CMAP Activity-Based Modeling System

User Guide

Prepared for:

Chicago Metropolitan Agency for Planning

Prepared by:

PB Americas, Inc.

PARSONS BRINCKERHOFF

June 6, 2013

Contents

Figures

Tables

Overview

The Chicago Metropolitan Agency for Planning (CMAP) Activity-Based Model (ABM) has been developed to ensure that the regional transportation planning process can rely on forecasting tools that will be adequate for new socioeconomic environments and emerging planning challenges. It is equally suitable for highway projects and transit projects. The CMAP model is based on the CT-RAMP (Coordinated Travel & Regional Activity-Based Modeling Platform) family of Activity-Based Models. This model system is an advanced, but operational, AB model that fits the needs and planning processes of CMAP.

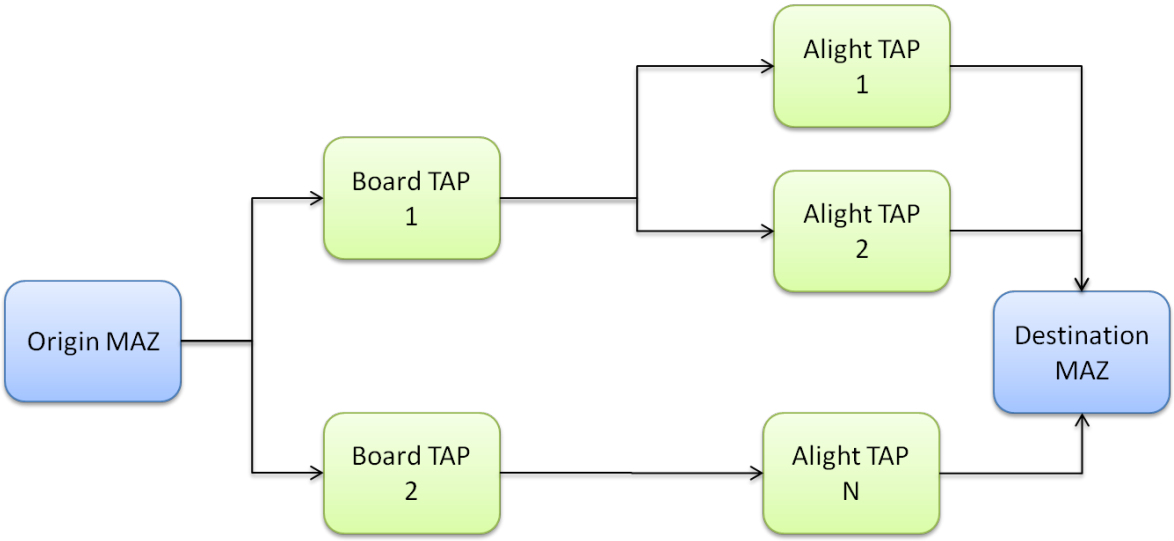
The CMAP ABM model is implemented in EMME 4 and Java and runs in a distributed fashion on four Windows 64bit machines. The overall model shell is run from DOS. The EMME components include all network assignment and skimming procedures. The Java components include the core activity-based demand model for microsimulating the travel of residents of the Chicago Metropolitan Region. This latest version of the model is similar to the previous, but with three significant revisions: 1) using one EMME databank instead of eight by time-of-day, 2) relying on MAZs instead of TAZs, and 3) using a transit virtual path builder (TVPB) and TAPs. These revisions are described below.

The first revision was to unify all the EMME databanks into one databank. This was possible with the upgrade to EMME 4, since matrices are now stored outside of the databank. Each time-of-day was assigned a range of 1000 matrices, as described later in this document.

The second revision was to migrate the model to essentially operate at the microzone (MAZ) level (the CMAP subzone level). Everywhere there was a TAZ in the previous version of the model, there is now a MAZ. The only exception to this is when calculating the level-of-service for auto modes, TAZ level skims are used.

The third major revision was the addition of the transit virtual path builder (TVPB). The TVPB relies on generalized transit stops called Transit Access Points (TAPs), which are like transit-specific TAZ centroids. Instead of building TAZ to TAZ transit skims in EMME, EMME now generates TAP to TAP transit skims and the model does all access and egress calculations in the TVPB. The TVPB calculates the full path (origin MAZ – boarding TAP – alighting TAP – destination MAZ) on-the-fly within the model. A graphical depiction of the paths for an OD pair is below.

Figure 1: Transit Virtual Path Builder Example Paths



The purpose of this User Guide is to describe the modeling system setup, how to run and setup the model, and how to manage the model inputs and the model outputs. Refer to the model specification document for details about model structure, parameters, highway and transit network procedures, etc.

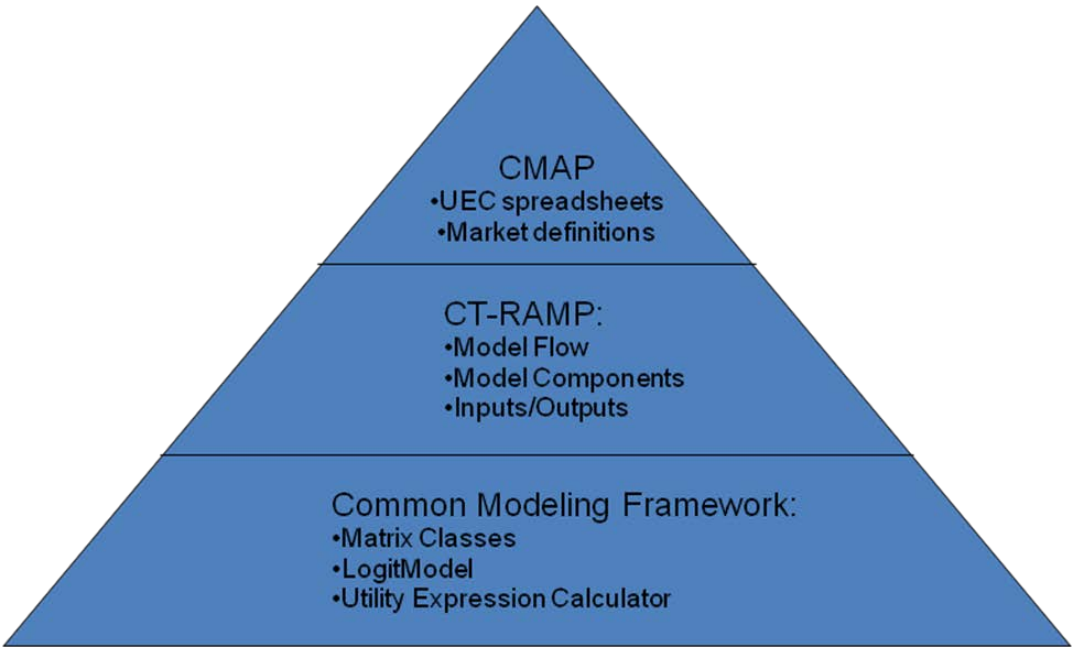
Hardware and Software Prerequisites

The CMAP ABM requires the following software and hardware configuration:

1) Four 64bit Windows (Server 2008) machines each with >= 144GB of RAM.
2) EMME 4 installed on every machine for running the highway skimming and assignment steps.
3) 64bit Java Development Kit 1.6 or later installed on every machine.
4) 64bit Python for running the MAZ to MAZ impedance calculations.

The modeling system requires 64bit Java, EMME 4, and the CT-RAMP Java package. The model requires a 64 bit OS in order to take advantage of larger (64-bit) memory addresses. As shown in Figure 2, 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 CMAP 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 CMAP 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 Java package is included in the CMAP ABM setup.

Figure 2: Disaggregate Travel Demand Model Software Components



Distributed Setup

There are two distributed processes in the CMAP ABM model. The first is the distribution of the EMME highway assignment and skimming steps. The second is the distribution of the core CT-RAMP model calculations.

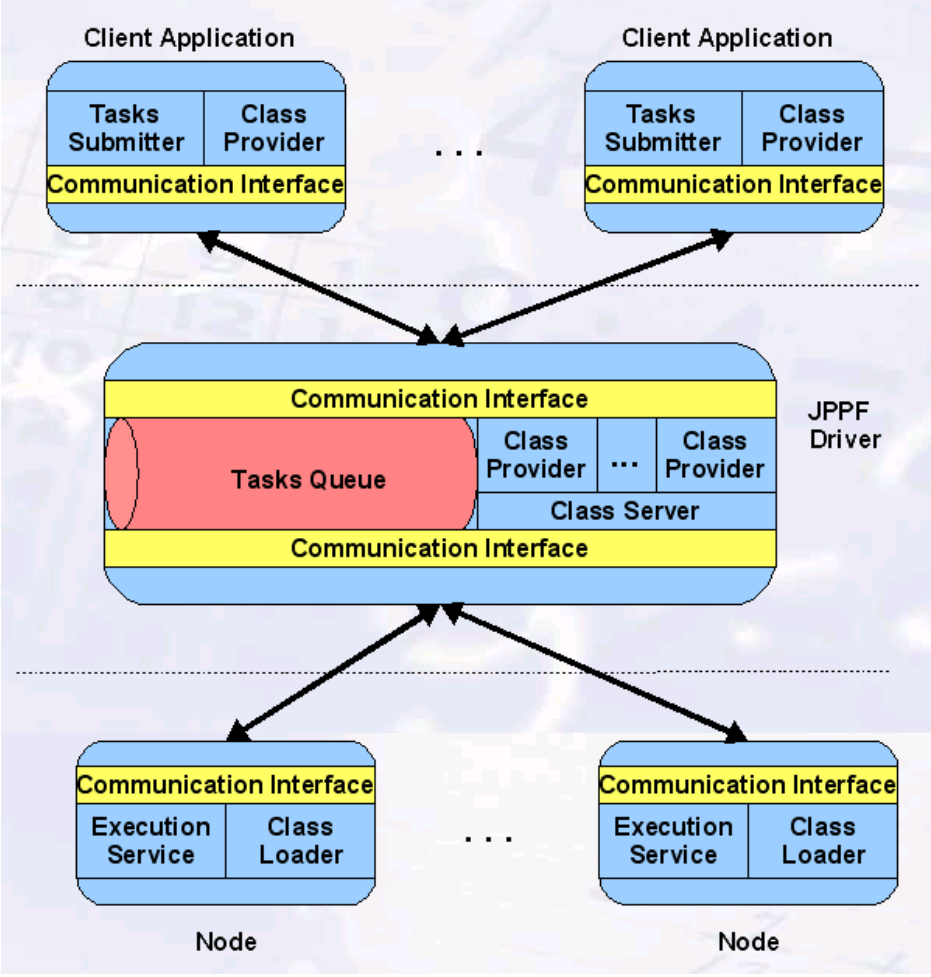
The EMME skimming and assignment can be done two ways: 1) by running all 8 EMME sessions by time-of-day period on one machine or 2) by starting 2 separate EMME sessions by time-of-day period on each of the four machines. Option two, the distributed EMME setup, is made possible by the Microsoft PsExec utility to execute processes remotely[1]. PsExec is a Windows executable that is called from the command line and calls a remote program such as a DOS bat file or EMME by referring to the machine name, program name on the remote machine, and requires a username and password to use for authentication. Details on both setups are provided later in this document.

CT-RAMP, the activity-based microsimulation component of the model, is also distributed across the four machines. The approach allows the utilization of one or more computers, each with multiple computing cores, and efficiently balances the computation among the computers. Experience with distributing large-scale CT-RAMP applications in Atlanta and San-Francisco Bay Area regions has shown that the resulted runtime is roughly inversely proportional to the computer power. In addition, the configuration of the cluster of computers is relatively flexible, allowing computers to be easily added or removed. The model results, whether solved entirely on one computer or cooperatively with many computers, are identical.

[1] http://technet.microsoft.com/en-us/sysinternals/bb897553

CT-RAMP threads the application of activity-based choice models to groups of households taking advantage of the fact that most of the behavioral models are applied for each household independently. Given that CT-RAMP is implemented in Java, it uses the Java Parallel Processing Framework (JPPF), a robust open source library, to manage the distribution of tasks. As illustrated in the Figure below, the JPPF framework consists of 3 main parts: a driver, a set of one or more nodes, and a client. The client is in this case CT-RAMP. The nodes are also additional separate processes, typically one per computer. The driver is a separate server process that is run on one of the cluster machines. The driver is a facilitator that receives tasks from the client application, sends them to node processes, receives results from nodes, and returns those back to the client.

Figure 3: JPPF Framework



Node processes receive tasks of calculations, perform those calculations, and return results. Nodes are configured through a properties file to communicate with the driver process upon their start-up. A typical configuration might be to set memory equal to 124GB and threads equal to 12 (for a 12 core machine). The majority of parallel computations in the CT-RAMP implementation occur through tasks executed in parallel on nodes.

The driver process uses logic contained in the JPPF framework to balance computational loads across Java Virtual Machines running on the nodes in the cluster. The driver receives tasks from the client application and submits them in bundles to the nodes. The driver also retrieves class files from the client application and passes those to the nodes, as needed by the nodes. Additional nodes can therefore be added by simply editing two properties files and running a Java command.

The client application, which is called by the main DOS model script and configured through a properties file, communicates with the driver as described above. The client application is responsible for creating task objects that can be run in parallel and submitting those to the driver. In the CT-RAMP implementation for CMAP, 3.9 million base year households are split into 1950 tasks of 2000 households each. The tasks are submitted to the driver and the driver assembles the tasks into bundles and submits them to nodes that have notified the driver that they are part of the cluster. As the nodes complete the tasks, the driver receives their results and submits new bundles, while balancing the submission of bundles to keep the nodes uniformly busy.

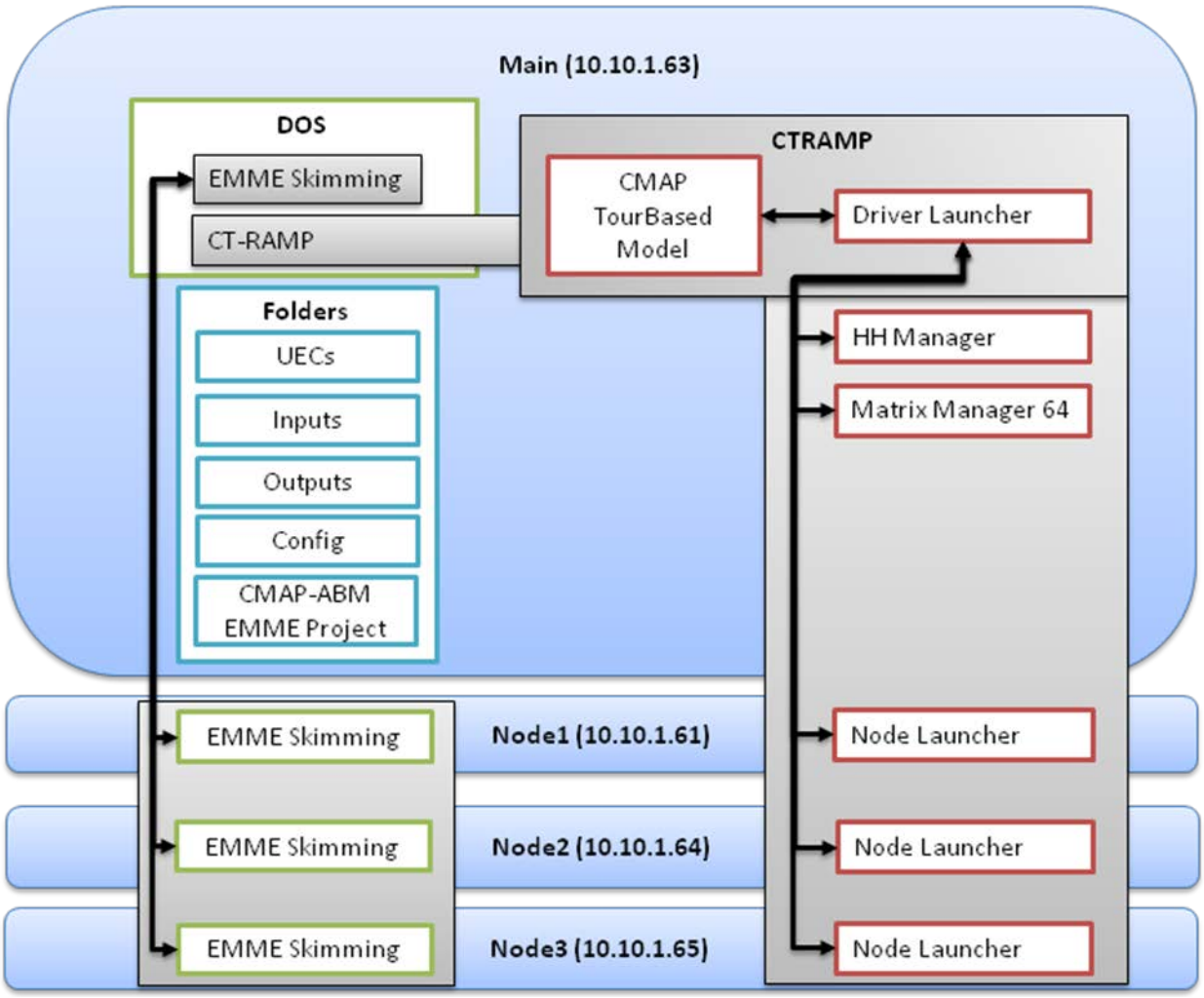
In addition to the JPPF components, CT-RAMP has a Household Manager and a Matrix Manager process. The Household Manager's purpose is to manage the household and person synthetic population in memory and to provide the JPPF nodes with all household and person related data. The Matrix Manager's purpose is to read the skims into memory and to provide the JPPF nodes with all requested skim values. These Java processes run on the main computer and have substantial memory footprints. To help reduce run time, the synthetic population is only created once in the Household Manager and then left in memory between feedback loops. It is updated with model results each iteration.

System Setup and Design

The ABM system is implemented with Java, EMME, and DOS. The Java JPPF library handles the distributed/threaded computing for CT-RAMP, whereas DOS and EMME handle the distribution of skimming and assignment. The CMAP system design consists of:

1) A main computer which:
    a. runs the main EMME skimming and assignment processes or manages the distributed EMME skimming and assignment process
    b. runs the main CT-RAMP (CMAPTourBasedModel) Java process
    c. runs the Household Manager Java process
    d. runs the Matrix Manager Java process (including the slave 32bit Java process)
    e. runs the JPPF Driver Launcher processes which is called by CMAPTourBasedModel, and which manages communicating with the nodes
2) Three additional node computers which:
    a. Listen for tasks from the JPPF driver
    b. Run the remote EMME skimming and assignment if desired

Figure 4: CMAP ABM System Design



Initial Machine Setup

1) Install EMME 4 with a 6000 zone license
2) Install 64 bit Python via the python-2.6.amd64.msi installer in the installers folder (see below). It is suggested to install it to C:\Python26\P64.
3) Configure the 64bit Python to include the following libraries:
    a. Run installer/matplotlib-1.2.0.win-amd64-py2.6.exe to install matplotlib in the 64bit Python setup.
    b. Run installer/numpy-MKL-1.6.2.win-amd64-py2.6.exe to install numpy in the 64bit Python setup.
    c. Copy the shapefile.py file from installers/pyshp-1.1.4-py2.6.egg folder to: C:\Python26\P64\Lib\site-packages
    d. Copy the installers/networks folder to: C:\Python26\P64\Lib\site-packages
4) Install Java 7 64bit using the jdk-7u13-windows-x64.exe installer in the installer folder.

Running the Model

Running the CMAP ABM requires configuring the secondary, or slave, computers first and then configuring the main computer to execute the model. The Java and slave processes must be started on the three additional computers. In addition, a remote EMME skimming folder must be setup. There is a startup script for each computer that starts all required Java processes on the machine. The remote EMME skimming machines do not require a startup script. The next step is to open and execute the main DOS run script. All the model components on each machine talk to one project directory on the network. All inputs are read from this folder, thereby simplifying model setup, inspection, and error detection. For the distributed EMME skimming, all inputs are copied from the project directory to the remote machines before running EMME.

Setting Up a Run

To set up a model run, the user has to do the following:

1) Create a project folder containing all input and output files
2) Create the initial databank
3) Create remote EMME skimming folders and share these on the network
4) Start JPPF services
5) Running the Overall Model

Create a Project Folder

5) Create a root folder on the main computer that will house scenarios. For example:
   D:\abm
6) Share the folder on the network.
7) Create a scenario specific project directory inside it. For example:
   D:\abm\cmap\model
8) Unzip the contents of cmap_abm_v2_<date>.zip to a project working folder such as c:\projects\cmap. This includes a template setup with the following folders and files:
   e. Folders
      i. CMAP-ABM – the EMME 4 project folder
      ii. Config – CT-RAMP JPPF distributed task processing configuration files
      iii. Exec – Java programs (JAR files)
      iv. Inputs – model inputs
      v. Installers – required Python libraries and installers
      vi. Outputs – model outputs
      vii. Scripts – Python scripts and EMME macros used by the model
      viii. UECs – CT-RAMP UEC files

f. Files
    i. cmap_tvpb.properties – properties file for running CT-RAMP ABM or TVPB for estimation
    ii. runBuildNetworks.py – Python script to build the EMME networks
    iii. runCTRAMP.bat – DOS bat file to run CT-RAMP ABM in distributed mode
    iv. runCTRAMP-SingleProcess.bat – DOS bat file to run CT-RAMP ABM in single process (i.e. one machine mode) for testing purposes
    v. runFinalAssignments.py – Python script to run the final assignments and skimming based on the CT-RAMP ABM demand
    vi. runInitialSkims.py – Python script to run the initial assignment and skimming procedures before CT-RAMP
    vii. runMAZSkimsInitial.py – Python script to setup creating the MAZ to MAZ/TAP impedances
    viii. runModel.bat – DOS bat file to run the MAZ skims scripts and overall model run script
    ix. runTapLines.py – Python script to produce the lines by TAP file from EMME for CT-RAMP
    x. runTVPB.bat – DOS bat file to run the TVPB for model estimation

9) Create a Y mapped network drive set to the project folder, such as \\10.10.1.63\abm

Creating the Initial Databank

The initial databank is created through the following steps. These steps result in an empty databank that is populated by running the runBuildNetworks.py script. This script is described later in this document.

1) Open EMME 4 desktop and create a new empty project
2) Create a new databank in a temporary location with the following dimensions:
    a. Accept the default setting for a size 24 license (6000 zones), except:
    b. 17 scenarios (8 for highway by time-of-day, 8 for transit by time-of-day, 1 for highway skimming)
    c. 40 transit vehicle types
    d. 9999 mf matrices, 9999 mo matrices, and 9999 md matrices
    e. 50,000,000 extra attribute size
    f. Set the user coordinates to mi and the user coordinate conversation factor 0.000189
3) After creating the databank, open it in EMME desktop and create mode “n” (not used). EMME 4 requires the databank have at least one mode defined before running macros or Modeller scripts.
4) Copy the created Database folder, which contains the EMME 4 databank and emmemat folder, from the temporary location to the template CMAP-ABM folder created by unzipping the template folder setup.

Matrix Conventions

The 9999 matrices are numbered according to the following convention:

1) Matrices <tod>XXX are for a specific time-of-day period. For example, matrix mf2101 is matrix number 101 for time-of-day period 2.
2) Matrices 0-999 are generic and not for a specific time-of-day period.
3) Matrices <tod>1-250 are for highway modeling
4) Matrices <tod>267-268, 271-276, 400+ are for transit modeling
5) Matrices are stored as 6000 by 6000 zone matrices in the emmemat folder and can be either TAZ or TAP level matrices depending on which scenario was referenced when writing the matrix.
6) All skim and demand matrix numbers passed from EMME to CT-RAMP are identified in the python scripts.

Create Remote Skimming Folders

1) On each remote machine, create a d:\abm folder
2) Share the folder on the network

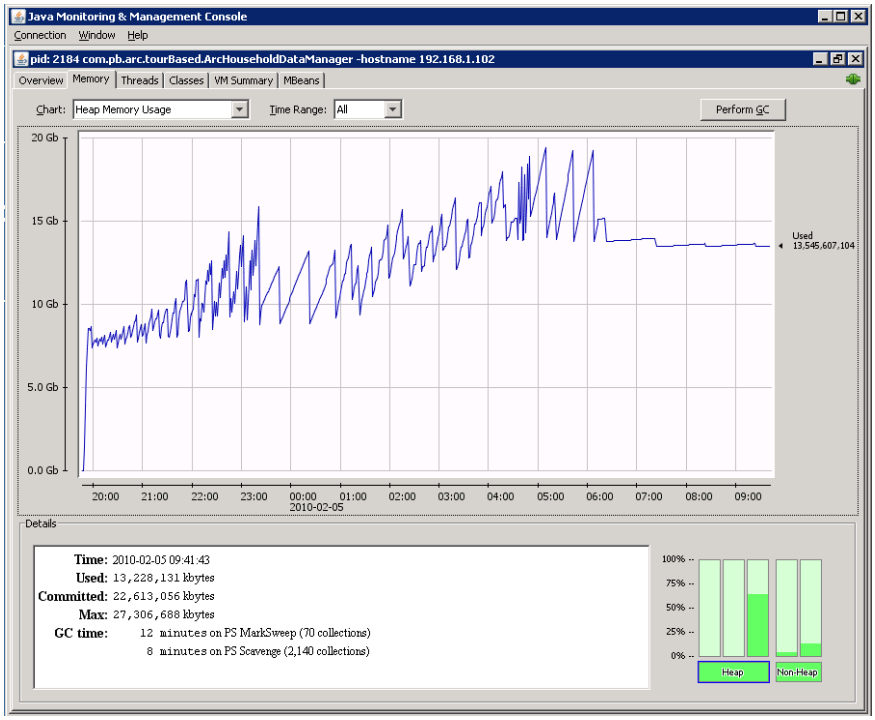
Start JPPF Services

After creating the project folder and copying over the template files, the JPPF services can be started on the main machine.

1) If necessary, edit the following properties files in the config folder:
    a. Main Computer DOS Command File - runMain.cmd
        i. set RUNTIME=Y:/cmap/model/ #project directory
        ii. set JAVA_PATH=C:/Program Files/Java/jdk1.6.0_25 #JDK 64bit path
        iii. set HOST_IP=10.10.1.63 #IP of main computer
    b. Node DOS Command File - runNode1.cmd, runNode2.cmd, runNode3.cmd
        i. set RUNTIME=Y:/cmap/model/ #project directory
        ii. set JAVA_PATH=C:/Program Files/Java/jdk1.6.0_25 #JDK 64bit path
    c. JPPF Client Driver Properties File - jppf-clientDistributed.properties
        i. driver1.jppf.server.host = 10.10.1.63 #IP of main computer
    d. JPPF Driver Properties File - jppf-driver.properties
        i. jppf.server.host = 10.10.1.63 #IP of main computer
    e. Remote JPPF Node Properties Files - jppf-node1.properties, jppf-node2.properties
        i. jppf.server.host = 10.10.1.63 #IP of main computer
        ii. processing.threads = 23 #number of cores
        iii. other.jvm.options = –Xmx64000m #memory on node
2) Change directory to the config folder and run the runMain.cmd DOS file to:
    a. Start the JPPF Driver Launcher on the main computer
    b. Start the HH Manager on the main computer
    c. Start the matrix Manager on the main computer

3) Remote desktop to the Node1 computer and run runNode1.cmd to start the Node1 process
4) Remote desktop to the Node2 computer and run runNode2.cmd to start the Node2 process
5) Remote desktop to the Node3 computer and run runNode3.cmd to start the Node3 process
6) *Optional*: Start jConsole sessions to track memory usage of the Java processes. In the config folder is the *runJConsole.cmd* file that starts a JConsole session which requires the user to select a Java process such as the HH manager. Running a JConsole session can be useful for troubleshooting memory problems.

Figure 5: Memory Usage for HH Manager in JConsole



Running the Overall Model

The runModel.bat file runs the overall model. The batch file runs the following procedures, which are described in more detail below.

1) Creates the MAZ to MAZ/TAP impedances
2) Creates the EMME networks
3) Creates the initial skims
4) Creates the tap lines file
5) Runs CT-RAMP
6) Runs the final assignments

The skimming and assignment procedures can be distributed to remote machines in order to improve overall model runtime. Like the pricing ABM, the distribution is done with PsExec. The overall model run scripts copies the databank to the remote machines, runs the skimming and assignment Python

script for two periods, and then copies back the skims. The parameters for configuring the distributed model setup are:

1) un =cmap\<username>
2) pwd =<password>
3) projDir =y:\cmap\model
4) IP1 =10.10.1.61
5) IP2 = 10.10.1.64
6) IP3 = 10.10.1.65
7) py32 ="C:\Program Files (x86)\INRO\Emme\Emme 4\Emme-4.0.4\Python26\python.exe"
8) py64 ="C:\Python26\P64\python.exe"

When running the assignment and skimming scripts, a list of comma separated 1/0 flags that identify which time periods to run is required. For example, 0,0,1,1,0,0,0,0, will run time periods 3 and 4. To run the assignment and skimming procedures only on the main machine, run the runInitialSkims.py and runFinalAssignments.py scripts with the time periods parameter set to 1,1,1,1,1,1,1,1.

In addition, before running the overall model, first run runMain.cmd in the Config folder to start the CT-RAMP HH and matrix data servers on the main machine. Then run runNodeX.cmd on each worker machine that will be used for the model run. The sampleRate parameter specifies the CT-RAMP household sample percent.

Running the MAZ to MAZ/TAP Impedances Script

The MAZ to MAZ/TAP impedances are created by running the runMAZSkimsInitial.py, SPWrapper.py, and cmapPostProcess.py script. These scripts require 64bit Python and are run in the project folder via the command line. This script creates the MAZ to MAZ and MAZ to TAP impedances files, which is written to outputs/SP_MAZ_to_MAZ_and_TAP.txt. To run it for future years, set the isBaseYear parameter to False in the runMAZSkimsInitial.py script. The max shortest path distance to generate MAZ to MAZ and MAZ to TAP impedances is set via the SPThreshold parameter.

Building the EMME Networks

The runBuildNetworks.py script builds the initial EMME database of highway and transit networks. This script requires the 32bit Python version included with EMME (since it uses the EMME Modeller libraries) and is run in the project folder via the command line. The only argument to the script is a list of comma separated 1/0 flags that identify which time periods to build. The command line call is:

```
"C:\Program Files (x86)\INRO\Emme\Emme 4\Emme-4.0.3\Python26\python.exe"
runBuildNetworks.py 1,1,1,1,1,1,1,1
```

The script is split into the following steps:

1) Settings

a. highwayScenarios – EMME highway assignment scenarios
b. transitScenarios – EMME transit assignment scenarios
c. tods – time-of-day period that corresponds to each highway and transit scenario
d. runTransitOnly – True or False to run only transit network build and skimming
e. transitImport – transit import scenario
f. previousBank – location of the EMME bank with matrices 1-99 to be imported
g. trnAssignIters – Number of transit capacity constrained assignment iterations
h. matNumConvDemand – EMME matrix number (without tod) for TAP to TAP conventional transit demand
i. matNumPremDemand – EMME matrix number (without tod) for TAP to TAP premium transit demand

2) Start EMME – starts an EMME desktop session and associates an EMME Modeller session with it

3) Time-of-day loop
    a. Highway:
        i. Creates the scenario if needed
        ii. Runs the initial highway network setup macros
        iii. Batches in matrices 1-99 from an existing EMME 4 bank
        iv. Runs the time-of-day tables macro to create the highway demand matrices for non-ABM user classes.
        v. Runs the macros to define a series of extra attributes and read in the toll extra attribute data.
    b. Transit:
        i. Creates the scenario if needed
        ii. Runs the transit network setup macro to create the GTFS network. This network contains TAZ nodes and TAZ connector links, which are replaced by TAP nodes and TAP connector links in the next step.
        iii. Deletes TAZ centroids and creates TAP centroids and connectors based on the input TAPs file.
        iv. Reads in the temporary conventional and premium TAP level demand matrices for all time periods.

Create the Initial Skims

The runInitialSkims.py script creates the initial set of highway and transit skims for use in CT-RAMP. Before running this script, the runModel.bat program copies the databank and macros to the remote machines in order to run remote skimming. Upon completion of the skimming, the skim matrix zmx files are copied back to the main machine. This script has the same setup as the build networks script, with the following differences.

1) Settings

a. bypassConventionalTransit – True or False to bypass conventional transit assignments and to run only premium transit assignments and skimming

2) Time-of-day loop
    a. Highway:
        i. Runs the highway skimming macro to create TAZ to TAZ highway skims.
        ii. Converts the required TAZ level skims to ZMX format for use by CT-RAMP.
    b. Transit:
        i. Runs the transit skimming macro to create TAP to TAP transit skims.
        ii. Converts the required TAP level skims to ZMX format for use by CT-RAMP.

Create the TAP Lines File

The runTapLines.py script creates the list of lines served by TAP for use in CT-RAMP. This file (tapLines.csv) is used by CT-RAMP to trim the near tap set by origin/destination by only including the nearest tap when more than one tap serves the same line. This script has the same setup as the build networks script except that it is run for all time periods at once and therefore does not have the time period 1/0 flags as an argument.

Running CT-RAMP

The runCTRAMP.bat file runs CT-RAMP. As described above, to run CT-RAMP in distributed mode, which is the default, do the following:

1) Before running the overall model, first run runMain.cmd in the Config folder to start the HH and matrix data servers on the main machine. Then run runNodeX.cmd on each worker machine that will be used for the model run.
2) The runModel.bat script then calls runCTRAMP.bat to run CT-RAMP across the distributed cluster of worker machines.

In single process testing mode:

1) Before running the overall model, first run runMain-SingleProcess.cmd in the Config folder to start the HH and matrix data servers on the main machine.
2) The runModel.bat script then calls runCTRAMP-SingleProcess.bat to run CT-RAMP on the main machine as well (using local worker threads within the main process).

Running the Final Assignments

The runFinalAssignments.py script runs the final assignments based on the CT-RAMP demand. Before running this script, the runModel.bat program copies the CT-RAMP generated demand matrices to the remote machines in order to run remote assignments. The resulting assignments are not copied back to the main machine, so the user needs to refer to the d:\abm\remote folders on the remote machines for any assignment summaries. This script has the same setup as the build networks script, with the following differences.

1) Time-of-day loop
    a. Loads the TAZ level demand matrices from CT-RAMP into the EMME databank
    b. Runs highway assignment and skimming
    c. Loads the TAP level demand matrices from CT-RAMP into the EMME databank
    d. Runs transit assignment and skimming

CT-RAMP Model Run Properties

The CT-RAMP properties file defines the properties specified to run the main CMAPTourBasedModel CT-RAMP Java process. Most of the CT-RAMP settings should not be changed when running a new scenario. The settings that are likely to change with a new scenario are:

1) Project directory
2) Population Synthesizer Household File
3) Population Synthesizer Person File
4) Population Synthesizer Sample Percents
5) Population Synthesizer Sample Seed

In addition, the following settings may need to be changed if a new model setup is configured:

1) Run Model – Matrix Server Address
2) Run Model – Household Server Address

What follows is an exhaustive list of all the CT-RAMP settings in the CMAP ABM properties file.

General Inputs

1) Project Directory – The project directory. This should be set to the location of the project directory (i.e. the directory that contains the runModel.bat file).
2) Debug – Trace Household ID List – HH IDs separated by a comma to write debug results for. Ensure these HHs are in the sampled HHs if a sample rate < 1.0 is used.
3) Taz Data – Location of the MAZ Subzone data input file
4) Zone Accessibilities – Location of the MAZ-based accessibility measures input file (note this would be calculated on-the-fly in a full ABM implementation).

Result Files and Database

5) Results – Write Data to Files – Write CT-RAMP output files to CSV files. This is required for writing the trip matrices to the EMME databanks. See the Model Outputs section for details.

6) Results – Write Data to Database – Writes CT-RAMP output tables to a SQLite database. This is default to FALSE and may not complete successfully due to memory limitations.

7) Results – <Model Name> - Specifies the location of output CSV files. See the Model Outputs section for details.

Result Matrices

1) Results – Write Trip Matrices– T/F to write the trip matrices to databanks

2) Results – Trips Matrices – LowVOTUpperBound – Max person value-of-time in order to classify their trips as low income in the output trip matrices.

3) Results – Trip Matrices – Num Zones – Number of zones in the output trip matrices.

4) Results – Trip Matrices – Folder – Location to wrote output matrices

5) Results – TimePeriodUpperBounds – Time period code upper bound for time periods 1 – 8

6) Results – Trips Matrices – Numbers – Matrix numbers to write. The order and number of these matrices cannot be changed without changes to the code.

7) Results – Trips Matrices – Names – Matrix names for each matrix number above.

8) Results – Trip Matrices – Num Transit Zones – Number of TAPs in the output trip matrices.

9) Results – Trips Matrices – TransitNumbers – Matrix numbers to write transit demand matrices.

10) Results – Trips Matrices – TransitNames – Matrix names for each demand matrix number above.

Run Model IP and Port Settings

11) Run Matrix Server Separate – This runs the CT-RAMP matrix server in is a separate Java process.

12) Run Model – Matrix Server Address – IP address of the CT-RAMP matrix server. This needs to be set to the actual IP address, not localhost or 127.0.0.1 as this confuses JPPF.

13) Run Model – Matrix Server Port – Port of the CT-RAMP matrix server.

14) Run Model – Household Server Address – IP address of the CT-RAMP household data server. This needs to be set to the actual IP address, not localhost or 127.0.0.1 as this confuses JPPF.

15) Run Model – Household Server Port – Port of the CT-RAMP household server.

Models to Run

The Models to Run section (RunModel.*) defines which CT-RAMP models to run. By default, all models are run except for the Population Synthesizer. These models correspond to CT-RAMP UECs stored in the uecs folder. Refer to the CMAP ABM Model Specification document for details on the individual models (more than 20 choice submodels with multiple segments by travel purpose and/or person type within most of them).

UEC Files

The UEC Files section (UecFile.*) defines the location of the UEC input files in the uecs folder. There are a number of other UEC files specified in the properties file as well. These are:

16) Stop.depart.arrive.proportions – lookup table for stop depart time period by tour purpose, is inbound stop, time period, and trip index on tour

17) TourDepartureAndDuration.AlternativeList.InputFile – alternative file

18) IndividualNonMandatoryTourFrequency.AlternativesList.InputFile – alternatives file

19) IndividualNonMandatoryTour.FrequencyExtension.ProbabilityFile – lookup table for individual non-mandatory tour frequency by person type, is mandatory tour, is joint tour, and non-mandatory tour type

20) UsualWorkAndSchoolLocationChoice.SizeCoefficients.InputFile – usual work and school location choice size term coefficients input file

21) UsualWorkAndSchoolLocationChoice.AlternativesList.InputFile – alternatives file

22) StopDestinationChoice.SizeCoefficients.InputFile – stop destination choice size term coefficients input file

23) StopPurposeLookup.Proportions – lookup table for stop purpose by tour purpose, is inbound stop, departure range start and end hour, and person type

24) CBDParkingAlternatives.file – CBD parking zone alternatives file

Distributed Value of Time Settings

25) Household Manager – Min Value of Time, MaxValueOfTime, MeanValueOfTime.Values, MeanValueOfTime.Income.Limits, Mean.ValueOfTime.Multiplier.Mu, ValueOfTime.Lognormal.Sigma, HH.ValueOfTime.Multiplier.Under18 – Settings to calculate the person value of time. Refer to the CMAP ABM Model Specification document for details on the calculation.

Distributed Model Run Settings

26) Distributed Task Packet Size – The size of packages to distribute and process in CT-RAMP.

27) Initialization Packet Size – The initialization packet size for distributing tasks in CT-RAMP.

28) Number Initialization Packets – The number of initialization packets in CT-RAMP

Population Synthesizer Inputs

29) Population Synthesizer Household File – Location of the HH population synthesizer file

30) Population Synthesizer Person File – Location of the person population synthesizer file

31) Population Synthesizer Work Occupation Code Files – The crosswalk from PUMS Industry Census codes to CMAP worker occupancy classes.

Usual Work and School Location Choice Model Settings

32) Usual Work and School Location Choice – Run Flag - Work – Run the usual work location model

33) Usual Work and School Location Choice – Run Flag - University – Run the usual university location model

34) Usual Work and School Location Choice – Run Flag - School – Run the usual school location model

35) Usual Work and School Location Choice – Shadow Pricing – Max Iterations – the maximum number of show pricing iterations for the run.

36) Usual Work and School Location Choice Shadow Pricing Input File – The input file to use for re-start with shadow pricing if desired. This file would have been calculated from previous model run and has the same format as the output shadow pricing file. The default is to comment out this property.

Sample of Alternatives Sample Size Settings

37) Usual Work and School Location Choice Sample of Alts – Sample Size – The number of alternatives.

38) Joint Tour Location Choice Sample of Alts – Sample Size – The number of alternatives.

39) Individual Non Mandatory Tour Location Choice Sample of Alts – Sample Size – The number of alternatives.

40) At Work Subtour Location Choice Sample of Alts – Sample Size – The number of alternatives.

41) Stop Location Choice Sample of Alts – Sample Size – The number of alternatives.

Departure Time Model Settings

42) The departTime.* properties in the properties file specify the tour departure and duration model UEC data pages for each activity.

CT-RAMP Output Files

43) CT-RAMP Output – Write to Disk – T/F to write the output files described below. See the Model Outputs section for details on each file (Households, Persons, and Tours).

Transit Virtual Path Builder Settings

The key settings related to running the TVPB as part of the ABM are:

44) tvpb.uec = uecs/TVPB.xls #path generalized cost settings file
45) tvpb.maxpaths.walk = 1 #max number of walk paths to output
46) tvpb.maxpaths.knr = 1 #max number of KNR paths to output
47) tvpb.maxpaths.pnr = 1 #max number of PNR paths to output
48) tvpb.maxwalkdist = 10560 #max MAZ to TAP walk distance in feet. Used to trim the walk alternatives since the maz2tapfile input file is for KNR as well.
49) tvpb.maxtotalwalkdist =15840 #max total walk (origin walk + destination walk)
50) tvpb.maxpnrdist = 105600 #max MAZ to TAP PNR distance in feet. Used to trim the PNR alternatives set.
51) tvpb.tripfile.debugRecord = -1 #debug record to output trace results for; -1 for no tracing.
52) tvpb.trace = False #NA; Traces TVPB results when running as part of the ABM
53) HouseholdManager.WalkPreferences.FileName = inputs/walkPreferences.csv #propensity to walk factors by age

EMME Macro Settings

The highway and transit skimming and assignment procedures are done with EMME. The settings are specified in the macros. The macros are described below. Note that the Python run scripts call these macros with the appropriate command line arguments.

1) hwySetup.mac – highway network setup macro. The command line argument are:
    a. Time period code, such as 1 for p1
2) HwayMatIn.mac – simple macro to read in matrices 1-99 from the previous bank.
    a. Previous databank to load matrices from
3) TOD_tables.mac – creates time of day period specific matrices for highway assignment, including the initial version of the auto trip matrices that will be further created by CT-RAMP (to generate starting sets of skims for each period). This macro is run once for each time-of-day period before the first global iteration. The command line argument are:
    a. Time period code, such as 1 for p1

4) extraclass.mac – creates extra attributes for CT_RAMP_skim macro. This macro is only run once for each time-of-day period before the first global iteration when running the initial skimming. The command line argument are:
    a. Time period code, such as 1 for p1
5) addtoll.mac – simple macro to read in the toll extra attribute data.
    a. Time period code, such as 1 for p1
6) Build_TOD_Transit_CT_RAMP.mac – creates time-of-day specific transit network for assignment and skimming. The command line arguments are:
    a. Three-digit transit scenario such as 100.
    b. TOD period (1-8) and highway scenario (1-8) from which auto times will be imported.
    c. Path to Input file folder (to be set to ../inputs).
7) TranMatIn.mac – simple macro to read in the temporary Premium and Conventional transit demand matrices by tod.
    a. Transit scenario
8) CT_RAMP_skim3.mac – toll road choice (for non CT-RAMP user classes) and skimming macro. Implements binary choice between toll and non-toll users and all necessary highway skimming procedures. Can be applied independently to generate starting skims and/or explore pricing scenarios. When applied to support CT-RAMP, auto trip tables mf101-mf106 are generated by CT-RAMP, Truck trip tables mf107-mf110, external autos mf111-mf116 and passenger autos to airports mf121-mf126 are split by this macro (between toll and non-toll users). This macro is run 4 times (internal iterations) for each global iteration (to equilibrate non-core traffic components). The command line arguments are:
    a. 0 = initialize split matrices mf131-mf174 (only for first global iteration), 1 = start with the previous set
    b. MSA factor for averaging matrices (0.0 = no update, 1.0 = full update)
    c. 0 = skip final assignment, 1 = implement final assignment (for last global iteration)
    d. base network scenario for assignment (p1-8)
    e. number of assignment iterations (normally set to 10, 20, 30, 40 by internal iterations)
    f. 0 = include auto split (for initial skims), 1 = exclude (when applied with CT-RAMP)
9) Transit_assignment_skimming_CT_RAMP3.mac – macro for running capacity constrained transit assignment and skimming. Needs to be run after highway skimming. The command line arguments are:
    a. Three digit transit scenario such as 100.
    b. TOD period (1-8) and highway scenario (1-8) from which auto times will be imported.
    c. Number of iterations for transit assignment equilibration, default is 3
    d. Matrix number for conventional transit demand for initial assignment
    e. Matrix number for premium transit demand for initial assignment
    f. 1=bypass conventional transit assignment and skims; 0=skim both conv and prem
    g. 1=Bypass create matrix segmentation by class; 0=create demand matrices by class

Scripts (and Macros)

The following scripts and macros are included in the Scripts folder. See the *Running the Model* section for more information on the script and macro parameters.

1) Python
    a. __init__.py – empty Python script to trick Python into thinking the scripts folder is a Python model that can be imported
    b. EMXtoZMX.py – module to convert EMME 4 EMX matrices to PB zipped matrix (ZMX) format and vice versa for use by CT-RAMP Java code.
    c. cmapTransportationNetwork.py – script to convert the NAVTEQ shapefile network to a text file format for later use. Used by the MAZ to MAZ/TAP impedances procedures.
    d. cmapTapConnector.py – script to add TAP nodes and connectors to the NAVTEQ network. Used by the MAZ to MAZ/TAP impedances procedures.
    e. cmapCentroidsConnectors.py – script to add MAZ nodes and connectors to the NAVTEQ network. Used by the MAZ to MAZ/TAP impedances procedures.
    f. cmapInputFileGen.py - script to convert the all streets network input files to the format required by the shortest path procedures. Used by the MAZ to MAZ/TAP impedances procedures.
    g. cmapShortestPath_NX<_future>.py - script to calculate shortest paths between MAZs and MAZ/TAPs. Generates an output file for each thread. Used by the MAZ to MAZ/TAP impedances procedures. The <_future> version is used for future year model runs.
    h. cmapParallelPP.py – script to merge the threaded outputs into one file.
    i. cmapRStoSubZone<_future>.py - script to create the final MAZ to MAZ/TAP impedance output file by removing TAP to TAP pairs. Used by the MAZ to MAZ/TAP impedances procedures. The <_future> version is used for future year model runs.
    j. cmapPostProcess.py – script that calls cmapParallelPP.py and cmapRStoSubZone<_future>.py scripts.
    k. SPwrapper.py – Script that distributes the shortest path computations to the different threads.
    l. Parameters.py – written on-the-fly by the runMAZSkimsInitial.py script and is read-only.

2) EMME Macro
    a. hwySetup.mac – highway network setup macro.
    b. addtoll.mac – simple macro to read in the toll extra attribute data.
    c. extraclass.mac – updated macro for defining extra attributes.
    d. HwayMatIn.mac – simple macro to read in matrices 1-99 from the previous bank.
    e. TOD_tables3.mac – updated macro for building time-of-day trip tables.
    f. CT_RAMP_skim3.mac – Highway skimming and assignment macro.
    g. TranMatIn.mac – simple macro to read in the temporary Premium and Conventional transit demand matrices by tod.

h. Build_TOD_Transit_CT_RAMP3.mac – build GTFS-based transit network macro.
i. Transit_assignment_skimming_CT_RAMP3.mac – Transit skimming and assignment macro.

Exec Folder

The Exec folder contains the following Java code files used by the model:

1) cmap.jar – the CMAP specific Java code that is the entry point for running the model. It requires the jar files specified below
2) ctrampIf.jar – the CT-RAMP Java code
3) synpop.jar – population synthesizer Java code
4) common-base.jar – PB's common modeling framework Java code
5) jxl.jar – J Excel Java code for reading Excel files
6) log4j-1.2.9.jar – Log4J Java code for logging
7) ssj.jar – University of Montreal stochastic simulation Java code
8) sawdust-util-1.0.jar – PB Java 7 utilities for threading calculations code.

The Exec folder also contains the Microsoft PsExec.exe program for remote execution, the Microsoft PsKill.exe program for remote process termination, and the cmap_abm.sql script to load the output tables into a MySQL database, as described in the Model Outputs section.

Model Inputs

Inputs to the CMAP ABM are stored in the following folders: uecs, inputs, and CMAP-ABM.

UECs Folder

The uecs folder stores all CT-RAMP Utility Expression Calculator (UEC) files and choice model alternatives files. The files are summarized in the table below. Many of the UEC files, such as the Destination Choice UEC, contain multiple models. In addition, the complete definition of one destination choice model is actually spread across multiple files – DestinationChoice.xls, DestinationChoiceAlternatives.csv, DestinationChoiceAlternativeSample.xls, and DestinationChoiceSizeCoefficients.csv. Refer to the CMAP ABM Model Specification for a description of each CT-RAMP model.

Table 1: CT-RAMP Model Input Files

File Name	Description
AtWorkSubtourFrequency.xls	At work subtour frequency model UEC
AutoOwnership.xls	Auto ownership model UEC
AutoDependecy.xls	Auxiliary auto dependency model called by auto ownership model
cbd_parking_zones.csv	CBD parking zone alternatives
CoordinatedDailyActivityPattern.xls	Coordinated daily activity pattern model UEC
DepartureTimeAndDurationAlternatives.csv	Tour departure time and duration models alternatives
DestinationChoice.xls	Destination choice model UECs (mandatory, maintenance, discretionary, at work subtours)
DestinationChoiceAlternatives.csv	Destination choice models alternatives (including 3 market segments by walk access to transit within each TAZ)
DestinationChoiceAlternativeSample.xls	Destination choice model sample of alternatives UECs
DestinationChoiceSizeCoefficients.csv	Destination choice models size term coefficients
FreeParkingEligibility.xls	Household free parking eligibility model UEC
IndividualMandatoryTourFrequency.xls	Individual mandatory tour frequency model UEC
IndividualNonMandatoryTourFrequency.xls	Individual non-mandatory tour frequency model UEC
IndividualNonMandatoryTourFrequency Alternatives.csv	Individual non-mandatory tour frequency alternatives by activity part one
IndividualNonMandatoryTourFrequencyExtension Probabilities.csv	Individual non-mandatory tour frequency alternatives part two
JointTours.xls	Joint tour frequency, party composition, person participation model UECs
ModeChoice.xls	Tour mode choice model UECs (mandatory, non-mandatory, at work subtour)
Parklocation.xls	Parking location choice model UEC (mandatory, non-mandatory)
stopDepartArriveProportions.csv	Stop departure time-of-day proportions
StopDestinationChoice.xls	Stop destination choice model UECs (mandatory, maintenance, discretionary, at work subtour)
StopDestinationChoiceAlternativeSample.xls	Stop destination choice model UECs sample of alternatives
StopDestinationChoiceCoefficients.csv	Stop destination choice models size term coefficients

File Name	Description
StopFrequency.xls	Stop frequency model UECs (mandatory, maintenance, discretionary, at work subtour)
StopPurposeLookup.csv	Stop purpose shares by primary tour purpose, direction, time-of-day, and person type
TourDepartureAndDuration.xls	Tour departure time and duration model UECs (mandatory, joint non-mandatory, individual non-mandatory, at work subtour)
TripDepartHourPercents.csv	Trip depart time hour percents by tour purpose, inbound/outbound, tour hour and trip index
TripModeChoice.xls	Trip mode choice model UECs (mandatory, non-mandatory, work subtour)
TVPB.xls	Transit virtual path builder UEC

Inputs Folder

The Inputs folder stores various inputs such as zonal land use data, the synthetic population, time-of-day factors for EMME, zonal accessibility measures, etc. Some of these inputs are expected to change for each new scenario/alternative. Each input file is described in detail below.

Table 2: Inputs Folder Input Files

File Name	Description
EMME - Highway	
Hwymodes.201	Highway modes file
Highway_p<tod>.211	Highway network for time-of-day period
Functions.411	EMME functions file
Directional.splits	PA to OD directional factors by time period
Kzone.mtx	Kzone park and ride lot matrix
Tod_factors.p{1-8}	Time period trip table factors for each trip matrix by time period
Tod_flags.in	Zone flags for trip table fractioning
TOD_OCC_INC_NON_WORK.p{1-8}	Time period distribution by occupancy & income for HBO & NHB
tollsys.flag	Toll facility link flag used by extraclass.mac
Toll	Link tolls
emmebank	EMME 4 databank with initial daily demand matrices 1-99
emmemat	EMME 4 matrix folder to go along with emmebank

File Name	Description
EMME – Transit	
17120001_parking.csv	Parking spaces and cost for park-and-ride transit mode
bus_node_extra_attributes.csv	Bust stop type and real time information
cermakbrt_parking.csv	BRT parking spaces and cost
hwy_node_pef.csv	Pedestrian environment factor scaled from 1 through 3
rail_node_extra_attributes.csv	Rail station parking spaces and costs
rail_node_pef.csv	Pedestrian environment factor scaled from 1 through 3
tranmodes.txt	Description of transit modes
transveh.txt	Description of transit vehicle types
Tap_attributes.csv	Transit access point (TAP) file
Tap2node.csv	TAP access/egress links
boarding_ease_by_line_id.csv	Line boarding ease
Line_capacity3.csv	line capacity
Conv<tod>_<tod>267.txt	TAP to TAP conventional demand matrix
Prem<tod>_<tod>268.txt	TAP to TAP premium demand matrix
CT-RAMP	
Accessibility.csv	Zonal accessibility measures (which would be calculated on-the-fly in a full ABM implementation)
ForecastHHFile_MAZ.csv	Synthesized household population file with households allocated to MAZs
ForecastPersonFile.csv	Synthesized person population file
totalpctwkt.csv	Percent of zone with no walk transit access, with short walk to transit access, and with long walk to transit access
workerOccupationCodes.csv	Crosswalk from Census Industry codes to CMAP ABM employment categories
SubZoneData.csv	Zonal level data such as population, employment by employment type, parking costs, area type, etc
ModesNames.csv	Mode code to Mode name lookup for MySQL database script
walkPreferences.csv	Propensity to walk to transit factors by age
MAZ to MAZ/TAP Impedances	
Chicago_Streets.shp/shx/dbf	NAVTEQ allstreets network
CMAP_MAZ_Cents.csv	MAZ centroids
cmapRStoSubZone.txt	Mapping between RS and sub-zone (MAZ)
nodes_in_MAZ_CMAP.txt	Nodes that are within a 50 foot buffer of each MAZ polygon
cmapDistFromHwy.txt	Nodes that are within 2500 feet of a highway (future year only)

File Name	Description
cmapMAZLandUseShare.txt	Land use share by MAZ (future year only)

Inputs\{Transit Scenario Number} Folder

The name of this subfolder (for example 100) corresponds to the transit scenario number used in the databank and is used by the transit macros. This folder contains input files required for running transit procedures such as transit network and attributes, and stop/station characteristics. These files are described in the table below.

Table 3: Inputs\100 Folder Input Files

File Name	Description
access.network_{1-8}	Pedestrian access links
bus.itinerary_{1-8}	Bus line itineraries
busnode.extatt_{1-8}	Extra attributes for bus stops
busseg.extatt_{1-8}	Extra attributes for bus segments
rail.itinerary_{1-8}	Rail line itineraries
rail.network_{1-8}	Network of rail links
railnode.extatt_{1-8}	Extra attributes for rail stations
railseg.extatt_{1-8}	Extra attributes for rail segments

Accessibilites.csv – Zonal accessibility measures

Table 4: Accessibility Measures File Fields

Field	Description
subzone09	Zone
autoPeakRetail	Peak Auto Retail accessibility
autoPeakTotal	Peak Auto Total accessibility
autoOffPeakRetail	Off-Peak Auto Retail accessibility
autoOffPeakTotal	Off-Peak Auto Total accessibility
transitPeakRetail	Peak Transit Retail accessibility
transitPeakTotal	Peak Transit Total accessibility
transitOffPeakRetail	Off-Peak Transit Retail accessibility
transitOffPeakTotal	Off-Peak Transit Total accessibility
nonMotorizedRetail	Non-motorized retail accessibility
nonMotorizedTotal	Non-motorized total accessibility
access17	Transit accessibilities to all non-mandatory activity locations
access18	Non-motorized accessibilities to all non-mandatory activity locations

ForecastHHFile.csv – Synthetic Population Household File

This file contains the synthesized household population for the model year. See the 2000 PUMS documentation for details on the possible values for the fields.

Table 5: Synthesized Household File Fields

Field	Description
HHID	Household ID
TAZ	TAZ
SERIALNO	Housing/Group Quarters (GQ)Unit Serial Number
PUMA5	Public Use Microdata Area Code (PUMA)
HINC	Household Total Income in 1999
PERSONS	Number of person records following this housing record
HHT	Household/Family Type
UNITTYPE	Type of unit
NOC	Number of own children under 18 years in household
BLDGSZ	Size of Building
TENURE	Home Ownership
hinccat1	HH annual income (4 categories)
hinccat2	HH annual income (9 1-digit categories)
hhagecat	age of householder (1--under 65, 2--65+)
hsizecat	household size category (1--1,2--2,3--3,4--4,5--5+)
hsizecat6	household size category (1--1,2--2,3--3,4--4,5--5,6--6+)
hfamily	HH is nonfamily(1) or family(2)
hunittype	Type of group quarters household (0--non-GQ, 1--inst. GQ, 2--noninst GQ, 3--empty housing unit)
hNOCcat	Presence of own children age 00-17 (0-none, 1-1 or more)
hNCcat6	Number children age 00-17 in household (0-0,1-1,2-2,3-3,4-4,5-5+)
hwrkrcat	Number in HH employed PT or FT (0--0,1--1,2--2,3--3+)
hwrkcat5	Number in HH employed PT or FT (0--0,1--1,2--2,3--3,4--4+)
h0005	Number in HH age under 6
h0611	Number in HH age 6-11
h1215	Number in HH age 12-15
h1617	Number in HH age 16-17
h1824	Number in HH age 18-24
h2534	Number in HH age 25-34
h3549	Number in HH age 35-49
h5064	Number in HH age 50-64

h6579	Number in HH age 65-79
h80up	Number in HH age 80+
hworkers	Number in HH employed PT or FT
hwork_f	Num in HH with ptype=1 (FT worker age 16+)
hwork_p	Num in HH with ptype=2 (PT worker nonstudent age 16+)
huniv	Num in HH with ptype=3 (university student)
hnwork	Num in HH with ptype=4 (nonworker nonstudent age 16-64)
hretire	Num in HH with ptype=5 (nonworker nonstudent age 65+)
hpresch	Num in HH with ptype=8 (under age 6)
hschpred	Num in HH with ptype=7 (age 6-15)
hschdriv	Num in HH with ptype=6 (age 16-19 student ; not FT wrkr or univ stud)
htypdwel	Type of dwelling unit: 1-detached SF / 2-attached / 3-mobile home
hownrent	own or rent dwelling
hadnwst	Number in HH not-employed students age 16+ (PUMS P.ESR in {3,6} and P.GRADE>0)
hadwpst	Number in HH employed students age 16+ (PUMS P.ESR in {1,2,4,5} and P.GRADE>0)
hadkids	Number adult children in HH (padkid=1)
bucketBin	Number of subgroups into which each PUMA's HHs of a single category are grouped
originalPUMA	Original PUMA
WIF	Workers in family
VEHICL	Number of Vehicles Available
MAZ	MAZ

ForecastPersonFile.csv - Synthetic Population Person File

This file contains the synthesized person population for the model year. See the 2000 PUMS documentation for details on the possible values for the fields.

Table 6: Synthesized Person File Fields

Field	Description
HHID	Household ID
PERID	Person ID
AGE	Age
RELATE	Relationship
ESR	Employment Status Recode
GRADE	School Enrollment: Grade Level Attending

PNUM	Person Sequence Number
PAUG	Augmented Person Flag
DDP	Data-defined Person Flag
SEX	Sex
WEEKS	Weeks Worked in 1999
HOURS	Hours Worked Per Week in 1999
RACE1	Race Recode 1
HISPAN	Hispanic or Latino Origin
MSP	Married, Spouse Present Recode
POVERTY	Person ’s Poverty Status
EARNS	Person ’s Total Earnings in 1999
pagecat	Age category (10 1-digit categories)
pemploy	Employment status (1-employed FT, 2-employed PT, 3-not employed, 4-under age 16)
pstudent	1-primary or secondary student age 3+; 2-post-secondary student; 3-age under 3 or not a student (from PUMS P.GRADE)
phispan	hispanic: 1-not hispanic, 2-hispanic (PUMS min(P.HISPAN,2))
ptype	Person type (8 categories)
padkid	young adult age 18-24 in family household with older adult age 35-64 in HH: 1-yes, 2-no
EDUC	Education attainment in years
INDCEN	Industry Census code

workerOccupationCodes.csv – Census Industry to Employment Categories

Table 7: Worker Occupation Code Fields

Field	Description
IndCenMin	Census person industry code range minimum
IndCenMax	Census person industry code range maximum
workerOccupation	CMAP ABM employment category

SubZoneData.csv – Zone data

Table 8: Zone Data Fields

Field	Description
Subzone	Subzone (MAZ)
Zone	TAZ

Construc	Construction Employment
Manufac	Manufacturing Employment
TCU	TCU (Transportation, Communication, Utilities)
Wholesl	Wholesale Employment
Retail	Retail Employment
FIRE	Fire Employment
Service	Service Employment
Private	Total Private Employment
Govt	Government Employment
Emp	Total Employment
Pop	Population
Hshld	Households
Univ	University Enrollment
Acres	Area (acres)
OtherEmp	Other employment
District	District code
Parktot	Total parking spaces
Parklng	Long term parking spaces
Propfree	Proportion free parking
Parkrate	Parking rate ($)
areatype	Area type
County	County code
CBD FLAG	CBD Flag
Highschool	High school code
Gradeschool	Grade school code
naics11	NAICS employment category employment
naics21	NAICS employment category employment
naics22	NAICS employment category employment
naics23	NAICS employment category employment
naics31_33	NAICS employment category employment
naics42	NAICS employment category employment
naics44_45	NAICS employment category employment
naics48_49	NAICS employment category employment
naics51	NAICS employment category employment
naics52	NAICS employment category employment
naics53	NAICS employment category employment
naics54	NAICS employment category employment
naics55	NAICS employment category employment

naics56	NAICS employment category employment
naics61	NAICS employment category employment
naics62	NAICS employment category employment
naics71	NAICS employment category employment
naics72	NAICS employment category employment
naics81	NAICS employment category employment
naics92	NAICS employment category employment
naics99	NAICS employment category employment
k8_enroll	School enrollment
Highschool_enroll	High school enrollment
small_coll	Small college enrollment
lg_college	Large college enrollment
B_hhden	Buffered household density
B_empden	Buffered employment density
Pef	Walkability measure for the zone
Ring	Area ring code

ModeNames.csv – Mode code to name lookup table

Table 9: Mode Name Fields for Loading Outputs into Database

Field	Description
CODE	Mode code (alternative number)
NAME	Mode name

walkPreferences.csv – Person walk preferences by age

Table 10: Walk Preferences File

Field	Description
Age	Age
Lower 0.5%	Lower 0.5% value
Median 50%	Median 50% value
Upper 99.5%	Upper 99.5% value

CMAP-ABM EMME Project Folder

The CMAP-ABM folder each contains the EMME databank. The databank is used for skimming and assignment. The matrix numbers are the same across databanks. The time periods are defined in the first table below. The second table defines the highway matrices and the third table the transit matrices.

Table 11: Time Periods

TOD Period	Time-Of-Day	Highway Scenario	Transit Scenario
1	8pm-6am	1	101
2	6am-7am	2	102
3	7am-9am	3	103
4	9am-10am	4	104
5	10am-2pm	5	105
6	2pm-4pm	6	106
7	4pm-6pm	7	107
8	6pm-8pm	8	108

Highway Skims

Table 12: Highway Skims

Variable		Matrix Number by Highway Mode					
		SOV (SV1**)		HOV2 (HV2**)		HOV3+ (HV3**)	
		Non-Toll (*n*)	Toll (*t*)	Non-Toll (*n*)	Toll (*t*)	Non-Toll (*n*)	Toll (*t*)
Congested time	**(**1c)**	<tod>175	<tod>180	<tod>185	<tod>190	<tod>195	<tod>200
Toll	**(**2t)**	<tod>176	<tod>181	<tod>186	<tod>191	<tod>196	<tod>201
Distance	**(**3d)**	<tod>177	<tod>182	<tod>187	<tod>192	<tod>197	<tod>202
Toll distance	**(**4m)**	<tod>178	<tod>183	<tod>188	<tod>193	<tod>198	<tod>203
Free-flow time	**(**5f)**	<tod>179	<tod>184	<tod>189	<tod>194	<tod>199	<tod>204

Transit Skims

Table 13: Transit Skims

Variable / User Class		Matrix Number by Transit Mode	
		Conventional (Not currently used)	Premium
Generalized Cost	**1**	<tod>X	<tod>430

Variable / User Class		Matrix Number by Transit Mode	
		Conventional (Not currently used)	Premium
Generalized Cost	**2**	<tod>X	<tod>492
Generalized Cost	**3**	<tod>X	<tod>554

Model Outputs

The core outputs of the CMAP ABM are summarized in the following table. The outputs are grouped into three sections – EMME, CSV Files, and Trip Matrices. In addition to the skims created and described above, the EMME outputs are matrices created by the EMME macros that remain in EMME. These are non-CT-RAMP matrices such as trucks, external model trips and airport trips. The CSV output files are the ABM model outputs and are described in more detail below. The Trip Matrices are also output by CT-RAMP and are written into the databanks.

Table 14: Model Output Files

Output	Description
EMME	
Mf135-142	Truck demand matrices by toll/non-toll and class (commercial, light truck, medium truck, heavy truck)
Mf143-154	External demand matrices by toll/non-toll and class (SOV1 low inc., SOV1 high inc., HOV2 low inc., HOV2 high inc., HOV3 low inc., HOV3 high inc.)
Mf155-166	Airport demand matrices by toll/non-toll and class (SOV1 low inc., SOV1 high inc., HOV2 low inc., HOV2 high inc., HOV3 low inc., HOV3 high inc.)
CSV Files	
hh_Data_{loop}.csv	household attribution results
personData_{loop}.csv	person attribution results
ShadowPricing_{shadowPricingIteration}.csv	shadow pricing results
wsLocResults_{loop}.csv	usual work and school location choice results
aoResults.csv	auto ownership results
cdapResults.csv	CDAP model results
indivTourData_{loop}.csv	individual tour records
jointTourData_{loop}.csv	joint tour records
indivTripData_{loop}.csv	individual trip records
jointTripData_{loop}.csv	joint trip records
tapLines.csv	Lines served by TAP

SP_MAZ_to_MAZ_and_TAP.txt	MAZ to MAZ/TAP all streets network shortest distance
Auto Trip Matrices	
Mf<tod>123	SOV1n1 - SOV non-toll, low income trips
Mf<tod>124	SOV1tl – SOV toll, low income trips
Mf<tod>125	SOV1nh – SOV non-toll, high income trips
Mf<tod>126	SOV1th – SOV toll, high income trips
Mf<tod>127	HOV2nl – HOV2 non-toll, low income trips
Mf<tod>128	HOV2tl – HOV2 toll, low income trips
Mf<tod>129	HOV2nh – HOV2 non-toll, high income trips
Mf<tod>130	HOV2th – HOV2 toll, high income trips
Mf<tod>131	HOV3nl – HOV 3+ non-toll, low income trips
Mf<tod>132	HOV3tl – HOV 3+ toll, low income trips
Mf<tod>133	HOV3nh – HOV 3+ non-toll, high income trips
Mf<tod>134	HOV3th – HOV 3+ toll, high income trips
Transit Trip Matrices	
Mf<tod>271	Local trips – class 1
Mf<tod>272	Local trips – class 2
Mf<tod>273	Local trips – class 3
Mf<tod>274	Premium trips – class 1
Mf<tod>275	Premium trips – class 2
Mf<tod>276	Premium trips – class 3

CSV Files

1) Household attribution results – hh_Data_{loop}.csv

The household data file for each feedback loop has the following fields:

Table 15: Household Output File Fields

Field	Description
hh_id	Household ID
taz	Home zone
walk_subzone	Walk subzone
income	HH income
autos	Autos
jtf_choice	Joint tour frequency choice
size	HH size
workers	Number of workers in HH

auto_suff	Auto sufficiency

2) Person attribution results – personData_{loop}.csv

The person data file for each feedback loop has the following fields:

Table 16: Person Output File Fields

Field	Description
hh_id	Household ID
person_id	Person ID
person_num	person number in HH
age	Age
gender	Gender
type	person type (worker, student, etc)
value_of_time	value of time
fp_choice	free parking choice
activity_pattern	activity pattern
imf_choice	individual mandatory tour freq choice
inmf_choice	individual non-mandatory tour freq choice
Walk_time_weight	Individual walk time weight
Walk_speed	Individual walk speed
Max_walk	Individual max walk distance
user_class_work_walk user_class_work_pnr user_class_work_knr user_class_non_work_walk user_class_non_work_pnr user_class_non_work_knr	Individual transit user class membership for work/non-work by walk, pnr, and knr

3) Shadow pricing result files – ShadowPricing_{shadowPricingIteration}.csv

Each iteration of the shadow pricing results for each taz and subzone by destination choice size term class are written to this file.

4) Usual work and school location choice results – wsLocResults_{loop}.csv

The usual work and school location choice results table consists of the following fields:

Table 17: Usual Work and School Location Choice Output File Fields

Field	Description
HHID	household ID

HomeTAZ	Home zone
HomeSubZone	Home subzone
PersonID	Person ID
PersonNum	Person number in HH
PersonWorkOcc	Worker occupancy code
PersonType	Person type
PersonAge	Age
EmploymentCategory	employment category
StudentCategory	student category
WorkLocation	work TAZ location
WorkSubZone	work subzone
SchoolLocation	school TAZ location
SchoolSubZone	school subzone

Table 18: Person Type Codes

Person Type Code	Description
1	Full-time worker
2	Part-time worker
3	Non-worker
4	Retired
5	University Student
6	Student of driving age
7	Student of non-driving age
8	Child too young for school

5) Auto ownership results – aoResults.csv

Table 19: Auto Ownership Output File Fields

Field	Description
hhid	Household ID
ao	Number of autos

6) Coordinated Daily Activity Pattern (CDAP) results – cdapResults.csv

The results of the CDAP model are written to this file.

Table 20: CDAP Output File Fields

Field	Description
HHID	household ID
PersonID	Person ID
PersonNum	Person number in HH
PersonType	Person type
ActivityString	Daily activity pattern (M = mandatory, N = non-mandatory, H = at home)

7) Individual tour records – indivTourData_{loop}.csv

Individual tours for each CT-RAMP feedback loop are written to this file.

Table 21: Individual Tours Output File Fields

Field	Description
hh_id	Household ID
person_id	Person ID
person_num	Person number in household
person_type	Numeric identifier of person type
tour_id	Tour number for each person (e.g. 0 = first tour made by a person, 1 = second tour, etc)
tour_category	tour category
tour_purpose	purpose of tour
orig_taz	tour origin zone location
orig_walk_segment	tour origin subzone
dest_taz	tour destination zone location
dest_walk_segment	tour destination subzone
depart_period	tour depart period
Arrive_period	tour arrival period
tour_mode	mode of tour
atWork_freq	number of at work subtours made
num_ob_stops	number of outbound stops on tour
num_ib_stops	number of inbound stops made on tour
util_1	utility for mode 1
util_2	utility for mode 2
…	
util_26	utility for mode 26
prob_1	probability of mode 1

prob_2	probability of mode 2
…	
prob_26	probability of mode 26

Table 22: Trip/Tour Mode Codes

Trip/Tour Mode Code	Description
1	Drive alone free
2	Drive alone pay
3	Shared ride 2 free
4	Shared ride 2 pay
5	Shared ride 3+ free
6	Shared ride 3+ pay
7	Walk
8	Bike
9	Walk to local transit
10	Walk to premium transit
11	Drive to local transit
12	Drive to premium transit
13	Taxi
14	School bus

8) Joint tour records – jointTourData_{loop}.csv

Joint tours for each CT-RAMP feedback loop are written to this file.

Table 23: Joint Tours Output File Fields

Field	Description
hh_id	Household ID
tour_id	Tour ID
tour_category	tour category (joint non-mandatory)
tour_purpose	purpose of tour
tour_composition	tour composition
tour_participants	household participants on tour (e.g. 1 2 4 = persons 1, 2, and 4 in the household)
orig_taz	tour origin zone location
orig_walk_segment	tour origin subzone
dest_taz	tour destination zone location
dest_walk_segment	tour destination subzone

depart_period	tour depart period
arrive_period	tour arrival period
tour_mode	mode of tour
num_ob_stops	number of outbound stops on tour
num_ib_stops	number of inbound stops made on tour
util_1	utility for mode 1
util_2	utility for mode 2
…	
util_26	utility for mode 26
prob_1	probability of mode 1
prob_2	probability of mode 2
…	
prob_26	probability of mode 26

9) Individual trip records – indivTripData_{loop}.csv

Individual trips for each CT-RAMP feedback loop are written to this file.

Table 24: Individual Trips Output File Fields

Field	Description
hh_id	Household ID
person_id	Person ID
person_num	Person number
tour_id	Tour ID
stop_id	Stop ID
inbound	Is Inbound Trip
tour_purpose	Tour Purpose
orig_purpose	Origin purpose
dest_purpose	Destination purpose
orig_maz	Origin MAZ
orig_walk_segment	Origin walk market segment
dest_maz	Destination MAZ
dest_walk_segment	Destination walk market segment
parking_taz	Parking zone used
stop_period	Stop period
trip_mode	Trip Mode
tour_mode	Tour mode
tour_category	Tour category
Board_tap	Boarding tap

Alight_tap	Alighting tap
Orig_taz	Origin TAZ
Dest_taz	Destination TAZ

10) Joint trip records – jointTripData_{loop}.csv

Joint trips for each CT-RAMP feedback loop are written to this file.

Table 25: Joint Trips Output File Fields

Field	Description
hh_id	Household ID
tour_id	Tour ID
stop_id	Stop ID
inbound	Is Inbound Trip
tour_purpose	Tour Purpose
orig_purpose	Origin purpose
dest_purpose	Destination purpose
orig_maz	Origin MAZ
orig_walk_segment	Origin walk market segment
dest_maz	Destination MAZ
dest_walk_segment	Destination walk market segment
parking_taz	Parking zone used
stop_period	Stop period
trip_mode	Trip Mode
Num_part	Number of participants on tour
tour_mode	Tour mode
tour_category	Tour category
Board_tap	Boarding tap
Alight_tap	Alighting tap
Orig_taz	Origin TAZ
Dest_taz	Destination TAZ

11) TAP Lines – tapLines.csv

This file contains a list of lines served for each TAP. It is used to trim the available TAPs considered by origin and destination micro-zone.

Table 26: Tap Lines File Fields

Field	Description
tap	TAP ID
Lines	Space separated list of line names

12) MAZ to MAZ/TAP Impedances – SP_MAZ_to_MAZ_and_TAP.txt

This file contains the distance between near MAZs and TAPs.

Table 27: MAZ to MAZ/TAP Impedances File Fields

Field	Description
Origin	Origin node
Destin	Destination node
SP	Shortest path distance
Euclid	Euclidean distance
Ratio	SP/Euclidean

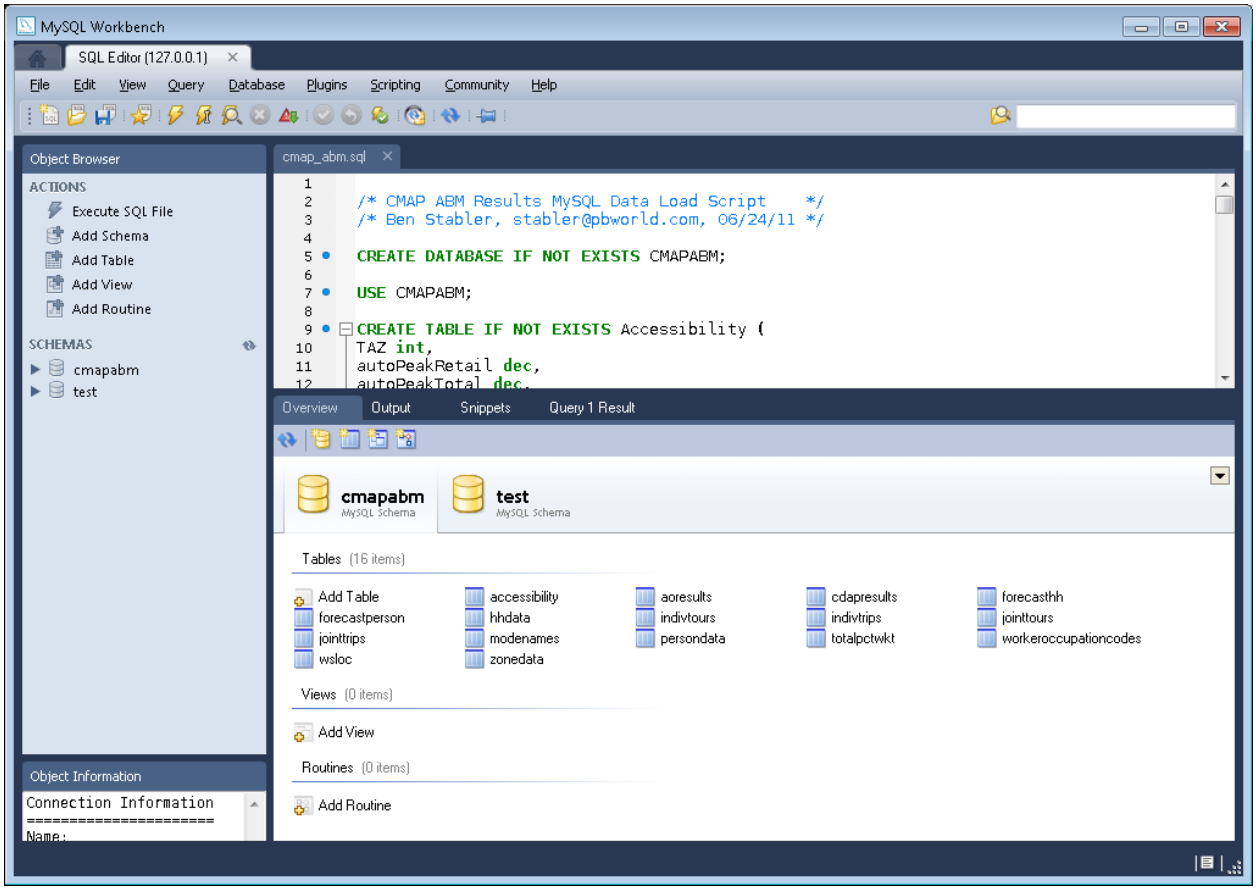
MySQL

The resulting CT-RAMP output files plus the input synthetic population and zone data files can be loaded into a MySQL database with the cmap_abm.sql script in the exec folder. To run the script, do the following:

1) Edit the LOAD DATA LOCAL FILE commands to reference the project directory folder.

2) Run the script with MySQL Workbench. The script will create a new database called CMAPABM with 16 tables, as shown below. The tables are the same tables described in the inputs and outputs sections above.

Figure 6: MySQL CMAPABM Database



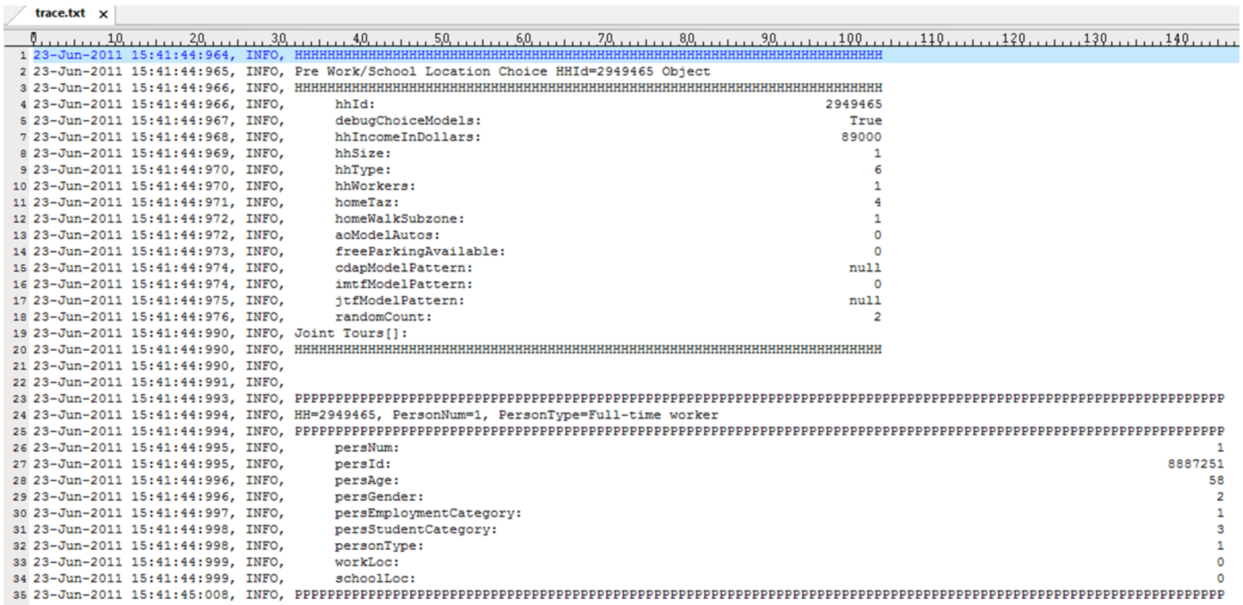
Model Logging/Trace Results

CT-RAMP writes a series of log files to the logFiles folder during a model run. These log files are extremely useful for understanding the model as well as for debugging a model run. The main log file is the event.log file and it is the starting point for reviewing the logs. In addition to event.log, the event_hh.log logs the household data manager work, event_mtx.log logs the matrix data manager work, and event-node<X>.log logs the work on each remote node. There are also model component specific log files for each node such as event-node1-ao.log for the auto ownership model being run on remote machine node 1.

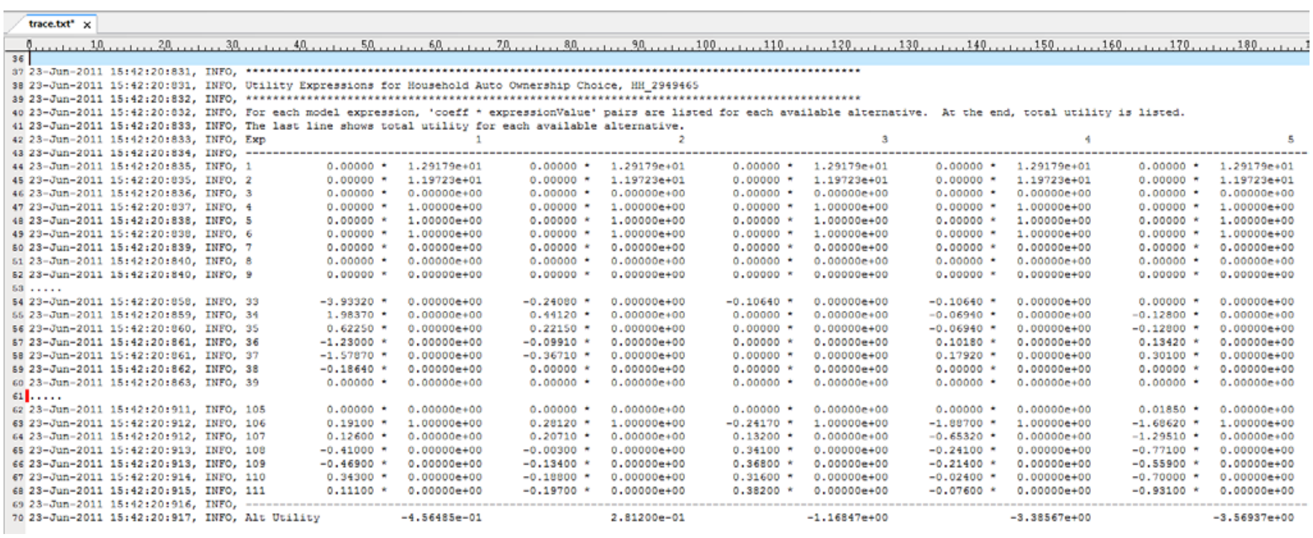
In addition to writing log files, CT-RAMP can trace model calculations for a user specified household. To trace results for household 2949465 for example, set Debug.Trace.HouseholdIdList = 2949465 in the cmap.properties file. This tells CT-RAMP to basically write out all calculations for every person in household 2949465, including the results of the UEC calculations for each model. Figure 7 below contains a sample of the household trace results. The first screenshot shows a sample of the trace results for the household and person number 1 in the household. As is shown below, key attributes of the household and person are traced such as the household income, size, number of workers, home

zone, person age, person employment category (worker, retired, etc). As is shown in the second screenshot, 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 111 expressions and five alternatives (0, 1, 2, 3, 4+ autos) traced correspond exactly with the expressions and alternatives in the auto ownership UEC file, which makes tracing and debugging easier.

Figure 7: Household Trace Results



```
23-Jun-2011 15:41:44:964, INFO, HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
23-Jun-2011 15:41:44:965, INFO, Pre Work/School Location Choice HHId=2949465 Object
23-Jun-2011 15:41:44:966, INFO, HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
23-Jun-2011 15:41:44:966, INFO,      hhId:                                                  2949465
23-Jun-2011 15:41:44:967, INFO,      debugChoiceModels:                                        True
23-Jun-2011 15:41:44:968, INFO,      hhIncomeInDollars:                                       89000
23-Jun-2011 15:41:44:969, INFO,      hhSize:                                                      1
23-Jun-2011 15:41:44:970, INFO,      hhType:                                                      6
23-Jun-2011 15:41:44:970, INFO,      hhWorkers:                                                   1
23-Jun-2011 15:41:44:971, INFO,      homeTaz:                                                     4
23-Jun-2011 15:41:44:972, INFO,      homeWalkSubzone:                                             1
23-Jun-2011 15:41:44:972, INFO,      aoModelAutos:                                                0
23-Jun-2011 15:41:44:973, INFO,      freeParkingAvailable:                                        0
23-Jun-2011 15:41:44:974, INFO,      cdapModelPattern:                                         null
23-Jun-2011 15:41:44:974, INFO,      imtfModelPattern:                                            0
23-Jun-2011 15:41:44:975, INFO,      jtfModelPattern:                                          null
23-Jun-2011 15:41:44:976, INFO,      randomCount:                                                 2
23-Jun-2011 15:41:44:990, INFO, Joint Tours[]:
23-Jun-2011 15:41:44:990, INFO, HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
23-Jun-2011 15:41:44:990, INFO,
23-Jun-2011 15:41:44:991, INFO,
23-Jun-2011 15:41:44:993, INFO, PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
23-Jun-2011 15:41:44:994, INFO, HH=2949465, PersonNum=1, PersonType=Full-time worker
23-Jun-2011 15:41:44:994, INFO, PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
23-Jun-2011 15:41:44:995, INFO,      persNum:                                                                                          1
23-Jun-2011 15:41:44:995, INFO,      persId:                                                                                     8887251
23-Jun-2011 15:41:44:996, INFO,      persAge:                                                                                         58
23-Jun-2011 15:41:44:996, INFO,      persGender:                                                                                       2
23-Jun-2011 15:41:44:997, INFO,      persEmploymentCategory:                                                                           1
23-Jun-2011 15:41:44:998, INFO,      persStudentCategory:                                                                              3
23-Jun-2011 15:41:44:998, INFO,      personType:                                                                                       1
23-Jun-2011 15:41:44:999, INFO,      workLoc:                                                                                          0
23-Jun-2011 15:41:44:999, INFO,      schoolLoc:                                                                                        0
23-Jun-2011 15:41:45:008, INFO, PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
```



```
23-Jun-2011 15:42:20:831, INFO, ****************************************************************************
23-Jun-2011 15:42:20:831, INFO, Utility Expressions for Household Auto Ownership Choice, HH_2949465
23-Jun-2011 15:42:20:832, INFO, ****************************************************************************
23-Jun-2011 15:42:20:832, INFO, For each model expression, 'coeff * expressionValue' pairs are listed for each available alternative.  At the end, total utility is listed.
23-Jun-2011 15:42:20:833, INFO, The last line shows total utility for each available alternative.
23-Jun-2011 15:42:20:833, INFO, Exp                          1                              2                              3                              4                              5
23-Jun-2011 15:42:20:834, INFO, ----------------------------------------------------------------------------------------------------------------------------------------------------------------
23-Jun-2011 15:42:20:835, INFO, 1         0.00000 *  1.29179e+01      0.00000 *  1.29179e+01      0.00000 *  1.29179e+01      0.00000 *  1.29179e+01      0.00000 *  1.29179e+01
23-Jun-2011 15:42:20:835, INFO, 2         0.00000 *  1.19723e+01      0.00000 *  1.19723e+01      0.00000 *  1.19723e+01      0.00000 *  1.19723e+01      0.00000 *  1.19723e+01
23-Jun-2011 15:42:20:836, INFO, 3         0.00000 *  0.00000e+00      0.00000 *  0.00000e+00      0.00000 *  0.00000e+00      0.00000 *  0.00000e+00      0.00000 *  0.00000e+00
23-Jun-2011 15:42:20:837, INFO, 4         0.00000 *  1.00000e+00      0.00000 *  1.00000e+00      0.00000 *  1.00000e+00      0.00000 *  1.00000e+00      0.00000 *  1.00000e+00
23-Jun-2011 15:42:20:838, INFO, 5         0.00000 *  1.00000e+00      0.00000 *  1.00000e+00      0.00000 *  1.00000e+00      0.00000 *  1.00000e+00      0.00000 *  1.00000e+00
23-Jun-2011 15:42:20:838, INFO, 6         0.00000 *  1.00000e+00      0.00000 *  1.00000e+00      0.00000 *  1.00000e+00      0.00000 *  1.00000e+00      0.00000 *  1.00000e+00
23-Jun-2011 15:42:20:839, INFO, 7         0.00000 *  0.00000e+00      0.00000 *  0.00000e+00      0.00000 *  0.00000e+00      0.00000 *  0.00000e+00      0.00000 *  0.00000e+00
23-Jun-2011 15:42:20:840, INFO, 8         0.00000 *  0.00000e+00      0.00000 *  0.00000e+00      0.00000 *  0.00000e+00      0.00000 *  0.00000e+00      0.00000 *  0.00000e+00
23-Jun-2011 15:42:20:840, INFO, 9         0.00000 *  0.00000e+00      0.00000 *  0.00000e+00      0.00000 *  0.00000e+00      0.00000 *  0.00000e+00      0.00000 *  0.00000e+00
.....
23-Jun-2011 15:42:20:858, INFO, 33       -3.93320 *  0.00000e+00     -0.24080 *  0.00000e+00     -0.10640 *  0.00000e+00     -0.10640 *  0.00000e+00      0.00000 *  0.00000e+00
23-Jun-2011 15:42:20:859, INFO, 34        1.98370 *  0.00000e+00      0.44120 *  0.00000e+00      0.00000 *  0.00000e+00     -0.06940 *  0.00000e+00     -0.12800 *  0.00000e+00
23-Jun-2011 15:42:20:860, INFO, 35        0.62250 *  0.00000e+00      0.22150 *  0.00000e+00      0.00000 *  0.00000e+00     -0.06940 *  0.00000e+00     -0.12800 *  0.00000e+00
23-Jun-2011 15:42:20:861, INFO, 36       -1.23000 *  0.00000e+00     -0.09910 *  0.00000e+00      0.00000 *  0.00000e+00      0.10180 *  0.00000e+00      0.13420 *  0.00000e+00
23-Jun-2011 15:42:20:861, INFO, 37       -1.57870 *  0.00000e+00     -0.36710 *  0.00000e+00      0.00000 *  0.00000e+00      0.17920 *  0.00000e+00      0.30100 *  0.00000e+00
23-Jun-2011 15:42:20:862, INFO, 38       -0.18640 *  0.00000e+00      0.00000 *  0.00000e+00      0.00000 *  0.00000e+00      0.00000 *  0.00000e+00      0.00000 *  0.00000e+00
23-Jun-2011 15:42:20:863, INFO, 39        0.00000 *  0.00000e+00      0.00000 *  0.00000e+00      0.00000 *  0.00000e+00      0.00000 *  0.00000e+00      0.00000 *  0.00000e+00
.....
23-Jun-2011 15:42:20:911, INFO, 105       0.00000 *  0.00000e+00      0.00000 *  0.00000e+00      0.00000 *  0.00000e+00      0.00000 *  0.00000e+00      0.01850 *  0.00000e+00
23-Jun-2011 15:42:20:912, INFO, 106       0.19100 *  1.00000e+00      0.28120 *  1.00000e+00     -0.24170 *  1.00000e+00     -1.88700 *  1.00000e+00     -1.68620 *  1.00000e+00
23-Jun-2011 15:42:20:912, INFO, 107       0.12600 *  0.00000e+00      0.20710 *  0.00000e+00      0.13200 *  0.00000e+00     -0.65320 *  0.00000e+00     -1.29510 *  0.00000e+00
23-Jun-2011 15:42:20:913, INFO, 108      -0.41000 *  0.00000e+00     -0.00300 *  0.00000e+00      0.34100 *  0.00000e+00     -0.24100 *  0.00000e+00     -0.77100 *  0.00000e+00
23-Jun-2011 15:42:20:913, INFO, 109      -0.46900 *  0.00000e+00     -0.13400 *  0.00000e+00      0.36800 *  0.00000e+00     -0.21400 *  0.00000e+00     -0.55900 *  0.00000e+00
23-Jun-2011 15:42:20:914, INFO, 110       0.34300 *  0.00000e+00     -0.18800 *  0.00000e+00      0.31600 *  0.00000e+00     -0.02400 *  0.00000e+00     -0.70000 *  0.00000e+00
23-Jun-2011 15:42:20:915, INFO, 111       0.11100 *  0.00000e+00     -0.19700 *  0.00000e+00      0.38200 *  0.00000e+00     -0.07600 *  0.00000e+00     -0.93100 *  0.00000e+00
23-Jun-2011 15:42:20:916, INFO, ----------------------------------------------------------------------------------------------------------------------------------------------------------------
23-Jun-2011 15:42:20:917, INFO, Alt Utility          -4.56485e-01                  2.81200e-01                 -1.16847e+00                 -3.38567e+00                 -3.56937e+00
```

Running the Transit Virtual Path Builder for Estimation

The runTVPB.bat file runs the transit virtual path builder (TVPB) for estimation purposes via the command line. Before running it, run runMain-SingleProcess.cmd in the Config folder to start the HH

and matrix data servers. The TVPB is configured with the cmap.properties. The key settings in this file are:

54) Project.Directory= d:/model/ #model run folder
55) tvpb.uec = uecs/TVPB.xls #path generalized cost settings file
56) tvpb.tripfile = inputs/trips.csv #input trip file to run TVPB on
57) tvpb.runthreaded = True #run TVPB in threaded mode. Set to false when debugging
58) tvpb.maxpaths.walk = 3 #max number of walk paths to output
59) tvpb.maxpaths.knr = 3 #max number of KNR paths to output
60) tvpb.maxpaths.pnr = 3 #max number of PNR paths to output
61) tvpb.maxwalkdist = 10560 #max MAZ to TAP walk distance in feet. Used to trim the walk alternatives since the maz2tapfile input file is for KNR as well.
62) tvpb.maxtotalwalkdist =15840 #max total walk (origin walk + destination walk)
63) tvpb.maxpnrdist = 105600 #max MAZ to TAP PNR distance in feet. Used to trim the PNR alternatives set.
64) tvpb.tripfile.debugRecord = -1 #debug record to output trace results for; -1 for no tracing.
65) tvpb.alts = 1 #1 alternative (premium) in the UEC
66) tvpb.trace = False #NA; Traces TVPB results when running as part of the ABM
67) HouseholdManager.WalkPreferences.FileName = inputs/walkPreferences.csv #propensity to walk factors by age

The TVPB program outputs the outputs/tvpb_paths.csv file when complete. The tvpb_paths.csv file contains path alternatives for each input trip record. The file contains the following format:

1) Each row is an alternative path for a trip record. If there are no alternatives, then the row has one alternative with no path data.
2) The recid field is the input trip file row id
3) Each record has a series of input trip file attributes
4) Each record has an alt – alternative number – field
5) Each record has an otap – origin tap – and dtap – destination tap – field
6) Each KNR/PNR record has an otaptaz – origin tap's taz and dtaptaz – destination tap's taz field
7) Each record has a isWalkTapPair, isPnrTapPair, isKnrTapPair field to indicate the type of path
8) Each record has the following path generalized cost attributes:
    a. Maz2tapUtils0 – origin MAZ to TAP utility alternative 0
    b. otapUtils0 – origin TAP utility alternative 0
    c. tap2tapUtils0 – origin TAP to destination TAP utility alternative 0
    d. dtapUtils0 – destination TAP utility alternative 0
    e. tap2MazUtils0 – destination TAP to destination MAZ utility alternative 0
    f. Maz2tapUtils1 – origin MAZ to TAP utility alternative 1 (if applicable)
    g. otapUtils1 – origin TAP utility alternative 1 (if applicable)
    h. tap2tapUtils1 – origin TAP to destination TAP utility alternative 1 (if applicable)
    i. dtapUtils1 – destination TAP utility alternative 1 (if applicable)

j. tap2MazUtils1 – destination TAP to destination MAZ utility alternative 1 (if applicable)

The event-tvpb.log file in the logFiles folder contains all the calculations for each potential tap pair for the debug record. The calculations for each sheet (MAZ2TAP, OTAP, TAP2TAP, DTAP, and TAP2MAZ) in the TVPB UEC are logged in order to easily compare the calculated values to the generalized cost settings.