Transportation Authority of Marin Demand Model (TAMDM) User's Guide

Prepared for:

TAM
Transportation Authority of Marin

September 17, 2020

WC16-3330

FEHR PEERS

RSG
the science of insight

Table of Contents

List of Figures

List of Tables

This page intentionally left blank.

1. Introduction

The Transportation Authority of Marin (TAM) has been working with RSG and Fehr & Peers on deploying the Metropolitan Transportation Commission (MTC) Travel Model Two (TM2) travel modeling system for its use. TM2 is a sophisticated activity-based modeling system that covers the entire nine-county San Francisco Bay Area. The model is a member of the Coordinated Travel – Regional Activity-based Modeling Platform (CT-RAMP) family of models, implemented in a number of large metropolitan areas including Atlanta, San Diego, Miami, and Chicago. The MTC version of CT-RAMP features a very detailed spatial system including an all-streets transportation network with 4,800 Transportation Analysis Zones (TAZs) and almost 40,000 Micro-Analysis Zones (MAZs) for precise measurements of walking and biking time. The model also utilizes 6,200 Transit Access Points (TAPs) for transit stop-to-stop level-of-service. TM2 simulates 2.7 million households and 7.4 million persons in 2015.

This report is intended to provide Transportation Authority of Marin Demand Model (TAMDM) users with recommended practices for using the travel demand model to evaluate the potential impacts to the transportation system of proposed land use development and infrastructure improvements within Marin County. Instructions for performing common modeling tasks associated with the recommended practices are provided, including instructions for how to modify the transportation networks and land use/socio-economic data (SED), install and run the model, and view the model outputs. This report also provides information on how TAMDM can be used to develop traffic volume and transit ridership forecasts, including guidelines for how to adjust raw model outputs to account for the discrepancy between a base year field count and a base year model volume. For more general information regarding Citilabs travel demand modeling, please refer to the Citilabs Learning Center[1].

[1] http://www.citilabs.com/support-services/learning-center

2. Model Overview

MTC TM2 served as the starting point for TAMDM, and includes all enhancements made by MTC to Travel Model One (TM1) as part of the TM2 development process. This chapter provides a brief overview of MTC TM2.

2.1 MTC Travel Model Two

MTC TM2 is based on the CT-RAMP (Coordinated Travel Regional Activity-Based Modeling Platform) family of Activity-Based Models (ABMs). This model system is an advanced, but operational, ABM that fits the needs and planning processes of MTC and Marin County. The CT-RAMP model adheres to the following basic principles:

- The CT-RAMP design attempts to model individual travel choices with behavioral realism. 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 travel time-of-day and peak spreading of traffic volume.
- Reflects and responds to detailed demographic information, including household structure, aging, changes in wealth, and other key attributes.
- Offers sensitivity to demographic and socio-economic changes observed or expected in the dynamic San Francisco Bay region. This is ensured by the enhanced and flexible population synthesis procedures as well as by the fine level of model segmentation.
- Accounts for the full set of travel modes including non-motorized travel and transit.
- Responds to accessibilities throughout the model system including effects of accessibility on auto ownership and frequency of travel.

CT-RAMP is a proven design, intensively tested in practice in several regions. MTC TM2 is based on a design that was transferred from the San Diego County (SANDAG) model. This model was used by SANDAG to develop the last San Diego RTP and has been successfully transferred to a number of other regions including Miami, Chicago, and Southern Oregon.

3. Installing the Model

This chapter describes the steps to install TAMDM including the computing environment and software/hardware requirements.

3.1 Computing Environment

The hardware and software used to execute TAMDM are described in this section. To date, MTC has not experimented enough with the model to define the minimum or ideal hardware configuration. As such, the description here is for a hardware set up that is sufficient – not optimal. It is important to note that both the software and model structure are highly configurable and flexible; depending on the analysis needs, the required computing power could vary dramatically.

3.1.1 Hardware

MTC uses four identical servers with the following characteristics:

- Operating system: Microsoft Windows Server 2007 with Service Pack 2, 64-bit edition;
- Processors: Two Intel Xeon X5570 @ 2.93 GHz (i.e., two quad-core processors with hyper-threading capability);
- Memory (RAM): 96.0 GB

These four computers act in different roles, each requiring different amounts of memory and computing power. The four computers are named as follows: mainmodel, satmodel (for satellite), satmodel2, and satmodel3. The mainmodel computer plays a specialized role in the system design; the satellite machines each play identical and completely interchangeable roles.

3.1.2 Software

The following software are required to execute TAMDM.

Citilabs Cube Voyager

TAMDM currently uses version 6.4.2 of Citilabs Cube software. The Cube software is used to build skims, manipulate networks, manipulate matrices, and perform assignments.

Citilabs Cube Voyager 64bit Matrix I/O DLL

The CT-RAMP software needs to access data stored in a format dictated by Cube. This is accomplished through a 64-bit DLL library specifically for matrix I/O, which must be accessible through the PATH environment variable. To install the DLL:

- Run VoyagerFileAPIInstaller.msi, which is included in the "CTRAMP\runtime" folder
- Ensure VoyagerFileAccess.dll is in the "CTRAMP\runtime" folder
- Ensure the Microsoft Visual C++ 2012 redistributable is installed on the matrix server machine. Make sure to get version "110" DLLs (MSVCR110.dll and MSVCP110.dll). These files can be obtained from the Microsoft. Download and install vcredist_x64.exe.

Citilabs Cube Cluster

The Cube Cluster software allows Cube scripts to be multi-threaded. In the current approach, the travel model uses 64 computing nodes across four machines. The Cube scripts can be manipulated to use any number of computing nodes across any number of machines, provided each machine has, at a minimum, a Cube Voyager node license (for the time being, MTC has found 64 nodes on a single machine to be the most effective approach – in terms of reliability and run time). Cube Cluster is not strictly necessary, as the Cube scripts can be modified to use only a single computing node. Such an approach would dramatically increase run times.

Java and CT-RAMP

TAMDM operates on the open-source Coordinated Travel - Regional Activity-based Modeling Platform (or CT-RAMP) developed by Parsons Brinckerhoff. The software is written in the Java programming language. CT-RAMP requires the 64-bit Java Development Kit version 1.8 or later to be installed on each computer running the CT-RAMP software. The Java Development Kit includes the Java Runtime Environment. The 64-bit version of the software allows CT-RAMP to take advantage of larger memory addresses. The details of setting up and configuring the software are presented in the Setup and Configuration section of this chapter.

Python

Certain network processing programs are written in Python. Python must be installed on the computer executing the Cube scripts – mainmodel in MTC's configuration.

Python Rtree Library

The open source Python rtree library is required for a script that dynamically codes link area type based on land use data. The rtree library provides an efficient spatial index for looking up all spatial units within a buffered distance from each spatial unit. To install, open a dos prompt, navigate to the directory and type: pip install Rtree-0.8.2-cp27-cp27m-win_amd64.whl.

Microsoft Excel

The CT-RAMP software allows discrete choice models to be specified via so-called Utility Expression Calculators. These files are Excel-based.

Remote Execution and Stop Utilities

The Microsoft PsKill and PsExec programs are used to remotely kill programs and execute programs.

3.2 System Design

This section describes the manner in which the software is configured to take advantage of the available hardware.

3.2.1 Distributed Computing

The MTC travel model uses two types of distributed applications. The first is facilitated by the Cube Cluster software and allows the skim building and assignment steps to utilize multiple threads. The second is facilitated by the CT-RAMP software, which allows the choice models to be distributed across multiple threads and multiple computers. A brief overview of both of these applications is provided below.

Cube Cluster

Citilabs Cube scripts facilitate two types of distribution, both of which are highly configurable through the Cube scripting language and the Cube Cluster thread management system; the two distinct types of multi-threading are as follows:

- Intra-step threading: The DistributeINTRAStep keyword allows calculations that are performed across a matrix of data to be performed in blocks – specifically rows of data – across multiple threads. TAMDM uses intra-step threading in highway assignment, allowing shortest paths to be constructed for more than one origin at a time. Complex matrix calculations can also benefit from intra-step threading.
- Multi-step threading: The DistributeMULTIStep keyword allows blocks of code to be distributed across multiple threads. For example, if the same calculations are being performed for five different time periods, the same block of code (with variables) can be distributed across computers for parallel processing. This type of Cube multi-threading is a bit less flexible than the intra-step threading as it requires threads to be identified *a priori* (e.g., thread one will do the calculations for time period A), where the intra-step threading can be given a list of available

processes and use what is available. TAMDM uses multi-step threading for highway skimming, transit skimming, highway assignment, the conversion of trip lists to trip matrices, highway assignment, and transit assignment.

As noted in the Computing Environment section, MTC specifies the Cube scripts to take advantage of 64 threads. A user can easily adjust the necessary scripts to take advantage of more or fewer processors.

CT-RAMP

The CT-RAMP software allows for the choice models to be distributed across multiple threads and machines. The MTC application currently uses four machines, but the CT-RAMP software can be configured fairly easy to utilize fewer or more machines. CT-RAMP uses the Java Parallel Processing Framework, or JPPF, to manage the distribution of tasks. JPPF is an open-source Java package. The JPPF framework consists of three main parts as follows:

1. a driver, also referred to as the JPPF server;
2. one or more nodes, typically one node is established on each machine; and,
3. a client, the CT-RAMP software in this case.

As noted in the Computing Environment section, MTC uses four computers with the names mainmodel, satmodel, satmodel2, and satmodel3. The JPPF driver process is executed on mainmodel and acts like a traffic cop by acquiring tasks from the client and distributing those tasks to the node processes. When the node processes complete tasks, the results are returned back to the client via the JPPF driver. Three nodes are used in the MTC application, one each on satmodel, satmodel2, and satmodel3 (each node runs 12 processes). These three nodes are created prior to executing a model run. After being created, each node listens for tasks from the JPPF driver.

Node processes receive tasks, perform those tasks, and return the results. Nodes are configured to communicate with the driver process when they are started. MTC configures the nodes to use 90 GB of memory and 12 threads (see the Setup and Configuration section for details on where these parameters are specified). The JPPF driver attempts to balance computational loads across available nodes. The driver also retrieves class files, i.e. sets of Java code, from the client application and passes those to the nodes as needed.

The CT-RAMP software, which serves as the client, is responsible for creating task objects that can be run in parallel and submitting those to the driver. Because TAMDM simulates households, the CT-RAMP software creates packets of N (a user-configurable quantity, e.g. 500) households and sends those packets to the nodes for processing. As the nodes complete tasks and returns them to the driver, the driver gives the nodes new tasks, attempting to keep each node uniformly busy.

3.2.2 Household Manager and Matrix Manager

Before executing a model run, the travel model requires a Household Manager and a Matrix Manager be created. In the MTC application, both the Managers reside on the satmodel computer during execution. The Household Manager is tasked with managing the simulated households, as well as each simulated person in each simulated household. The Household Manager provides the JPPF nodes with information regarding the households for which the JPPF nodes are applying choice models and stores the resulting information computed by the JPPF nodes. To help keep run time down, the synthetic population is read from disk and stored in memory at the beginning of the application and then continuously updated as choice models are completed and iterations are performed. When the last iteration is complete, the necessary information is written to disk.

The Matrix Manager is tasked with managing all of the skim matrices used by the choice models. When a skim is needed, a request is made to the Matrix Manager, which then reads the required skim from disk and stores it in memory. Once in memory, each matrix is available to any other JPPF node process that may need it.

Both the Household Manager and Matrix Manager have substantial memory footprints, currently 35GB and 44GB, respectively.

3.2.3 TAZ, MAZ, and TAP Data Manager

The main CT-RAMP model process includes the following internal data management interfaces for managing zone-type data:

- TAZ data manager,
- MAZ data manager, and
- TAP data manager.

These data managers provide TAZ, MAZ, and TAP level data to the various sub-models. The TAZ data manager provides TAZ attribute data. The MAZ data manager provides MAZ attribute data, as well as MAZ to MAZ impedances. The TAP data manager provides TAP attribute data, as well as MAZ to TAP impedances. These data managers are copied to JPPF nodes during execution, which increases the memory required by JPPF nodes.

3.3 Setup and Configuration

This section provides details on setting up TAMDM to run on a cluster of computers, including descriptions of the necessary configuration files.

Step 1: Create the required folder structure

The TAMDM folder structure is described below. Typically the parent directory will be named for the scenario to be run (e.g. '2010_base'). The parent directory contains two subfolders ctramp and input, and one MS-DOS batch file, RunModel.bat. This folder can be placed in any directory on a computer designated as the main controller of the program flow. On MTC's setup, these files are placed on the mainmodel computer.

The CTRAMP directory contains all of the model configuration files, Java instructions, and Cube scripts required to run the travel model, organized in the following three folders:

- model – contains all of the Utility Expression Calculators files that specify the choice models;
- runtime – contains all of the Java configuration and JAR (executable) files, as well as the files necessary for Java to communicate with Cube;
- scripts – contains all of the Cube scripts and associated helper files.

The INPUT directory contains all of the input files (see the Input Files section) required to run a specific scenario. TAMDM will be delivered with a set of scenario-specific inputs. When configuring the model on a new computing system, one should make sure that the results from an established scenario can be recreated before developing and analyzing a new scenario. The INPUT directory contains the following folders:

- hwy – contains the input master network with all zone centroids as well (TAZ, MAZ, and TAP);
- trn – contains all of the input transit network files;
- landuse – contains the MAZ and TAZ level socio-economic input land use files;
- nonres – contains the fixed, year-specific internal/external trip tables, the fixed, year-specific air passenger trip tables, and files used to support the commercial vehicle model;
- popsyn – contains the synthetic population files;
- warmstart – contains trip tables for warm-starting the model.

The RunModel.bat script contains a list of MS-DOS instructions that control model flow.

Step 2: Map a network drive to share across computers

As noted in the previous section, the MTC model files can be placed within any directory. After establishing this location, the user must map a network drive to a shared folder to allow other computers access. On MTC's machine, the directory E:\MainModelShare is first mapped to the letter drive M:\ and this directory is then shared across on the network (M:\ = \\MainModel\MainModelShare\). Satellite computers should also map the letter drive M:\ to this network location. Please note that the model components running on the main machine should use the local version of the directory (i.e. M:\Projects\) rather than the network version (i.e. \\MainModel\MainModelShare\Projects\).

Step 3: Configure the CT-RAMP and JPPF Services

Much of the configuration of the CT-RAMP software is done automatically by the RunModel.bat batch file. However, prior to executing a model run, the files controlling the CT-RAMP and JPPF services may need to be configured. Please see the System Design section for a broad overview of these services. When executing the travel model, the following start-up scripts need to be run separately on each computer. Each script specifies the tasks assigned to each computer and need not be configured exactly as described in the System Design section (below is a description of MTC's setup; numerous other configurations are possible). In the MTC setup, the following commands are executed:

1. runDriver.cmd starts the JPPF Driver required for distributed model running;
2. runHhMgr.cmd starts the household manager on satmodel1;
3. runMtxMgr.cmd starts the matrix manager on satmodel1;
4. runNode{X}.cmd starts up JPPF worker nodes on the remaining node {X} machine(s);

The .cmd files are MS-DOS batch scripts and can be edited in a text editor, such as Notepad.

Each program requires an explicit amount of memory to be allocated to it. The amount of memory allocated to each program is identified by the -Xmx parameter (XXXm allocates XXX megabytes; Xg allocates X gigabytes). This may need to be adjusted depending on the model and hardware configurations. An example is as follows:

```
.. %JAVA_PATH%\bin\java -server -Xmx35000m ...
```

Most of the JPPF-related configuration parameters have been optimized for the MTC travel model application and, as such, need not be modified. There are, however, a handful of parameters described in the table below that may need to be modified to meet the specifications of the computing environment upon which the model is being executed. Each of the files listed below can be found in the "CTRAMP\runtime\config" directory.

Table 3-1: Model Parameter Files

File Name	File Function	Statement	Purpose
jppf-clientDistributed.properties	JPPF Client Driver Control file	driver1.jppf.server.host = 192.168.1.200	IP address of the main computer (mainmodel at MTC)
jppf-clientLocal.properties	JPPF Client Local Control file	jppf.local.execution.threads = 22	Number of threads to use for running the model on one machine for testing (mainly for debugging)
jppf-driver.properties	JPPF Driver Control file	jppf.server.host = 192.168.1.200	IP address of the main computer (mainmodel at MTC)
jppf-node{x}.properties	Remote JPPF Node Control file	jppf.server.host = 192.168.1.200	IP address of the main computer (mainmodel at MTC)
		processing.threads = 12	Number of computing cores on node {X}
		other.jvm.options = -Xms48000m -Xmx48000m -Dnode.name=node0	Maximum amount of memory, in MB, to allocate to node {X} and node name for logging

A file that needs to be edited prior to executing a model run is the mtctm2.properties file located in the "CTRAMP\runtime" folder. This file serves as the general control module for the entire CT-RAMP application. At this stage, the following variables need to be modified for the software to execute the model properly.

Table 3-2: General Control Module File

Statement	Purpose
RunModel.MatrixServerAddress = 192.168.1.200	The IP address of the machine upon which the Matrix Manager is being executed (satmodel1 at MTC)
RunModel.HouseholdServerAddress = 192.168.1.200	The IP address of the machine upon which the Household Manager is being executed (satmodel1 at MTC)

Step 4: Configure RunModel.bat and CTRampEnv.bat

The file RunModel.bat MS-DOS batch file that executes the model stream needs to be consistent with the network and if a specialized model run is being executed, the model flow logic and/or sample rates may need to be adjusted. The following statements may need to be configured within this file:

Table 3-3: RunModel.bat MS-DOS Batch File

Statement	Purpose
set SAMPLERATE_ITERATION{iteration}=0.1	Set choice model household sample rate by iteration
model run	
set MODEL_YEAR=2015	Set model year
set BASE_SCRIPTS=CTRAMP\scripts	Set scripts folder
set /A MAX_ITERATION=3	Set the model feedback iterations
set TAZ_COUNT=4509	The number of tazs
set TAZ_EXTS_COUNT=4530	The number of tazs + externals
set TAP_COUNT=6172	The number of transit access point zones

The file CTRampEnv.bat MS-DOS batch file points to locations of executables and contains some additional information on machine configuration. The following statements may need to be configured within this file:

Table 3-4: CTRampEnv.bat MS-DOS Batch File

Statement	Purpose
set JAVA_PATH=C:\Program Files\Java\jdk1.8.0_111	Specify the 64-bit Java path; version 1.8.0+
set TPP_PATH=C:\Progam Files\Citilabs\CubeVoyager	Specify the Cube Voyager path
set CUBE_PATH=C:\Progam Files (x86)\Citilabs\Cube	Specify the Cube path
set PYTHON_PATH=C:\Program Files\anaconda2	Specify the Python path
set RUNTIME=CTRAMP\runtime	Specify the location of the CT-RAMP software (relative to the project directory)
set JAVA_32_PORT=1190	Specify the port for Java 32 bit matrix reader\writer (not currently used)
set MATRIX_MANAGER_PORT=1191	Specify the port for the matrix manager
set HH_MANAGER_PORT=1129	Specify the port for the household manager
set HHMGR_IP=172.24.0.100	Specify IP address of household manager
set MATRIX_SERVER=\\w-ampdx-d-sag02	Machine running matrix data manager
set MATRIX_SERVER_BASE_DIR=%MATRIX_SERVER%\c$\projects\mtc\%SCEN%	Machine running matrix data manager base directory

Table 3-4: CTRampEnv.bat MS-DOS Batch File

Statement	Purpose
set MATRIX_SERVER_ABSOLUTE_BASE_DIR=c:\projects\mtc\%SCEN%	Machine running matrix data manager absolute directory
set MATRIX_SERVER_JAVA_PATH=C:\Program Files\Java\jdk1.7.0_25	Machine running matrix data manager Java install
set HH_SERVER=\\w-ampdx-d-sag02	Machine running household data manager
set HH_SERVER_BASE_DIR=%HH_SERVER%\c$\projects\mtc\%SCEN%	Machine running household data manager base directory
set HH_SERVER_ABSOLUTE_BASE_DIR=c:\projects\mtc\%SCEN%	Machine running household data manager absolute directory
set HH_SERVER_JAVA_PATH=C:\Program Files\Java\jdk1.7.0_25	Machine running household data manager Java install
set UN=	Username for logging in to remote machines
set PWD=	Password for logging in to remote machines

Now that the model is configured, the user can run the model, as described below.

3.4 Model Execution

This section describes how RunModel.bat executes the travel model.

Step 1: Set globally-available environmental variables

See Setup and Configuration for complete details.

Step 2: Set relevant paths to access software

See Setup and Configuration for complete details.

Step 3: Create and populate a working directory of input files

A working directory is created and populated with the input files, leaving the input files untouched in the process. This step also creates the necessary directory structure for the household and matrix data servers on the remote machine and copies over needed files.

- mkdir hwy
- mkdir trn
- mkdir skims
- mkdir landuse
- mkdir popsyn
- mkdir nonres
- mkdir main
- mkdir logs
- mkdir database
- mkdir ctramp_output

After the directories are created, copies are made to the remote machines for access by the household and matrix servers.

Step 4: Pre-process steps

Several steps are needed to prepare the inputs for use in the model. The following Cube scripts are executed to perform the following:

- zone_seq_net_builder.job – build an internal numbering scheme for the network nodes to play nice with Cube
- CreateNonMotorizedNetwork.job – convert the roadway network into bike and ped networks
- tap_to_taz_for_parking.job – create the transit access point (TAP) data
- SetTolls.job – set network prices (i.e., bridge tolls, express lane tolls) in the roadway network
- SetHovKferPenalties.job – add a penalty of X seconds for dummy links connecting HOV/express lanes and general purpose lanes
- SetCapClass.job – compute area type and populate the CAPCLASS network variable
- CreateFiveHighwayNetworks.job – create time-of-day-specific roadway networks
- BuildTazNetworks.job – create TAZ-scale networks for TAZ-scale roadway assignment

Step 5: Build walk, bicycle, and nearby automobile level-of-service matrices

Two scripts create the level-of-service information for the non-motorized modes and nearby automobile skims (for which, as a simplification, congestion is constant). The following Cube scripts do the job:

- NonMotorizedSkims.job – skim the walk and bicycle networks
- MazMazSkims.job – builds short-distance MAZ-to-MAZ automobile skims

Step 6: Build air passenger demand matrices

The BuildAirPax.job Cube script creates the air passenger demand estimates. Air passenger demand is assumed to be independent of roadway level-of-service and, as such, can be computed a single time.

Step 7: Build highway and transit skims

The following steps create the highway and transit level-of-service matrices:

- Copy data files to the remote househould data manager machine
- Set the sampling rate based on `SAMPLERATE_ITERATION global variable
- HwySkims.job – build the roadway skims
- BuildTransitNetworks.job – build the transit networks using the congested roadway times
- TransitSkims.job – build the transit skims
- Copy the skims and related files to the remote household and matrix data manager machines

Step 8: Execute the CT-RAMP models

The core passenger travel demand models are executed via the CT-RAMP Java code via the following steps:

- Remote worker node(s), as specified, are started using psexec
- Remote household and matrix servers are started using psexec
- JPPF driver, as needed, is started via CTRAMP/runtime/runDriver.cmd
- CT-RAMP models are executed via CTRAMP/runMTCTM2ABM.cmd
- Stops remote servers using pskill
- Copies output matrices from the matrix manager machine back to the main machine
- merge_demand_matrices.s – merges the output demand matrices

Step 9: Execute the internal/external and commercial vehicle models

These ancillary demand models are executed via a series of Cube scripts as follows:

- IxForecasts.job – create the internal/external demand matrices
- IxTimeOfDay.job – apply diurnal factors to the daily internal/external demand matrices
- IxTollChoice.job – apply a toll choice model for express lanes to the internal/external demand
- TruckTripGeneration.job – apply the commercial vehicle generation models
- TruckTripDistribution.job – apply the commercial vehicle distribution models
- TruckTimeOfDay.job – apply the commercial vehicle diurnal factors
- TruckTollChoice.job – apply a toll choice model for express lanes to the commercial vehicle demand

Step 10: Network Assignment

Demand is located on mode-specific paths through the networks in the assignment step via the following steps:

- build_and_assign_maz_to_maz_auto.job – nearby automobile demand assigned to best path on MAZ-scale network
- HwyAssign.job – using nearby demand as background demand, demand assigned to TAZ-scale network
- AverageNetworkVolumes.job – method of successive averages (MSA) applied across overall model iterations
- CalculateAverageSpeed.job – using the averaged volumes, compute speeds
- MergeNetworks.job – merge time-of-day-specific networks into a single network
- IF additional feedback ITERATIONs are needed, GOTO Step 7: Build highway and transit skims
- ELSE perform transit assignment with TransitAssign.job

Step 11: Clean up

The final step of the model run moves all the TP+ printouts to the /logs folder and deletes all the temporary Cube printouts and cluster files.

4. Key Model Input Files

This chapter briefly describes the key model input files.

4.1 Input File List

The table below contains brief descriptions of the key input files required to execute TAMDM.

Table 4-1: Key Model Input Files

Filename	Purpose	Folder location	File type	File format
mtc_final_network.net	Highway, bike, walk network	hwy\	Citilabs Cube	Roadway Network
mazData.csv	Micro zone data	landuse\	CSV	Micro Zonal Data
tazData.csv	Travel analysis zone data	landuse\	CSV	Zonal Data
truckFF.dat	Friction factors for the commercial vehicle distribution models	nonres\	ASCII	Truck Distribution
truckkfact.k22.z1454.mat	"K-factors" for the commercial vehicle distribution models	nonres\	Citilabs Cube	Truck Distribution
truck_kfactors_taz.csv	"K-factors" for the commercial vehicle distribution models	nonres\	CSV	Truck Distribution
ixDailyYYYY.tpp	Internal-external fixed trip table for year YYYY	nonres\	Citilabs Cube	Fixed Demand
IXDaily2006x4.may2208.new	Internal-external input fixed trip table	nonres\	Citilabs Cube	Fixed Demand
YYYY_fromtoAAA.csv	Airport passenger fixed trips for year YYYY and airport AAA	nonres\	CSV	Fixed Demand
households.csv	Synthetic population household file	popsyn\	CSV	PopSynHousehold
persons.csv	Synthetic population person file	popsyn\	CSV	
transitLines.lin	Transit lines	trn\transit_lines	Citilabs Cube	TransitNetwork
transitFactors_MMMM.fac	Cube Public Transport (PT) factor files by transit line haul mode MMMM	trn\transit_support	Citilabs Cube	TransitNetwork

4.2 Synthetic Population

Decision-makers in the TAMDM model system include both persons and households. These decision-makers are created (synthesized) for each simulation year based on tables of households and persons from census data and forecasted TAZ-level distributions of households and persons by key socio-economic categories. These decision-makers are used in the subsequent discrete-choice models to select a single alternative from a list of available alternatives according to a probability distribution. The probability distribution is generated from a logit model which takes into account the attributes of the decision-maker and the attributes of the various alternatives. The decision-making unit is an important element of model estimation and implementation, and is explicitly identified for each model specified (long-term, mobility, travel pattern, stop-level, and trip-level).

4.3 Roadway Networks

The TAMDM highway network was built from an all-streets network from the TomTom (previously TeleAtlas) North America routable network database. The projection is NAD 1983 StatePlane California VI FIPS 0406 Feet.

The TAMDM bike network was built from the highway network and the MTC Bike Mapper network. The Bike Mapper network is a framework in which local cities update a master database of bicycle infrastructure and bicycle lane attributes, from which MTC has built and now maintains a trip planner application.

The TAMDM walk network was built from the highway network and the open source Open Street Map (OSM) network.

4.3.1 County Node Numbering System

The TAMDM highway network uses a numbering system whereby each county has a reserved block of nodes. Within each county's block, nodes 1 through 9,999 are reserved for TAZs, 10,001 through 89,999 are for MAZs, and 90,001 through 99,999 for transit access points or TAPs. The blocks are assigned to the nine counties per MTC's numbering scheme, as shown in the table below.

TeleAtlas network nodes are numbered by county as well and range from 1,000,000 to 10,000,000 as shown below. HOV lane nodes are those nodes corresponding to general purpose lane nodes.

Table 4-2: County Node Numbering System

Code	County	TAZs	MAZs	TAPs	Network Node	HOV Lane Node
1	San Francisco	1 - 9,999	10,001 – 89,999	90,001 – 99,999	1,000,000 – 1,500,000	5,500,000 – 6,000,000
2	San Mateo	100,001 – 109,999	110,001 – 189,999	190,001 – 199,999	1,500,000 – 2,000,000	6,000,000 – 6,500,000
3	Santa Clara	200,001 – 209,999	210,001 – 289,999	290,001 – 299,999	2,000,000 – 2,500,000	6,500,000 – 7,000,000
4	Alameda	300,001 – 309,999	310,001 – 389,999	390,001 – 399,999	2,500,000 – 3,000,000	7,000,000 – 7,500,000
5	Contra Costa	400,001 – 409,999	410,001 – 489,999	490,001 – 499,999	3,000,000 – 3,500,000	7,500,000 – 8,000,000
6	Solano	500,001 – 509,999	510,001 – 589,999	590,001 – 599,999	3,500,000 – 4,000,000	8,000,000 – 8,500,000
7	Napa	600,001 – 609,999	610,001 – 689,999	690,001 – 699,999	4,000,000 – 4,500,000	8,500,000 – 9,000,000
8	Sonoma	700,001 – 709,999	710,001 – 789,999	790,001 – 799,999	4,500,000 – 5,000,000	9,000,000 – 9,000,000
9	Marin	800,001 – 809,999	810,001 – 889,999	890,001 – 899,999	5,000,000 – 5,500,000	9,500,000 – 9,999,999
	External	900,001 - 999,999				

5. Other Model Components

This chapter discusses other TAMDM model components.

5.1 Time-of-Day Model

Time periods in TAMDM are consistent with MTC's Travel Model One:

Table 5-1: TAMDM Time Periods

Time Period	Times	Duration
EA (early AM)	3 am to 6 am	3 hours
AM (AM peak period)	6 am to 10 am	4 hours
MD (mid-day)	10 am to 3 pm	5 hours
PM (PM peak period)	3 pm to 7 pm	4 hours
EV (evening)	7 pm to 3 am	8 hours

5.2 Truck Model

TAMDM uses a simple three-step (generation, distribution, and assignment) commercial vehicle model to generate estimates of four types of commercial vehicles. The four vehicle types are very small (two-axle, four-tire), small (two-axle, six-tire), medium (three-axle), and large or combination trucks (four-or-more-axle).

5.3 Friction Factors

The trip distribution step uses a standard gravity model with a blended travel time impedance measure. This file sets the friction factors, which are vehicle type specific, using an ASCII fixed format text file with the following data:

- Travel time in minutes (integer, starting in column 1, left justified);
- Friction factors for very small trucks (integer, starting in column 9, left justified);
- Friction factors for small trucks (integer, starting in column 17, left justified);

- Friction factors for medium trucks (integer, starting in column 25, left justified); and,
- Friction factors for large trucks (integer, starting in column 33, left justified).

5.4 Fixed Demand

TAMDM uses representations of internal/external and air passenger demand that is year-, but not scenario-, specific – meaning simple sketch methods are used to estimate this demand from past trends. This demand is then fixed for each forecast year and does not respond to changes in land use or the transport network.

5.4.1 Internal/External

So-called internal/external demand is travel that either begins or ends in the nine-county Bay Area. This demand is based on Census journey-to-work data and captures all passenger (i.e. non-commercial) vehicle demand. This demand is introduced to the model via a matrix that contains the following four demand tables in production-attraction format:

- Daily single-occupant vehicle flows;
- Daily two-occupant vehicle flows;
- Daily three-or-more occupant vehicle flows; and,
- Daily vehicle flows, which is the sum of the first three tables and not used by the travel model.

5.4.2 Air Passenger

Air passenger demand is based on surveys of air passenger and captures demand from the following travel modes: passenger vehicles, rental cars, taxis, limousines, shared ride vans, hotel shuttles, and charter buses. This demand is introduced to the model via specific matrices that contain the following six flow tables:

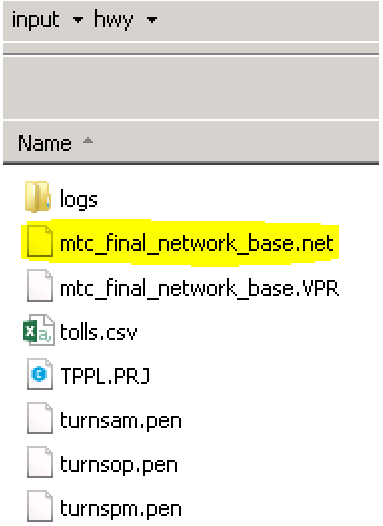
- Single-occupant vehicles;
- Two-occupant vehicles;
- Three-occupant vehicles;
- Single-occupant vehicles that are willing to pay a high-occupancy toll lane fee;
- Two-occupant vehicles that are willing to pay a high-occupancy toll lane fee; and,
- Three-occupant vehicles that are willing to pay a high-occupancy toll lane fee.

6. Common Modeling Tasks

Instructions for performing common modeling tasks are provided in this chapter.

6.1 Highway Network Modifications

In order to update the model highway network, such as to adjust the number of travel lanes on a particular roadway segment, the "mtc_final_network_base.net" file located in the "Inputs\Hwy" folder needs to be modified using the CUBE modeling software available from Citilabs. The location of the file is shown in the image below.



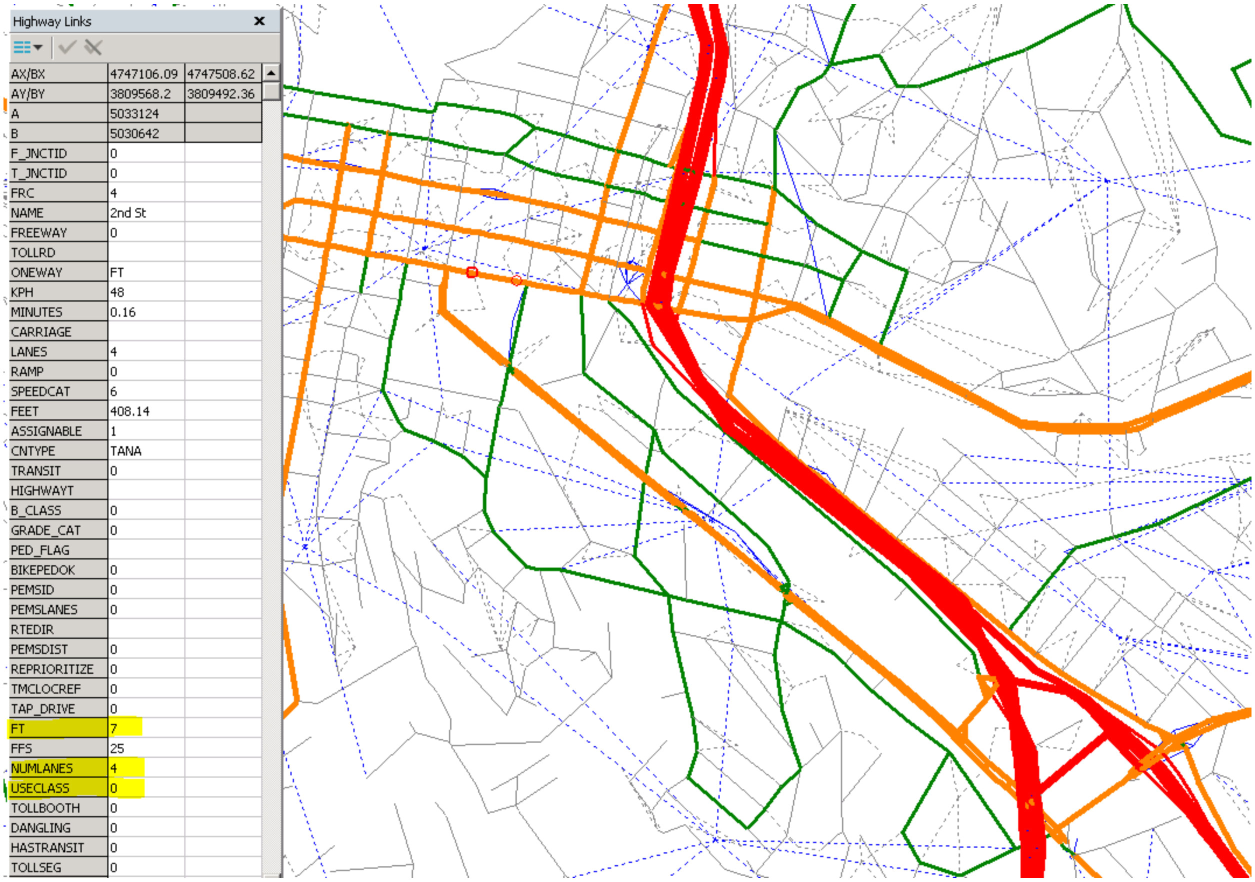
logs
mtc_final_network_base.net
mtc_final_network_base.VPR
tolls.csv
TPPL.PRJ
turnsam.pen
turnsop.pen
turnspm.pen

Please note TAMDM utilizes a year-specific highway network for each model scenario, in which all base year (in this case 2015) and future year (in this case 2040) network inputs are contained in separate network files.

6.1.1 Modifying Highway Network Attributes

Upon opening the roadway network input file in the CUBE modeling software, a network of links and nodes will be displayed as shown in the image below.

Figure 6-1: Highway Network Attributes



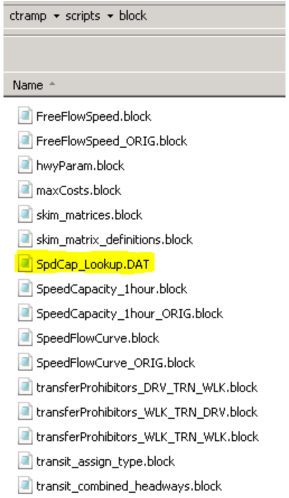
Highway Links

AX/BX	4747106.09	4747508.62
AY/BY	3809568.2	3809492.36
A	5033124	
B	5030642	
F_JNCTID	0	
T_JNCTID	0	
FRC	4	
NAME	2nd St	
FREEWAY	0	
TOLLRD		
ONEWAY	FT	
KPH	48	
MINUTES	0.16	
CARRIAGE		
LANES	4	
RAMP	0	
SPEEDCAT	6	
FEET	408.14	
ASSIGNABLE	1	
CNTYPE	TANA	
TRANSIT	0	
HIGHWAYT		
B_CLASS	0	
GRADE_CAT	0	
PED_FLAG		
BIKEPEDOK	0	
PEMSID	0	
PEMSLANES	0	
RTEDIR		
PEMSDIST	0	
REPRIORITIZE	0	
TMCLOCREF	0	
TAP_DRIVE	0	
FT	7	
FFS	25	
NUMLANES	4	
USECLASS	0	
TOLLBOOTH	0	
DANGLING	0	
HASTRANSIT	0	
TOLLSEG	0	

A list of network attributes and their input values can be viewed or modified for individual links or nodes by using the "Pointer" tool. A list and description of key link attribute fields is provided below.

- NAME – the name of the roadway the link represents
- FEET – the distance, in miles, from the beginning to the end of the link
- NUMLANES - the number of travel lanes, by direction, on the roadway link
- FT - the facility type associated with the link. TAMDM utilizes a speed-capacity lookup table based on each roadway link's Area Type and Facility Type. Are Type is calculated by the model using a Python script while Facility Type is input within the highway network.
- FFS - the free-flow travel speed used to estimate impedance during the network skimming process. However, this field is only used for freeway links to allow for posted freeway speeds which vary by location to be input. All other links receive their free-flow travel speed from the speed-capacity lookup table.
- USECLASS – field used to restrict the vehicle types that may travel on the link. A value of 0 allows all vehicles to use the link. A value of 2 only allows HOV2 and HOV3+ vehicles. A value of 3 only allows HOV3+ vehicles.

When modifying the highway network the most common modifications will be to the "NUMLANES" field to change the number of travel lanes and to the "FT" field to change the facility type associated with the link. The speed and capacity values can be viewed in the "SpdCap_Lookup.dat" file located in the "Ctramp\Scripts\block" folder. It is recommended that this file not be modified as the values were calibrated during the model validation process. The location of the file is shown in the image below.



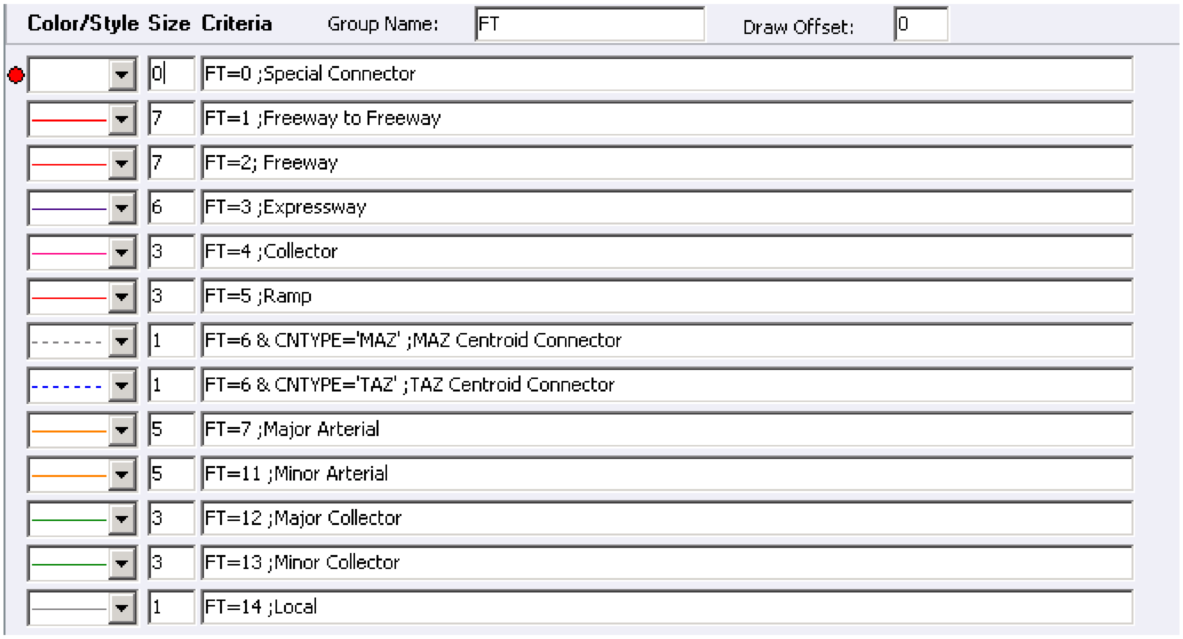
ctramp ▸ scripts ▸ block

Name

- FreeFlowSpeed.block
- FreeFlowSpeed_ORIG.block
- hwyParam.block
- maxCosts.block
- skim_matrices.block
- skim_matrix_definitions.block
- SpdCap_Lookup.DAT
- SpeedCapacity_1hour.block
- SpeedCapacity_1hour_ORIG.block
- SpeedFlowCurve.block
- SpeedFlowCurve_ORIG.block
- transferProhibitors_DRV_TRN_WLK.block
- transferProhibitors_WLK_TRN_DRV.block
- transferProhibitors_WLK_TRN_WLK.block
- transit_assign_type.block
- transit_combined_headways.block

The figure below shows the roadway link facility types. It is recommended that Facility Type 4 not be used within Marin County as additional roadway classifications consistent with General Plan designations were added to TAMDM.

Figure 6-2: Roadway Link Facility Types



Group Name: FT Draw Offset: 0

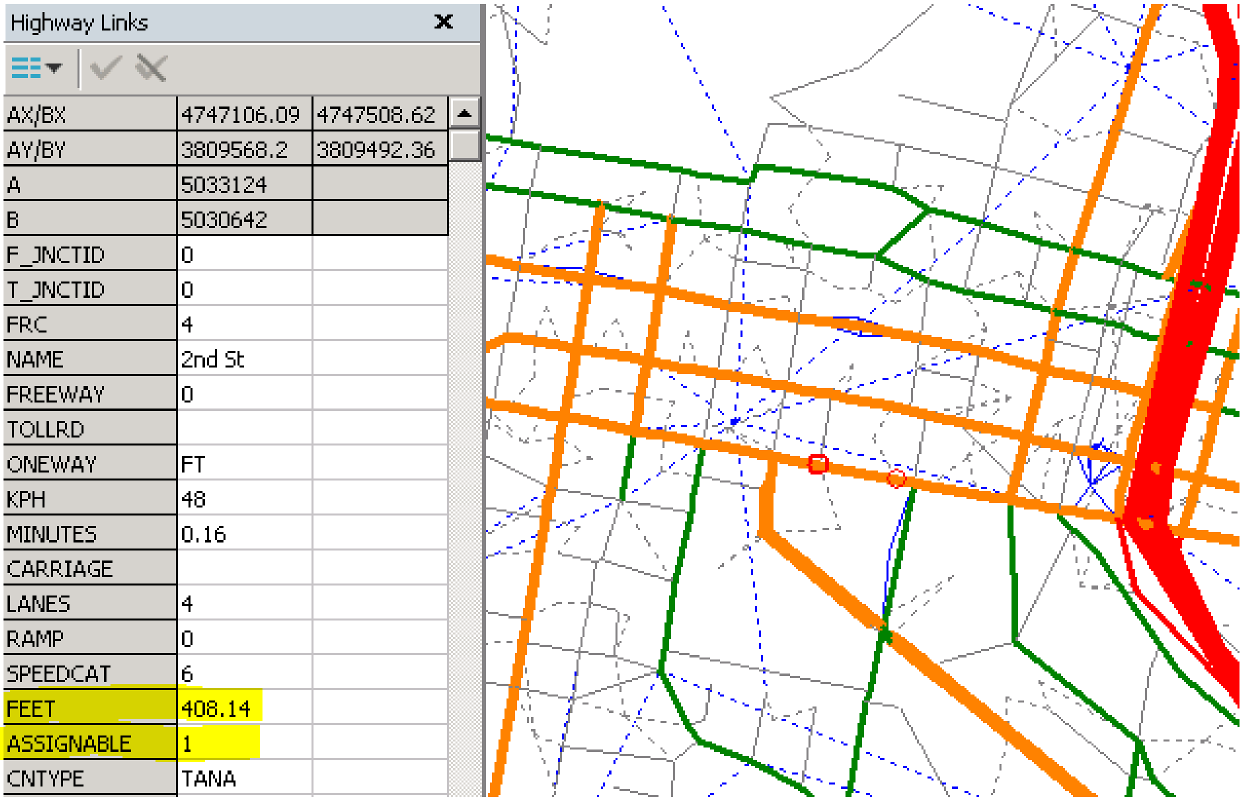
Color/Style	Size	Criteria
	0	FT=0 ;Special Connector
——	7	FT=1 ;Freeway to Freeway
——	7	FT=2; Freeway
——	6	FT=3 ;Expressway
——	3	FT=4 ;Collector
——	3	FT=5 ;Ramp
- - - - - -	1	FT=6 & CNTYPE='MAZ' ;MAZ Centroid Connector
- - - - - - -	1	FT=6 & CNTYPE='TAZ' ;TAZ Centroid Connector
——	5	FT=7 ;Major Arterial
——	5	FT=11 ;Minor Arterial
——	3	FT=12 ;Major Collector
——	3	FT=13 ;Minor Collector
——	1	FT=14 ;Local

6.1.2 Adding a New Roadway

The easiest way to add a new roadway in Cube is to use the Cube "Pointer" tool to select an existing link similar to the link you wish to create. The link can then be copied using "Control-C' and pasted using "Control-V." The new link will contain all attribute data from the previous link which can then be modified as described above.

Please note when adding a new roadway the "FEET" field needs to be manually modified to reflect the new distance and the "ASSIGNABLE" field needs to be set to "1" (roadway links with a value of "0" are not used during TAZ assignment).

Figure 6-3: Important New Roadway Link Fields



Highway Links

AX/BX	4747106.09	4747508.62
AY/BY	3809568.2	3809492.36
A	5033124	
B	5030642	
F_JNCTID	0	
T_JNCTID	0	
FRC	4	
NAME	2nd St	
FREEWAY	0	
TOLLRD		
ONEWAY	FT	
KPH	48	
MINUTES	0.16	
CARRIAGE		
LANES	4	
RAMP	0	
SPEEDCAT	6	
FEET	408.14	
ASSIGNABLE	1	
CNTYPE	TANA	

6.2 Transit Network Modifications

Transit network modifications involve editing of transit network components to reflect changes in the transit system. There are several interconnected components that are used to represent transit accessibility and level-of-service in TAMDM. The transit network is made up of three core components – transit lines, transit modes and transit fares. The lines are coded with a line-haul mode, an operator and serve a series of stop nodes. **Table 6-1** shows the transit line-haul modes. The line-haul mode code ranges in TAMDM are defined by transit technology. For example, line-haul mode codes from 10 to 79 are assigned to all local buses. **Table 6-2** shows the transit operator codes.

Table 6-1: Line-Haul Modes

Line-Haul Mode	Technology
10-79	Local Bus
80-99	Express Bus
100-109	Ferry Service
110-119	Light Rail
120-129	Heavy Rail
130-139	Commuter Rail

Source: Fehr & Peers.

Table 6-2: Operator Codes

Operator Number	Operator Name
1	Santa Rosa CityBus
2	St. Helena VINE
3	Blue and Gold
4	TriDelta Transit
5	ACE
6	samTrans
7	Stanford Marguerite Shuttle
8	AirBART
9	Benicia Transit
10	Emery Go-Round
11	Alameda/Oakland Ferry
12	Vallejo Transit

Table 6-2: Operator Codes

Operator Number	Operator Name
13	Rio Vista Delta Breeze
14	Vacaville City Coach
15	The County Connection
16	South SF/Oakland Ferry
17	Caltrain
18	Dumbarton Express
19	Sonoma County Transit
20	Golden Gate Ferry
21	Petaluma Transit
22	San Francisco MUNI
23	Amtrak Capitol Cor. & Reg. Svc
24	WHEELS
25	Alameda Harbor Bay Ferry
26	BART
27	Vallejo Baylink Ferry
28	Fairfield-Suisun Transit
29	American Canyon Transit
30	AC Transbay
31	Santa Clara VTA
32	AC Transit
33	Oakland/South SSF Ferry
34	Angel Island - Tiburon Ferry
35	Golden Gate Transit
36	Union City Transit
37	WestCAT
38	Napa VINE

Source: Fehr & Peers.

The transit lines (routes) are defined in the transit-line file "transitLines.LIN" located in the "Inputs\Trn" folder. The transit-line file is a text file containing line records that store transit line attributes. **Table 6-3** shows the line attributes specified in the transit-line file.

Table 6-3: Transit Line Attributes

Field	Description
NAME	RTD CPT_AGENCYID and SCH_ROUTEDESIGNATOR
USERA1	Transit operator name
USERA2	Line haul mode, one of - Local Bus - Express Bus - Ferry Service - Light Rail - Heavy Rail - Commuter Rail
MODE	10 – 139, The line-haul mode code defined in Table 1
OPERATOR	1 – 38. Operator Code defined in Table 2
ONEWAY	set to TRUE since each route is coded by direction
XYSPEED	set to 15 by default
HEADWAY[1]	early AM headway (3AM to 6AM)
HEADWAY[2]	AM peak headway (6AM to 10AM)
HEADWAY[3]	Midday headway (10AM to 3PM)
HEADWAY[4]	PM peak headway (3PM to 7PM)
HEADWAY[5]	Evening headway (7PM to 3AM)
N	List of stops served. Lines are coded through stops. A negative stop is not served.
NNTIME	Travel time, in minutes, between the most recent node and the node preceding the last NNTIME keyword. This is an optional parameter used to set travel times for fixed guideway services.

Source: Fehr & Peers.

The data on modes, operators and wait curves are stored in the Cube Public Transportation System (PTS) input file "transitSystem.PTS". The Cube factor files "transitFactors_SETXXX.FAC" contain the generalized cost factors and control information for the route enumeration and evaluation processes. The fare systems are defined in the "fares.far" fare file. Each transit mode is assigned a fare system in the Cube factor files. Each fare system is either FROMTO (fare zone based) or FLAT (initial + transfer in fare). For FROMTO fare systems, each stop node is assigned a FAREZONE ID in the master network. The fare is looked up from the fare matrix file "fareMatrix.txt". The additional cost to transfer to a mode from other modes is defined via the FAREFROMFS array in the "fares.far" file. For FLAT fare systems, the initial fare is defined via the IBOARDFARE in the "fares.far" file.

Before attempting to edit transit networks, it is important to understand how transit skimming and assignment works in TAMDM. TAMDM uses Transit Access Points (TAPs) as 'dummy' zone centroids for transit skimming and assignment. Transit trips are assigned between TAPs, which represent individual stops or collections of transit stops for transit access/egress. TAPs are essentially transit specific TAZs that must be coded by the user if new routes are added or routes are changed. Each stop should be connected to a TAP or it will not be available as an initial boarding or final alighting location. Stops serving premium services (express bus, ferry, light rail, heavy rail, and commuter rail) should be coded with their own TAP while other stops can share a TAP. Transit walk times and drive times are automatically calculated by Python scripts between each MAZ and TAP within a certain distance. Three sets of transit skims are built:

1. Set 1: 'Local only': This TAP set only includes local buses
2. Set 2: 'Premium only': This TAP set only includes premium (express bus, ferry, light rail, heavy rail, and commuter rail) modes
3. Set 3: 'Local plus Premium': This TAP set includes both local and premium modes and requires a transfer between local and premium.

The TAMDM software reads the three sets of TAP skims and the distance and times between each MAZ and each TAP it is accessible to. The accessibility, tour, and trip mode choice components of the software use this data to calculate the transit mode choice logsum, and if transit is chosen at the trip level, the boarding and alighting TAP pair is selected from one of the three skim sets.

Below are instructions for performing some common transit network editing tasks.

6.2.1 Adding a New Transit Line

To code a new transit line in the transit network, follow these steps:

1. Define mode and operator for the new transit line in the input PTS file "transitSystem.PTS". The mode code must be within the ranges defined in **Table 6-1**.
2. Define the fare system for the new transit line in "fares.far" file. Specify the fare system in the factor files "transitFactors_SETXXX.FAC".
3. If existing nodes do not represent stops for the new transit line, add new nodes and links to the master network representing stops on the new transit line.
4. Appropriately set the following attributes for the new nodes or an existing node to be used to represent a stop on the new transit line:
    a. MODE: Mode ID of the best transit mode served by this stop
    b. STOP: Transit stop or terminal name of the node

   c. FAREZONE: Unique sequential fare zone ID for transit skimming and assignment.

5. Appropriately set the following attributes for the new links or an existing link to be used to represent the new transit line route:
   a. CNTYPE: link connector type
   b. TRANSIT: transit link dummy
6. Add a line record to the "transitLines.LIN" file representing the attributes of the new transit line as shown in **Table 6-3**. These include mode and operator codes, headways for each period, list of node IDs representing the new transit line on the network.
7. Each stop must be assigned to a TAP. Stops serving premium modes should be assigned to their own TAP while other stops can share a TAP. TAPs should generally be within 500 feet from the stop. TAPs should be created in case that they are not close to a new stop. The stop must be connected to the TAP via a TAP connector link.

6.2.2 Modifying an Existing Transit Line

An existing transit line may need to be edited to update a line attribute such as headways, fare or realign a transit route. Follow the steps below to modify an existing transit line:

Update Line Attributes

The most common line attributes that may need an update are the headways and fare. The headways by period are specified in the "transitLines.LIN" file. Headways are specified in minutes. Headways must represent average frequency of the service for each period.

Fare systems are specified in the "fares.far" and the "fareMatrix.txt" file. Edit the "fares.far" and "fareMatrix.txt" files to update an existing fare system or add a new fare system. To assign a new or an existing fare system to a transit mode, edit the Cube factors file "transitFactors_SETXXX.FAC".

Update the mode or operator codes in the Cube PTS file "transitSystem.PTS". Ensure that the new codes are defined as per the ranges specified in **Table 6-1**.

Realigning a Transit Route

Realigning of transit routes includes addition of stops to an existing route or removal of stops. A stop can be removed from an existing route by updating the node IDs under "N" keyword for the link record in the "transitLines.LIN" file. Simply making the node ID negative makes the stop inactive (not served) for the transit route being modified.

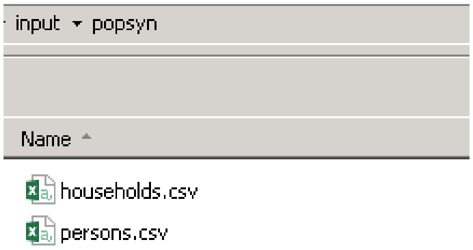
To add new stops to an existing transit route, follow these steps:

1. If existing nodes cannot be used as additional stops on the transit route, add new nodes and links to the master network representing stops on the new transit line.
2. Appropriately set the following attributes for the new nodes or an existing node to be used to represent a stop on the new transit line:
    a. MODE: Mode ID of the best transit mode served by this stop
    b. STOP: Transit stop or terminal name of the node
    c. FAREZONE: Unique sequential fare zone ID for transit skimming and assignment.
3. Appropriately set the following attributes for the new links or an existing link to be used to represent the new transit line route:
    a. CNTYPE: link connector type
    b. TRANSIT: transit link dummy
4. Update the list of node IDs under the "N" keyword for the transit line in the "transitLines.LIN" file.
5. Each stop must be assigned to a TAP. Stops serving premium modes should be assigned to their own TAP while other stops can share a TAP. TAPs should generally be within 500 feet from the stop. TAPs should be created in case that they are not close to a new stop. The stop must be connected to the TAP via a TAP connector link.

6.3 Residential Land Use Modifications

The synthetic persons and households are stored in separate .csv files that are used as inputs to TAMDM. These files are not generated by TAMDM, but are inputs to TAMDM, created by a separate software platform referred to as a "population synthesizer." TAMDM does not use a traditional SED input file where population and households are specified by TAZ/MAZ. These inputs do exist but are not used by TAMDM.

In order to update the model residential land use and SED, such as to adjust the number of households or persons in a particular MAZ/TAZ, the "households.csv" and "persons.csv" file located in the "Input\Popsyn" folder must be modified. The location of the files is shown in the image below.



households.csv
persons.csv

The synthetic persons and households files may be edited manually using a text editor such as Notepad or recreated using a population synthesizer.

6.3.1 Population Synthesizer

TAMDM synthetic persons and households files were generated using PopulationSim, which is an open platform for population synthesis. It emerged from Oregon DOT's desire to build a shared, open, platform that could be easily adapted for statewide, regional, and urban transportation planning needs. PopulationSim is implemented in the ActivitySim framework.

PopulationSim files can be obtained from the Bay Area Metro Travel Model Two GitHub at the link below.

https://github.com/BayAreaMetro/populationsim

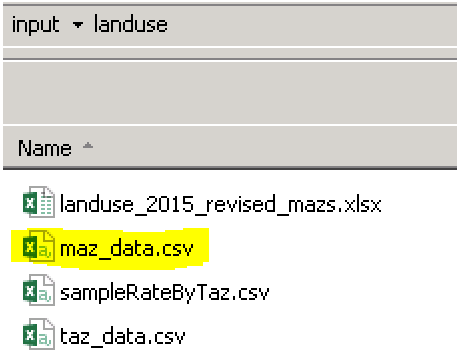
Instructions for installing and running PopulationSim are at the link below.

https://activitysim.github.io/activitysim/index.html

Please note that ActivitySim is currently under development and the process for developing synthetic persons and households files for TAMDM has not been finalized and therefore not discussed in this model user's guide. Fehr & Peers recommends using a text editor such as Notepad to modify the 2015 and 2040 synthetic population and household files provided in the "Input\Popsyn" folder until the process is finalized.

6.4 Non-Residential Land Use Modifications

In order to update the model non-residential land use and SED, such as to adjust the number of employees in a particular MAZ/TAZ, the "maz_data.csv" file located in the "Input\Landuse" folder needs to be modified. The location of the file is shown in the image below.



landuse_2015_revised_mazs.xlsx
maz_data.csv
sampleRateByTaz.csv
taz_data.csv

A list and description of key non-residential SED categories for each model MAZ is provided in **Table 6-4**. All employment fields use jobs as an input.

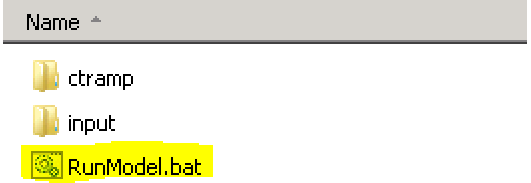
Table 6-4: TAMDM Employment Categories

Field Name	Employment Description
Ag	Agriculture
Art_rec	Arts, entertainment, and recreation
Constr	Construction
Eat	Food services
Ed_high	Higher education
Ed_k12	K-12 education
Ed_oth	Other education
Fire	Finance and insurance, and real estate and rental and leasing
Gov	Public administration
Health	Health care and social assistance
Hotel	Accommodation services
Info	Information services
Lease	Financial & leasing
Logis	Logistics services
Man_bio	Bio manufacturing
Man_lgt	Light manufacturing
Man_hvy	Heavy manufacturing
Man_tech	Technological manufacturing
Natres	Forestry, fishing and hunting, and mining
Prof	Professional, scientific, and waste management services
Ret_loc	Local serving retail
Ret-reg	Regional serving retail
Serv_bus	Business services
Serv_pers	Personal services
Serv_soc	Social services
Transp	Transportation and warehousing
Util	Utilities
Emp_total	**Total employment**

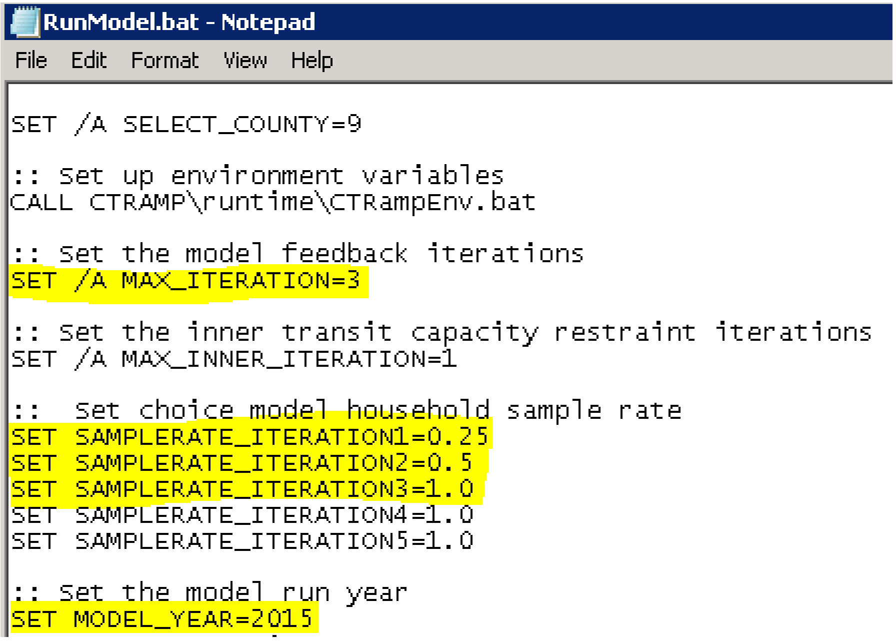
Source: Fehr & Peers.

7. Running the Model

Once TAMDM has been properly installed and input files modified for the desired scenario, the model can be run by executing RunModel.bat. The base year (2015) model run time is around 26 hours on a single server-based computer with 32 cores and 256 GB of RAM. The location of the execution file to run TAMDM is shown in the image below.



The image below illustrates parameters that can be modified when performing a model run. The user can modify the model year by adjusting "MODEL_YEAR" and the number of feedback loops performed by adjusting "MAZ_ITERATIONS" (the default is three feedback loop iterations). The user can also specify the household sample rate for each feedback loop iteration (the default is 25 percent sampling for the first feedback iteration, 50 percent sampling for the second feedback iteration, and 100 percent sampling for the third and subsequent feedback iterations). Fehr & Peers recommends 100 percent sampling be used for the final feedback iteration.



```
SET /A SELECT_COUNTY=9

:: Set up environment variables
CALL CTRAMP\runtime\CTRampEnv.bat

:: Set the model feedback iterations
SET /A MAX_ITERATION=3

:: Set the inner transit capacity restraint iterations
SET /A MAX_INNER_ITERATION=1

::  Set choice model household sample rate
SET SAMPLERATE_ITERATION1=0.25
SET SAMPLERATE_ITERATION2=0.5
SET SAMPLERATE_ITERATION3=1.0
SET SAMPLERATE_ITERATION4=1.0
SET SAMPLERATE_ITERATION5=1.0

:: Set the model run year
SET MODEL_YEAR=2015
```

8. Model Outputs

This chapter discusses key model output files and level-of-service indicators.

8.1 Level of Service Indicators

TAMDM employs a tiered spatial system to allow level-of-service indicators to be computed at a fine or coarser geography, as appropriate. Two spatial systems are defined: 1) a travel analysis zone (TAZ) system and 2) a micro-analysis zone (MAZ) system. MAZs nest within TAZs. For travel done at a "micro" scale (in the regional context, meaning less than five miles or so), the MAZ system is used; for travel done at a larger scale, the TAZ system is used.

Further, transit travel is represented as a combination of the following three movements:

1. Access. An access movement from an MAZ to a so-called transit access point (or TAP), which is a single transit stop or an abstract location representing a collection of bus stops.
2. Line haul. A line-haul movement from a boarding TAP to an alighting TAP, which can include a transfer (moving from one TAP to another TAP) between services.
3. Egress. An egress movement from the alighting TAP to the destination MAZ.

The motivation for the MAZ and TAP model design was to more precisely represent neighborhood-level travel while avoiding the steep computational price required to maintain a full set of MAZ-to-MAZ level-of-service matrices. This design concept originated at the San Diego Association of Governments (SANDAG) and is being adopted by other planning organizations.

The MAZ and TAP model design requires transit paths be built "on the fly" (i.e. outside the commercial software used by the travel model) by intelligently building and combining the MAZ-to-TAP, TAP-to-TAP, and TAP-to-MAZ trip components into logical, efficient potential paths for evaluation in probabilistic models of mode/transit route choice.

The table below presents the manner in which level-of-service indicators are extracted from the model network.

Table 8-1: Level-of-Service Indicators

Level-of-Service Indicator	Separation of Origin and Destination	Geography	Source
Automobile times, distances, and costs	Near	MAZ to MAZ	MAZ-scale single best least-cost path
Automobile times, distances, and costs	Far	TAZ to TAZ	TAZ-scale equilibrium assignment path
Transit line-haul	All	TAP to TAP	N least-cost path determination
Transit walk access and egress	All	MAZ to TAP	N least-cost path determination
Transit bicycle access and egress	All	MAZ to TAP	N least-cost path determination
Transit drive access and egress	All	TAZ to nearest TAP TAZ	TAZ-scale equilibrium assignment path
Walk	Near (assume all walk travel is near)	MAZ to MAZ	MAZ-scale single best least-cost path
Bicycle	Near	MAZ to MAZ	MAZ-scale single best least-cost path
Bicycle	Far	TAZ to TAZ	TAZ-scale single best least-cost path

Three distinct methods of extracting times from the network are employed, as follows:

1. Equilibrium assignment. For automobile travel, congestion effects impact path choice, so a traditional equilibrium assignment is performed at the TAZ-scale.

2. N best least-cost paths. For transit movements, the model builds, "on the fly", the N best least-cost paths between MAZ pairs. The N best least-cost paths are then evaluated in the mode/transit route choice model.

3. Single best least-cost path. For close-proximity automobile, bicycle, and walk travel, a single best mode-specific least-cost path is computed from the MAZ level all streets network. Because the full MAZ level network is not assigned due to computational cost, the impact of congestion on MAZ level path decisions is not taken into account. As a compromise (for gaining the spatial fidelity offered by the MAZ level network), the model implicitly assumes that automobile, pedestrian, and bicycle congestion have a negligible impact on path choice decisions and assigns each MAZ-to-MAZ movement to a single best least-cost path.

8.2 Traffic Volumes

The vehicle assignment results are provided for the time periods listed below.

- EA – Early morning period (midnight to 6 AM)
- AM – AM peak period (6 AM to 10 AM)
- MD – Mid-day period (10 AM to 3 PM)
- PM – PM peak period (3 PM to 7 PM)
- EV – Evening period (7 PM to midnight)
- 24HR – sum of the five time periods above to represent daily conditions

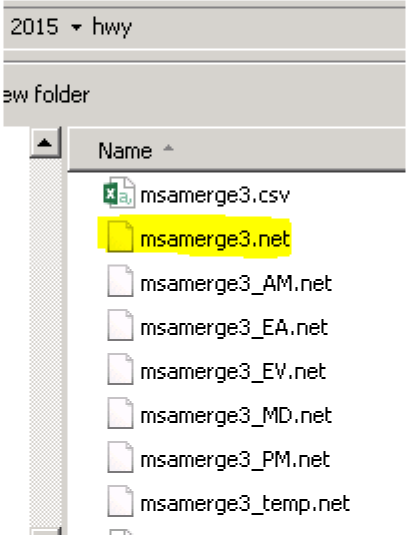
Ten vehicle classes are tracked in each time period assignment, with value-toll-eligible and not-value-toll-eligible classes, as follows:

1. drive alone, no value toll
2. shared ride 2, no value toll
3. shared ride 3+, no value toll
4. very small, small, and medium trucks, no value toll
5. large trucks, no value toll
6. drive alone, value toll eligible
7. shared ride 2, value toll eligible
8. shared ride 3+, value toll eligible
9. very small, small, and medium trucks, value toll eligible
10. large trucks, value toll eligible

Four types of trips are assigned, with separate trip tables read in by the script for each of the following travel types:

1. personal, intra-regional travel
2. personal, inter-regional travel
3. commercial travel
4. air passenger travel

Separate files with output data for each time period are located in the "Hwy" folder after the vehicle assignment run is complete. Data for all model time periods is compiled into a single file "msamerge3.net," shown in the image below.


2015 ▾ hwy

ew folder

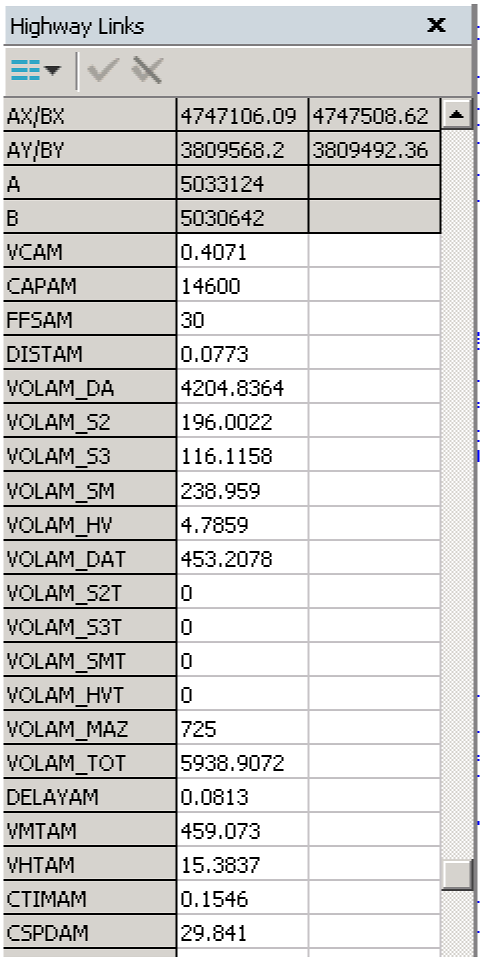
Name

msamerge3.csv
msamerge3.net
msamerge3_AM.net
msamerge3_EA.net
msamerge3_EV.net
msamerge3_MD.net
msamerge3_PM.net
msamerge3_temp.net

A list and description of key link attribute fields included in all files regardless of the time period or scenario year being modeled is provided below. "XX" represents the model time period listed above.

- VOLXX_TOT – the total directional traffic volume for all vehicular modes of travel
- CSPDXX – the congested travel speed, calculated based on the model volume-delay functions
- CTIMXX – the amount of time, in minutes, required to travel from one end of the link to the other
- CAPXX - the total capacity for the link for the specified time period
- VCXX – the volume-to-capacity ratio, calculated by dividing the total traffic volume by the capacity

The key attribute fields are shown in the image below.

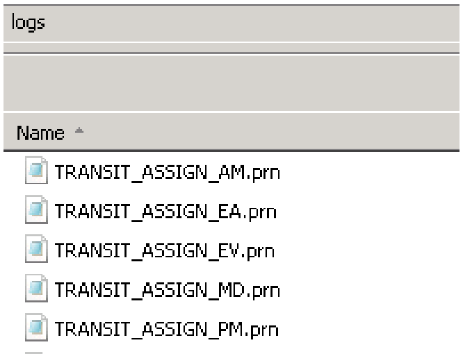


Highway Links

AX/BX	4747106.09	4747508.62
AY/BY	3809568.2	3809492.36
A	5033124	
B	5030642	
VCAM	0.4071	
CAPAM	14600	
FFSAM	30	
DISTAM	0.0773	
VOLAM_DA	4204.8364	
VOLAM_S2	196.0022	
VOLAM_S3	116.1158	
VOLAM_SM	238.959	
VOLAM_HV	4.7859	
VOLAM_DAT	453.2078	
VOLAM_S2T	0	
VOLAM_S3T	0	
VOLAM_SMT	0	
VOLAM_HVT	0	
VOLAM_MAZ	725	
VOLAM_TOT	5938.9072	
DELAYAM	0.0813	
VMTAM	459.073	
VHTAM	15.3837	
CTIMAM	0.1546	
CSPDAM	29.841	

8.3 Transit Ridership

The transit ridership results are provided for all model time periods listed above in separate "TRANSIT_ASSIGN_XX.prn" print files located in the "Logs" folder. "XX" represents the model time period listed above. The print files are shown in the image below.

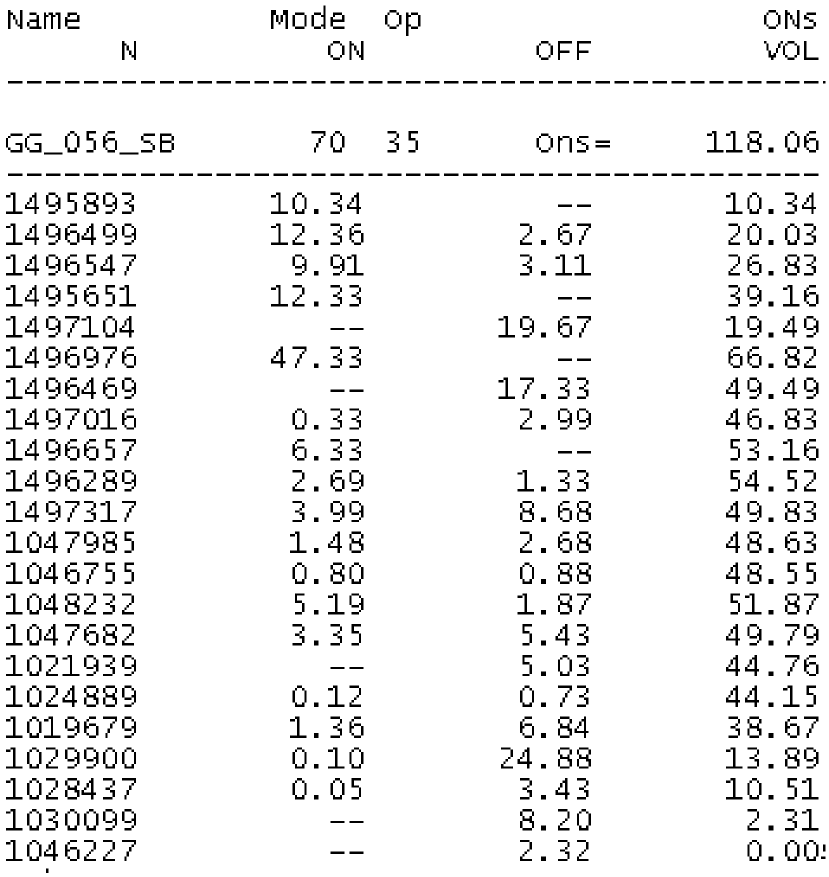


logs

Name

TRANSIT_ASSIGN_AM.prn
TRANSIT_ASSIGN_EA.prn
TRANSIT_ASSIGN_EV.prn
TRANSIT_ASSIGN_MD.prn
TRANSIT_ASSIGN_PM.prn

Each print file provides the ONs and OFFs for each transit line by transit stop, as shown in the image below.



```
Name              Mode  Op                     ONs
        N              ON            OFF           VOL
-------------------------------------------------------

GG_056_SB           70  35          Ons=       118.06
-------------------------------------------------------
1495893         10.34              --          10.34
1496499         12.36            2.67          20.03
1496547          9.91            3.11          26.83
1495651         12.33              --          39.16
1497104            --           19.67          19.49
1496976         47.33              --          66.82
1496469            --           17.33          49.49
1497016          0.33            2.99          46.83
1496657          6.33              --          53.16
1496289          2.69            1.33          54.52
1497317          3.99            8.68          49.83
1047985          1.48            2.68          48.63
1046755          0.80            0.88          48.55
1048232          5.19            1.87          51.87
1047682          3.35            5.43          49.79
1021939            --            5.03          44.76
1024889          0.12            0.73          44.15
1019679          1.36            6.84          38.67
1029900          0.10           24.88          13.89
1028437          0.05            3.43          10.51
1030099            --            8.20           2.31
1046227            --            2.32           0.00
```

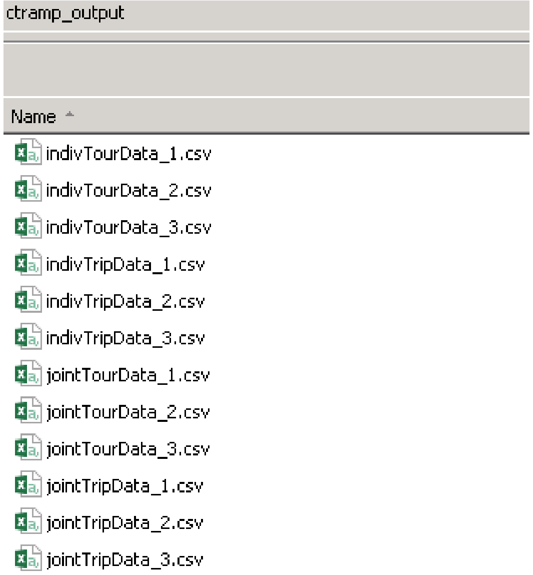
The print files can be imported into Excel using the "Text to Columns" tool to summarize transit boardings and alightings by mode, carrier, line (route), or stop.

8.4 Activity Tables

The daily activity pattern type of each household member is the first travel-related sub-model in the modeling hierarchy. This model classifies daily patterns by three types: 1) mandatory (that includes at least one out-of-home mandatory activity), 2) non-mandatory (that includes at least one out-of-home non-mandatory activity, but does not include out-of-home mandatory activities), and 3) home (that does not include any out-of-home activity and travel). The pattern type model also predicts whether any joint tours will be undertaken by two or more household members on the simulated day. Daily pattern-type choices of the household members are linked in such a way that decisions made by some members are reflected in the decisions made by the other members.

The next set of sub-models define the frequency, time-of-day, and mode for each mandatory tour. The scheduling of mandatory activities is generally considered a higher priority decision than any decision regarding non-mandatory activities for either the same person or for the other household members. The exact number of joint tours are then predicted by travel purpose and party composition (adults only, children only, or mixed) for the entire household. It is followed by choice of destination, time-of-day, and mode. This is followed by the prediction of individual maintenance (escort, shopping and other household-related errands) and discretionary (eating out, social/recreation, and other discretionary) tours. All of these tours are generated by person and their destination, time of day, and mode are then chosen.

The resulting tours and trips are written to activity tables located in the "Ctramp_output" folder (shown in the image below). Separate tables are provided for individual and joint tours. Separate tables are also provided for the individual and joint trips that comprise the tours. The number at the end of the file represents the feedback loop from which the data was predicted.



ctramp_output

Name

indivTourData_1.csv
indivTourData_2.csv
indivTourData_3.csv
indivTripData_1.csv
indivTripData_2.csv
indivTripData_3.csv
jointTourData_1.csv
jointTourData_2.csv
jointTourData_3.csv
jointTripData_1.csv
jointTripData_2.csv
jointTripData_3.csv

8.4.1 Tour Activity Tables

Below is a discussion of key fields included in the tour activity tables.

- Hh_id – the unique household identification number
- Person_id – the unique person identification number
- Person_number – the household person number
- Tour_id – the unique tour identification number
- Tour_category – defines whether the tour is mandatory, non-mandatory, or an at-work tour
- Tour_purpose – defines the purpose of the tour (e.g. work, shop, maintenance)
- Orig_mgra – origin MAZ of the tour
- Dest_mgra – destination MAZ of the tour
- Start_period – starting time period of the tour
- End_period – ending time period of the tour
- Tour_mode – mode of the tour
- Tour_distance – distance travelled along the tour
- Tour_time – time duration of the tour

8.4.2 Trip Activity Tables

Below is a discussion of key fields included in the trip activity tables.

- Hh_id – the unique household identification number
- Person_id – the unique person identification number
- Person_number – the household person number
- Tour_id – the unique tour identification number
- Stop_id - the unique trip identification number
- Tour_purpose – defines the purpose of the tour (e.g. work, shop, maintenance)
- Orig_purpose – defines the purpose for the start of the trip
- Dest_purpose – defines the purposes for the end of the trip
- Orig_mgra – origin MAZ of the trip
- Dest_mgra – destination MAZ of the trip
- Trip_mode – mode of the trip
- Tour_distance – distanced travelled along the trip
- Tour_time – time duration of the trip

8.4.3 Time Periods

The tour/trip start and end half-hourly time periods are provided in **Table 8-2**.

Table 8-2: Model Time Periods

TAMDM Period	Actual Time Period	TAMDM Period	Actual Time Period
1	3 to 5 AM	21	2:30 PM
2	5:00 AM	22	3:00 PM
3	5:30 AM	23	3:30 PM
4	6:00 AM	24	4:00 PM
5	6:30 AM	25	4:30 PM
6	7:00 AM	26	5:00 PM
7	7:30 AM	27	5:30 PM
8	8:00 AM	28	6:00 PM
9	8:30 AM	29	6:30 PM
10	9:00 AM	30	7:00 PM
11	9:30 AM	31	7:30 PM
12	10:00 AM	32	8:00 PM
13	10:30 AM	33	8:30 PM
14	11:00 AM	34	9:00 PM
15	11:30 AM	35	9:30 PM
16	12:00 PM	36	10:00 PM
17	12:30 PM	37	10:30 PM
18	1:00 PM	38	11:00 PM
19	1:30 PM	39	11:30 PM
20	2:00 PM	40	12 to 3 AM

Source: Fehr & Peers.

8.4.4 Travel Modes

There are 17 travel modes in TAMDM, including auto by occupancy and toll/non-toll choice, walk and bike non-motorized modes, and walk and drive access to transit service. Pay modes are those that involve paying a choice or "value" toll. Tolls on bridges are counted as a travel cost with the trip mode being considered "free" or non-toll. The TNC modes were added to the model in order to provide functionality for potential scenario-tests in which it is envisioned that TNC mode shares would increase significantly beyond their current levels.

The 17 trip modes in TAMDM are provided in **Table 8-3**.

Table 8-3: Model Trip Modes

Number	Trip Mode
1	Auto SOV (Non-Toll)
2	Auto SOV (Toll)
3	Auto 2 Person (Non-Toll, General Purpose Lane)
4	Auto 2 Person (Non-Toll, HOV)
5	Auto 2 Person (Toll)
6	Auto 3+ Person (Non-Toll, General Purpose Lane)
7	Auto 3+ Person (Non-Toll, HOV)
8	Auto 3+ Person (Toll)
9	Walk
10	Bike
11	Walk-Transit
12	PNR-Transit
13	KNR-Transit
14	KNR-TNC
15	Taxi
16	TNC
17	School Bus (only available for school purpose)

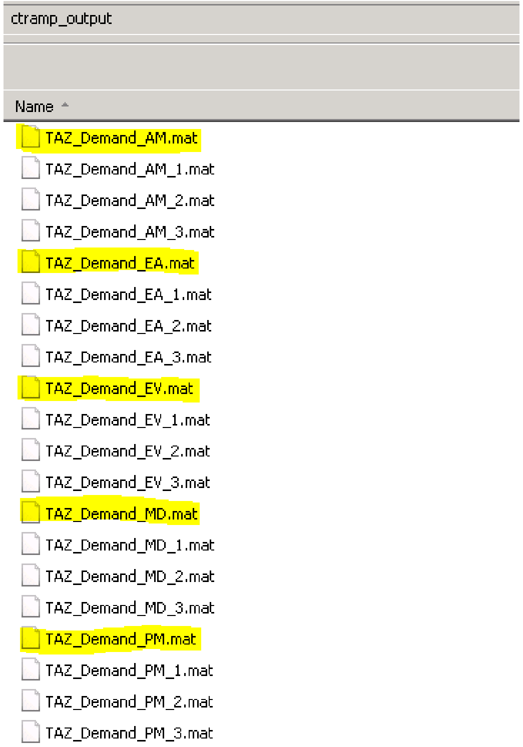
Source: Fehr & Peers.

8.5 Origin-Destination Data

The daily activity table vehicle trips are aggregated into origin-destination trips tables for the following time periods to assign vehicle trips to the highway network. Each table includes the number of vehicle trips from each TAZ to every other TAZ in the model.

- EA – Early morning period (midnight to 6 AM)
- AM – AM peak period (6 AM to 10 AM)
- MD – Mid-day period (10 AM to 3 PM)
- PM – PM peak period (3 PM to 7 PM)
- EV – Evening period (7 PM to midnight)
- 24HR – sum of the five time periods above to represent daily conditions

The origin-destination trip tables are shown in the image below. The number at the end of the files indicate the feedback loop that produced the file. The files with no numbers at the end are copied from the final feedback loop and represent the final model vehicle origin-destination data.



ctramp_output

Name

TAZ_Demand_AM.mat
TAZ_Demand_AM_1.mat
TAZ_Demand_AM_2.mat
TAZ_Demand_AM_3.mat
TAZ_Demand_EA.mat
TAZ_Demand_EA_1.mat
TAZ_Demand_EA_2.mat
TAZ_Demand_EA_3.mat
TAZ_Demand_EV.mat
TAZ_Demand_EV_1.mat
TAZ_Demand_EV_2.mat
TAZ_Demand_EV_3.mat
TAZ_Demand_MD.mat
TAZ_Demand_MD_1.mat
TAZ_Demand_MD_2.mat
TAZ_Demand_MD_3.mat
TAZ_Demand_PM.mat
TAZ_Demand_PM_1.mat
TAZ_Demand_PM_2.mat
TAZ_Demand_PM_3.mat

9. Recommended Practices

TAMDM is designed for use in evaluating the potential impacts to the transportation system of proposed land use development and infrastructure improvements. This chapter provides an overview of the recommended practices when using the model in this role.

9.1 Local Calibration

TAMDM was calibrated to meet the validation criteria developed by Caltrans and the CTC for roadway segments of highest significance within Marin County. Because the model includes a highly detailed roadway network and TAZ structure, it can be adapted for more localized studies. We recommend that this adaption include additional local calibration to ensure the model is able to replicate traffic conditions on the roadway segments or at the intersections of most interest to the particular project under study, as well as to ensure that the model responds in the correct direction and magnitude when making changes to land use and the roadway network in the vicinity of the project. Performing a local calibration should be standard practice prior to using any travel demand model, not just TAMDM.

A local calibration typically involves the following:

- Collecting multi-day field counts at key locations in the project study area, and checking the model's base year traffic volumes against those field counts.
- Checking the roadway network coding for roadways affected by the project, including determining whether the centroid connectors are loading at the proper locations in the roadway network.
- Checking that the base year land uses and SED are correct for zones affected by the project.
- Running the model and iteratively adjusting model parameters until all validation criteria established by Caltrans and the CTC are met or exceeded.

9.2 Scenario Input Data

Once the local calibration is complete, the next step is to input the data for the scenario being tested. The mechanics of inputting scenario data are very similar to the steps in the local calibration.

- Modify the roadway network if the scenario includes changes such as the creation of new roadways or intersections or the addition of travel lanes.
- Edit the land use and SED if the scenario includes changes such as the addition of households or employment.

- Run the model with the modified roadway network and land use inputs and compare the outputs to the outputs from the locally calibrated model.

9.3 Adjusting Model Outputs

Travel demand models such as TAMDM provide volume outputs that need to be adjusted in order to develop volume forecasts for the scenario being tested. Raw volume outputs from a travel demand model should rarely be applied directly in analysis, only being used after adjustments are made. The rationale for adjusting raw model volume outputs is that observed travel behavior is the result of a highly complex mixture of variables, only some of which are included in any given travel demand model, and so an adjustment is needed to capture the variables not captured by the model itself. The adjustment takes the form of changing the model outputs to correct for discrepancies between the base year field counts and the base year model volumes identified during the local calibration process, as it is assumed that the discrepancy will likely affect all scenarios in the same order of magnitude. This can be done several ways, as defined in the *National Cooperative Highway Research Program Report 255: Highway Traffic Data for Urbanized Area Project Planning and Design*, Transportation Research Board, December 1982[2]. The three most common industry-standard procedures for adjusting model traffic forecasts for both link and turning movement volumes are described below.

Difference Method

The difference between the base year field count and the base year model volume is added to the output model volume to develop the forecasted volume for the scenario being tested. For example, if the base year model volume for a roadway segment was 650 ADT while the field count was 700 ADT, then the difference method would suggest the output model volume on that roadway segment should be increased by 50 ADT to develop the forecasted volume for the scenario being tested. The difference method adjustment is summarized in the formula below.

Scenario Forecast = Output Model Volume + (Field Count – Base Year Model Volume)

[2] http://teachamerica.com/tih/PDF/nchrp255.pdf

Ratio Method

The ratio method is similar, except that it uses the ratio of the base year field count and the base year model volume to make the adjustment. For example, if the base year model volume for a roadway segment was 650 ADT while the field count was 700 ADT, then the ratio method would suggest the output model volume should be increased by 7.7% (700 / 650 = 1.077) to develop the forecasted volume for the scenario being tested. The ratio method adjustment is summarized in the formula below.

Scenario Forecast = Output Model Volume * (Field Count / Base Year Model Volume)

Blended Method

The blended method takes the average of the ratio method and the difference method scenario forecasts. The blended method adjustment is summarized in the formula below.

Scenario Forecast = (Difference Method Scenario Forecast + Ratio Method Scenario Forecast) / 2

The most appropriate adjustment method is left to the judgment of the engineer for each project. However, there are guidelines that the Transportation Research Board has published based on the difference between base year field counts and base year model volumes: use the ratio method if the difference is less than 50%, use the difference method if the difference is greater than 150%, otherwise use the blended method. No matter which method is selected, the reasonableness of the scenario forecast should always be checked based on knowledge of the area, comparisons to forecasts at adjacent locations, off-model calculations of the trip generation, distribution, and assignment of adjacent land use, or other methodologies, as deemed appropriate.

FEHR PEERS