects in SOABM is presented in Section 4.1 |Networks and Zones. The SOABM version file is using the following VISUM network objects. This section assumes familiarity with highway and transit network modeling in VISUM. For more information on VISUM, refer to VISUM’s HTML Help, which is available via Help + HTML Help.

## Non-Motorized Network

The non-motorized network is managed in the VISUM network as well. The Bike and Walk transport systems are used for generating bike and walk network impedances. The TransitWalk transport system is used for walk access to transit.

In order to code a link for walk, set the TSysSet to include Walk. The walk distance and time between MAZs is calculated using link length at the user-defined walk speed.

In order to code a link for bike, set the TSysSet to include Bike. The bike distance and time between MAZs is calculated using link length at the user-defined bike speed.

Walk access/egress to transit is modeled with the TransitWalk transport system, which is available to the Transit mode when generating skims and assigning trips. Since skimming and assignment is done from TAP to TAP, the actual walk access/egress transit connectors are quite short. Short little connectors are created from each TAP zone centroid, which started as a StopArea centroid, to each StopArea’s network access node. These connectors are automatically created by the skimming and assignment procedures and the travel time is calculated using the user defined walk speed.

## Issues to Be Aware of When Editing the Network

Unlike a traditional trip-based model, which has one zone system, the SOABM has three zone systems. Since VISUM’s network data model only supports one zone system for skimming and assignment, the VISUM Zones need to be changed on-the-fly from TAZs to MAZs or TAPs. This means that each version file created during the model run may have a different set of zones and connectors depending on the purpose. Thus, it is important to always start with the input master version file when making editing since it contains all three zone systems. The VISUM Zones are TAZs, the VISUM MainZones are MAZs, and the VISUM StopAreas are TAPs.

After editing the auto or non-motorized network, it is a good idea to check network connectivity. To do so, run the following steps:

1. Go to Calculate + Network Check
2. Unclick all the procedure except for Check network consistency between “All zones” for each transport system edited
3. Run the procedure
4. Click the ! to review the errors (i.e. OD pairs that are not connected)
5. Close Calculate + Network Check
6. Go To Graphics + Shortest Path Search (Figure 27)
7. Try to find a shortest path for the transport system edited for an OD pair that is not connected. Use distance for walk and bike and t0 for auto modes.
8. Work from the origin zone or destination zone toward the other end of the trip to find the unconnected links, turns, or nodes.
9. Repair the connection and repeat until all OD pairs are connected.

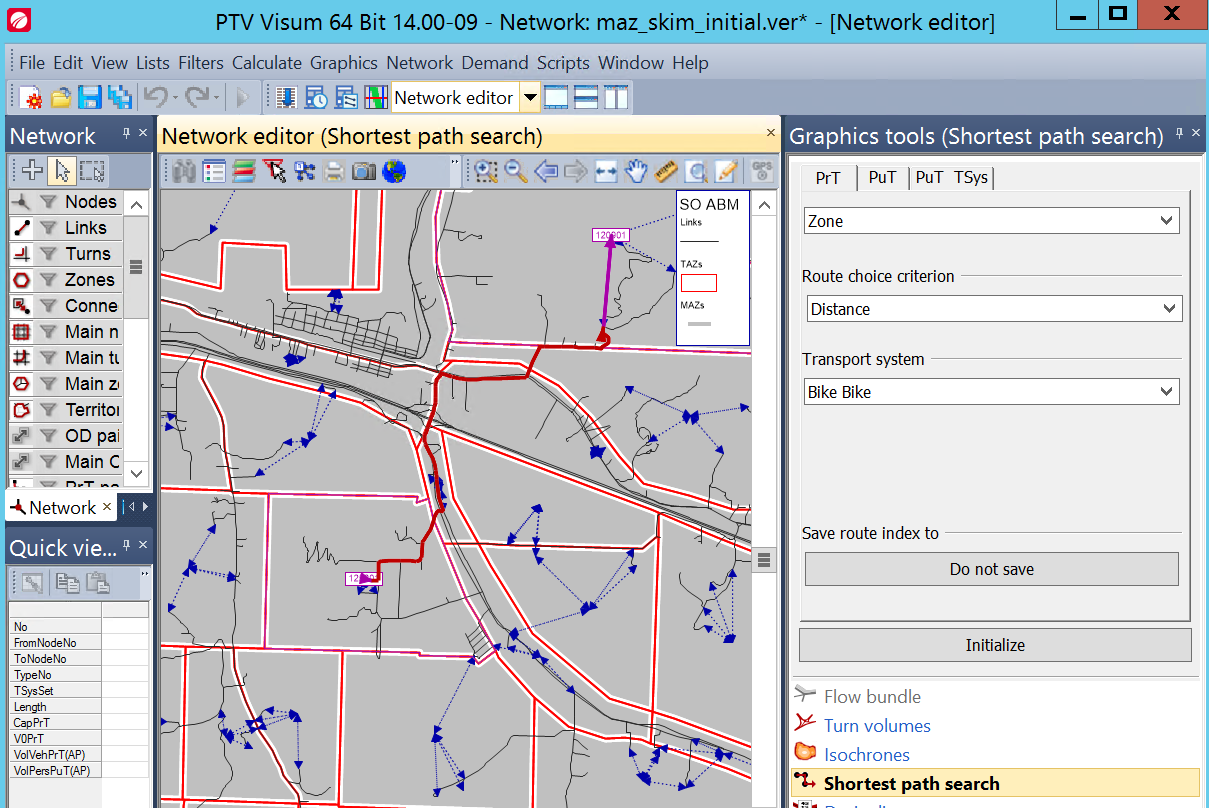


Figure – Network Shortest PATH Search CheCk

Transit line run and dwell times are set using congested link travel times. In cases where auto assignment is not possible, line route travel time based on auto assignment time is also not possible. In this case, such as transit lines with dedicated right-of-way (i.e. links), run and dwell time must be set manually for the line route. To check, or set, the transit route run and dwell times, do the following:

1. Set Lines as the active network objects
2. Switch the dialog to Line Routes and double click the line route of interest
3. Switch to the Items and time profiles tab as shown in Figure 28 below.
4. Make sure the Board, Alight, Stop time, and Run time values are as expected. It is important to do this after the Set Run and Dwell Times procedure is run by the TAP skimming and assignment procedures since it updates these values.

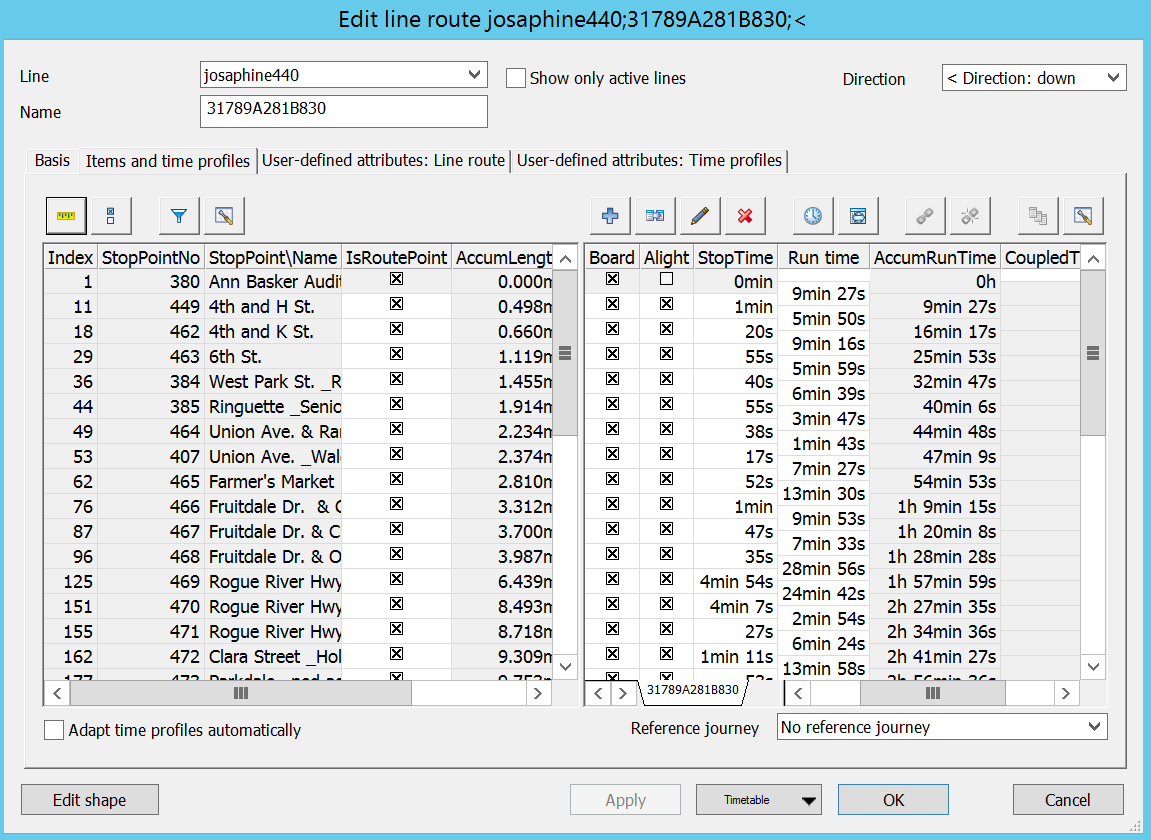


Figure - Line Route Run and Dwell Times

When editing bike and walk paths, it is important to add nodes at various points along the paths in order to eventually get the MAZ connectors that are likely desired. MAZ connectors are built on-the-fly (as described earlier) to available nearby nodes, and if bike or walk paths are not broken-up into a series of links, then there are few nodes for the destination connector node.

1. https://en.wikipedia.org/wiki/Well-known\_text [↑](#footnote-ref-1)
2. https://usa.ipums.org/usa/resources/codebooks/DataDict0711.pdf [↑](#footnote-ref-2)
3. http://analytics.mtc.ca.gov/foswiki/Main/UtilityExpressionCalculator [↑](#footnote-ref-3)