July 2017

Activity-Based Model User Guide

Coordinated Travel – Regional Activity-Based Modeling Platform (CT-RAMP) for Atlanta Regional Commission

Atlanta Regional Commission
229 Peachtree St., NE, Suite 100
Atlanta, Georgia 30303

WSP | Parsons Brinckerhoff
3340 Peachtree Road NE
Suite 2400, Tower Place
Atlanta, GA 30326

Atkins
1600 RiverEdge Parkway
Atlanta, Georgia 30328

Table of Contents

1 Overview 1
1.1 Hardware and Software Prerequisites 2
1.2 Distributed Setup 3
2 System Setup and Design 6
3 Running Population Synthesizer 7
3.1 PUMS Data Tables Setup 7
3.2 Control Data Tables Setup 8
3.3 Control Totals 8
3.4 File Setup 9
3.5 PopSynIII Configuration File 11
3.6 Running PopSynIII 14
4 Running the Model 16
4.1 Setting Up a Run 16
4.1.1 Create a Project Folder 16
4.1.2 Start JPPF Services 17
4.1.3 Start Cube Cluster 19
4.1.4 Configure and Run the Model 19
4.2 Warm Start 22
4.3 GUI Settings 22
4.3.1 General Settings and Modules to Run 22
4.3.2 Feedback Modules 26
4.3.3 CT-RAMP Settings I 29
4.3.4 CT-RAMP Settings II 31
4.3.5 CT-RAMP Settings III 33
4.3.6 Model Summary Settings 35
5 Model EXEC Folder 36
6 Model Inputs 36
6.1 Input Files 36
6.2 Traffic Analysis Zone 38
6.3 Highway Networks 39
6.3.1 Network Attributes 39
6.3.2 Capacities 43
6.3.3 Speeds 44
6.3.4 Lane Coding 45
6.3.5 Toll Facilities 47
6.3.6 Prohibitions 48
6.4 Transit Networks/System 50
6.4.1 Transit Route Files 50
6.4.2 Transit System Data File 50
6.4.3 Transit Factor Files 51
6.4.4 Transit Fare File 52
6.5 Socioeconomic Data Files 54
6.6 Air Passenger Model 56
6.7 Externals 56

6.8 Commercial Vehicle and Truck Model 58
6.9 CT-RAMP Input Files 59
6.10 Miscellaneous File 61
7 Model Outputs 62
7.1 Highway Networks 62
7.2 Highway and Transit Skims 63
7.3 Air Passenger 66
7.4 Externals 66
7.5 Commercial Vehicle and Trucks 68
7.6 CT-RAMP Output Files 69
7.6.1 Shadow Pricing Results 69
7.6.2 Usual Work and School Location Choice Results 69
7.6.3 Auto Ownership Results 70
7.6.4 Household Attribution Results 70
7.6.5 Person Attribution Results 72
7.6.6 Coordinated Daily Activity Pattern (CDAP) Results 72
7.6.7 Individual Tour Records 72
7.6.8 Joint Tour Records 74
7.6.9 Individual Trip Records 76
7.6.10 Joint Trip Records 77
7.6.11 Trip Data 78
7.6.12 CT-RAMP Tables 79
7.7 Time-of-Day Trip Tables 80
7.8 Transit Assignment Output 81
8 Highway Network Coding 83
8.1 Cube Geodatabase 83
8.2 Mapping Attributes 86
8.3 Coding Projects 89
8.4 Notes on Link Attributes 96
8.5 Toll Optimization 97
9 Transit Network Coding 103
9.1 Non-Transit and Transit Modes 103
9.2 Transit Route File Structure 105
9.3 Transit Route Coding 106
9.3.1 Dwell Time Coding 113
9.3.2 Circular Route Coding 114
9.4 Transit Related Highway Network Coding 117
9.4.1 Station Coding 117
9.4.2 Transit Coding Highway Node Attributes 119
9.4.3 Fixed Guideway Coding 119
10 Shadow Pricing 120
10.1 Running Shadow Pricing 121
10.2 Issue with School Location Choice 122
11 Accessibility Calculator 123
11.1 Running the Calculator 123
11.2 Required Files 123
11.3 Output Files 124
12 Transit Project Level Forecast 125

12.1 No Build Scenario 125
12.2 Build Scenario 126
12.3 Step by Step Application 127
13 Activity-Based Model Visualization Tool (ABMVIZ) 128
13.1 Introduction to ABMVIZ 128
13.1.1 ABMVIZ v.2 128
13.1.2 Definitions and Acronyms 128
13.1.3 Scenario Explanations 129
13.2 Visuals 129
13.2.1 Persons Not At Home 129
13.2.2 Person Group, by Day Pattern 131
13.2.3 Trip Mode, by County 133
13.2.4 Time Use 135
13.2.5 Trip Mode 137
13.2.6 Radar Charts 138
Appendix 141

List of Tables

Table 1: System Setup for Running ARC ABM on the ModCalc Clusters 7
Table 2: PopSyn Database Tables 7
Table 3: ARC Population Synthesis Controls 9
Table 4: Working Directory Contents 9
Table 5: *runtime* Directory Contents 9
Table 6: *outputs* Directory Contents 10
Table 7: *scripts* Directory Contents 10
Table 8: *lib* Directory Contents 10
Table 9: *config* Directory Contents 10
Table 10: Database Connection Setting (<database> attribute) 11
Table 11: PUMS Table setting (<pumsData> attribute) 11
Table 12: Control Table Settings ([maz/taz/meta]ControlsTable attributes) 12
Table 13: Specifying Controls (<target>attribute) 13
Table 14: INPUTS Folder Required Files 36
Table 15: PARAMETERS Folder Required Files 37
Table 16: Zone Ranges by County 38
Table 17: Required Highway Network Attributes 40
Table 18: Recommended Highway Network Attributes 41
Table 19: Hourly Capacities 43
Table 20: Auxiliary Lane Capacities 43
Table 21: Free-Flow Speeds 44
Table 22: AM Congested Speeds – First Feedback Loop 45
Table 23: Time of Day Lanes Example 1 46
Table 24: Time of Day Lanes Example 2 46
Table 25: Time of Day Lanes Example 3 46
Table 26: Toll ID Database 48
Table 27: Population and Employment Input 54
Table 28: Household Size by Income Group Input 55
Table 29: University Enrollment 55
Table 30: Enplanements File 56
Table 31: Externals File 57
Table 32: External Counties File 57
Table 33: External Station Interstate Flag File 58
Table 34: External Station Passenger Car Friction Factors 58
Table 35: EE Passenger Car File 58
Table 36: EE Truck File 58
Table 37: Truck Zone Flag File 59
Table 38: Truck Delta Matrices Example 59
Table 39: Truck Model Friction Factors 59
Table 40: CT-RAMP Model Input Files 60
Table 41: I-285 Zone File 61
Table 42: List of Highway Networks 62
Table 43: Highway Network Added Attributes 63
Table 44: Highway and Transit Skim Files 64
Table 45: Highway Non-Toll Skim Tables 64

Table 46: Highway Toll Eligible Skim Tables 64
Table 47: Walk Access / Walk Egress Transit Skim Tables 65
Table 48: Drive Access or Egress Transit Skim Tables 65
Table 49: Air Passenger Vehicle Tables 66
Table 50: Air Passenger Transit Tables 66
Table 51: Internal-External Passenger Car Productions 66
Table 52: Internal-External Passenger Car Attractions 67
Table 53: Internal-External Passenger Car Trip Tables 67
Table 54: External-External Passenger Car Trip Table 67
Table 55: External Station Truck Trip File 67
Table 56: External-External Truck Trip Tables 68
Table 57: Initial Internal-Internal / Internal-External Truck Trip Table 68
Table 58: Final Truck Tables 68
Table 59: CT-RAMP Model Output Files 69
Table 60: Usual Work and School Location Choice Output File Fields 70
Table 61: Person Type Codes 70
Table 62: Household Output File Fields 71
Table 63: Person Output File Fields 72
Table 64: CDAP Output File Fields 72
Table 65: Individual Tours Output File Fields 73
Table 66: Trip/Tour Mode Codes 74
Table 67: Joint Tours Output File Fields 75
Table 68: Individual Trips Output File Fields 76
Table 69: Joint Trips Output File Fields 77
Table 70: CT-RAMP Trips Data Output File Fields 78
Table 71: Trip Table Format 80
Table 72: Passenger Car TOD Trip Table File 80
Table 73: TOD Trip Table Assignment File 81
Table 74: Transit Assignments 82
Table 75: Non-Transit Modes 103
Table 76: Transit Modes 104
Table 77: Files Needed to Run the Accessibility Calculator 123
Table 78: Accessibility Files Produced by the Accessibility Calculator 124
Table 79: Content of Accessibility Files 124
Table 80: ABMVIZ Definitions and Acronyms 128
Table 81: Scenario Explanations 129
Table 82: MAZ (ARC TAZ) Level Control Totals 141
Table 83: TAZ (ARC PECAS Zone) Level Control Totals 141
Table 84: Meta-Control (ARC County) Totals 142
Table 85: PUMS Processed Household table 143
Table 86: PUMS Processed Person Table 143
Table 87: Synthetic Population Household Table in Unexpanded Form 144
Table 88: Synthetic Population Person Table in Unexpanded Form 144
Table 89: Synthetic Population Household Table in Expanded Form 145
Table 90: Synthetic Population Person Table in Expanded Form 145

List of Figures

Figure 1: ARC ABM Model Structure 1
Figure 2: Disaggregate Travel Demand Model Software Components 3
Figure 3: JPPF Framework 5
Figure 4: ARC ABM System Design 6
Figure 5: Computer Usage for Node Runner in JConsole 18
Figure 6: Starting Cube Cluster Nodes 19
Figure 7: ARC ABM GUI 20
Figure 8: Execute Main Run Script 21
Figure 9: Additional Files Needed for a Warm Start Run 22
Figure 10: General Settings and Modules to Run 25
Figure 11: Model Preparation, Commercial Vehicle and Truck Model, Feedback Loops, Transit Assignment, and Post Process Assignments 27
Figure 12: CT-RAMP Settings 30
Figure 13: CTRAMP Models to Run and Distributed Model Run Settings 32
Figure 14: Other CT-RAMP Settings 33
Figure 15: Model Summary Settings 35
Figure 16: Traffic Analysis Zones 39
Figure 17: Toll ID Example 47
Figure 18: Prohibition Coding 49
Figure 19: System Data File Example Modes 50
Figure 20: System Data File Example Wait Curves 51
Figure 21: System Data File Example Operators 51
Figure 22: Transit Factor File Walk Access Example 52
Figure 23: Transit Fare File Example 53
Figure 24: Fare Structure Cube Help 53
Figure 25: Accessing Data Manager 83
Figure 26: Adding Data 84
Figure 27: Browse to Geodatabase 84
Figure 28: Data Manager with Geodatabase 84
Figure 29: Files Stored in Geodatabase 85
Figure 30: Table of Contents 85
Figure 31: Layer Properties 86
Figure 32: Highway Layer Properties Menu 87
Figure 33: Advanced Properties 87
Figure 34: Single Symbol Symbology 88
Figure 35: Categories Symbology 88
Figure 36: Selecting Layer for Editing 89
Figure 37: Enable Editing 89
Figure 38: Editor Functions 90
Figure 39: Feature Explorer 91
Figure 40: Save Edits 91
Figure 41: New Facility with Vertices 92
Figure 42: Feature Template 93
Figure 43: Two-Way vs. One-Way Editing 93
Figure 44: HOV Merge Coding 95

Figure 45: Loop Ramp Coding 96
Figure 46: Weave Flag 96
Figure 47: Toll Optimization Process 97
Figure 48: Network Coding for Toll Optimization 98
Figure 49: Toll Optimization Directory Structure 99
Figure 50: Period for Optimization 99
Figure 51: Optimization Loops 100
Figure 52: Toll IDs for Optimization 100
Figure 53: Sample Optimization Output File 102
Figure 54: New Tolls Sample File (Optimized Only) 102
Figure 55: New Toll Input File 102
Figure 56: Route Structure Text Format 105
Figure 57: Opening Route File 106
Figure 58: Route File TOC 107
Figure 59: Displaying Transit Lines 107
Figure 60: Selecting Transit Line for Display 107
Figure 61: Selected Transit Line for Editing 108
Figure 62: Route Attributes 108
Figure 63: Route Edit 109
Figure 64: Route Alignment 110
Figure 65: Route Modification 110
Figure 66: Exporting from Data Manager 111
Figure 67: Transit Line Layer Export 111
Figure 68: Importing from Data Manager 112
Figure 69: Transit Line Layer Import 112
Figure 70: Dwell Time Coding 1 113
Figure 71: Dwell Time Coding 2 114
Figure 72: Circular Coding Example 1 115
Figure 73: Circular Coding Example 2 116
Figure 74: Circular Coding Example 3 116
Figure 75: Station Coding Example 118
Figure 76: Station Aerial 118
Figure 77: Node Attributes 119
Figure 78: Java Node Window 126
Figure 79: Locations of the interactive features of the Persons Not At Home visual 130
Figure 80: Locations of the interactive features of the Person Group, by Day Pattern visual 131
Figure 81: Result of hovering over a bar on the Person Group, by Day Pattern visual 132
Figure 82: Locations of the interactive features of the Trip Mode, by County visual 133
Figure 83: Result of hovering over a bar on the Trip Mode, by County visual 134
Figure 84: Locations of the interactive features of the Time Use visual 135
Figure 85: Results of hovering over the Time Use visual 136
Figure 86: Locations of the interactives features of the Trip Mode visual 137
Figure 87: Result of hovering over a ring on the Trip Mode visual 138
Figure 88: Locations of the interactive features of the Radar Charts visual 139
Figure 89: Result of hovering over the Radar Charts visual 139

1 Overview

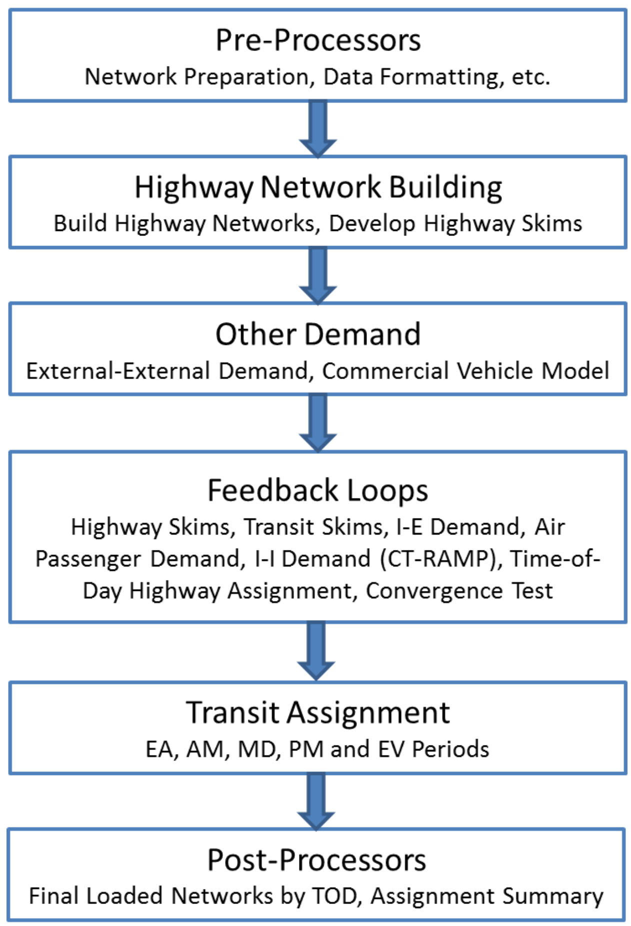
The Atlanta Regional Commission (ARC) Activity-Based Model (ABM) has been developed to ensure that the regional transportation planning process can rely on forecasting tools that will be adequate for new socioeconomic environments and emerging planning challenges. It is also equally suitable for conventional highway projects, transit projects, and various policy studies such as highway pricing and HOV analysis.

The ARC 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 ARC.

The ARC ABM model is implemented in Cube and Java and runs in a distributed fashion on three Windows 64-bit machines. The overall model structure is similar to the existing trip-based model. The primary difference between the trip-based model and the ABM is that the trip-based internal-internal (I-I) demand models were replaced with tour-based microsimulation models. The micro-simulated trip lists are converted to demand matrices and assigned just like before. **Figure 1** illustrates the overall model structure.

The purpose of this User Guide is to describe the modeling system setup, how to setup and run the model, the model inputs and the model outputs. Refer to the Activity-Based Model Specification Report and the Activity-Based Model Calibration Report for details about model structure, model coefficients, etc.

Figure 1: ARC ABM Model Structure



1.1 Hardware and Software Prerequisites

The machines used to run ARC ABM need to be set up with the following software and hardware configuration:

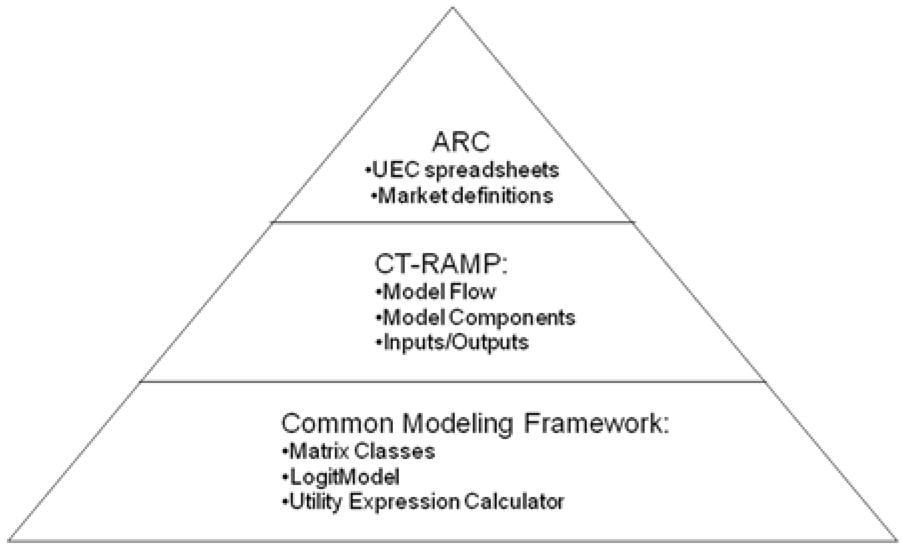
1) Equipped with at least 128 GB of RAM and running 64-bit Windows OS.
2) Installed with 64-bit Java Development Kit 1.7 or later.
3) Installed with Cube Voyager 6.1 or later, with at least one installation with at least one seat Cube Cluster license[1].
4) Installed with VoyagerFileAPIInstaller.msi, which places 64-bit Voyager matrix access API files under the *Citilabs\VoyagerFileAPI* directory.
5) Include *C:\Program Files\Citilabs\VoyagerFileAPI* in the system %PATH%.
6) Installed with Microsoft Visual C++ 2012 (x64) redistributable package (downloadable from *http://www.microsoft.com/en-us/download/details.aspx?id=30679*).

The ABM requires a 64-bit OS in order to take advantage of larger (64-bit) memory addresses. Additional 64-bit API files provided by Citilabs are required on the machines to natively read binary Cube matrices because Cube is currently a 32-bit application. Additional library files are also shipped with the CT-RAMP package for the same reason.

As shown in **Figure 2**, the CT-RAMP software for the microsimulation components of the model, which has been co-developed for Metropolitan Transportation Commission for San Francisco Bay Area, relies on the Common Modeling Framework (CMF), a collection of Java libraries specifically designed for the implementation of disaggregate travel demand models. The ARC 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 ARC are contained in Utility Expression Calculator (UEC) files. These spreadsheet-based files open up the models so that the parameters, input filenames, 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 ARC ABM setup.

[1] If reinstalling Cube, remove all "Citilabs" folders found on the system in addition to running "Uninstall".

Figure 2: Disaggregate Travel Demand Model Software Components



1.2 Distributed Setup

The starting point for the ARC ABM was the ARC trip-based model. The trip-based model's internal-internal demand model was replaced by CT-RAMP, while the other components were used as is or updated as needed. The other components include network processing, a commercial vehicle model, an air passenger model, an external model, a time-of-day model, transit network building, and highway and transit assignment and skimming. All of these models were threaded with Cube Cluster's *DistributeINTRAStep* and *DistributeMULTIStep* commands in order to improve runtimes.

The first type of threading/distribution threaded all calculations in an origin zone loop using Cube Cluster's *DistributeINTRAStep* command, which distributes the calculations in blocks of origin zones to waiting Cube Cluster processes. When Cube Cluster completes, it writes an end text file and the main Cube process reads this file and continues to the next step. This type of threading is independent of the number of processes available and is flexible for adding/removing processes. Distribution by origin zone loop is allowed for threading highway assignments and matrix processing.

The second type of threading/distribution was performed using Cube Cluster's *DistributeMULTIStep* command, which essentially distributes code blocks across multiple processes and then waits for all of them to complete before continuing. Unlike the first type of threading/distribution, this type requires explicitly assigning the tasks to specific processes, thereby being less flexible to adding/removing processes. This was implemented for highway assignments by time-of-day, conversion of trip lists to time-of-day matrices, and transit assignments by access mode, local/premium, and time-of-day.

CT-RAMP, the activity-based microsimulation component of the model, is also distributed across the three machines. The approach allows the utilization of one or more computers, each with multiple computing cores, and efficiently balances the computation among the computers. 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.

CT-RAMP threads the application of activity-based choice models to groups of households. 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 **Figure 3**, the JPPF framework consists of three 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.

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 32 GB and threads equal to 8 (for an 8 core machine). The majority of parallel computations in the CT-RAMP implementation occur through tasks executed in parallel on nodes.

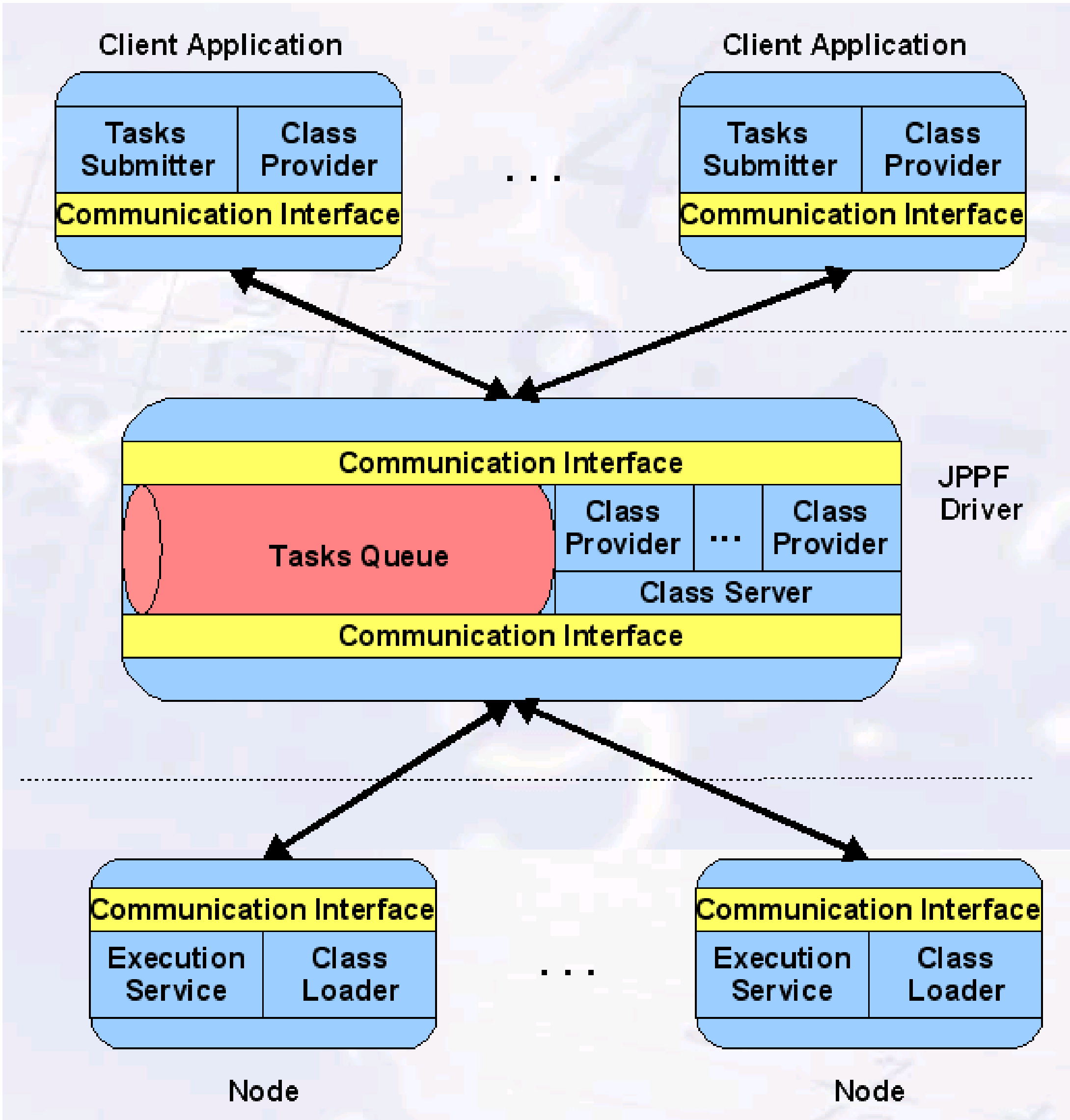
The driver process uses logic contained in the JPPF framework to balance computational loads across Java Virtual Machines running on the nodes in the cluster. The driver receives tasks from the client application and submits them in bundles to the nodes. The driver also retrieves class files from the 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 Cube model script and configured through a *properties* file dynamically written by the model GUI, 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. 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 that manages the household and person synthetic population in memory and provides the JPPF nodes with all household and person related data. 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.

The remote node processes are started and terminated manually in the current setup.

Figure 3: JPPF Framework



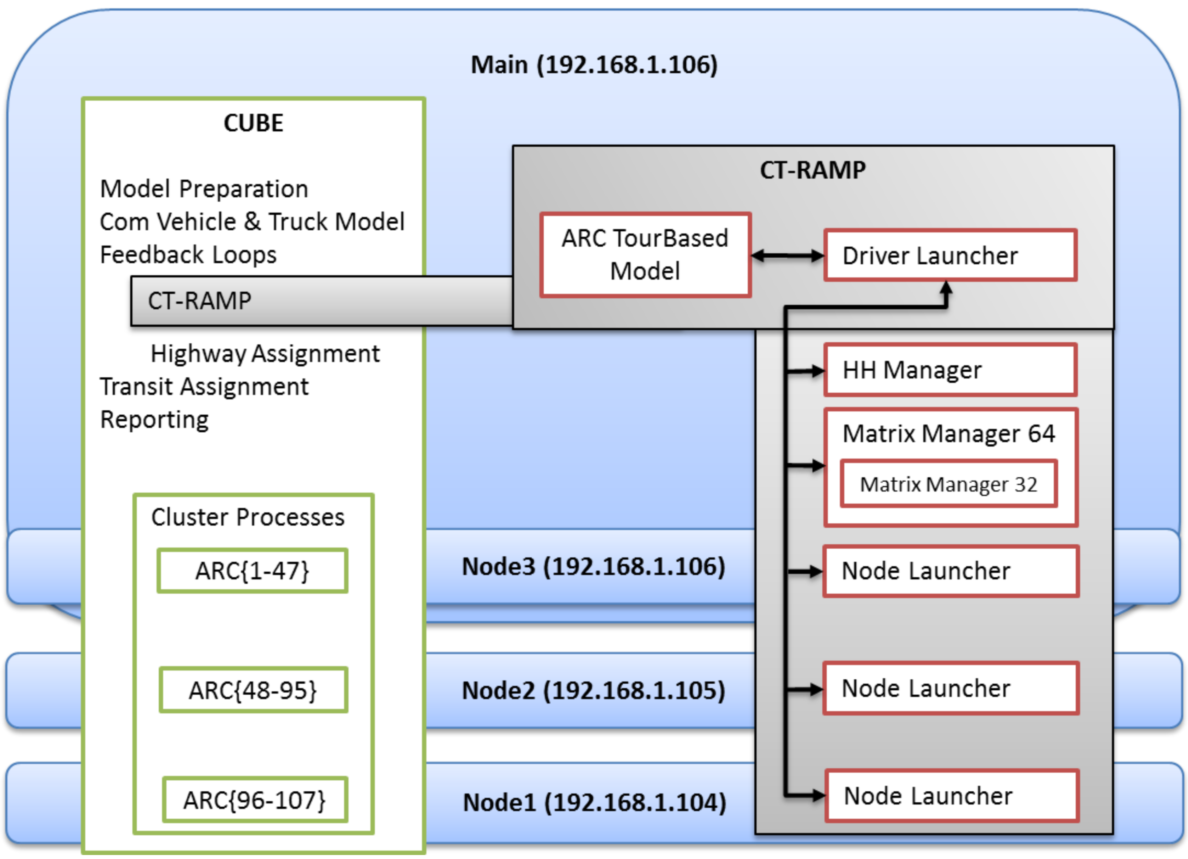
2 System Setup and Design

The ABM system is implemented with Java and Cube, with JPPF and Cube Cluster for distributed and threaded computing respectively. As depicted in **Figure 4**, the ARC system design consists of:

1) A main computer which:
   a. runs the main Cube work, including some of the Cube Cluster processes,
   b. runs the main CT-RAMP (*ArcTourBasedModel*) Java process,
   c. runs the Household Manager Java process,
   d. runs the JPPF Driver Launcher processes which is called by *ArcTourBasedModel*, and which manages communicating with the nodes, and
   e. optionally runs a JPPF node process which listens for tasks from the JPPF driver.

2) Additional node computers which:
   a. listen for tasks from the JPPF driver and
   b. listen for Cube Cluster tasks.

Figure 4: ARC ABM System Design



The current setup involves using ARC's ModCalc6 server as the main computer and ModCalc4, ModCalc5 and ModCalc6 servers as the node computers. Currently, the Cube Cluster processes and JPPF nodes are allocated on these computers as following:

Table 1: System Setup for Running ARC ABM on the ModCalc Clusters

Computer	Cube Cluster Processes	JPPF Nodes	No. Threads for CT-RAMP
ModCalc4	ARC {96-107}	Node1	24
ModCalc5	ARC {48-95}	Node2	96
ModCalc6	ARC {1-47}	Node3	96

3 Running Population Synthesizer

The new population synthesizer software, called PopSyn III, is a Java-based program that uses Microsoft SQL Server 2008 R2 or later database. It requires Census PUMS household and person data that is read from the database. The final synthetic population calculated by the software is written into the database as well. All the data tables are maintained in one database called *ARCPopSyn*. The database tables are described in **Table 2**. The details of each table can be found in the **Appendix**.

Table 2: PopSyn Database Tables

Schema	Name	Description
dbo	psam_h13	2007-2011 ACS 5-year PUMS Household table for the region
dbo	psam_p13	2007-2011 ACS 5-year PUMS Person table for the region
dbo	hhtable	HH input table processed from PUMS HH table
dbo	perstable	Person input table processed from PUMS HH table
dbo & scenario	control_totals_maz	Table with different control totals at MAZ level
dbo & scenario	control_totals_taz	Table with different control totals at TAZ level
dbo & scenario	control_totals_meta	Table with different meta-control totals at meta level
dbo & scenario	households	Synthetic population households output table
dbo & scenario	persons	Synthetic population persons output table

3.1 PUMS Data Tables Setup

The first step to setup the database is to load all the Census ACS tables. The user should download the 2007-2011 5 year ACS household and person csv data file for Georgia:

- Household file: http://www2.census.gov/acs2011_5yr/pums/csv_hga.zip
- Person file: http://www2.census.gov/acs2011_5yr/pums/csv_pga.zip

The user also needs the file *socPECASCwlk.csv*, which is a crosswalk between the SOC and PECAS occupation categories, in order to prepare the PUMS tables. Once the user has extracted the PUMS files from the archive, the user must run the *PUMS Table creation.sql* script to create the *hhtable* and *perstable* tables. This script only needs to be run once. This script depends on the CSV file names specified in the *csv_filenames* table created by the run batch file described later. It can also be run with the run batch file.

3.2 Control Data Tables Setup

In order to setup the control tables, the following input files are required:

- *hshld{year}g.dat*: Number of households segmented by income and household size for each TAZ.
- *workerPercentages.csv*: Households by number of workers as a percentage for each household size income group segment.
- *personByAge.csv*: Persons by age group for each county in the model area.
- *LaborMakeAndUse.csv*: Amount of labor make/use by occupation by PECAS zone.
- *avgWagePUMS00.csv*: Average wage by occupation and household type as per 2000 PUMS
- *geographicCwalk.csv*: The geographic correspondence between MAZ, TAZ, PUMA (Census 2000 definitions) and META geographies.

The *importControls.SQL* script run by the run batch file setups up the control tables: *control_totals_maz*, *control_total_taz,* and *control_totals_meta*.

3.3 Control Totals

The new version of the PopSyn matches both household and person distributions, as well as controls at multiple levels of geography. There are three main levels of geography at which controls can be set – microzones (MAZ), traffic analysis zones (TAZ), and district. For ARC, these correspond to TAZs, PECAS zones, and County. Controls at the district-level (ARC County) are known as meta-controls. In addition, PUMAs must nest within meta-geographies. As a result, some of the counties were grouped since each meta-geography must be at least as big as a PUMA. The controls being used for ARC are summarized in **Table 3** below.

The raw PUMS data needs to be processed to generate the required control attributes. The household and person datasets are read into a temporary table. All group quarter (GQ) records are dropped from the datasets. Using employment status attributes in the person database, the number of workers is assigned to each household. The SOC code is extracted for each person and *socPECASCwlk.csv* crosswalk is used to generate the occupation category.

The ARC PopSyn uses the persons by occupation at the PECAS zone level as a control. In order to map that data into PopSyn, the PECAS total wages by occupation by PECAS zone are divided by the average wage by occupation from Census 2000 to create the total persons by occupation by PECAS zone. The occupation totals thus generated are scaled to match the number of households by numbers of worker controls defined earlier for the MAZ geography.

Table 3: ARC Population Synthesis Controls

Control	Categories	Geography	Data Source
Number of HHs	N/A	Region	ARC socio-economic forecast
Number of HH by income	0-25k, 25k-60k, 60k-120k, 120k+	MAZ (ARC TAZ)	ARC socio-economic forecast
Number of HH by HH size	1,2,3,4,5,6+	MAZ (ARC TAZ)	ARC socio-economic forecast
Number of HH by workers	0,1,2,3+	MAZ (ARC TAZ)	ARC workers per household shares from trip-based model
Number of persons by age	0-14, 15-24, 25-34, 35-44, 45-54, 55-64, 65-74, 75-84, 85+	District (ARC County group)	ARC socio-economic forecast
Number of persons by occupation	CL23WhiteCollar, CL24Services, CL25Health, CL26Retail, CL27BlueCollar	TAZ (ARC PECAS zone)	PECAS model

3.4 File Setup

PopSyn III is configured to run using the batch file runPopSynIII.bat. The files that are critical for the population synthesizer are all housed in the directory *runtime.* Both the batch file and the *runtime* directory should also be present in the working directory at the time of program execution. A brief description of the file directory setup follows.

- *Working directory –*

Table 4: Working Directory Contents

File	Description
runPopSynIII.bat	Batch file for running PopSynIII
/runtime	Sub-directory with all critical files
/outputs	Sub-directory with log file and scenario specific output

- *runtime*

Table 5: *runtime* Directory Contents

File	Description
popsyn3Unsigned.jar	Java archive containing PopSynIII source code
common-base.jar	Java archive contains common modeling framework (CMF) code
/lib	Sub-directory with external libraries
/config	Sub-directory with configuration files

- *outputs*

Table 6: *outputs* Directory Contents

File	Description
event.log	Event log file containing details of the latest run
persons.csv	Synthesized person CSV file
households.csv	Synthesized household CSV file

- *scripts*

Table 7: *scripts* Directory Contents

File	Description
importControls.sql	Import and process ARC control data
PUMS Table creation.sql	Import and process PUMS data
Outputs.sql	Create expanded person and household tables

- *runtime/lib* – This sub-directory has all the necessary external dependencies for PopSynIII. The details are tabulated below:

Table 8: *lib* Directory Contents

File	Description
sqljdbc4.jar	Microsoft Java Database Connectivity driver
com.google.ortools.linearsolver.jar	Linear programming solver – Java libraries
sqljdbc_auth.dll	SQL Server "Windows authentication" dependency
jnilinearsolver.dll	Linear programming solver dependency
/JPFF-3.2.2	JPFF libraries for distributed setup (not yet implemented)

- *runtime/config*

Table 9: *config* Directory Contents

File	Description
log4j.xml	Configuration file for java logging library
settings.xml	PopSynIII configuration file
jppf-clientLocal.properties	JPPF configuration file

3.5 PopSynIII Configuration File

Using the file *settings.xml*, the user can configure the database connection settings as well as specify database attributes that are to be used for balancing the controls. The settings available to the user are discussed below:

Table 10: Database Connection Setting (<database> **attribute)**

ATTRIBUTE	DESCRIPTION
<server>	Name of the server where the database is stored and the TCP/IP port number for connecting the client. For Microsoft SQL Server, Windows sets the default port to 1433.
<type>	Specifies the database engine being used. Currently PopSynIII allows the user to specify: MS_SQL – Microsoft SQL Server My_SQL – MySql
<user>	Username for logging into the database server. Comment out if Windows authentication is being used.
<password>	Password for the user specified above. Leave it empty if Windows authentication is being used.
<dbName>	Name of the database where all the relevant tables are stored.

XML Instance

```xml
<database>
  <server>W-AMPDX-D-TFS05:1433</server>
  <type>MS_SQL</type>
  <!--<user></user>-->
  <password></password>
  <dbName>ARCPopSyn</dbName>
</database>
```

Table 11: PUMS Table setting (<pumsData> **attribute)**

ATTRIBUTE	DESCRIPTION
<idField>	User generated unique ID for households in the PUMS table
<pumaFieldName>	The attribute in control tables which identifies the PUMA
<metaFieldName>	The attribute in control tables which identifies the meta geography
<tazFieldName>	The attribute in control table which identifies the TAZ
<mazFieldName>	The attribute in control table which identifies the MAZ
<weightField>	Weight field to be used in the PUMS tables as initial weights
<hhTable>	Name of the processed PUMS household table
<pumsHhIdField>	PUMS HH ID field name
<persTable>	Name of the processed PUMS persons table
<pumsHhTable>	Name of the processed PUMS household table
<pumsPersTable>	Name of the processed PUMS persons table
<maxExpansionFactor>	Maximum HH expansion factor weight setting
<synpopOutputHhTableName>	Name of synthesized households table (synpop_hh)

<synpopOutputPersTableName>	Name of synthesized persons table (synpop_per)
<outputHhAttributes>	PUMS HH Attributes to write out for the synthesized household
<outputPersAttributes>	PUMS Person Attributes to write out for persons in synthesized HHs

XML Instance

```xml
<pumsData>
  <idField>            hhnum       </idField>
  <pumaFieldName>          PUMA        </pumaFieldName>
  <metaFieldName>          DISTRICT     </metaFieldName>
  <tazFieldName>           TAZ         </tazFieldName>
  <mazFieldName>           MAZ          </mazFieldName>
  <weightField>            WGTP        </weightField>
  <hhTable>                hhtable      </hhTable>
  <persTable>              perstable    </persTable>
  <pumsHhTable>            hhtable      </pumsHhTable>
  <pumsHhIdField>          hhnum        </pumsHhIdField>
  <pumsPersTable>          perstable    </pumsPersTable>
  <maxExpansionFactor>       3          </maxExpansionFactor>
  <synpopOutputHhTableName>  synpop_hh    </synpopOutputHhTableName>
  <synpopOutputPersTableName> synpop_person </synpopOutputPersTableName>
  <outputHhAttributes>       serialno, np, nwrkrs_esr, hincp, adjinc, veh, hht, bld,
   type </outputHhAttributes>
  <outputPersAttributes>       sporder, agep, employed, pecasOcc, sex, esr, wkw, wkhp,
   mil, schg, schl, indp02, indp07, occp02, occp10 </outputPersAttributes>
</pumsData>
```

Table 12: Control Table Settings ([maz/taz/meta]ControlsTable **attributes)**

ATTRIBUTE	DESCRIPTION
<table_name>	Name of the table where the [MAZ/TAZ/META] level controls are stored.
<id_field_name>	The attribute in control table that identifies [MAZ/TAZ/META]
<aggregation_level>	The geographic level to aggregate the data to. MAZ = MAZ, TAZ = TAZ, Meta = PUMA

XML Instance

```xml
 <mazControlsTable>
  <mazTable id="1">
    <table_name>control_totals_maz</table_name>
    <id_field_name>MAZ</id_field_name>
    <aggregation_level>MAZ</aggregation_level>
  </mazTable>
 </mazControlsTable>
```

Table 13: Specifying Controls (<target>**attribute)**

ATTRIBUTE	DESCRIPTION
<marginals>	
<id>	The control ID. IDs are from 0 to the number of controls.
<description>	The description of the control being configured.
<totalHouseholdsControl>	This attribute needs to be made "*true*" when configuring the file for controlling the total households at the regional level. For all other controls this attribute does not appear in the configuration.
<controlType>	♦ *simple*: Comparison to the control total is just a simple check to see if the number of synthesized records matches the control. All HH controls are *simple* controls. ♦ *count:* Comparison to the control totals involves counting up the number of matching person records from synthesized HHs and ensuring consistency with the control totals. All person controls are *count* controls.
<geographyType>	The geography at which the control has been specified. [MAZ/TAZ/META]
<table>	The seed table
<constraint>	
<importance>	Weights to adjust the importance of the control
<field>	Attribute in the PUMS table that specifies the initial weights.
<controlField>	The attribute in the PUMS table that corresponds to the control being set.
<type>	Depending on nature of comparison being carried out a household/person record will qualify into a control category. The comparison types are: ♦ *interval* – If the values in the controlField needs to be compared to a range in order to qualify into the control category. ♦ *equality* – If the values in the controlField needs to be equal to a particular value to qualify into the control category.
If <type> is *interval*	
<lo_value>	Lower bound of the range that defines the control category
<lo_type>	♦ *closed* – if the range includes lo_value ♦ *open* – if the range does not include lo_value
<hi_value>	Upper bound of the range that defines the control category
<hi_type>	♦ *closed* – if the range includes hi_value ♦ *open* – if the range does not include hi_value
If <type> is *equality*	
<value>	Value that defines the control category

Table 13 (Cont'd): Specifying Controls (<target>attribute)

XML Instance
``<!-- Defining a 2 category control for Number of HHs by Persons per HH at MAZ level -->``` ```<!-- Category 1: One person household <type> equality-->``` ```<!-- Category 2: 2+ person household <type> interval-->``` ```<target>``` ```<marginals>``` ```<id>2</id>``` ```<description>Number of HHs by Persons per HH Category (1, 2+)</description>``` ```<geographyType>MAZ</geographyType>``` ```<controlType>simple</controlType>``` ```<table>hhtable</table>``` ```<constraint id="1">``` ```<importance>1000</importance>``` ```<field>NP</field>``` ```<controlField>HHSIZE1</controlField>``` ```<type>equality</type>``` ```<value>1</value>``` ```</constraint>``` ```<constraint id="2">``` ```<importance>1000</importance>``` ```<field>NP</field>``` ```<controlField>HHSIZE2</controlField>``` ```<type>interval</type>``` ```<lo_value>2</lo_value>``` ```<lo_type>closed</lo_type>``` ```<hi_value>infinity</hi_value>``` ```<hi_type>open</hi_type>``` ```</constraint>``` ```</marginals>``` ```</target>```

3.6 Running PopSynIII

PopSyn III is run with the *runPopSynIII.bat* file. The batch file first creates the control tables, runs PopSyn III, copies the outputs to the scenario schema, and creates the household and persons CSV files for use in the model. The batch file can also be used to import the PUMS data. The batch file requires the following parameters:

```
SET SCENARIO                    =      BaseYear
SET SQLSERVER                   =      PRODDB
SET DATABASE                    =      ARCPopSyn
SET MY_PATH                     =      %CD%

SET tazData_File                =      %MY_PATH%\data\HSHLD10G.txt'
SET personsByAge_File           =      %MY_PATH%\data\personsByAge.csv'
SET ageScaleFactors_File        =      %MY_PATH%\data\ageScalingFactors.csv'
SET makeUse_File                =      %MY_PATH%\data\LaborMakeAndUse.csv'
SET pumsOccDist_File            =      %MY_PATH%\data\persByOccpCountyDist.csv'

SET avgWage_File                =      %MY_PATH%\data\avgWagePUMS00.csv'
SET workerPercentage_File       =      %MY_PATH%\data\workerPercentages.csv'

SET geographicCWalk_File        =      %MY_PATH%\data\geographicCwalk.csv'
SET pumsHH_File                 =      %MY_PATH%\data\ss11hga.csv'
SET pumsPersons_File            =      %MY_PATH%\data\ss11pga.csv'
SET socPECASCwalk_File          =      %MY_PATH%\data\socPECASCwlk.csv'
```

The *SCENARIO* parameter is used to create a schema of the same name within the database and all the key input and output tables for the scenario gets stored in the schema. For instance, if *SCENARIO* is set to *BaseYear*, the following tables are created:

BaseYear.control_totals_maz
BaseYear.control_totals_taz
BaseYear.control_totals_meta
BaseYear.households
BaseYear.persons

The *SQLSERVER* and *DATABASE* parameters refer to the database location, and the *MY_PATH* parameter identifies the working folder. The rest of the parameters refer to file locations and are copied into the *csv_filenames* table in the database for use by various SQL scripts.

The DOS bat file calls the PopSyn III Java program via the command line:

```
%JAVA_64_PATH%\bin\java -showversion -server –Xms8000m –Xmx15000m
-cp "%CLASSPATH%" -Djppf.config=jppf-clientLocal.properties
-Djava.library.path= %LIBPATH% popGenerator.PopGenerator
runtime/config/setting.xml
```

In addition to the output tables in the database, the synthesized household and person tables described in the **Appendix** are saved as CSV files in the output directory.

To run a new scenario, including future year scenarios, the following inputs need to be adjusted: *SCENARIO, tazData_File, personsByAge_File,* and *makeUse_File*. The *avgWage_File* and *workerPercentage_File* inputs may be adjusted as well if data is available.

4 Running the Model

The main model run script starts all required Java and Cube processes on the remote machines. The main run script is executed in Cube. All the model components on each machine communicate to one project directory on the network. All inputs are read, and all outputs are written to this folder, thereby simplifying model setup, inspection, and error detection.

4.1 Setting Up a Run

To set up a model run:

1) Create a project folder,
2) Start JPPF services,
3) Start Cube Cluster, and
4) Configure and run the model in Cube Base.

4.1.1 Create a Project Folder

1) Create a root folder on the main computer (ModCalc6) that will house scenarios. For example:
   C:\ ARCTourBasedModel

2) Create a scenario specific project directory inside the root folder. For example:
   C:\ ARCTourBasedModel\Base2010

3) Unzip the contents of the base year template scenario to the project folder created above (*ARCTourBasedModel_Main.zip*). This zip file, which is versioned by a date stamp, contains all the required files to setup and run the base year model. The resulting folders and files in the project directory should be:
   a. *config* folder – contains configuration files for setting up and running JPPF distributed task processes. Also contains JPPF java code files (jar files).
   b. *ctrampModels* folder – contains all the model utility expression calculator (UEC) model description files.
   c. *Exec* folder – contains the CT-RAMP ARC.jar java code file.
   d. *Inputs* folder – contains model input files such as the population and employment data by TAZ, the person and household population synthesizer files and other model data inputs.
   e. *Parameters* folder – contains parameter files used by the non-CT-RAMP Cube components of the model.
   f. *ARCTourBasedModel_{date}.s* – the overall run script/GUI for running the model.
   g. *Accessibilities.properties* – contains property settings for Accessibility Calculator.

4) On each remote computer (ModCalc4, ModCalc5, and ModCalc6), create a project folder with the same path and file name as the one on the main machine. For example:
   C:\ ARCTourBasedModel\Base2010

5) Unzip the contents of *ARCTourBasedModel_Node.zip* to the project folder created on each remote computer.

6) On the main computer, map drives Y, Z and T to the root folder on the remote computers, as follows:

   Y:\ = \\ModCalc4\ARCTourBasedModel
   Z:\ = \\ModCalc5\ARCTourBasedModel
   T:\ = \\ModCalc6\ARCTourBasedModel

   Note that these drive names are embedded in the Cube script to copy necessary input files during runtime from the main computer to the remote computers.

4.1.2 Start JPPF Services

After creating the project folder and copying over the template files, the JPPF services may need to be configured. If necessary, edit the following properties files in the *config* folder:

1) Main Computer DOS Command File - runMain.cmd
   a. set *JAVA_PATH=C:/Program Files/Java/jdk1.7.0_45* #JDK 64bit path
   b. set *HOST_IP=192.168.1.106* #IP of main computer

2) Node DOS Command File - *runNode1.cmd*, *runNode2.cmd*, *runNode3.cmd*
   a. set *JAVA_PATH=C:/Program Files/Java/jdk1.7.0_45* #JDK 64bit path

3) JPPF Client Driver Properties File - *jppf-clientDistributed.properties*
   a. *driver1.jppf.server.host = 192.168.1.106* #IP of main computer

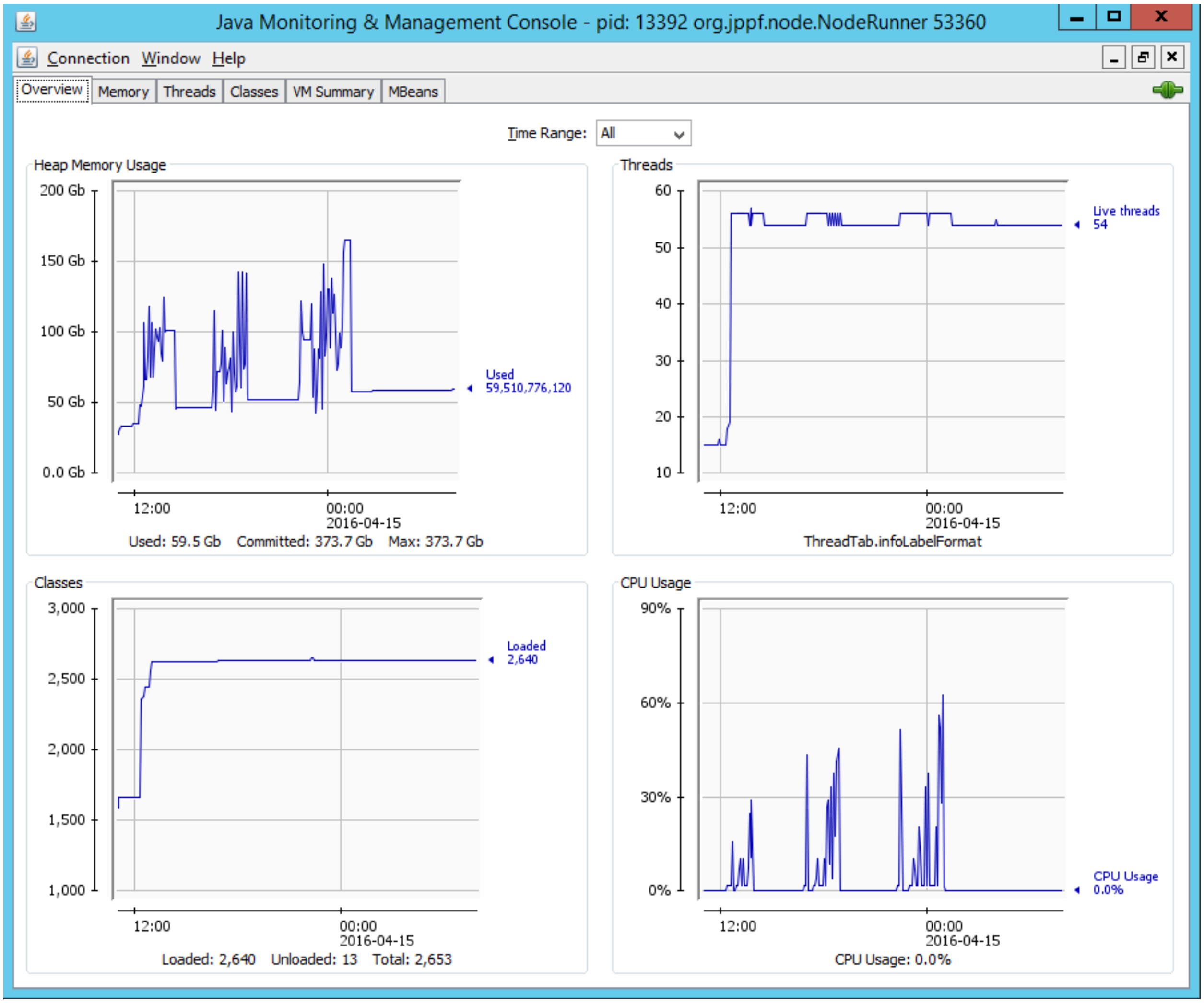
4) JPPF Driver Properties File - *jppf-driver.properties*
   a. *jppf.server.host = 192.168.1.106* #IP of main computer

5) Remote JPPF Node Properties Files - *jppf-node1.properties*, *jppf-node2.properties*, *jppf-node3.properties*
   a. *jppf.server.host = 192.168.1.106* #IP of main computer
   b. *processing.threads = 16* #number of computing cores on node
   c. *other.jvm.options = -Xms100g –Xmx100g* #memory on node

Once the JPPF properties files are configured, start the JPPF Driver Launcher and the HH Manager on the main computer (ModCalc6) by running the *runMain.cmd* file. Next, run the DOS files *runNode1.cmd*, *runNode2.cmd* and *runNode3.cmd* on ModCalc4, ModCalc5 and ModCalc6, respectively, to start the JPPF Node on the worker machines.

As an option, start jConsole sessions to track computer usage of the JPPF services. In the *config* folder is the *runJConsole.cmd* file that starts a JConsole session which requires the user to select a Java process. Running a JConsole session can be useful for troubleshooting memory problems.

Figure 5: Computer Usage for Node Runner in JConsole



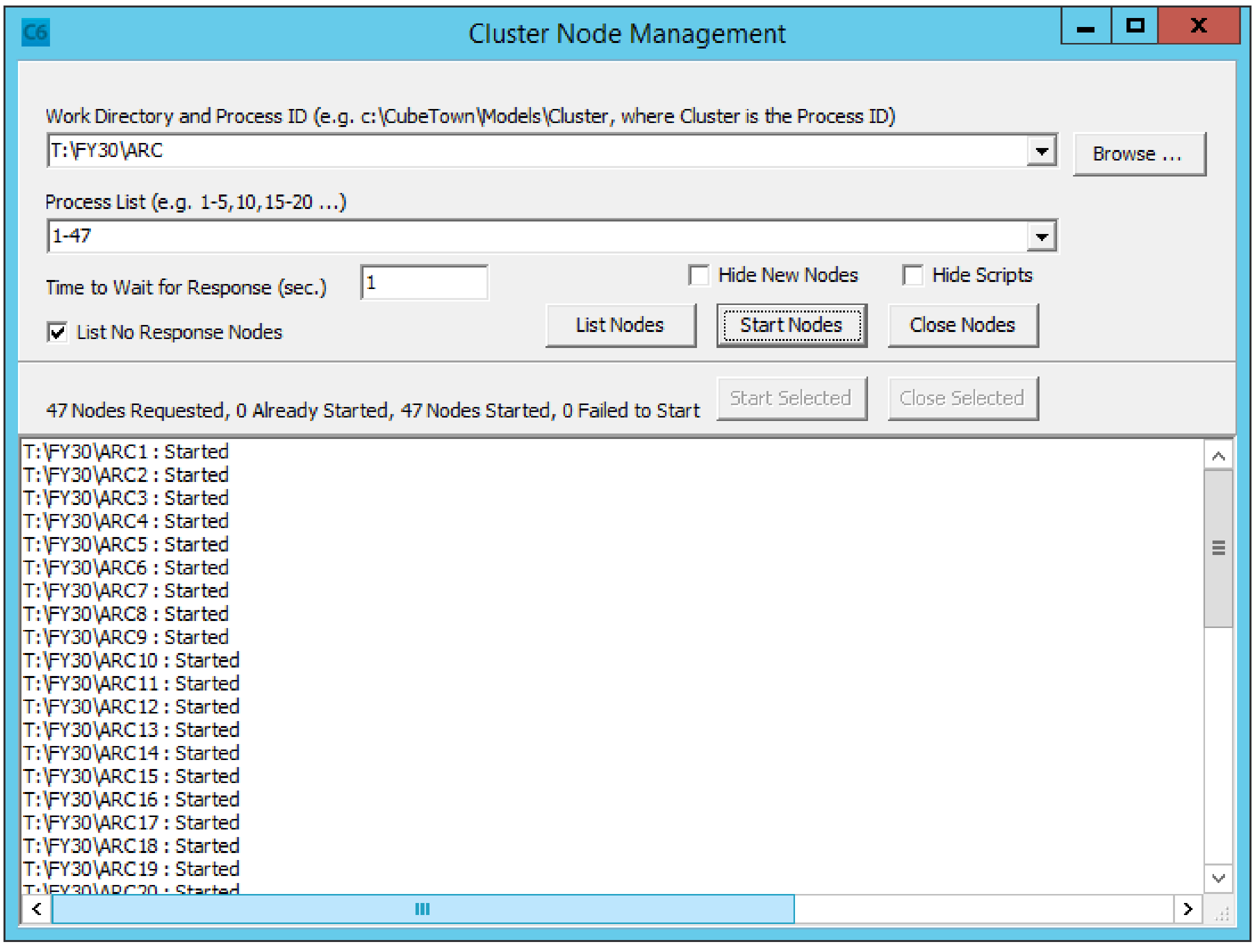
4.1.3 Start Cube Cluster

On the main and the node computers, map drive T to the root folder on the main computer:

T:\ = \\ModCalc6\ARCTourBasedModel

Start Cube on all the computers and start the Cube Cluster nodes through Cube's Cluster Node Management dialog box using the recommended setting outlined in **Table 1**. Set the work directory to the project directory on the T drive.

Figure 6: Starting Cube Cluster Nodes

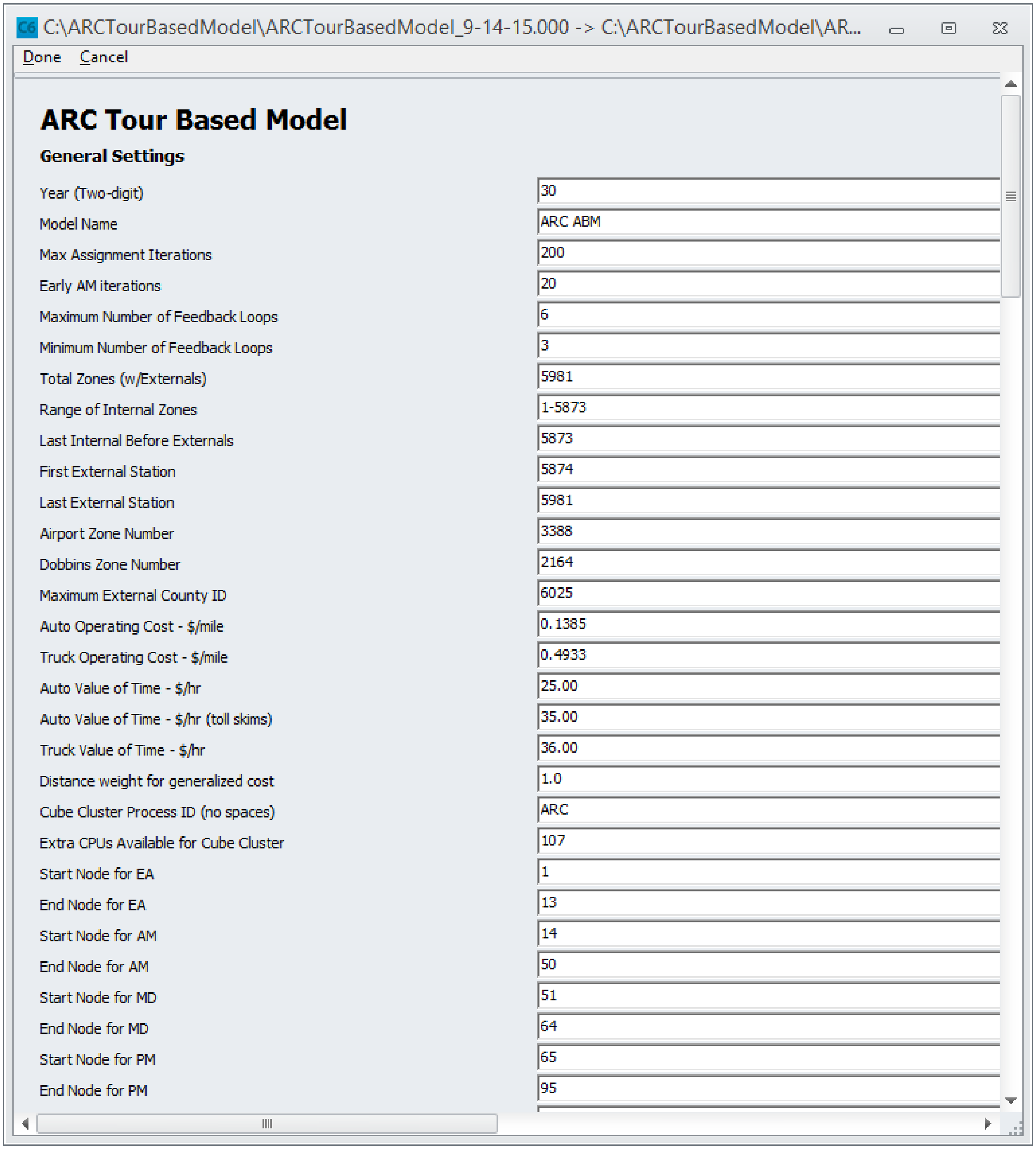


4.1.4 Configure and Run the Model

To run the main model script (*ARCTourBasedModel_{date}.s*), which creates a TP+ macro processor GUI in Cube, do the following:

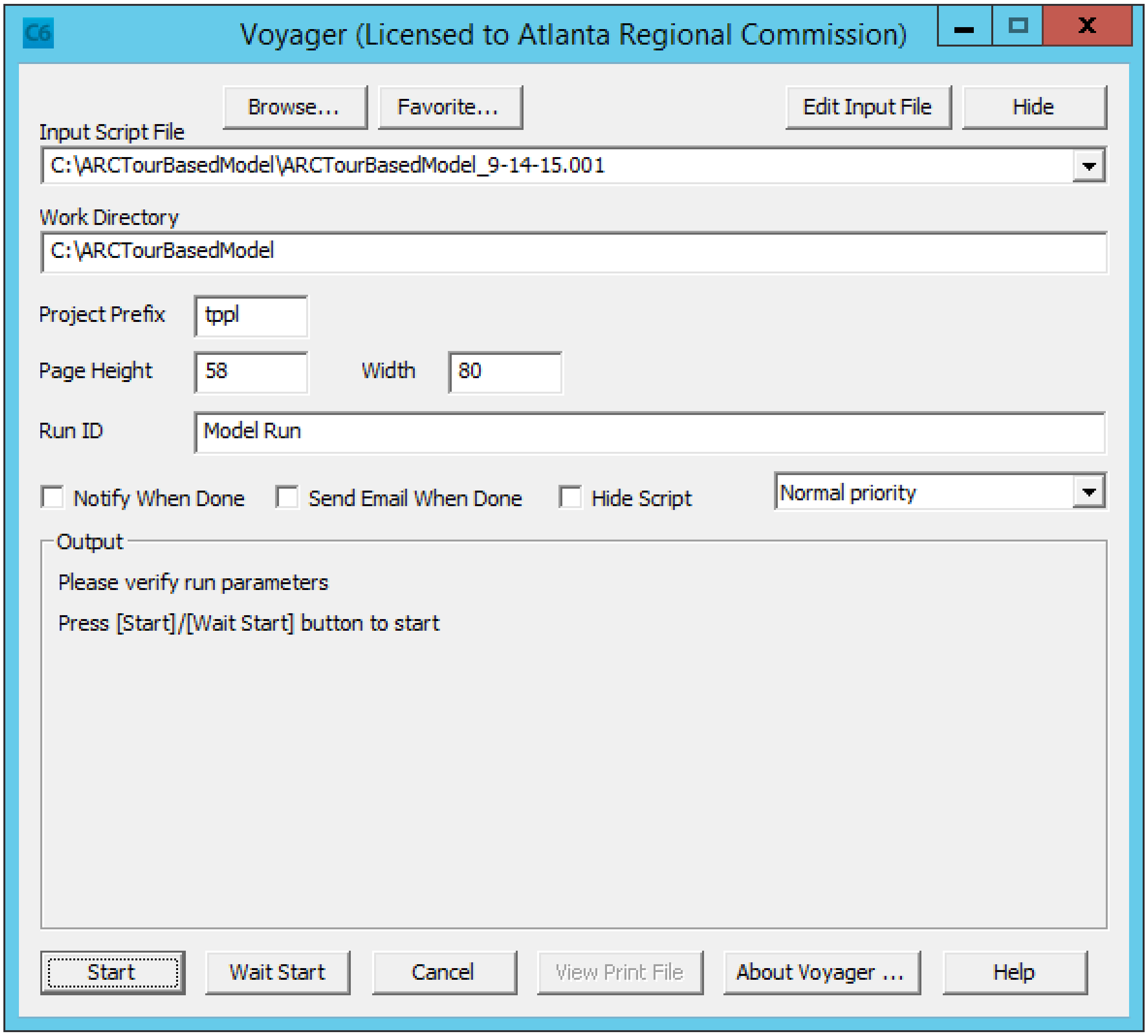
1) Open *ARCTourBasedModel_{date}.s* via File + Open.
2) Set the Model System to Voyager in order to use Cube Cluster via Run + Set Model System.
3) Run the script via Run + Current File to start the GUI, which is shown in **Figure 7**.

Figure 7: ARC ABM GUI



4) Set the various settings, including the project directory, and click *Done*. Note that the project directory should point to the folder on C:\ drive so that the input files for CT-RAMP, particularly the skims, are read from the local drive. See the GUI Settings section below for more information on the GUI settings. Cube then creates an *ARCTourBasedModel_{date}.00X* Cube script with the tokens in the script replaced with the GUI settings.

Figure 8: Execute Main Run Script



5) Click Start to run the model. As discussed in the GUI Settings and Model Outputs sections of this document, the Cube script dynamically writes the CT-RAMP properties file and calls CT-RAMP for each feedback loop via a System call. The Cube script parses the CT-RAMP *indivTripData_<feedback_iteration>.csv* file and *jointTripData_<feedback_iteration>.csv* file to create the demand matrices used for assignment.

Before running the model again, the Java JPPF services should be re-started. In addition, if the model fails, it's better to kill any Java processes still running on the main computer. The *killjava.cmd* file located in the *config* folder can be run on the main computer to kill all running Java processes. The *killjava.cmd* can be used to kill the JPPF nodes running on the slave machines as well.

4.2 Warm Start

The Cube GUI allows for the option of running the model with a warm start (using existing trip tables and shadow prices). This differs from a cold start model run in two major ways.

- The following files need to be copied from a previous run to the current project directory prior to start running the model:

Figure 9: Additional Files Needed for a Warm Start Run

Name	Date modified	Type	Size
com	10/9/2015 11:12 AM	TRP File	716,337 KB
htkbypass	10/9/2015 11:12 AM	TRP File	434,632 KB
htknobypass	10/9/2015 11:12 AM	TRP File	299,155 KB
mtk	10/9/2015 11:12 AM	TRP File	706,928 KB
ShadowPricing_1_0	9/29/2015 12:47 PM	Microsoft Office Excel Comma Separated Values File	3,454 KB
TODAM30_asgn	10/10/2015 7:10 AM	VTT File	12,462 KB
TODEA30_asgn	10/10/2015 7:08 AM	VTT File	3,813 KB
TODEV30_asgn	10/10/2015 7:16 AM	VTT File	9,446 KB
TODMD30_asgn	10/10/2015 7:12 AM	VTT File	11,711 KB
TODPM30_asgn	10/10/2015 7:14 AM	VTT File	13,472 KB

The *{year}* in the naming of the trip tables "*TOD{period}{year}_asgn.VTT*" need to be consistent with the two-digit year specified in the GUI for the model run.

- In the GUI, turn on "*Warm Start Peak Period Highway Assignment*" and "*Use Initial Shadow Pricing File for 1st Feedback Loop*". Set "*Initial Usual Work And School Location Choice Shadow Pricing Input File*" to the appropriate file name.

4.3 GUI Settings

The ARC ABM GUI contains various settings for running the model. The GUI contains groups of settings, which are summarized below.

4.3.1 General Settings and Modules to Run

<u>General Settings</u>

- Year – The year for current model run. Only the last two digits of the year should be specified. This value is used to name input and output filenames and to specify parameters in model control files.

- Max Assignment Iterations – Describes the maximum number of iterations the highway assignments will apply until the network reaches equilibrium. This parameter applies to all highway assignments within the model stream (i.e., Feedback Loop and Time-of-Day). The default is 200 iterations.

- Early AM iterations – Describes the maximum number of iterations the highway assignment for the early AM period will apply until the network reaches equilibrium.

- Maximum Number of Feedback Loops – Describes the maximum number of iterative loops the model will go through during the Feedback Loop Module.

- Minimum Number of Feedback Loops – Describes the minimum number of feedback loops. The first loop can begin by calculating new AM and PM trip tables for assignment or by using seed AM and PM trip tables. Each successive loop uses new assignment trip tables calculated by the model. The default number of overall model loops is three. Convergence is met if the percent RMSE (root mean square error) between the current and previous loop AM assignment link volumes is less than 5.0 percent.

- Total Zones (w/Externals) – Total Number of zones in current model including external stations.

- Range of Internal Zones – Provides the list of internal zones, for specific model steps that are performed only on internal zones.

- Last Internal Before Externals – The highest value internal zone that is less than the lowest external station. This is used as an input to model control files and is particularly important if external stations are imbedded within the internal zone range.

- First External Station – The lowest value external station. This is used as an input to model control files and is particularly important if external stations are imbedded within the internal zone range.

- Last External Station – The highest value external station. This is used as an input to model control files and is particularly important if external stations are imbedded within the internal zone range.

- Airport Zone Number – The TAZ number for the Hartsfield-Jackson Atlanta International Airport. This is used as a parameter for model control files and functions.

- Dobbins Zone Number – The TAZ number for Dobbins Air Reserve Base. This is used as a parameter for model control files and functions.

- Maximum External County ID – The maximum external station number for the surrounding external counties that are used to develop external trips.

- Auto Operating Cost – used to compute generalized cost. The default is 13.85 cents per mile.

- Truck Operating Cost – used to compute generalized cost. The default is 49.33 cents per mile.

- Auto Value of Time – used to compute generalized cost. The default is 25 dollars per hour.

- Auto Value of Time (toll skims) – used to compute generalized cost. The default is 35 dollars per hour.

- Truck Value of Time – used to compute generalized cost. The default is 36 dollars per hour.

- Distance weight for generalized cost – applied to generalized cost calculation. The default is 1.0.
- Cube Cluster Process ID – The process label to use for Cube Cluster. The default is "*ARC*."
- Extra CPUs Available for Cube Cluster – The number of additional CPUs available for distributing/threading tasks with Cube Cluster.
- Start Node for EA – The start cluster node assigned for the EA highway assignment.
- End Node for EA – The end cluster node assigned for the EA highway assignment.
- Start Node for AM – The start cluster node assigned for the AM highway assignment.
- End Node for AM – The end cluster node assigned for the AM highway assignment.
- Start Node for MD – The start cluster node assigned for the MD highway assignment.
- End Node for MD – The end cluster node assigned for the MD highway assignment.
- Start Node for PM – The start cluster node assigned for the PM highway assignment.
- End Node for PM – The end cluster node assigned for the PM highway assignment.
- Start Node for EV – The start cluster node assigned for the EV highway assignment.
- End Node for EV – The end cluster node assigned for the EV highway assignment.
- Cluster Node Restart Wait Time – The length of time allowed to wait for cluster nodes to restart.
- Archive (Zip Results) – After deleting extra files, the model will compress the model results folder using 7-zip command line tool.

Figure 10: General Settings and Modules to Run

ARC Tour Based Model

General Settings

Year (Two-digit)	30
Model Name	ARC ABM
Max Assignment Iterations	200
early am iterations	20
Maximum Number of Feedback Loops	6
Minimum Number of Feedback Loops	3
Total Zones (w/Externals)	5981
Range of Internal Zones	1-5873
Last Internal Before Externals	5873
First External Station	5874
Last External Station	5981
Airport Zone Number	3388
Dobbins Zone Number	2164
Maximum External County ID	6025
Auto Operating Cost - $/mile	0.1385
Truck Operating Cost - $/mile	0.4933
Auto Value of Time - $/hr	25.00
Auto Value of Time - $/hr (toll skims)	35.00
Truck Value of Time - $/hr	36.00
Distance weight for generalized cost	1.0
Cube Cluster Process ID (no spaces)	ARC
Extra CPUs Available for Cube Cluster	107
Start Node for EA	1
End Node for EA	13
Start Node for AM	14
End Node for AM	50
Start Node for MD	51
End Node for MD	64
Start Node for PM	65
End Node for PM	95
Start Node for EV	96
End Node for EV	107
Cluster Node Restart Wait Time	20

- [ ] Archive (Zip Results)

Modules to Run

- [x] Model Preparation
- [x] Commercial Vehicle and Truck Model
- [x] Feedback Loops (includes CTRAMP)
- [x] Transit Assignment
- [x] Assignment Post-Processing

Modules to Run

- Modules to Run – Each model component to be run includes a checkbox.
  - Model Preparation – This module will update new area type, highway speed, and capacity for any changes made to the initial highway network links and will rebuild the updated network. The module will also update any toll costs and build the peak and off-peak highway (free-flow and midday) skims. The module also has an option to run a warm start highway assignment with user specific demand matrix.
  - Commercial Vehicle and Truck Model – This module will build the commercial vehicle and truck model.
  - Feedback Loops – Prepares time-of-day trips tables. Assigns trips to the highway network for each of the five time-of-day (TOD) periods: EA, AM, MD, PM, and EV. This module will build congested highway travel time skims. The module also includes functions that build transit support links, calculate TAZ transit walk accessibility and estimates transit walk times. It will also run CT-RAMP and the IE/EI models.
  - Transit Assignment – Assigns trips from the person trip tables to the transit network.
  - Assignment Post-Processing – This module will also sum the TOD networks to create a total daily network as well as VMT and VHT statistics.

4.3.2 Feedback Modules

Within each feedback loop, each module to run has specific settings that are configured in the following section. It is recommended to run the entire model anytime a network or land use change is made to the model.

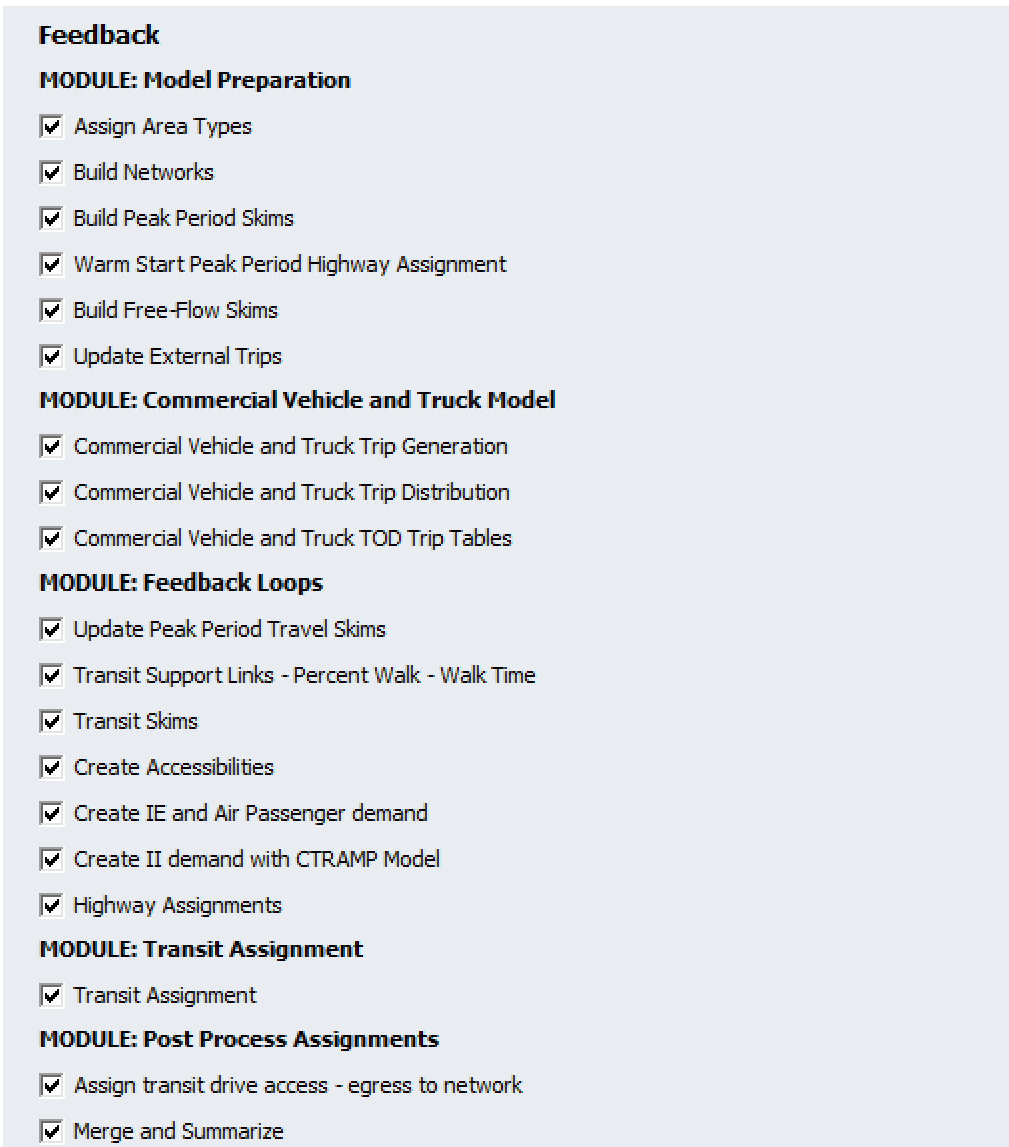
Model Preparation

- Assign Area Types – This procedure calculates a floating zone population and employment density for each TAZ, assigns an area type to each TAZ based on the densities, determines the nearest TAZ to each link in the network and then assigns the associated TAZ area type to the each link.
- Build Networks – This procedure sets the proper highway capacity and speed based on a lookup table that references the area type and facility type of each link.
- Build Peak Period Skims – Creates matrices of toll, distance and time for every zone-to-zone combination within the peak network. Initially the peak travel time skims are based on lookup tables, but they are updated during the feedback loop.
- Warm Start Peak Period Highway Assignment – Warm start the peak period highway assignment by assigning a previously calculated trip matrix.

- Build Free-Flow Skims – Creates matrices of toll, distance, and time for every zone-to-zone combination.

- Update External Trips – This flag is to indicate whether or not to update the external trips. Whenever the highway network changes and/or the socio-economic data for both internal and external surrounding areas changes, the external trip tables need to be updated as well.

Figure 11: Model Preparation, Commercial Vehicle and Truck Model, Feedback Loops, Transit Assignment, and Post Process Assignments



Commercial Vehicle and Truck Model

- Commercial Vehicle and Truck Trip Generation – Generate commercial vehicle and truck trips.

- Commercial Vehicle and Truck Trip Distribution – Distribute commercial vehicle and truck trips.

- Commercial Vehicle and Truck TOD Trip Tables – Create trip tables for commercial vehicle and truck trips by time-of-day.

Feedback Loops

- Update Peak Period Travel Skims – Generate new travel time skims (SOV and HOV) based on the current TOD loaded networks in the feedback loop.

- Transit Support Links – Bus Miles - Percent Walk – Walk Time – Creates walk-to-transit, drive-to-transit, and transfer support links. Calculates the number of peak bus miles within one-mile of each TAZ (for mode choice model). Estimates the portion of each TAZ that has walk access to transit. Calculates the estimated average time it takes to walk to transit from each TAZ that has walk access to transit.

- Transit Skims – Creates travel time tables for passengers using the transit system.

- Create Accessibilities – Creates the accessibility measures based on the skims for use in CT-RAMP.

- Create IE and Air Passenger demand matrices – Creates IE/EI and airport demand matrices.

- Create II demand with CT-RAMP Model – Runs CT-RAMP model via a system call to generate internal travel demand. This step also runs the Cube script code to create trip tables for assignment from the CT-RAMP trip lists.

- Highway Assignment – Runs highway assignments using the TOD trip table to obtain TOD period congested networks.

Transit Assignment

- Transit Assignment – Assigns transit trips to transit networks for five time periods.

Post Process Assignments

- Assign transit drive access – egress to network – Assign transit drive access trips to loaded highway networks from feedback.

- Merge and Summarize – Summarizes the five TOD highway networks into a total daily network. Also computes the VMT and VHT statistics for each TOD period and total daily statistics.

4.3.3 CT-RAMP Settings I

The CT-RAMP settings define the parameters required to run the main *ARCTourBasedModel* CT-RAMP Java process, which is the main CT-RAMP run script. CT-RAMP reads its settings from a Java properties file that is written dynamically by the ARC ABM GUI. The properties in this file are set in the GUI.

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:

- Project Directory
- Population Synthesizer Household File
- Population Synthesizer Person File
- Population Synthesizer Sample Percents
- Population Synthesizer Sample Seed

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

- Directory to 64bit JDK
- Directory to 64bit Cube Voyager File Access API
- Run Model – Household Server Address

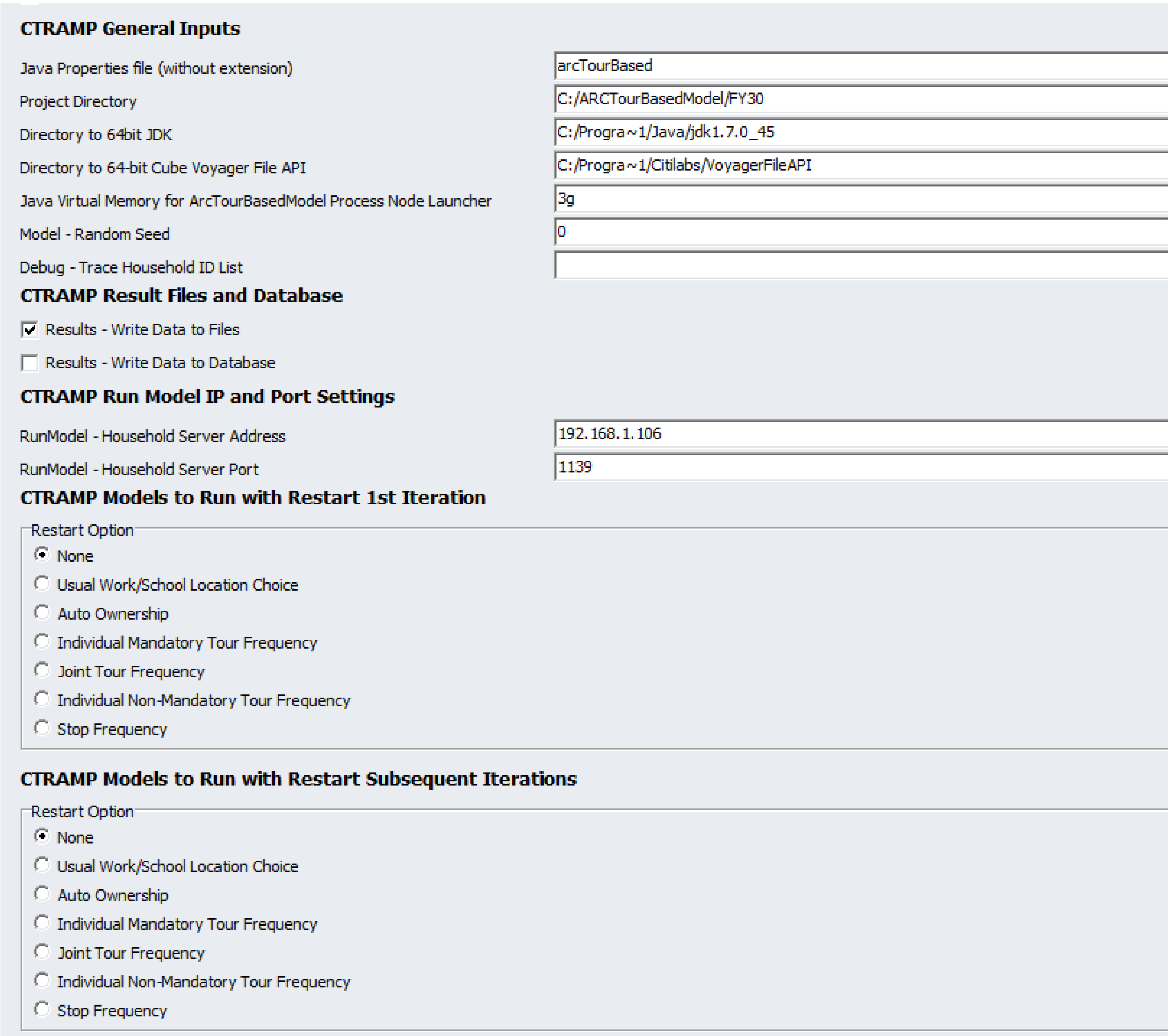
What follows is an exhaustive list of all the CT-RAMP settings in the ARC ABM GUI.

<u>CT-RAMP General Inputs</u>

- Java Properties File – The name of the CT-RAMP properties file to create that will be read by CT-RAMP.
- Project Directory – The project directory. This should be set to the location of the project directory (i.e. the directory that contains the *ARCTourBasedModel_{date}.s* script). Use the local project directory reference instead of the network drive to improve the runtimes.
- Directory to 64bit JDK – The path to the 64bit JDK installed on the main computer.
- Directory to 64-bit Cube Voyager File API – The path to 64-bit Cube Voyager File Access API installed on the main computer.
- Java Virtual Memory for ArcTourBasedModel Process Node Launcher – The gigabytes of memory allocated to the main *ArcTourBasedModel* process.
- Model – Random Seed – The random number seed for CT-RAMP.

- Debug – Trace Household ID List – HH IDs separated by a comma to write debug results for.

Figure 12: CT-RAMP Settings



<u>CTRAMP Result Files and Database</u>

- Results – Write Data to Files – Writes CT-RAMP output files to CSV files. This is required for integration with Cube.
- Results – Write Data to Database – Writes CT-RAMP output tables to a SQL database.

CTRAMP Run Model IP and Port Settings

- 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.

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

CTRAMP Models to Run with Restart 1st Iteration

- Restart Option – restart CT-RAMP at the following location:

    - None – no restart
    - Usual Work/School Location Choice
    - Auto Ownership
    - Individual Mandatory Tour Frequency
    - Joint Tour Frequency
    - Individual Non-Mandatory Tour Frequency
    - Stop Frequency

CTRAMP Models to Run with Restart Subsequent Iterations

- Restart Option – restart CT-RAMP at the following location – Same options as the first iteration.

4.3.4 CT-RAMP Settings II

CTRAMP Models to Run

This section defines which CT-RAMP models to run. By default, all models are run. These models correspond to CT-RAMP UECs stored in the *ctrampModels* folder.

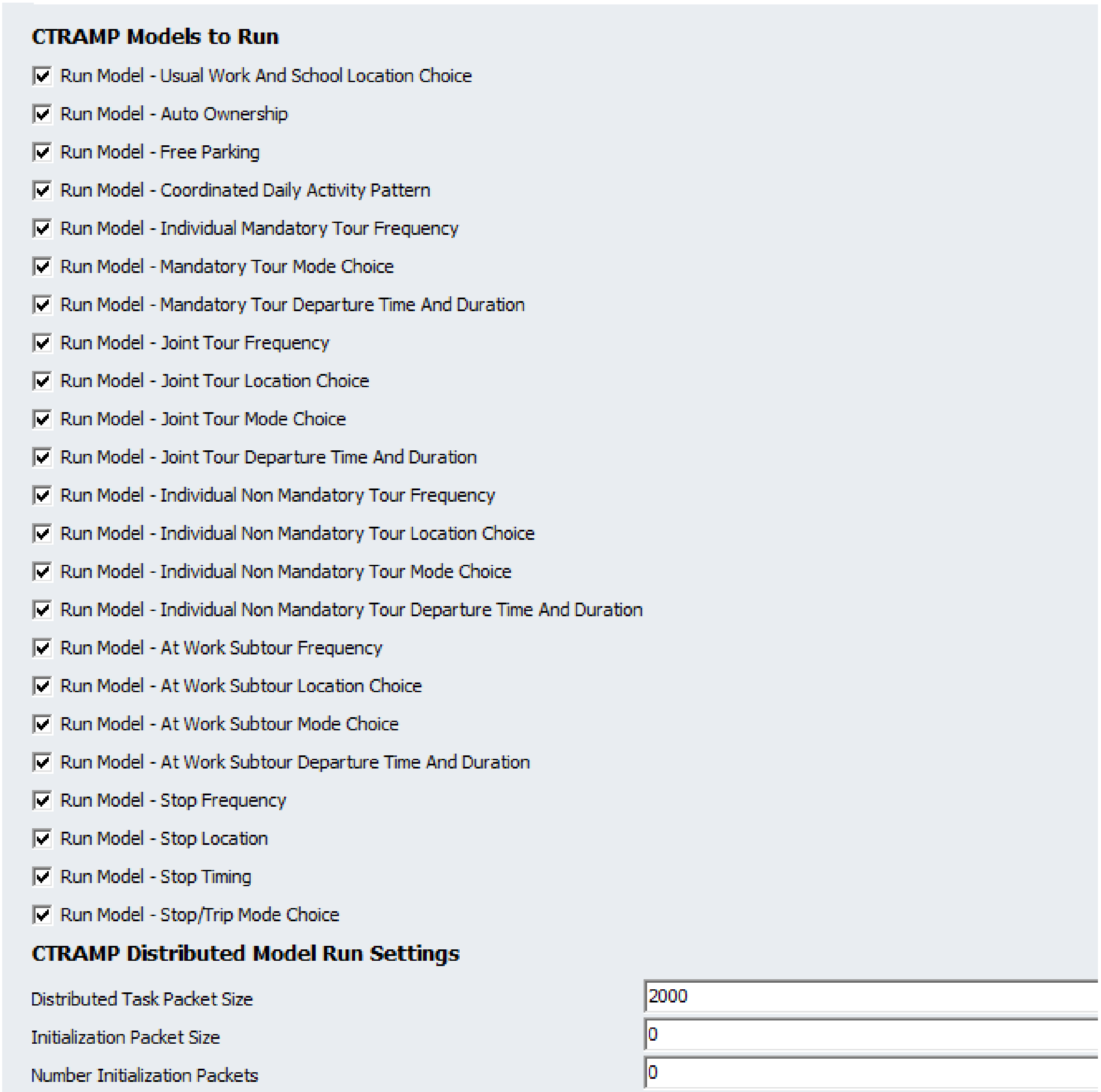
CTRAMP Distributed Model Run Settings

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

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

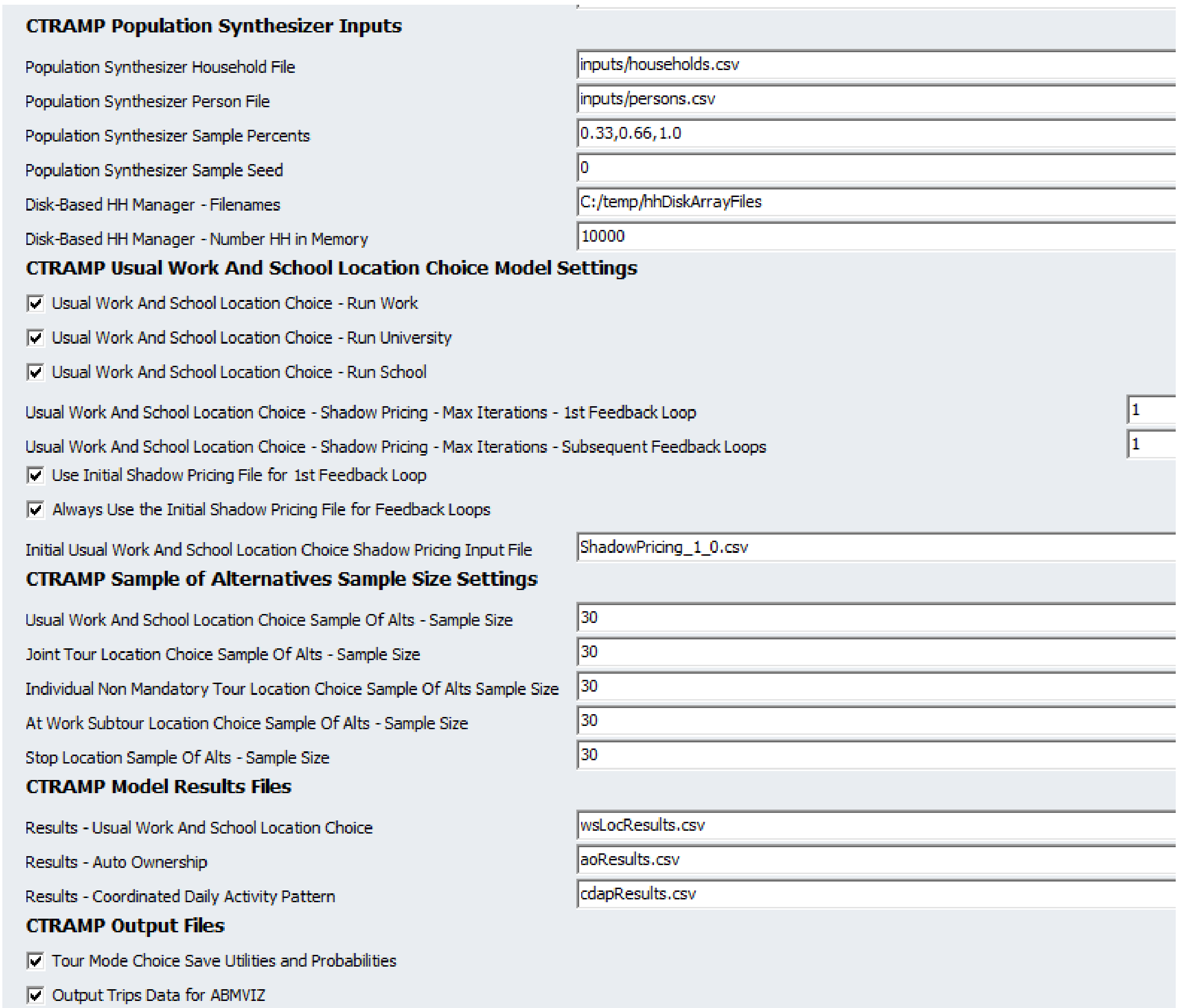
- Number Initialization Packets – The number of initialization packets in CT-RAMP.

Figure 13: CTRAMP Models to Run and Distributed Model Run Settings



4.3.5 CT-RAMP Settings III

Figure 14: Other CT-RAMP Settings



<u>CTRAMP Population Synthesizer Inputs</u>

- Population Synthesizer Household File – Location of the household population synthesizer file.
- Population Synthesizer Person File – Location of the person population synthesizer file.
- Population Synthesizer Sample Percents – An array, equal in the length to the number of overall model feedback loops, with the HH population sample percent. A value of 0.5 means that 50% of the household population will be run through CT-RAMP. The resulting demand matrices will be scaled by 2 (i.e., 1/0.5) before being assigned to the network.
- Population Synthesizer Sample Seed – The sample seed for sampling households.

CTRAMP Usual Work and School Location Choice Model Settings

- Usual Work and School Location Choice – Run Work – Run the usual work location model.

- Usual Work and School Location Choice – Run University – Run the usual university location model.

- Usual Work and School Location Choice – Run School – Run the usual school location model.

- Usual Work and School Location Choice – Shadow Pricing – Max Iterations – 1st Feedback Loop – The maximum number of show pricing iterations for the 1st feedback loop. The default is 1 iteration.

- Usual Work and School Location Choice – Shadow Pricing – Max Iterations – Subsequent Feedback Loop – the maximum number of show pricing iterations for subsequent feedback loops. The default is 1 iteration.

- Use Initial Shadow Pricing File for 1st Feedback Loop – Use an input shadow pricing file as a starting point for the 1st feedback loop. In order to improve run times, a previously calculated shadow pricing file can be used as a starting point.

- Always Use the Initial Shadow Pricing File for Feedback Loops – Use an input shadow pricing file for all feedback loops.

- Initial Usual Work and School Location Choice Shadow Pricing Input File – The input file to use for re-start with shadow pricing. This file would have been calculated from previous model run and has the same format as the output shadow pricing file.

CTRAMP Sample of Alternatives Sample Size Settings

- Usual Work and School Location Choice Sample of Alts – Sample Size – The number of alternatives. The default is 30.

- Joint Tour Location Choice Sample of Alts – Sample Size – The number of alternatives. The default is 30.

- Individual Non Mandatory Tour Location Choice Sample of Alts – Sample Size – The number of alternatives. The default is 30.

- At Work Subtour Location Choice Sample of Alts – Sample Size – The number of alternatives. The default is 30.

- Stop Location Choice Sample of Alts – Sample Size – The number of alternatives. The default is 30.

CTRAMP Model Results Files

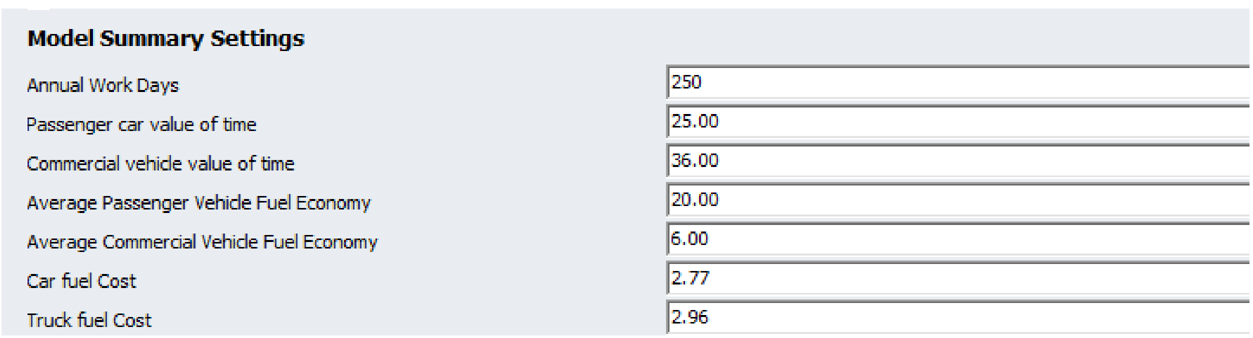
- Results – Usual Work and School Local Choice – Results from the usual work and school location choice model.
- Results – Auto Ownership Choice – Results from the auto ownership choice model.
- Results – Coordinated Daily Activity Pattern – Results from the CDAP choice model.

CTRAMP Output Files

- Tour Mode Choice Save Utilities and Probabilities – Include the alternative choice utilities and probabilities from the tour mode choice model in the tour data output file.
- Output Trips Data for ABMVIZ – Writes out the trips data output file used for ABMVIZ.

4.3.6 Model Summary Settings

Figure 15: Model Summary Settings



- Annual Work Days – The number of work days in a year. The default is 250.
- Passenger car value of time – The value of time for passenger cars in $/hour. The default is 25 dollars per hour.
- Commercial vehicle value of time – The value of time for commercial vehicles. The default is 36 dollars per hour.
- Average Passenger Vehicle Fuel Economy – The fuel efficiency of passenger vehicles. The default is 20 miles/gallon.
- Average Commercial Vehicle Fuel Economy – The fuel efficiency of commercial vehicles. The default is 6 miles/gallon.
- Car Fuel Cost – The fuel cost of cars. The default is $2.77/gallon.
- Truck Fuel Cost – The fuel cost of trucks. The default is $2.96/gallon.

5 Model EXEC Folder

The *EXEC* folder contains the following executables used by the CT-RAMP model:

- *arc.jar* – the ARC CT-RAMP Java archive file containing all the Java code.
- *filesToKeep.bat* – the DOS batch file used for archiving files. It reads *fileToKeep.txt* for a list of files to keep.
- *7z920-x64.exe* – 7-zip command line tool installer.

6 Model Inputs

Inputs to the ARC ABM are stored in the following folders: *INPUTS*, *PARAMETERS*, and *ctrampModels*. The *INPUTS* and *PARAMETERS* folders are required for the model to function and contain a number of files. The model script must be run from within the root folder as the paths are relative. The *INPUTS* folder contains files that typically vary by model scenario while the *PARAMETERS* folder contains files that should not be changed. The *ctrampModels* folder stores all CT-RAMP Utility Expression Calculator (UEC) files and choice model alternatives files.

6.1 Input Files

Within each of the required folders, there are expected files that are read by the model script. Any file missing from these folders will cause the model to fail in application. **Table 14** includes the names and description of files required in the *INPUTS* folder. Some, if not all, of these files will vary between model scenarios. However, the file naming structure must remain the same. **Table 15** provides the list of files required in the *PARAMETERS* folder. Except under special circumstances (e.g. splitting zones), these files should not be altered.

Table 14: INPUTS Folder Required Files

File	Description
ARC_20{year}.GDB	Cube geodatabase (highway and transit networks)
HSH{year}.CSV	households stratified by income and size
EMP{year}.CSV	population and employment by type
UNIV{year}.CSV	university enrollment / zonal acres
PERSONS.CSV	PopSyn person output
HOUSEHOLDS.CSV	PopSyn household output
ExtraZoneData{year}.PRN	zone data file (parking information, CBD, etc.)
TOLLS{year].DBF	toll rates by toll ID

Table 15: PARAMETERS Folder Required Files

File	Description
CAPACITY.DBF	hourly capacity lookup
AUXLANE.DBF	hourly capacity for auxiliary lanes
FFSPEED.DBF	free-flow speed lookup
AMSPEED.DBF	AM peak period speed lookup (first feedback loop)
ExternalCounties.DBF	external county population data
Externals.DBF	external station data (counts, % truck, etc.)
TruckZones.DBF	truck zone flag for commercial model
I285_ZONES.DBF	I-285 zone flag for truck trips inside I-285
ENPLANEMENTS.DBF	annual enplanement forecasts for Hartsfield-Jackson
IEPCSTA.DAT	interstate flag for external stations
IEPCFF.NEW	I-E passenger car friction factors
NWTAZ10G.PRN	2010 population / employment data
EEPC00.VTT	year 2000 E-E passenger car trip table
EETRK05.VTT	year 2005 E-E truck trip table
FFactors.PRN	I-I and I-E truck friction factors
DeltaAM.TRP	delta truck trip matrix for AM
DeltaMD.TRP	delta truck trip matrix for MD
DeltaPM.TRP	delta truck trip matrix for PM
DeltaNT.TRP	delta truck trip matrix for NT
Transit_Walk.FAC	transit path parameters for walk access/egress
Transit_KNR.FAC	transit path parameters for KNR access / walk egress
Transit_KNR_INBOUND.FAC	transit path parameters for walk access / KNR egress
Transit_PNR.FAC	transit path parameters for PNR access / walk egress
Transit_PNR_OP.FAC	transit path parameters for walk access / PNR egress
Transit_System.DAT	transit system data file (modes, operators, etc.)
Transit_Fares.FAR	transit fare structures

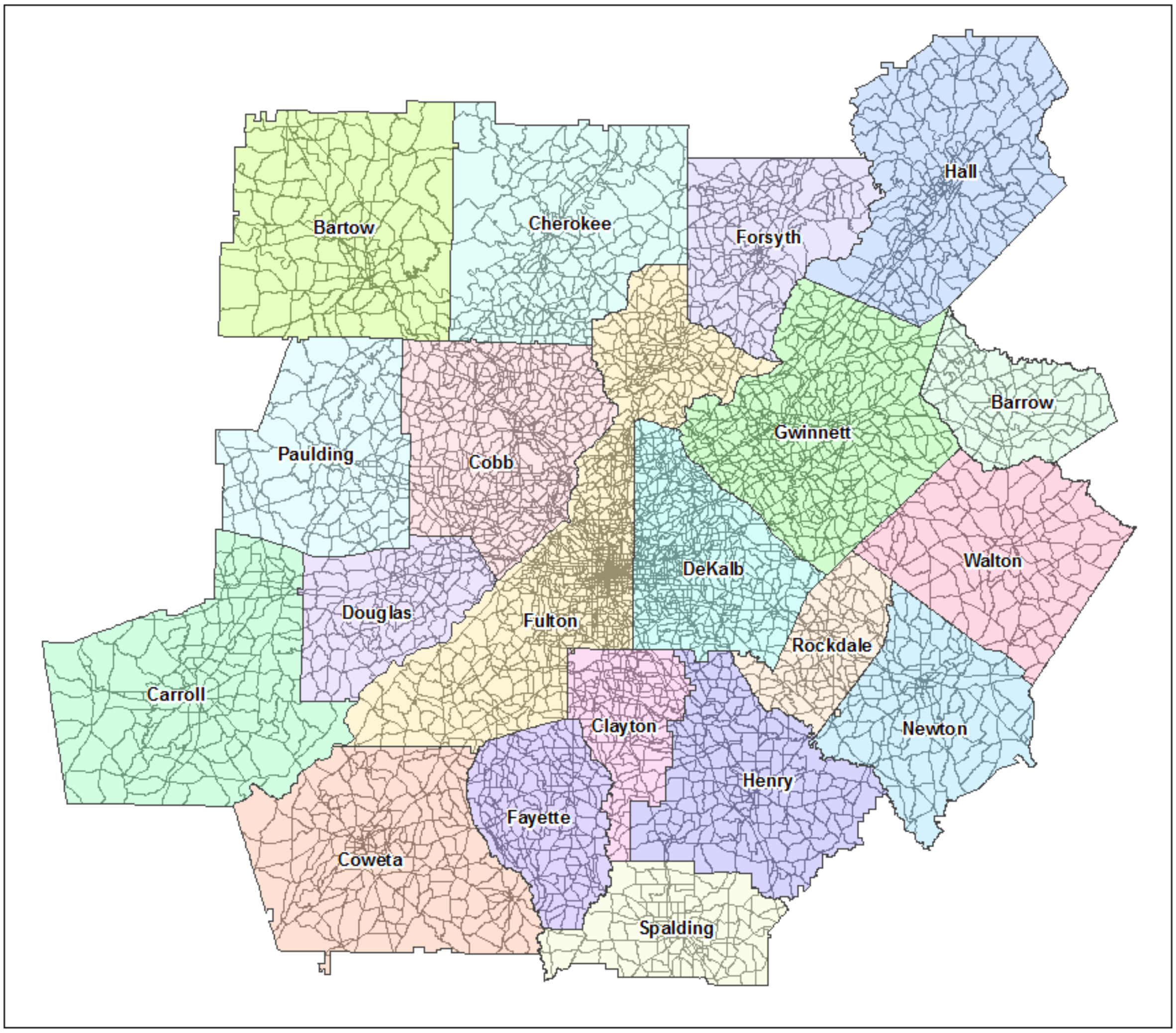
6.2 Traffic Analysis Zone

There were a significant number of zones added to the model. There are now 5,873 internal zones within the ARC boundary. The zones were built from Census 2010 block geographies and are nested within Census tracts. **Table 16** provides the zone numbering by county and a graphic of the zone boundaries is illustrated by **Figure 16**.

Table 16: Zone Ranges by County

County	FIPS Code	Zone Range
Fulton	13121	1-1296
DeKalb	13089	1297-1943
Cobb	13067	1944-2508
Gwinnett	13135	2509-3006
Rockdale	13247	3007-3141
Henry	13151	3142-3386
Clayton	13063	3387-3624
Fayette	13113	3625-3830
Douglas	13097	3831-3988
Cherokee	13057	3989-4209
Coweta	13077	4210-4437
Forsyth	13117	4438-4631
Paulding	13223	4632-4771
Bartow	13015	4772-4942
Carroll	13045	4943-5119
Spalding	13255	5120-5263
Newton	13217	5264-5407
Walton	13297	5408-5530
Barrow	13013	5531-5635
Hall	13139	5636-5873

Figure 16: Traffic Analysis Zones



6.3 Highway Networks

This section details the attributes in the highway networks, speeds, capacities and how to treat reversible lanes and toll facilities.

6.3.1 Network Attributes

The network attributes necessary for the model to function properly are provided in **Table 17**. These attributes include items such as the number of lanes, facility type, distance, etc. The attributes in **Table 18** are also recommended when coding networks to facilitate summarizing the output networks. Special attention should be given to the traffic count locations especially when copying/pasting link attributes or splitting existing links as it could result in traffic counts in incorrect locations.

Table 17: Required Highway Network Attributes

Variable	Description
A / B	Beginning node / ending node
DISTANCE	Link distance (miles)
	0 = No Restrictions
	1 = Trucks Prohibited
	2 = HOV 2+
	3 = SOV toll – HOV 2+ free – no trucks
	4 = Truck Only
	5 = I-285 Bypass
	6 = HOV 3+
	7 = SOV and HOV 2 toll; HOV 3+ free; no trucks
	8 = SOV and truck toll; HOV 2+ free
	9 = SOV, HOV 2, and truck toll; HOV 3+ free
	10 = Truck only toll
	11 = HOV 2 toll; HOV 3+ free; No trucks or SOV
	12 = SOV and HOV 2+ toll; no trucks
	13 = All vehicles allowed and tolled
LANES	Number of through lanes in one direction (including the number of auxiliary lanes)
AUXLANE	Number of auxiliary lanes
TOLLID	Toll identifier
LANESEA LANESAM LANESMD LANESPM LANESNT	Period lanes (if > 0, replaces Lanes)
FACTYPE	0 Centroid Connectors
	1 Interstate / Freeway
	2 Expressway
	3 Parkway
	4 Freeway HOV Buffer Separated
	5 Freeway HOV Barrier Separated
	6 Freeway truck only
	7 System to system ramp
	8 Exit ramp
	9 Entrance Ramp
	10 Principal arterial
	11 Minor arterial
	12 Arterial HOV
	13 Arterial truck only
	14 Collector
	50 Transit Only Link: Neighborhood Local
	51 Transit Only Link: Local Roads and Collectors
	52 Transit Only Link: Park-n-ride lot connector
	53 Transit Only Link: Transfer links between rail and bus
	54 Associated with BRT Routes (Future year coding)
	98 Connections to Transit Stations (pedestrian bridges, direct walk ups, etc)
	99 MARTA Heavy Rail Line

Table 17 (Cont'd): Required Highway Network Attributes

Variable	Description
TRNDIST	Fixed guideway transit distance (miles)
TRNSPD	Fixed guideway transit speed (MPH)
TRNTIME	Fixed guideway transit time (minutes)
HOVMERGE	Flag for HOV/HOT lane merge links (0 or 1 with 1 indicating the link is a merge link)
RAMPFLAG	Flag for loop ramps (0 or 1 with 1 indicating a loop ramp)
WEAVEFLAG	Flag for approaches to major freeway to freeway interchanges (0 or 1 with 1 indicating link is a major weave link)

Table 18: Recommended Highway Network Attributes

Variable	Description
NAME	Facility name
COUNTY	Barrow = 13
	Bartow = 15
	Carroll = 45
	Cherokee = 57
	Clayton = 63
	Cobb = 67
	Coweta = 77
	DeKalb = 89
	Douglas = 97
	Fayette = 113
	Forsyth = 117
	Fulton = 121
	Gwinnett = 135
	Hall = 139
	Henry = 151
	Newton = 217
	Paulding = 223
	Rockdale = 247
	Spalding = 255
	Walton = 297

Table 18 (Cont'd): Recommended Highway Network Attributes

Variable	Description
FCLASS	1 - Rural Interstate
	2 - Rural Principal Arterial
	6 - Rural Minor Arterial
	7 - Rural Major Collector
	8 - Rural Minor Collector
	9 - Rural Local
	11 - Urban Interstate
	12 - Urban Freeway
	14 - Urban Principal Arterial
	16 - Urban Minor Arterial
	17 - Urban Collector
	19 - Urban Local
CNTSTATION	GDOT traffic count station identifier
DIRAADT00 DIRAADT05 DIRAADT08 DIRAADT10	Directional average annual daily traffic by year
DIRAWDT00 DIRAWDT05 DIRAWDT08 DIRAWDT10	Directional average weekday traffic by year
SCREENLINE	Screen-line number for base year validation
ATR	flag indicating GDOT permanent count station
ATR_DIR	directionality of count
HR1 to HR24	2010 hourly volume (ATR locations)
CNTFAC	flag indicating HOV/GP count factoring
FACTOR	factor applied to split counts between HOV / GP
GPID	Corridor identifier for toll optimization
EASPD AMSPD MDSPD PMSPD EVSPD	Observed NPMRDS travel speeds

6.3.2 Capacities

In previous model versions, the lookups for link capacities were embedded in the model script. In this newer version, the lookup has been transferred to a DBF structure which is called within the code. The table is called *CAPACITY.DBF* and is located in the *PARAMETERS* folder. **Table 19** shows the hourly capacities by facility type and area type. Similarly, the capacity for the auxiliary lanes is called *AUXLANE.DBF* (**Table 20**) and is also located in the *PARAMETERS* folder.

Table 19: Hourly Capacities

NAME	FACTYPE	ATYPE 1	ATYPE 2	ATYPE 3	ATYPE 4	ATYPE 5	ATYPE 6	ATYPE 7
centroid connector	0	10000	10000	10000	10000	10000	10000	10000
interstate/freeway	1	1900	1900	2000	2000	2050	2100	2100
expressway	2	1200	1200	1300	1350	1400	1450	1450
parkway	3	1150	1150	1250	1300	1350	1400	1400
freeway HOV (concurrent)	4	1900	1900	2000	2000	2050	2100	2100
freeway HOV (barrier sep)	5	1900	1900	2000	2000	2050	2100	2100
freeway truck only	6	1900	1900	2000	2000	2050	2100	2100
system to system ramp	7	1300	1400	1500	1600	1700	1700	1700
exit ramp	8	800	850	850	850	850	900	900
entrance ramp	9	900	900	950	950	1000	1050	1100
principal arterial	10	1000	1050	1100	1150	1200	1250	1300
minor arterial	11	900	900	950	1000	1000	1050	1100
arterial HOV	12	1000	1050	1100	1150	1200	1250	1300
arterial truck only	13	900	900	950	1000	1000	1050	1100
collector	14	750	800	800	850	850	900	900

Note: ATYPE1 – CBD; ATYPE2 – Urban Commercial; ATYPE3 – Urban Residential; ATYPE4 – Suburban Commercial; ATYPE5 – Suburban Residential; ATYPE6 – Exurban; ATYPE7 – Rural.

Table 20: Auxiliary Lane Capacities

NAME	FACTYPE	AUXCAP	NAME	FACTYPE	AUXCAP
centroid connector	0	0	system to system ramp	7	900
interstate/freeway	1	1200	exit ramp	8	0
expressway	2	900	entrance ramp	9	0
parkway	3	600	principal arterial	10	300
freeway HOV (concurrent)	4	900	minor arterial	11	300
freeway HOV (barrier sep)	5	900	arterial HOV	12	300
freeway truck only	6	900	arterial truck only	13	300
			collector	14	300

6.3.3 Speeds

Similar to the capacities, link speeds were also previously embedded in the model script. These have also been converted to a DBF structure residing in the *PARAMETERS* folder. **Table 21** provides the free-flow speeds by facility type and area type while **Table 22** depicts the AM congested speeds for the first feedback loop.

Table 21: Free-Flow Speeds

NAME	FACTYPE	ATYPE 1	ATYPE 2	ATYPE 3	ATYPE 4	ATYPE 5	ATYPE 6	ATYPE 7
centroid connector	0	7	11	11	11	11	14	14
interstate/freeway	1	62	63	63	63	64	65	66
expressway	2	43	46	49	52	55	58	61
parkway	3	43	46	49	52	55	58	61
freeway HOV (concurrent)	4	64	65	65	65	66	67	68
freeway HOV (barrier sep)	5	64	65	65	65	66	67	68
freeway truck only	6	62	63	63	63	64	65	66
system to system ramp	7	50	50	50	55	55	55	55
exit ramp	8	50	50	50	50	50	50	50
entrance ramp	9	50	50	50	50	50	50	50
principal arterial	10	23	26	31	35	41	48	53
minor arterial	11	21	26	29	33	38	43	48
arterial HOV	12	21	26	29	33	38	43	48
arterial truck only	13	21	26	29	33	38	43	48
collector	14	17	23	24	26	30	35	45
Transit Only: Neighborhood Local	50	12	12	12	12	12	12	12
Transit Only: Locals and Collectors	51	20	20	20	20	20	20	20
Transit Only: PNR lot connector	52	20	20	20	20	20	20	20
Transit Only: Transfer between rail and bus	53	20	20	20	20	20	20	20

Note: ATYPE1 – CBD; ATYPE2 – Urban Commercial; ATYPE3 – Urban Residential; ATYPE4 – Suburban Commercial; ATYPE5 – Suburban Residential; ATYPE6 – Exurban; ATYPE7 – Rural.

Table 22: AM Congested Speeds – First Feedback Loop

NAME	FACTYPE	ATYPE 1	ATYPE 2	ATYPE 3	ATYPE 4	ATYPE 5	ATYPE 6	ATYPE 7
centroid connector	0	7	11	11	11	11	14	14
interstate/freeway	1	37	38	47	51	55	60	68
expressway	2	38	40	42	42	44	49	52
parkway	3	28	30	41	40	46	48	49
freeway HOV (concurrent)	4	50	54	60	65	64	70	70
freeway HOV (barrier sep)	5	50	54	60	65	64	70	70
freeway truck only	6	50	54	60	65	64	70	70
system to system ramp	7	33	33	34	43	46	48	55
exit ramp	8	12	15	19	22	22	25	30
entrance ramp	9	25	25	30	30	30	35	35
principal arterial	10	24	25	27	30	34	41	43
minor arterial	11	22	23	27	28	31	36	40
arterial HOV	12	22	23	27	28	31	36	40
arterial truck only	13	22	23	27	28	31	36	40
collector	14	22	22	26	27	29	36	39
Transit Only: Neighborhood Local	50	12	12	12	12	12	12	12
Transit Only: Locals and Collectors	51	20	20	20	20	20	20	20
Transit Only: PNR lot connector	52	20	20	20	20	20	20	20
Transit Only: Transfer between rail and bus	53	20	20	20	20	20	20	20

Note: ATYPE1 – CBD; ATYPE2 – Urban Commercial; ATYPE3 – Urban Residential; ATYPE4 – Suburban Commercial; ATYPE5 – Suburban Residential; ATYPE6 – Exurban; ATYPE7 – Rural.

6.3.4 Lane Coding

When coding the number of lanes, there are certain guidelines to follow for auxiliary lanes and when the facility may operate with a different number of lanes by time period. The LANES field should include auxiliary lanes. There are several attributes in the network that allow for varying the number of lanes by time period:

- LANESEA: available lanes during early AM period (3:00 AM to 6:00 AM)
- LANESAM: available lanes during AM Period (6:00 AM to 10:00 AM)
- LANESMD: available lanes during midday Period (10:00 AM to 3:00 PM)
- LANESPM: available lanes during PM Period (3:00 PM to 7:00 PM)
- LANESEV: available lanes during evening/late night Period (7:00 PM to 3:00 AM)

If the lanes do not vary by time period, the LANES attribute should be the only lanes attribute coded (i.e. the period lanes should all be set to zero). However, if the lanes do vary by period, then the appropriate values should be entered by time period. The code is written such that if the period lanes are zero, the program defaults to LANES when computing capacity. If the period lanes are greater than zero, the program uses the period lanes to compute capacity. Some examples are provided below in **Table 23** through **Table 25**. In example 1, four lanes (two lanes each direction) are available throughout the day which requires only LANES to be coded. The period lanes are left as zero.

Table 23: Time of Day Lanes Example 1

Attribute	AB Direction	BA Direction
LANES	2	2
LANESEA	0	0
LANESAM	0	0
LANESMD	0	0
LANESPM	0	0
LANESEV	0	0

Example 2 shows how one type of reversible lane could be handled. An additional lane is coded in the AB direction for the AM period; however, the BA direction for AM lanes is set to zero. For the AM period, this facility would operate as 3 lanes in the AB direction and 2 lanes in the BA direction (because LANES = 2). The reverse is true in the PM period (BA direction = 3 lanes, AB direction = 2 lanes). With no values coded in the EA, MD, and EV periods, the lanes would default to 2 lanes in each direction.

Table 24: Time of Day Lanes Example 2

Attribute	AB Direction	BA Direction
LANES	2	2
LANESEA	0	0
LANESAM	3	0
LANESMD	0	0
LANESPM	0	3
LANESEV	0	0

Example 3 shows a case where a one-way facility has lanes that are only available in the EA and AM period. In this example, only the LANES EA and LANESAM should have values. All other lane attributes should be coded as 0 (including LANES).

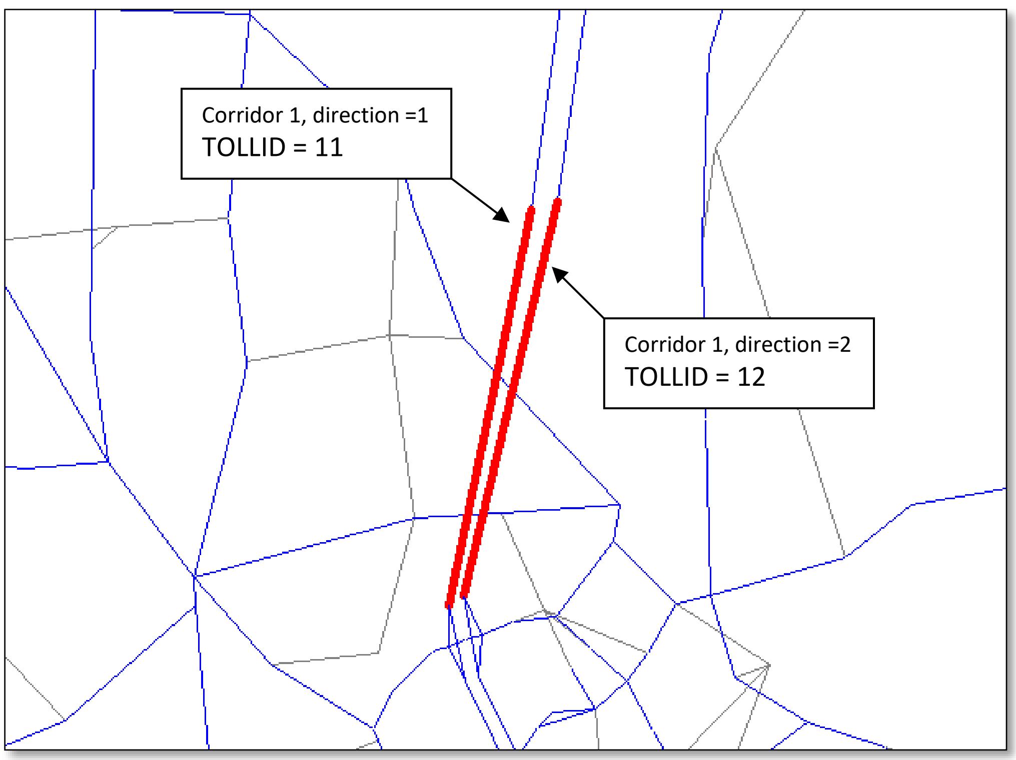
Table 25: Time of Day Lanes Example 3

Attribute	AB Direction
LANES	0
LANESEA	2
LANESAM	2
LANESMD	0
LANESPM	0
LANESEV	0

6.3.5 Toll Facilities

The coding of toll facilities has not changed in the new model version. The user must code the TOLLID attribute on the appropriate links to represent a toll facility. The value on this attribute is at the user's discretion; however, it is recommended that the user follow a defined set of guidelines when coding. One method is to make the first digit of the toll identifier unique to the corridor it represents and the second digit a directional flag. An example is provided below in **Figure 17**. Discussed later in this document is the process of optimizing tolls for managed lane facilities. To optimize tolls, a comparison between the travel times of the general purpose lanes and managed lanes is necessary. To accommodate this, a link attribute called GPID can be used to inform the toll optimization algorithm of the level-of-service characteristics of the competing lanes. The GPID should be coded with the same number as the competing managed lane facility (i.e. TOLLID = 11 and GPID = 11 for a competing route). Further information is provided in the highway network coding portion of this document.

Figure 17: Toll ID Example



The toll rates are defined by the file *TOLLS{year}.DBF* residing in the *INPUTS* folder. The *{year}* should be representative of the last two digits of the model run year. Within the file, there should be six column attributes labeled as follows:

- TOLLID: identifier used to provide the linkage to the highway network
- TOLLEA: toll rate for early AM
- TOLLAM: toll rate for AM period
- TOLLMD: toll rate for midday period

- TOLLPM: toll rate for PM period
- TOLLNT: toll rate for evening/late night period
- FIXED: flag which informs the model if the toll is fixed or distance based

The TOLLID in the database should correspond to the toll identifier coded in the highway network. Within the model this identifier is used as a lookup value to determine the toll rates to be assessed on the highway links. The toll rate should be entered in cents (e.g. one dollar is 100) and applied to the appropriate time period. If the toll rate is unchanged throughout the day, each period should have the same value.

The FIXED attribute is a simple Boolean logic applying a '1' or '0' which informs the program how to treat the toll ('1' is true and '0' is false). If the toll rate is fixed, the value should be set to '1'. If the toll rate is a distance based toll, the FIXED attribute should be set to '0' which tells the program to compute the toll based on the rate and link distance. The entered toll values should be reflective of the rates applied per mile for each time period. An example toll file setup is provided below.

Table 26: Toll ID Database

TOLLID	TOLLEA	TOLLAM	TOLLMD	TOLLPM	TOLLEV	FIXED
21	5	20	5	10	5	0
22	5	10	5	20	5	0
51	5	25	5	10	5	0
52	5	10	5	25	10	0

6.3.6 Prohibitions

The ARC model includes the ability to test various lane restrictions (e.g. HOT lanes, truck lanes, etc.). This is handled through coding a link attribute in the network called PROHIBITION. The path building procedures in the model script utilize this link attribute to determine what vehicle types can use the links. **Figure 18** provides a list of the prohibition codes and how each vehicle type is treated.

Figure 18: Prohibition Coding

Prohibition	SOV	HOV-2	HOV-3+	Trucks	
1	✓	✓	✓	🚫	No Trucks
2	🚫	✓	✓	🚫	HOV 2+
3	$	✓	✓	🚫	Managed Lanes - SOV Toll - HOV 2+ Free - No Trucks
4	🚫	🚫	🚫	✓	Truck Only Lanes
5	✓	✓	✓	✓ / 🚫	I-285 Bypass
6	🚫	🚫	✓	🚫	HOV 3+
7	$	$	✓	🚫	Managed Lanes - SOV & HOV2 Toll - HOV 3+ Free - No Trucks
8	$	✓	✓	$	Managed Lanes - SOV & Truck Toll - HOV 2+ Free
9	$	$	✓	$	Managed Lanes - SOV, HOV2 and Truck Toll - HOV 3+ Free
10	🚫	🚫	🚫	$	Truck Only Toll
11	🚫	$	✓	🚫	Managed Lanes - HOV2 Toll - HOV 3+ Free - No Trucks or SOV
12	$	$	$	🚫	Managed Lanes - SOV and HOV2+ Toll - No Trucks
13	$	$	$	$	Managed Lanes – SOV, HOV2+ and Trucks Toll

6.4 Transit Networks/System

The transit data in the new model is significantly different with the conversion to Public Transport (PT). There are still routes stored in text format, but the manner in which the modes, fares, transfer characteristics, etc. are stored has completely changed. This section provides information regarding the associated transit input files.

6.4.1 Transit Route Files

The transit route files are similar to the previous model versions but are now stored in the Cube geodatabase. There are two transit networks stored in the geodatabase:

- *NONPREMIUM_TRN{year}*: All non-premium transit routes
- *PREMIUM_TRN{year}*: Premium only transit routes

The route files can still be viewed in a text format containing line information and node numbers. The primary differences in the route files are some attribute name changes, required variables, and PT specific available attributes. More information is provided about the route files in the transit coding section of this document.

6.4.2 Transit System Data File

The structure of how the transit system is defined in PT is substantially different than in TRNBUILD. The transit modes, first and second wait curves, and operators are defined in the transit system data file called *TRANSIT_SYSTEM.DAT* located in the *PARAMETERS* folder. This file contains mode numbers and names as provided in **Figure 19**. The wait curves are stored in this file with an example shown in **Figure 20.** Also, the operator numbers are stored here as well. The operator numbers are set so that if an operator has varying fares by mode, each mode would have a unique operator. For example, if an operator charged $2.00 for local bus usage but $3.00 for rail, there would be an operator number for both local bus and rail. **Figure 21** shows an example of how the operators are specified.

Figure 19: System Data File Example Modes

```
;;<<PT>><<SYSTEM>>;;
MODE NUMBER=1 LONGNAME="WALK CONNECTOR" NAME="WALK"
MODE NUMBER=2 LONGNAME="KNR CONNECTOR" NAME="KNR"
MODE NUMBER=3 LONGNAME="PNR CONNECTOR" NAME="PNR"
MODE NUMBER=4 LONGNAME="STATION PLATFORM" NAME="STACON"
MODE NUMBER=5 LONGNAME="TRANSFER LINK" NAME="XFER"

MODE NUMBER=14 LONGNAME="MARTA LOCAL BUS" NAME="MARTA LBUS"
MODE NUMBER=15 LONGNAME="MARTA HEAVY RAIL" NAME="MARTA HRT"
MODE NUMBER=16 LONGNAME="MARTA EXPRESS BUS" NAME="MARTA XBUS"
MODE NUMBER=17 LONGNAME="MARTA LIGHT RAIL" NAME="MARTA LRT"
MODE NUMBER=18 LONGNAME="MARTA BUS RAPID TRANSIT" NAME="MARTA BRT"
```

Figure 20: System Data File Example Wait Curves

```
WAITCRVDEF NUMBER=1 LONGNAME="InitialWait" NAME="InitWait" ,
           CURVE=0-2,4-2,60-30,180-30
WAITCRVDEF NUMBER=2 LONGNAME="TransferWait" NAME="XferWait" ,
           CURVE=0-2,4-2,60-30,160-30
```

Figure 21: System Data File Example Operators

```
OPERATOR NUMBER=1 LONGNAME="MARTA" NAME="MARTA"
OPERATOR NUMBER=2 LONGNAME="COBB CO" NAME="CCT"
OPERATOR NUMBER=3 LONGNAME="CLAYTON CO" NAME="CTRAN"
OPERATOR NUMBER=4 LONGNAME="GWINNETT CO LOCAL BUS" NAME="GCT LBUS"
OPERATOR NUMBER=5 LONGNAME="GWINNETT CO EXPRESS BUS" NAME="GCT XBUS"
OPERATOR NUMBER=6 LONGNAME="GRTA" NAME="GRTA"
OPERATOR NUMBER=7 LONGNAME="CHEROKEE CO" NAME="CATS"
OPERATOR NUMBER=8 LONGNAME="HALL CO" NAME="HAT"
OPERATOR NUMBER=9 LONGNAME="PROJECT" NAME="PRJ"
OPERATOR NUMBER=10 LONGNAME="COMRAIL" NAME="CRAIL"
OPERATOR NUMBER=11 LONGNAME="FREE SHUTTLES" NAME="SHUTTLE"
```

6.4.3 Transit Factor Files

The transit factor files are in text format and contain information regarding the access/egress modes, path building parameters, and linkage between the fare systems and operators. The wait curves specified in the transit system data file are also linked here for the initial wait time and transfer wait time. There are six transit factor files located in the *PARAMETERS* folder:

- *TRANSIT_WALK.FAC*: walk access factors
- *TRANSIT_KNR.FAC*: KNR access factors (KNR access mode)
- *TRANSIT_KNR_INBOUND.FAC*: KNR access factors (KNR egress mode)
- *TRANSIT_PNR.FAC*: PNR access factors (PNR access mode)
- *TRANSIT_PNR_INBOUND.FAC*: PNR access factors (PNR egress mode)

An example of the format of the factor files is provided in **Figure 22.**

Figure 22: Transit Factor File Walk Access Example

```
;Walk access transit factors
;ARC - October 2012, Atkins

;Global Settings
 BESTPATHONLY=T
 MAXFERS=5
 SERVICEMODEL=FREQUENCY
 RECOSTMAX=400.0
 FREQBYMODE=T
 DELACCESSMODE = 2,3
 DELEGRESSMODE = 2,3

;Fare
 FARESYSTEM=1, OPERATOR=1
 FARESYSTEM=2, OPERATOR=2
 FARESYSTEM=3, OPERATOR=3
 FARESYSTEM=4, OPERATOR=4
 FARESYSTEM=5, OPERATOR=5
 FARESYSTEM=6, OPERATOR=6
 FARESYSTEM=7, OPERATOR=7
 FARESYSTEM=8, OPERATOR=8
 FARESYSTEM=9, OPERATOR=9
 FARESYSTEM=10, OPERATOR=10
 FARESYSTEM=11, OPERATOR=11
;Wait times
 IWAITCURVE=1,NODES=1-99999
 XWAITCURVE=2,NODES=1-99999
 WAITFACTOR=2.00,NODES=1-99999

;Run factors
 RUNFACTOR[1]=2.0
 RUNFACTOR[2]=2.0
 RUNFACTOR[3]=2.0
 RUNFACTOR[4]=2.0
 RUNFACTOR[5]=2.0
 RUNFACTOR[15]=0.75
```

6.4.4 Transit Fare File

The transit fare structures are also specified in a text file located in the *PARAMETERS* folder and is titled *TRANSIT_FARES.FAR*. An example format is shown in **Figure 23**. This file specifies the fare amount as well as the how the fare is assessed for each of the fare systems defined in the transit factor files. Linking the fare systems between the two files is handled by the test: FARESYTEM, NUMBER=# where the number is equal to the fare system. The fare systems also include names (e.g. MARTA FARES) to make interpretation easier. The way in which the fares are implemented for a given fare system are handled by STRUCTURE. There is much more flexibility in PT for how the fares are assessed. For example, fares can be applied as a single boarding fare, distance based, or from fare zone to fare zone. IBOARDFARE is the boarding fare for the system and is entered in dollars. To define how the fares are treated between fare systems, the FAREFROMFS is used. The FAREROMFS is specified from each of the fare systems to the system that is being defined. The value entered represents the fare incurred when transferring from another fare system to the current fare system. For example, if all operators allow for free transfers between systems, then the FAREFROMFS would all be set to zero. However, if fare system 1 charges a $1.00 and fare system 2 charges $2.00, there could be an up-charge from fare system to 1 to fare system 2 of $1.00. This would be specified using the FAREFROMFS.

An example of free transfers versus two different up charge scenarios is illustrated below:

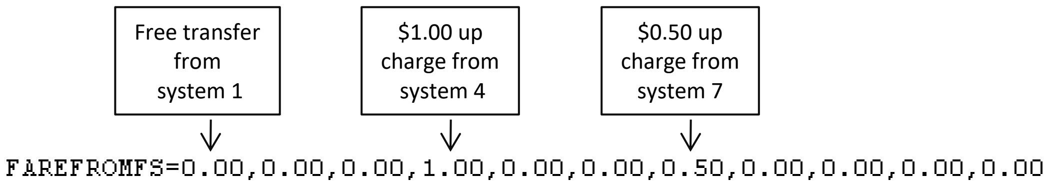


Figure 23: Transit Fare File Example

```
FARESYSTEM, NUMBER=1, LONGNAME="MARTA FARES", NAME="MARTA FARE",
  STRUCTURE="FLAT"  SAME="CUMULATIVE",
  IBOARDFARE=1.75,
  FAREFROMFS=0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00
FARESYSTEM, NUMBER=2, LONGNAME="COBB CO FARES", NAME="CCT FARE",
  STRUCTURE="FLAT"  SAME="CUMULATIVE",
  IBOARDFARE=1.75,
  FAREFROMFS=0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00
FARESYSTEM, NUMBER=3, LONGNAME="CLAYTON CO FARES", NAME="CTRAN FARE",
  STRUCTURE="FLAT"  SAME="CUMULATIVE",
  IBOARDFARE=1.75,
  FAREFROMFS=0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00
FARESYSTEM, NUMBER=4, LONGNAME="GWINNETT CO LOCAL BUS FARES", NAME="GCT LBUS FARE",
  STRUCTURE="FLAT"  SAME="CUMULATIVE",
  IBOARDFARE=1.75,
  FAREFROMFS=0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00
FARESYSTEM, NUMBER=5, LONGNAME="GWINNETT CO EXPRESS BUS FARES", NAME="GCT XBUS FARE",
  STRUCTURE="FLAT"  SAME="CUMULATIVE",
  IBOARDFARE=2.00,
  FAREFROMFS=0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00
```

Figure 24: Fare Structure Cube Help

FLAT	Trip length not used for fare calculation. Fares derived by leg using the boarding and transfer costs specified with IBOARDFARE and FAREFROMFS.
DISTANCE	Trip length measured as in-vehicle distance, in user-specified units. Fares calculated by leg or for sets of consecutive legs using the same fare system. Fares based on: • Boarding and transfer costs, specified with IBOARDFARE and FAREFROMFS. • Trip cost computed by multiplying in-vehicle distance by UNITFARE. • Trip cost obtained from the fare table specified by FARETABLE. The value of INTERPOLATE determines whether the program uses linear interpolation or a step function between coded points.
HILOW	Trip length measured by the highest and lowest fare-zones crossed. Appropriate for an annular fare-zone structure. Fares calculated by leg or for sets of consecutive legs using the same fare system. Fares based on: • Boarding and transfer costs, specified with IBOARDFARE and FAREFROMFS (boarding costs are added to the fare matrix) • Trip cost, extracted from the fare matrix specified with FAREMATRIX (rows contain the lower fare zone, and columns contain the higher fare zone)
COUNT	Trip length measured by the number of fare zones crossed plus 1 for the initial zone. Best suited to a sequential zone system. Fares calculated by leg or for sets of consecutive legs using the same fare system. Fares based on: • Boarding and transfer costs, specified with IBOARDFARE and FAREFROMFS • Trip cost, extracted from the fare table specified with FARETABLE. In this case, the program interpolates the fare table as a step function.
FROMTO	Trip length measured as function of the boarding and alighting fare zones. Best suited to a sequential zone system. Fares calculated by leg or for sets of consecutive legs using the same fare system. Fares based on: • Boarding and transfer costs, specified with IBOARDFARE and FAREFROMFS (boarding costs are added to the fare matrix). • Trip cost, extracted from the fare matrix specified with FAREMATRIX (rows contain the lower fare zone, and columns contain the higher fare zone)
ACCUMULATE	Trip length measured by accumulating fares associated with each fare zone traversed. Differs from COUNT, where program counts the number of fare zones traversed to compute fare. Best suited to a sequential zone system. Fares calculated by leg or for sets of consecutive legs using the same fare system. Fares based on: • Boarding and transfer costs, specified with IBOARDFARE and FAREFROMFS • Trip cost, extracted from the fare table specified with FARETABLE. The fare table has a fare for each fare zone in the system.

6.5 Socioeconomic Data Files

The input socioeconomic (SE) files have been modified and are now stored in CSV files located within the *INPUTS* folder. Employment by category and population is stored in *EMP{year}.CSV* with the format shown in **Table 27.** The household file (*HSH{year}.CSV*) is split by income group and household size using the format in **Table 28.** The university enrollment and acres are now stored in a separate file called *UNIV{year}.CSV* with the format provided in **Table 29**.

Table 27: Population and Employment Input

Column	Name	Description
1	TAZ	Zone
2	FIPS	County FIPS
3	EMP	Total employment
4	N11	NAICS - agriculture, forestry, fishing, and hunting
5	N21	NAICS - mining, quarrying, and oil and gas extraction
6	N22	NAICS - utilities
7	N23	NAICS - construction
8	N313233	NAICS - textile miles /
9	N42	NAICS - wholesale trade
10	N4445	NAICS - retail trade
11	N4849	NAICS - transportation and warehousing
12	N51	NAICS - information
13	N52	NAICS - finance and insurance
14	N53	NAICS - real estate and rental and leasing
15	N54	NAICS - professional, scientific, and technical services
16	N55	NAICS - management of companies and enterprises
17	N56	NAICS - administrative and support and waste management and remediation services
18	N61	NAICS - educational services
19	N62	NAICS - health care and social assistance
20	N71	NAICS - arts, entertainment, and recreation
21	N72	NAICS - accommodation and food services
22	N81	NAICS - other services (except public administration)
23	N92	NAICS - public administration
24	Cons	SIC - construction
25	Manu	SIC - manufacturing
26	TCU	SIC - transportation, communication, utilities
27	Whol	SIC - wholesale
28	Retail	SIC - retail
29	FIRE	SIC - financial, insurance, real estate
30	Serv	SIC - service
31	Govt	SIC - government

Table 28: Household Size by Income Group Input

Column	Name	Description
1	TAZ	Zone
2	I1H1	Income Group 1; Size 1
3	I2H1	Income Group 2; Size 1
4	I3H1	Income Group 3; Size 1
5	I4H1	Income Group 4; Size 1
6	I1H2	Income Group 1; Size 2
7	I2H2	Income Group 2; Size 2
8	I3H2	Income Group 3; Size 2
9	I4H2	Income Group 4; Size 2
10	I1H3	Income Group 1; Size 3
11	I2H3	Income Group 2; Size 3
12	I3H3	Income Group 3; Size 3
13	I4H3	Income Group 4; Size 3
14	I1H4	Income Group 1; Size 4
15	I2H4	Income Group 2; Size 4
16	I3H4	Income Group 3; Size 4
17	I4H4	Income Group 4; Size 4
18	I1H5	Income Group 1; Size 5
19	I2H5	Income Group 2; Size 5
20	I3H5	Income Group 3; Size 5
21	I4H5	Income Group 4; Size 5
22	I1H6	Income Group 1; Size 6+
23	I2H6	Income Group 2; Size 6+
24	I3H6	Income Group 3; Size 6+
25	I4H6	Income Group 4; Size 6+

Table 29: University Enrollment

Column	Name	Description
1	TAZ	Zone
2	UNIV	Enrollment
3	ACRES	Acres

6.6 Air Passenger Model

This section describes the input file required for running the air passenger model. The file is called *ENPLANEMENTS.DBF* and is located in the *PARAMETERS* folder. It contains the annual enplanements for several model years which are used in the model for the generation of air passenger trips. For intermediate years, the model interpolates between the two nearest years. The format of the file is shown in **Table 30**.

Table 30: Enplanements File

Column	Name	Description
1	Year	Forecast year
2	Enplane	Annual enplanements

6.7 Externals

This section describes the input files required for the external models. There are seven files located in the *PARAMETERS* folder which are used for forecasting the external station volumes. The file formats are provided in **Table 31** through **Table 36**. The file names and descriptions are:

- *EXTERNALS.DBF*: Base year external station data (e.g. station number, road name, lanes, traffic counts, percent IE, percent EE, etc.)
- *EXTERNALCOUNTIES.DBF*: Population forecasts for counties surrounding ARC model boundary
- *IEPCSTA.DAT*: Text file classifying stations as interstate and non-interstate
- *IEPCFF.NEW*: External station passenger car friction factors for work/non-work and interstate/non-interstate
- *EEPC00.VTT*: External to external (EE) passenger car matrix used for frataring future year EE passenger car trips
- *EETRK05.VTT*: EE truck trip tables for frataring future year EE truck trips
- *NWTAZ10G.PRN*: Year 2010 base year SE data file for determining population/employment growth

Table 31: Externals File

Column	Name	Description
1	N	Station number
2	Name	Road name
3	FIPS	County FIPS code
4	EXTCNTSTA	GDOT count station
5	LANES	Number of lanes
6	EXTFLAG	External station group ID
7	INTFLAG	Interstate flag (0 = non-interstate, 1 = interstate)
8	AADT2000	Year 2000 Average Annual Daily Traffic
9	AADT2005	Year 2005 Average Annual Daily Traffic
10	AADT2010	Year 2010 Average Annual Daily Traffic
11	AWDT2000	Year 2000 Average Weekday Traffic
12	AWDT2005	Year 2005 Average Weekday Traffic
13	AWDT2010	Year 2010 Average Weekday Traffic
14	PCINTWK	% IE Passenger car interstate work
15	PCINTNW	% IE Passenger car interstate non-work
16	PCNINTW	% IE Passenger car non-interstate work
17	PCNINTN	% IE Passenger car non-interstate non-work
18	CAREE	% EE Passenger car
19	COMIE	% IE Commercial vehicle
20	COMEE	% EE Commercial vehicle
21	MTKIE	% IE Medium duty truck
22	MTKEE	% EE Medium duty truck
23	HTKIE	% IE Heavy duty truck
24	HTKEE	% EE Heavy duty truck

Table 32: External Counties File

Column	Name	Description
1	N	External county number
2	COUNTY	County name
3	POP00	Year 2000 county population
4	POP05	Year 2005 county population
5	POP10	Year 2010 county population
6	POP15	Year 2015 county population
7	POP20	Year 2020 county population
8	POP25	Year 2025 county population
9	POP30	Year 2030 county population
10	POP40	Year 2040 county population

Table 33: External Station Interstate Flag File

Column	Description
1	Station node
2	Interstate flag (I = interstate, N = non-interstate)

Table 34: External Station Passenger Car Friction Factors

Column	Description
1	Travel Time
2	Friction factors for interstate work trips
3	Friction factors for interstate non-work trips
4	Friction factors for non-interstate work trips
5	Friction factors for non-interstate non-work trips

Table 35: EE Passenger Car File

Table	Name	Description
1	EEPC	EE Passenger Cars

Table 36: EE Truck File

Table	Name	Description
1	COMEE	EE Commercial vehicle
2	MTKEE	EE Medium truck
3	HTKEE	EE Heavy truck

6.8 Commercial Vehicle and Truck Model

This section describes the input files required for the commercial vehicle, medium duty truck, and heavy duty truck models. There are six files located in the *PARAMETERS* folder which are used for forecasting truck trips in the region. These include a truck zone flag, the delta matrices applied to the forecasts, and the friction factors for distribution. File formats are provided in **Table 37** through **Table 39** and are described below:

- *TruckZones.DBF*: Zonal file flagging zones with higher than normal truck activity
- *DeltaAM.TRP*: AM period truck delta matrices
- *DeltaMD.TRP*: Midday period truck delta matrices
- *DeltaPM.TRP*: PM period truck delta matrices
- *DeltaNT.TRP*: Night period truck delta matrices
- *FFactors.PRN*: Friction factors for I-I and I-E trips by truck type

Table 37: Truck Zone Flag File

Column	Name	Description
1	Zone	Zone number
2	Flag	Truck zone flag (1 = truck zone, 0 = non-truck zone)

Table 38: Truck Delta Matrices Example

Table	Name	Description
1	COM	Commercial vehicle delta matrix
2	MTK	Medium duty truck delta matrix
3	HTK	Heavy duty truck delta matrix

Table 39: Truck Model Friction Factors

Column	Name
1	Time
2	Commercial vehicle I-I friction factors
3	Medium duty truck I-I friction factors
4	Heavy duty truck I-I friction factors
5	Commercial vehicle I-E friction factors
6	Medium duty truck I-E friction factors
7	Heavy duty truck I-E friction factors

6.9 CT-RAMP Input Files

The CT-RAMP input files are located in the *ctrampModels* folder. The files are summarized in **Table 40**. 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*.

Table 40: CT-RAMP Model Input Files

File Name	Description
Accessibility_utility.xls	Accessibility calculation UEC
Accessibility_utility_constants.xls	Additional constants for accessibility calculation
AtWorkSubtourFrequency.xls	At work subtour frequency model UEC
AutoOwnership.xls	Auto ownership model UEC
cbd_parking_zones.csv	CBD parking zone alternatives
CoordinatedDailyActivityPattern.xls	Coordinated daily activity pattern model UEC
DestinationChoice.xls	Destination choice model UECs (mandatory, maintenance, discretionary, at work subtour)
DestinationChoiceAlternatives.csv	Destination choice models alternatives (including walk market segments)
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
Mandatory_accessibility_alts.csv	Accessibilities defined for mandatory purpose
ModeChoice.xls	Mode choice model UECs (mandatory, non-mandatory, at work subtour)
nonMandatory_accessibility_alts.csv	Accessibilities defined for non-mandatory purpose
Parklocation.xls	Parking location choice model UEC (mandatory, non-mandatory)
StopDepartureAndDurationStage1.xls	Stop timing, stage 1 model UEC
StopDepartureAndDurationStage2.xls	Stop timing, stage 2 model UEC
StopDepartureTimeStage1Alternatives.csv	Stop timing choice alternatives for stage 1 model
StopDepartureTimeStage2Alternatives.csv	Stop timing choice alternatives for stage 2 model

Table 40 (Cont'd): CT-RAMP Model Input Files

File Name	Description
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
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)
TourDepartureAndDurationAlternatives.csv	Tour departure time and duration models alternatives
TripDepartHalfHourPercents.csv	Trip depart time percents by tour purpose, inbound/outbound, tour hour and trip index
TripModeChoice.xls	Trip mode choice model UECs (mandatory, non-mandatory, work subtour)
TravelTime.xls	Travel time, distance, toll/fare, transit wait time, transit access time, transit boarding, and transit in-vehicle time skim lookup by mode and time period

6.10 Miscellaneous File

The highway assignment procedures within the model include the requirement that pass-through heavy duty trucks make use of I-285 rather than the facilities inside I-285. This is done with a database flagging the zones inside and outside of I-285 and is located in the *PARAMETERS* folder (*I285_ZONES.DBF*). The format is shown in **Table 41**.

Table 41: I-285 Zone File

Column	Name	Description
1	ZONE	Zone Number
2	I285	I-285 flag (1 = inside I-285, 2 = outside I-285)

7 Model Outputs

This section documents the output files that are used on a regular basis when reviewing model results. It includes attributes added to the highway networks, travel time skims, CT-RAMP output files, and trip tables.

7.1 Highway Networks

The list of highway networks created throughout the model process is provided in **Table 42**. Most of the networks are intermediate files which are referenced in later model steps. The attributes that are added during a model run are provided in **Table 43.** Note that not all of the attributes can be found in all the networks listed. For example, the volumes by vehicle type will only be located in the loaded highway networks.

Table 42: List of Highway Networks

Network	Description
ARC{year}HY_A.NET	Intermediate network with area types
HWY{year}FF.NET	Intermediate network with speeds, capacities
HWY{year}LOOP1_PK.NET	Intermediate network for first loop (peak)
HWY{year}LOOP1_OP.NET	Intermediate network for first loop (off-peak)
LOD{year}EA_LOOP#.NET (AM,MD,PM,EV)	Loaded network by feedback loop number and period
AVGLOAD_EA#.NET (AM,MD,PM,EV)	Average loaded network between loops by period
LOD{year}EA.NET (AM,MD,PM,EV)	Last loaded feedback loop network by period
ARC{year}EA_TRN.NET (AM,MD,PM,EV)	Includes congested bus speeds and bus times
LOD{year}EA_FINAL.NET, (AM,MD,PM)	Loaded time of day networks with drive access to transit trips loaded

Table 43: Highway Network Added Attributes

Variable	Description
WALKTIME	link walk time (minutes)
TAZ	associated TAZ
ATYPE	1 = CBD / Very High Density Urban
	2 = High Density Urban
	3 = Medium Density Urban
	4 = Low Density Urban
	5 = Suburban
	6 = Exurban
	7 = Rural
LINKCLASS	attribute for assignment (FACTYPE + 1)
FIXED	flag for fixed or per mile toll (0 = per mile, 1 = fixed)
TOLLEA, AM, MD, PM, EV	toll charge for link (in $)
EACAPACITY, AM, MD, PM, EV	link hourly capacity by time period
SPEED	free-flow speed
TIME1	free-flow time
TIME_1	congested time
VC_1	volume to capacity ratio
V_SOVEA, AM, MD, PM, EV	single occupancy vehicle volumes by period
V_HOVEA, AM, MD, PM, EV	total shared ride vehicle volumes by period
V_HOV2EA, AM, MD, PM, EV	2 person per car vehicle volumes by period
V_HOV3EA, AM, MD, PM, EV	3+ person per car vehicle volumes by period
V_COMEA, AM, MD, PM, EV	commercial vehicle volumes by period
V_MTKEA, AM, MD, PM, EV	medium duty truck volumes by period
V_HTKEA, AM, MD, PM, EV	heavy duty truck volumes by period
V_TOTEA, AM, MD, PM, EV	total vehicular volume by time period
V_TRKEA, AM, MD, PM, EV	total truck volume by time period (com + mtk + htk)
V_PNREA, AM, MD, PM	PNR access volume
V_KNREA, AM, MD, PM	KNR access volume
CGSTDSPD	congested speed
LOS	level of service
BUSSPD	bus speed
BUSTIME	bus time (minutes)

7.2 Highway and Transit Skims

During model application, a number of highway and transit skim files are created by time period which provide information regarding zone to zone travel times, distances, etc. for both highway and transit travel. A list of the most important files is provided in **Table 44.** There are other matrices created for different applications in the model; however, all of them are built from the files listed. **Table 45** through **Table 48** shows the format of the highway and transit skims. The distance and IVT tables include intra-zonal distance and time.

The format of the transit skim tables is different because the model now uses PT rather than TRNBUILD. Previously, all transit travel times were in hundreds (i.e. 1 minute was expressed as 100). With the switch to PT, the travel times are in real units. The transit skims are stratified differently in this model. Before the transit skims were stratified by local service only and combined local and premium service. In the new version, the all transit (ALLTRN) skims represents combined local bus and premium service while premium (PREMIUM) skims represents premium service only.

Table 44: Highway and Transit Skim Files

Name	Description
SOV_FREE_EA.SKM (AM,MD,PM,EV)	SOV non-toll highway skims by period
SOV_TOLL_EA.SKM (AM,MD,PM,EV)	SOV toll available highway skims by period
HOV2_FREE_EA.SKM (AM,MD,PM,EV)	HOV2 non-toll highway skims by period
HOV2_TOLL_EA.SKM (AM,MD,PM,EV)	HOV2 toll available highway skims by period
HOV3_FREE_EA.SKM (AM,MD,PM,EV)	HOV3+ non-toll highway skims by period
HOV3_TOLL_EA.SKM (AM,MD,PM,EV)	HOV3+ toll available highway skims by period
WLK_ALLTRN_WLK_EA.SKM (AM,MD,PM,EV)	Walk access/egress to all transit
WLK_PRMTRN_WLK_EA.SKM (AM,MD,PM,EV)	Walk access/egress to premium transit
KNR_ALLTRN_WLK_EA.SKM (AM,MD)	KNR access/ walk egress to all transit
KNR_PRMTRN_WLK_EA.SKM (AM,MD)	KNR access/ walk egress to premium transit
WLK_ALLTRN_KNR_MD.SKM (PM)	walk access/ KNR egress to all transit
WLK_PRMTRN_KNR_MD.SKM (PM)	walk access/ KNR egress to premium transit
PNR_ALLTRN_WLK_EA.SKM (AM,MD)	PNR access/ walk egress to all transit
PNR_PRMTRN_WLK_EA.SKM (AM,MD)	PNR access/ walk egress to premium transit
WLK_ALLTRN_PNR_MD.SKM (PM)	walk access/ PNR egress to all transit
WLK_PRMTRN_PNR_MD.SKM (PM)	walk access/ PNR egress to premium transit

Table 45: Highway Non-Toll Skim Tables

Name	Description
Toll	toll cost ($)
Distance	distance (miles)
Time	time (minutes)

Table 46: Highway Toll Eligible Skim Tables

Name	Description
Toll	toll cost ($)
Distance	distance (miles)
Time	time (minutes)
TollDistance	Toll distance (miles)

Table 47: Walk Access / Walk Egress Transit Skim Tables

Table	Name	Description
1	WALK	walk time
2	IWAIT	initial wait time
3	XWAIT	transfer wait time
4	LOCAL	local bus time
5	XBUS	express bus time
6	HRT	heavy rail time
7	BRT	bus rapid transit time
8	LRT	light rail time
9	COMRAIL	commuter rail time
10	BRDS	boardings
11	FARE	fare
12	XPEN	transfer penalties
13	IVT	transit in-vehicle time
14	DIST	OD transit distance

Table 48: Drive Access or Egress Transit Skim Tables

Table	Name	Description
1	WALK	walk time
2	AUTO	KNR/PNR time
3	IWAIT	initial wait time
4	XWAIT	transfer wait time
5	LOCAL	local bus time
6	XBUS	express bus time
7	HRT	heavy rail time
8	BRT	bus rapid transit time
9	LRT	light rail time
10	COMRAIL	commuter rail time
11	BRDS	boardings
12	FARE	Fare
13	DRVDIST	KNR/PNR distance
14	XPEN	transfer penalties
15	IVT	transit in-vehicle time
16	DIST	OD transit distance

7.3 Air Passenger

The air passenger model has also been converted to Cube script from FORTRAN. However, the file formats are the same as in previous versions. The air passenger model generates a vehicle and transit trip table. The vehicle trip table is called *VEHOUT{year}.MTT* and the format is provided in **Table 49**. The transit trip table is called *TRNOUT{year}.MTT* with the format provided in **Table 50**.

Table 49: Air Passenger Vehicle Tables

Table	Name	Description
1	BusRes	Business residents
2	BusNR	Business non-residents
3	NBRes	Non-business residents
4	NBNR	Non-business non-residents
5	TotVeh	Total vehicles

Table 50: Air Passenger Transit Tables

Table	Name	Description
1	BusRes	Business residents
2	BusNR	Business non-residents
3	NBRes	Non-business residents
4	NBNR	Non-business non-residents
5	WlkTrn	Walk to transit
6	PnrTrn	PNR transit
7	KnrTrn	KNR transit

7.4 Externals

Portions of the external model were previously applied in FORTRAN executables. In the new model, all components of the externals are handled within the Cube script. The internal-external station passenger car productions are created in a text file called *IEPRD{year}.PRN* with the format shown in **Table 51.** The internal-external passenger car attractions are stored in a text file called *IEPA_B{year}_ATT.NEW* with the format shown in **Table 52.**

Table 51: Internal-External Passenger Car Productions

Column	Description
1	Zone
2	IE work – interstate
3	IE non-work interstate
4	IE work - non-interstate
5	IE non-work non-interstate

Table 52: Internal-External Passenger Car Attractions

Column	Description
1	Zone
2	IE work – interstate
3	IE non-work interstate
4	IE work - non-interstate
5	IE non-work non-interstate

After trip distribution, the internal-external passenger car trips are stored in two trip tables:

- *IEWRKM{year}.VTT*: internal-external passenger car work trips
- *IENWKM{year}.VTT*: internal-external passenger car non-work trips

The basic format of the two trip tables is the same although the table names are slightly different. The formats are provided in **Table 53**. The external-external passenger car output is a trip table called *EEPC20{year}.VTT* shown in **Table 54**.

Table 53: Internal-External Passenger Car Trip Tables

Table	Name	Description
1	PCWRK1 / PCNWK1	I-E interstate passenger car
2	PCWRK2 / PCNWK2	I-E non-interstate passenger car

Table 54: External-External Passenger Car Trip Table

Table	Name	Description
1	EEPC	E-E passenger car

The internal-external trip ends for commercial vehicle, medium truck, and heavy trucks are created and stored in the text file *CMHEXT.PRN*. The format of this file is shown in **Table 55**. The external model also creates the initial external-external vehicle tables which are later modified in the commercial vehicle and truck model. The name of the initial file is called *CMHEE.TRP* and the format is shown in **Table 56**.

Table 55: External Station Truck Trip File

Column	Description
1	Zone
2	External station commercial vehicle
3	External station medium duty truck
4	External station heavy duty truck

Table 56: External-External Truck Trip Tables

Table	Name	Description
1	COMEE	E-E commercial vehicles
2	MTKEE	E-E medium trucks
3	HTKEE	E-E heavy trucks

7.5 Commercial Vehicle and Trucks

The file formats of the commercial vehicle and truck model have not changed in the newer model version. The program produces an initial internal-internal and internal-external file called *CMH.TRP* prior to implementing the truck delta matrices. Note that this file is not used for assignment. The file format is provided in **Table 57**. After the delta matrix application, the model creates three files for each truck type:

- *COM.TRP*: commercial vehicles
- *MTK.TRP*: medium duty trucks
- *HTK.TRP*: heavy duty trucks

Table 57: Initial Internal-Internal / Internal-External Truck Trip Table

Table	Name	Description
1	COMII	I-I commercial vehicle
2	MTKII	I-I medium truck
3	HTKII	I-I heavy truck
4	COMEXT	I-E commercial vehicle
5	MTKEXT	I-E medium truck
6	HTKEXT	I-E heavy truck

Table 58: Final Truck Tables

Table	Name	Description
1	EACOM, EAMTK, EAHTK	EA period trips
2	AMCOM, AMMTK, AMHTK	AM period trips
3	MDCOM, MDMTK, MDHTK	MD period trips
4	PMCOM, PMMTK, PMHTK	PM period trips
5	EVCOM, EVMTK, EVHTK	EV period trips

7.6 CT-RAMP Output Files

The core outputs from the CT-RAMP model component are summarized in **Table 59.**

Table 59: CT-RAMP Model Output Files

File Name	Description
ShadowPricing_{loop}_{shadowPricingIterations}.csv	Shadow pricing results
wsLocResults.csv	Usual work and school location choice results
aoResults.csv	Auto ownership results
cdapResults.csv	CDAP model results
hhData_{loop}.csv	Household attribution results
personData_{loop}.csv	Person attribution 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
tripData_{loop}.csv	Individual + expanded joint trips with select HH, person, and travel time fields (optional output to be used for ABMVIZ)

7.6.1 Shadow Pricing Results

The shadow pricing results for each shadow pricing iteration are written to the file, *ShadowPricing_ {loop}_{shadowPricingIterations}.csv*.

7.6.2 Usual Work and School Location Choice Results

The usual work and school location choice results are written to *wsLocResults_{loop}.csv*. The results file consists of the data fields shown in **Table 60**. The person type in the table is defined in **Table 61**.

Table 60: Usual Work and School Location Choice Output File Fields

Field	Description
HHID	household ID
HomeTAZ	Home TAZ
HomeSubZone	Home subzone
Income	Income
PersonID	Person ID
PersonNum	Person number in HH
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 61: Person Type Codes

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

7.6.3 Auto Ownership Results

The number of autos per household is written to *aoResults.csv*.

7.6.4 Household Attribution Results

The household attribution results for each feedback loop are written to *hhData_{loop}.csv*. The file has the following fields:

Table 62: Household Output File Fields

Field	Description
hh_id	Household ID
taz	Origin TAZ
walk_subzone	Walk subzone
income	HH income
autos	Number of autos
jtf_choice	Joint tour frequency choice
size	HH size
workers	Number of workers in HH
auto_suff	Auto sufficiency
ao_rn	Random number used in the auto ownership model
fp_rn	Random number used in the free parking model
cdap_rn	Random number used in the daily activity pattern model
imtf_rn	Random number used in the individual mandatory tour frequency model
imtod_rn	Random number used in the individual mandatory tour time-of-day model
immc_rn	Random number used in the individual mandatory tour model choice model
jtf_rn	Random number used in the joint tour frequency model
jtl_rn	Random number used in the joint tour location choice model
jtod_rn	Random number used in the joint tour time-of-day model
jmc_rn	Random number used in the joint tour model choice model
inmtf_rn	Random number used in the individual non-mandatory tour frequency model
inmtl_rn	Random number used in the individual non-mandatory tour location choice model
inmtod_rn	Random number used in the individual non-mandatory tour time-of-day model
inmmc_rn	Random number used in the individual non-mandatory tour model choice model
awf_rn	Random number used in the after work tour frequency model
awl_rn	Random number used in the after work location choice model
awtod_rn	Random number used in the after work tour time-of-day model
awmc_rn	Random number used in the after work tour model choice model
stf_rn	Random number used in the stop frequency model
stl_rn	Random number used in the stop location choice model

7.6.5 Person Attribution Results

The person attribution results for each feedback loop are written to *personData_{loop}.csv*. The file has the following fields:

Table 63: Person Output File Fields

Field	Description
hh_id	Household ID
person_id	Person ID
person_num	person number in HH
age	age
gender	gender
person_type	person type (worker, student, etc)
pecas_occ	occupation category defined in PECAS
fp_choice	free parking choice
activity_pattern	activity pattern
imf_choice	individual mandatory tour freq choice
inmf_choice	individual non-mandatory tour freq choice

7.6.6 Coordinated Daily Activity Pattern (CDAP) Results

The results of the CDAP model are written to *cdapResults_{loop}.csv*.

Table 64: 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)

7.6.7 Individual Tour Records

Individual tours for each CT-RAMP feedback loop are written to *indivTourData_{loop}.csv*. The final feedback loop version is written to *indivTourData.csv*. **Table 65** lists data fields in the output file. The tour mode used in this file is categorized in **Table 66**.

Table 65: 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 taz location
orig_walk_segment	tour origin subzone
dest_taz	tour destination taz location
dest_walk_segment	tour destination subzone
start_period	tour starting time period
end_period	tour ending time 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
outstop_#_mode	mode of outbound stop # of tour
outstop_#_period	time period of outbound stop # of tour
outstop_#_taz	taz location of outbound stop # of tour
outstop_#_purpose	purpose of outbound stop # of tour
instop_#_mode	mode of inbound stop # of tour
instop_#_period	time period of inbound stop # of tour
instop_#_taz	taz location of inbound stop # of tour
instop_#_purpose	purpose of inbound stop # of tour
util_#; prob_#	utility for mode #; probability for mode #

Table 66: Trip/Tour Mode Codes

Trip/Tour Mode Code	Description	UEC Label
1	Drive alone free	DRIVEALONEFREE
2	Drive alone pay	DRIVEALONEPAY
3	Shared ride 2 free	SHARED2FREE
4	Shared ride 2 pay	SHARED2PAY
5	Shared ride 3+ free	SHARED3FREE
6	Shared ride 3+ pay	SHARED3PAY
7	Walk	WALK
8	Bike	BIKE
9	Walk to local/premium transit	WALK_ALLTRN
10	Walk to premium transit	WALK_PRMTRN
11	Park-and-Ride to local/premium transit	PNR_ALLTRN
12	Park-and-Ride to premium transit	PNR_PRMTRN
13	Kiss-and-Ride to local/premium transit	KNR_ALLTRN
14	Kiss-and-Ride to premium transit	KNR_PRMTRN
15	School bus	SCHOOL_BUS

7.6.8 Joint Tour Records

Joint tours for each CT-RAMP feedback loop are written to *jointTourData_{loop}.csv*. The final feedback loop version is written to *jointTripData.csv*. **Table 67** lists data fields in the output file.

Table 67: 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 taz location
orig_walk_segment	tour origin subzone
dest_taz	tour destination taz location
dest_walk_segment	tour destination subzone
start_period	tour starting period
end_period	tour ending 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
outstop_#_mode	mode of outbound stop # of tour
outstop_#_period	time period of outbound stop # of tour
outstop_#_taz	taz location of outbound stop # of tour
outstop_#_purpose	purpose of outbound stop # of tour
instop_#_mode	mode of inbound stop # of tour
instop_#_period	time period of inbound stop # of tour
instop_#_taz	taz location of inbound stop # of tour
instop_#_purpose	purpose of inbound stop # of tour
util_#	utility for mode #
prob_#	probability of mode #

7.6.9 Individual Trip Records

Individual trips for each CT-RAMP feedback loop are written to *indivTripData_{loop}.csv*. The final feedback loop version is written to *indivTripData.csv*. **Table 68** lists data fields in the output file.

Table 68: 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_taz	Origin TAZ
orig_walk_segment	Origin walk market segment
dest_taz	Destination TAZ
dest_walk_segment	Destination walk market segment
parking_zone	Parking TAZ used
depart_period	Departure time period
trip_mode	Trip Mode
tour_mode	Tour mode
tour_category	Tour category

7.6.10 Joint Trip Records

Joint trips for each CT-RAMP feedback loop are written to *jointTripData_{loop}.csv*. The final feedback loop version is written to *jointTripData.csv*. **Table 69** lists data fields in the output file.

Table 69: 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_taz	Origin TAZ
orig_walk_segment	Origin walk market segment
dest_taz	Destination TAZ
dest_walk_segment	Destination walk market segment
parking_zone	Parking TAZ used
depart_period	Departure time period
trip_mode	Trip Mode
Num_participants	Number of participants on tour
tour_mode	Tour mode
tour_category	Tour category

7.6.11 Trip Data

The trips table, *tripData_{loop}.csv*, consists of the individual trips and the joint trips in expanded form. It also contains joined household and person fields.

Table 70: CT-RAMP Trips Data Output File Fields

Field	Description
hh_id	Unique HH ID
person_id	Unique Person ID
person_num	Person number within household
tour_id	Non-unique tour ID given person_id and tour_category
stop_id	Trip stop ID on the tour (-1 = no stop, else 0-3)
inbound	Inbound half tour (1=True, 0=False)
tour_category	Tour type such as INDIVIDUAL_NON_MANDATORY
tour_purpose	Tour purpose such as social
orig_purpose	Activity at trip origin
dest_purpose	Activity at trip destination
orig_taz	Trip origin taz
dest_taz	Trip destination taz
orig_walk_segment	Trip origin taz walk subzone type
dest_walk_segment	Trip destination taz walk subzone type
parking_taz	Destination parking taz (0=does not apply)
depart_period	Trip departure time period
trip_mode	Trip mode code
tour_mode	Tour mode code
num_participants	Number of persons on the trip
tour_participants	Space concatenated person_num(s) on the tour
tour_start_period	Tour start time period
tour_id_uniq	Unique tour ID (tour_category + “ “ + tour_id + “ “ + tour_purpose) given person_id
trip_id	Trip ID within each full tour (1 to N).
orig_purpose_start_period	Origin activity start time period (if first trip, then equal to first half-hour of the simulation day)
home_taz	Home taz
hh_income	Household income
hh_autos	Household autos

Table 70 (Cont'd): CT-RAMP Trips Data Output File Fields

Field	Description
hh_fp_choice	Household free parking choice model result (1 =free, 2=pay)
hh_inc_bin	Household income bin
hh_size	Household size
hh_wkrs	Household workers (person types "Full-time worker" and "Part-time worker")
hh_auto_suff	Household Auto sufficiency (cars<wkrs, cars>wkrs, nocars)
age	Person age
gender	Person gender
person_type	Person type such as non-worker
activity_pattern	Person activity pattern
tour_mode_name	Tour mode name
trip_mode_name	Trip mode name
travel_time	Trip travel time
distance	Trip distance
cost	Trip auto toll or transit fare
first_wait_time	Transit trip first wait time
xfer_wait_time	Transit trip transfer wait time
walk_aux_time	Transit trip walk access and egress time
auto_aux_time	Transit trip auto access and egress time
boardings	Transit boardings
ivt_local_bus	Transit trip in-vehicle time on local bus
ivt_premium_modes	Transit trip in-vehicle time on premium modes

7.6.12 CT-RAMP Tables

The CT-RAMP trip files are converted from CSV format to Cube matrices via Cube Voyager script. For highway modes, the trips represent vehicles whereas other modes represent person trips. The tables are named as follows:

- *TRIPS_INDIV_EA.TPP (AM, MD, PM, EV)*: individual trip tables by period
- *TRIPS_JOINT_EA.TPP (AM, MD, PM, EV)*: joint trip tables by period
- *TRIPSEA.TPP (AM, MD, PM, EV)*: combined trip tables by period

The tables are split by travel mode and are formatted as shown in **Table 71.**

Table 71: Trip Table Format

Table	Name	Description
1	DA	non-toll drive alone
2	DATOLL	toll eligible drive alone
3	SR2	non-toll shared ride 2
4	SR2TOLL	toll eligible shared ride 2
5	SR3	non-toll shared ride 3+
6	SR3TOLL	toll eligible shared ride 3+
7	WALK	walk
8	BIKE	bicycle
9	WLKALLWLK	walk access/egress all transit
10	WLKPRMWLK	walk access/egress premium transit
11	PNRALLWLK	PNR access / walk egress all transit
12	WLKALLPNR	walk access / PNR egress all transit
13	PNRPRMWLK	PNR access / walk egress premium transit
14	WLKPRMPNR	walk access / PNR egress premium transit
15	KNRALLWLK	KNR access / walk egress all transit
16	WLKALLKNR	walk access / KNR egress all transit
17	KNRPRMWLK	KNR access / walk egress premium transit
18	WLKPRMKNR	walk access / KNR egress premium transit
19	SCHLBUS	school bus

7.7 Time-of-Day Trip Tables

This section details the format of the time-of-day (TOD) vehicle trip tables which are used in the period level highway assignments. The passenger car trip tables are stored in the following five files in the format shown in **Table 72**.

- *TODEA{year}.VTT*: EA period passenger car trip table
- *TODAM{year}.VTT*: AM period passenger car trip table
- *TODMD{year}.VTT*: Midday period passenger car trip table
- *TODPM{year}.VTT*: PM period passenger car trip table
- *TODEV{year}.VTT*: Evening / late night period passenger car trip table

Table 72: Passenger Car TOD Trip Table File

Table	Name	Description
1	SOV	Single occupancy vehicles
2	HOV	Total shared ride vehicles
3	HOV2	2 / car vehicles
4	HOV3	3+ / car vehicles

The TOD vehicle trip files utilized for the highway assignment include the split by occupancy and toll/non-toll for each period.

- *TODEA{year}_asgn.VTT*: EA period vehicle trip table
- *TODAM{year}_asgn.VTT*: AM period vehicle trip table
- *TODMD{year}_asgn.VTT*: Midday period vehicle trip table
- *TODPM{year}_asgn.VTT*: PM period vehicle trip table
- *TODEV{year}_asgn.VTT*: Evening / late night period vehicle trip table

Table 73: TOD Trip Table Assignment File

Table	Name	Description
1	SOVF	Non-toll SOV
2	SOVT	Toll eligible SOV
3	HOV2F	Non-toll HOV 2
4	HOV2T	Toll eligible HOV 2
5	HOV3F	Non-toll HOV 3+
6	HOV3T	Toll eligible HOV 3+

7.8 Transit Assignment Output

This section details the format of the transit assignment output which include time of day, mode of access/egress, and the mix of transit modes. The primary assignment outputs used for route analysis are named as follows:

@access mode@_@transit mix@_@egress mode@_@period@.DBF

- Access mode = WLK, KNR, or PNR
- Transit mix = ALLTRN or PRMTRN
- Period = EA, AM, MD, PM, EV, AIRPASS

The output files are standard Public Transport output files from assignment loadings. Please refer to the Cube User guide for an explanation of the attributes in the output file.

Table 74: Transit Assignments

Time Period	Access	Egress	Transit Mix
Early AM	Walk	Walk	All transit
	Walk	Walk	Premium transit only
	KNR	Walk	All transit
	KNR	Walk	Premium transit only
	PNR	Walk	All transit
	PNR	Walk	Premium transit only
AM Peak	Walk	Walk	All transit
	Walk	Walk	Premium transit only
	KNR	Walk	All transit
	KNR	Walk	Premium transit only
	PNR	Walk	All transit
	PNR	Walk	Premium transit only
Midday	Walk	Walk	All transit
	Walk	Walk	Premium transit only
	KNR	Walk	All transit
	KNR	Walk	Premium transit only
	PNR	Walk	All transit
	PNR	Walk	Premium transit only
	Walk	KNR	All transit
	Walk	KNR	Premium transit only
	Walk	PNR	All transit
	Walk	PNR	Premium transit only
PM Peak	Walk	Walk	All transit
	Walk	Walk	Premium transit only
	Walk	KNR	All transit
	Walk	KNR	Premium transit only
	Walk	PNR	All transit
	Walk	PNR	Premium transit only
Evening/ Late Night	Walk	Walk	All transit
	Walk	Walk	Premium transit only
Air Passenger	Walk	Walk	Premium transit only
	KNR	Walk	Premium transit only
	PNR	Walk	Premium transit only

8 Highway Network Coding

This section provides further information regarding the coding of highway projects using the new Cube geodatabase format.

8.1 Cube Geodatabase

The new model network was converted to a Cube geodatabase allowing for more efficient storage of networks and the ability to include true shape geometry for facilities. The geodatabase network is based on a conflated network to an underlying shapefile (NAVSTREETS). This network and underlying shapefile were imported into Cube's geodatabase format.

Newer versions of Cube include a Data Manager which allows managing multiple geodatabases efficiently. To access the Data Manager, click the Data Manager from the toolbar as shown in **Figure 25.** This opens the Data Manager on the right side of the Cube window. To add a geodatabase, click the Add Data button as illustrated by **Figure 26**. This will prompt the user to browse for the geodatabase location as shown in **Figure 27**. Once selecting the geodatabase, the Data Manager window will contain the geodatabase and contents of it as shown in **Figure 28** and **Figure 29**.

Figure 25: Accessing Data Manager

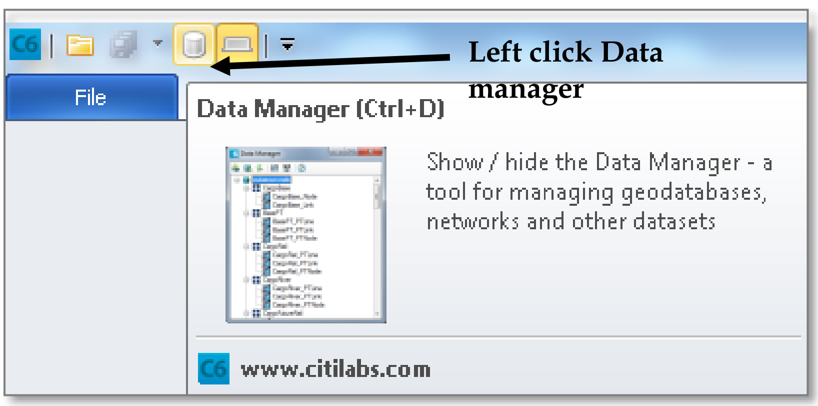


Figure 26: Adding Data

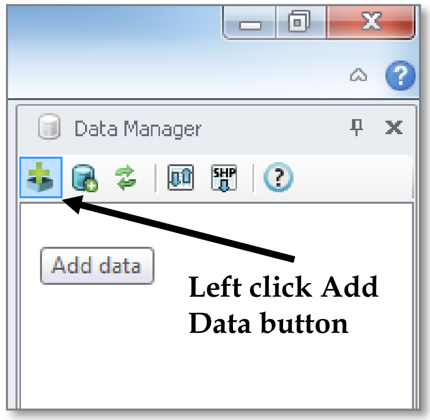


Figure 27: Browse to Geodatabase

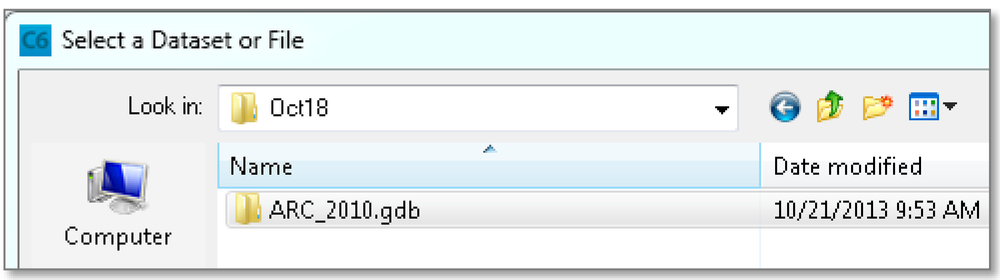


Figure 28: Data Manager with Geodatabase

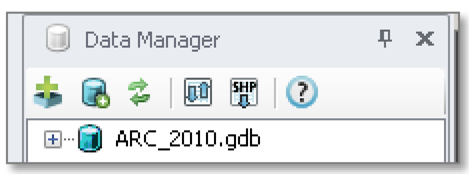
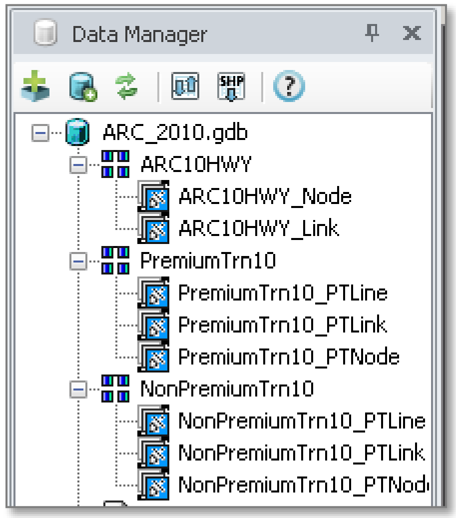
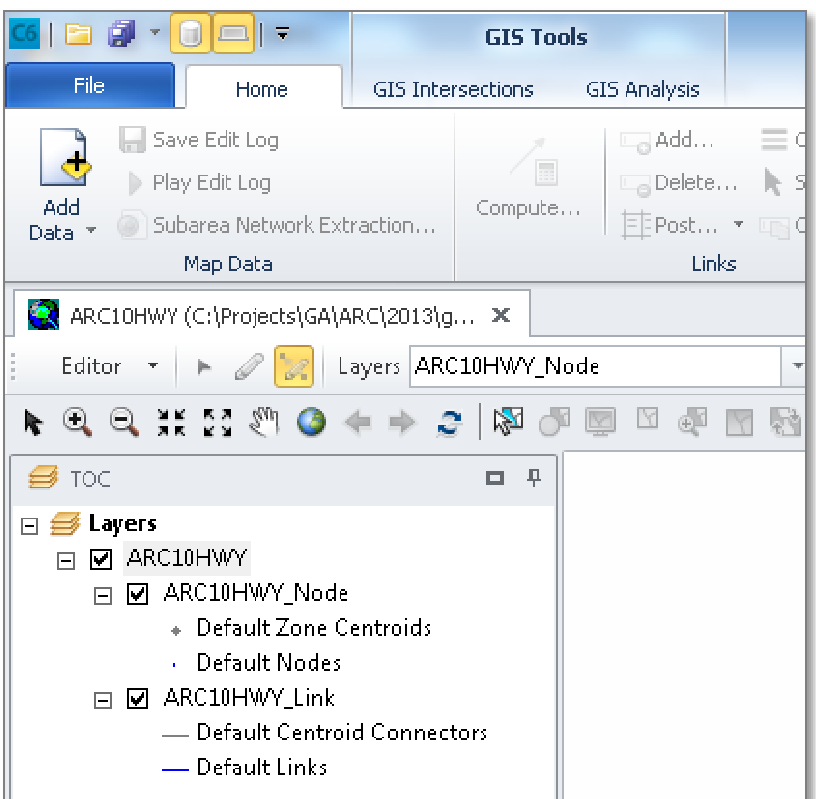


Figure 29: Files Stored in Geodatabase



To open one of the networks stored in the geodatabase, double click any one of the files. This will add the network to the Table of Contents as illustrated in **Figure 30** as well as add the data to the GIS window.

Figure 30: Table of Contents



8.2 Mapping Attributes

Many of the Cube features for mapping attributes remain the same; however, there is added functionality for users more accustomed to ArcGIS. This section covers how to map an attribute with new functionality for users that may not have experience with ArcGIS.

To utilize the ArcGIS functionality for mapping, right click on the link layer in the TOC and select Properties as illustrated in
Figure 31. After the window opens as shown in **Figure 32**, click Advanced Properties.

This will open the Advanced Properties window shown in **Figure 33**. There are a number of tabs that can be useful for mapping. The Symbology tab is useful for mapping link attributes. When this tab is selected, the user will see Features, Categories, Quantities, Charts, and Multiple Attributes in the window to the left for Symbology options. To make the network attributes one single color, select Features – Single Symbol. This will prompt the user to select the line type, color, width, etc. as shown in **Figure 34**.

A more likely scenario is creating a color coded map that differentiates between attributes such as facility types or number of lanes. In these cases, the Categories Symbology is more useful. To color code facility types for example, click Categories and select Unique Values, then select Add All Values as shown in **Figure 35**. From here, the user can modify the line type, color, width, etc. for each facility type.

Figure 31: Layer Properties

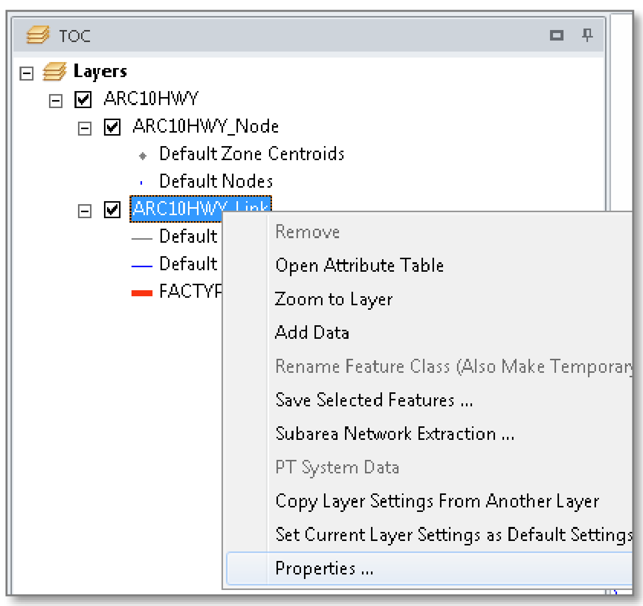


Figure 32: Highway Layer Properties Menu

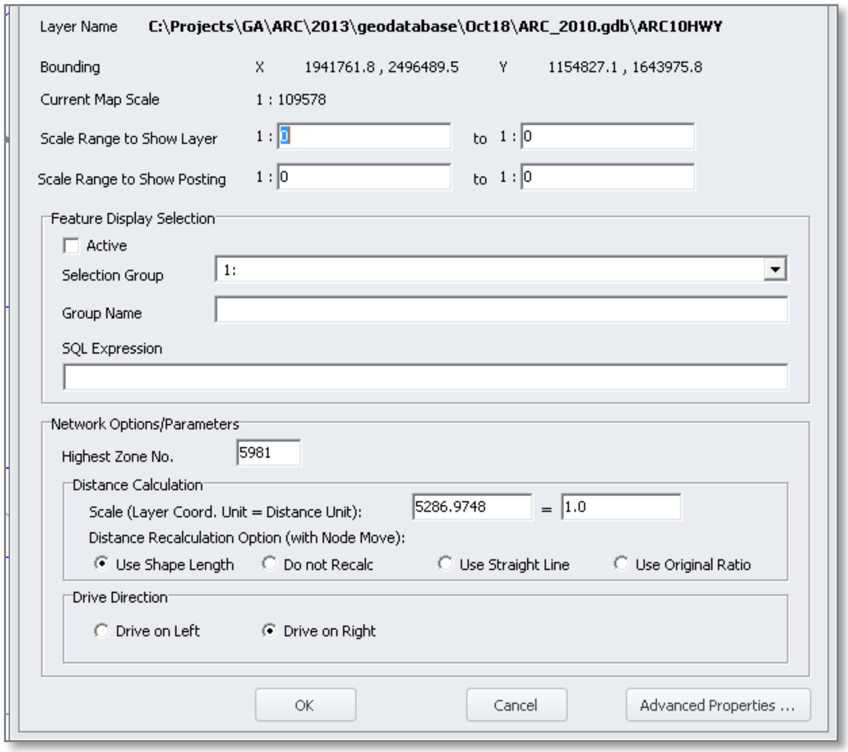


Figure 33: Advanced Properties

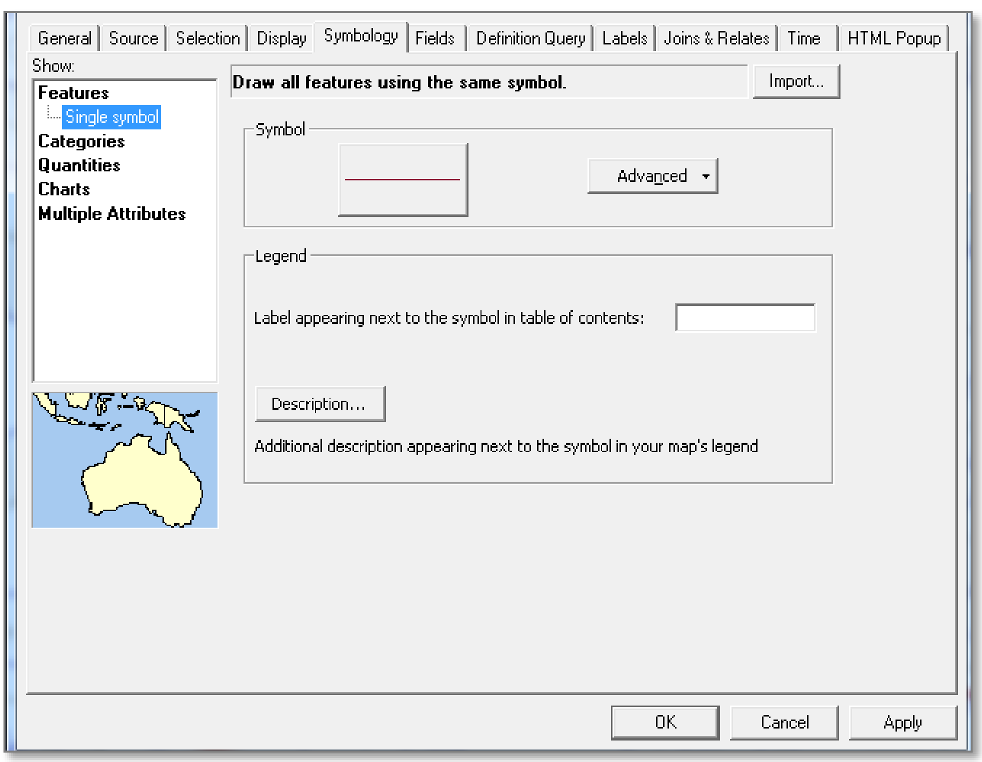


Figure 34: Single Symbol Symbology

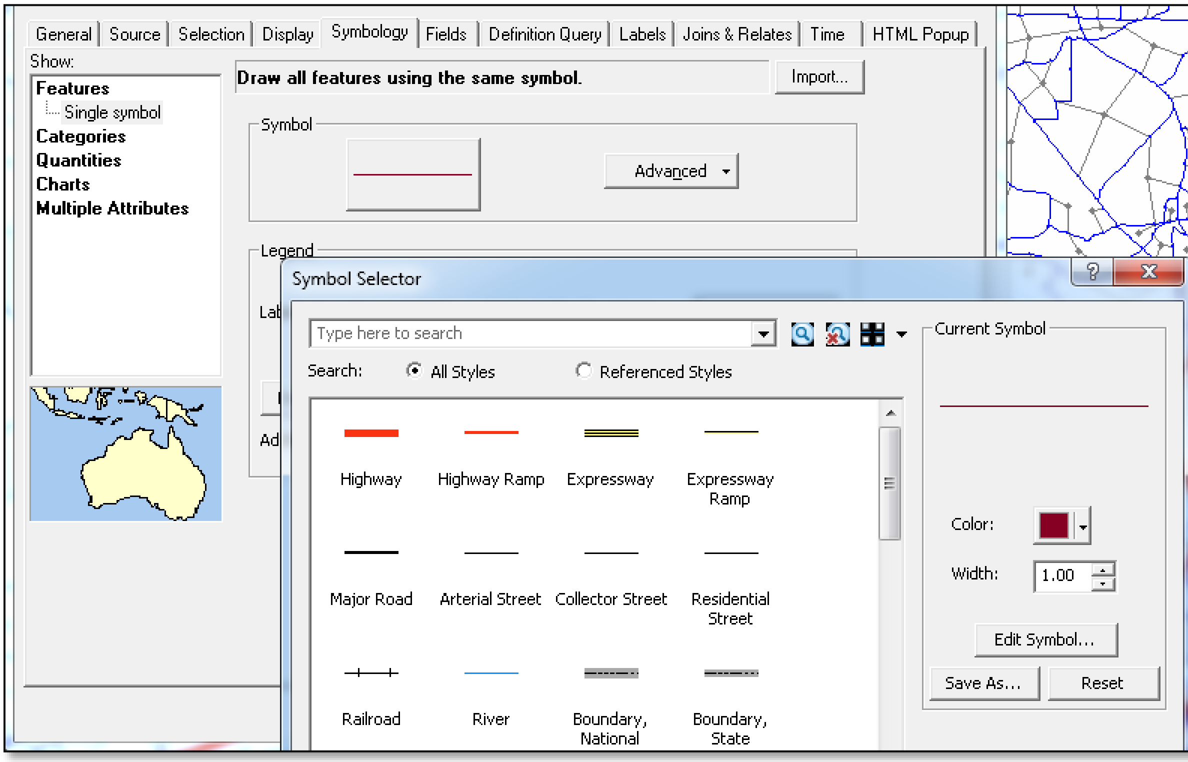
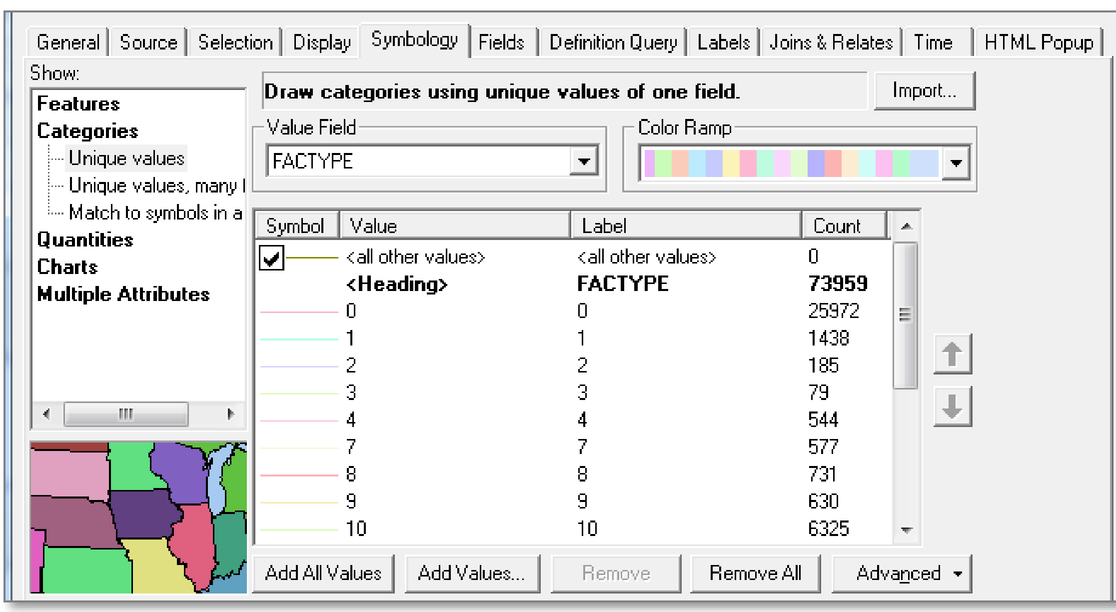


Figure 35: Categories Symbology



8.3 Coding Projects

Coding highway projects in the geodatabase structure is much different than the previous coding format. This section describes how to enable the geodatabase editor, changing link attributes, creating new facilities, one-way versus two-editing, and the use of geodatabase log files.

When working in the geodatabase structure, the user must select the layer to edit in both the Layers window and in the TOC to make it the active layer as shown in **Figure 36**. To enable editing, the user must select Editor – Start Editing as illustrated in **Figure 37**. This will activate the editor functions allowing the user to make modifications to the active layer.

Figure 36: Selecting Layer for Editing

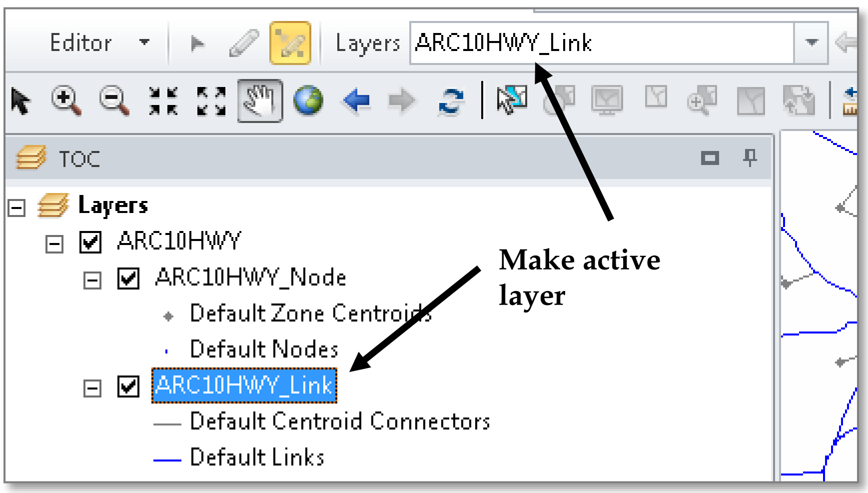


Figure 37: Enable Editing

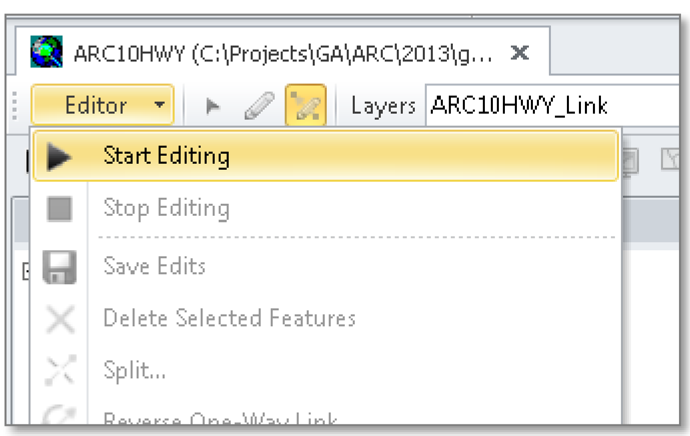
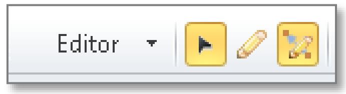


Figure 38: Editor Functions



Edit feature: for changing existing attributes (e.g. 2 to 4 lanes). The user would use this pointer to select a link to modify.

Create feature: for adding new facilities.

Two-way editing: Determines whether or not the one-way or two-way coding is enabled. If only coding one-direction, de-select this icon.

To change existing attributes, the user would select the Edit Feature icon. To select a link for editing, use the pointer and left click. This will highlight the link and shape vertices as shown in **Figure 39**. Once selected, this will also open the Feature Explorer window for the link. The Feature Explorer window contains the general directionality of the link for the a-node/b-node architecture along with the links attributes by A-B Value and B-A Value. The A-B Value is always based on which directionality is highlighted on the left side of the Feature Explorer window. Therefore, when the user is modifying only one-direction of a two-way link, careful attention must be paid to which direction is selected. To change an attribute, left click in the A-B or B-A values and type the changes.

There are two steps necessary to make the edits permanent in the geodatabase. The save and undo buttons will become activated whenever attributes are modified. To save the modifications, click the checkmark. Note that this has not permanently saved the edits to the geodatabase, but rather, temporarily saves the edits to allow the user to continue editing other links. To make the edits permanent, the user must click Editor – Save Edits as shown in **Figure 40**.

Figure 39: Feature Explorer

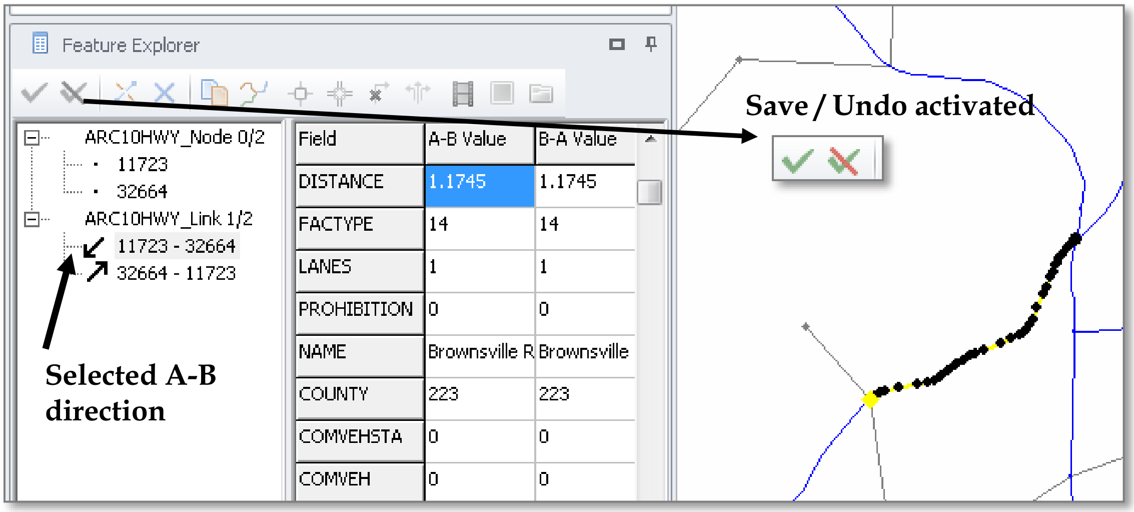
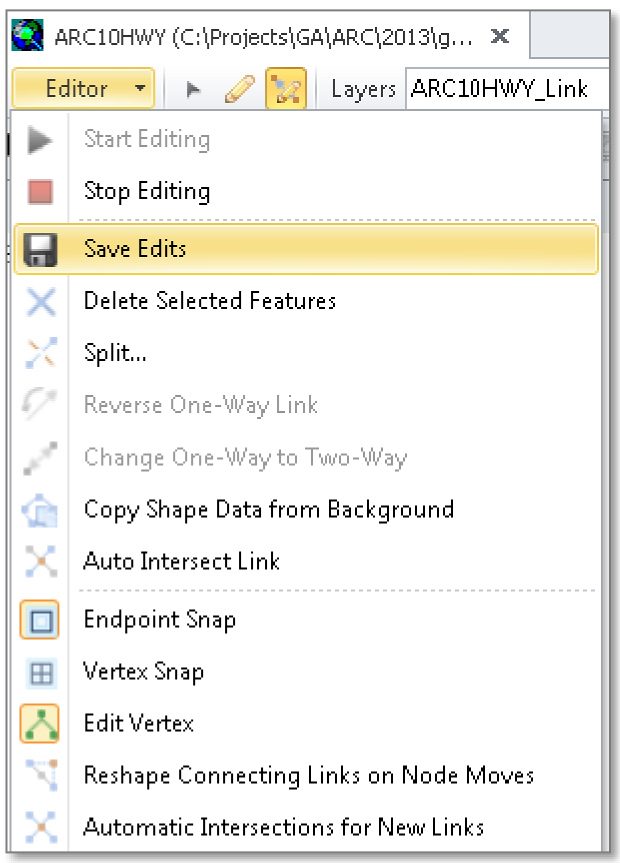


Figure 40: Save Edits



To code new facilities, the user must select the Create Feature button which looks like a pencil. Once selected, the pointer will change into a pencil. To add a link from an existing node, left click once on the existing node. To add shape vertices, left click once for each desired location. To create a link terminus, double click either at an existing node or the desired endpoint location. These steps are illustrated in **Figure 41**. To save the link, the user must click the checkmark as well as the save edits from the editor menu.

In the old editing format, the user could copy a link and then use the paste feature to ensure the attributes were the same for entire facility. The geodatabase still provides this option; however, the process is different. In the Feature Explorer window, there is an icon to Set Current Feature Attributes as Template which is shown in **Figure 42**. When adding new facilities, the user could select a similar facility and use this feature to use the attributes. Then all new links will have the same attributes as in the template. If no template is used, the new facilities will have all of its attributes set to zero for numeric fields or blank for character fields.

Figure 41: New Facility with Vertices

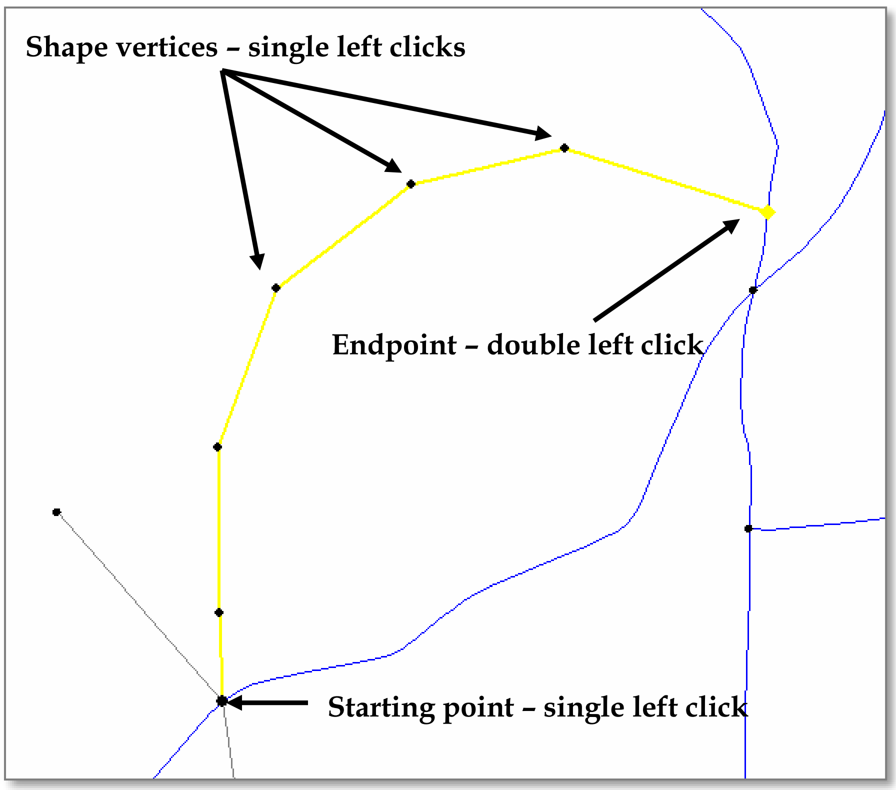
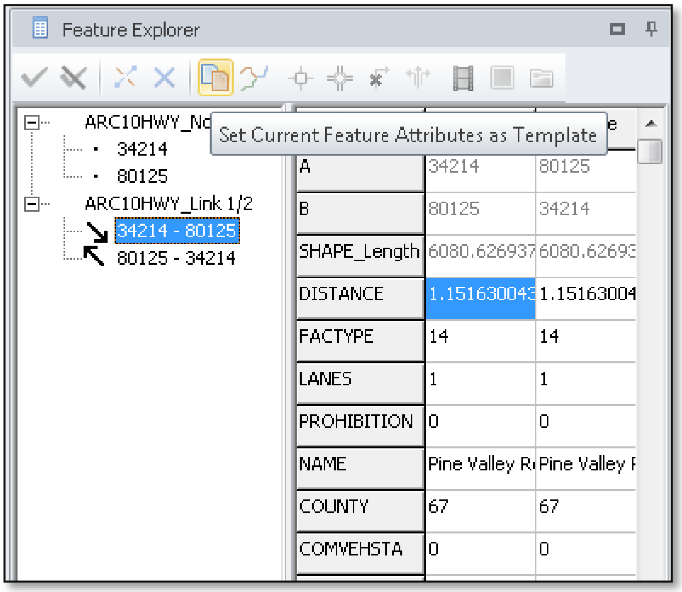
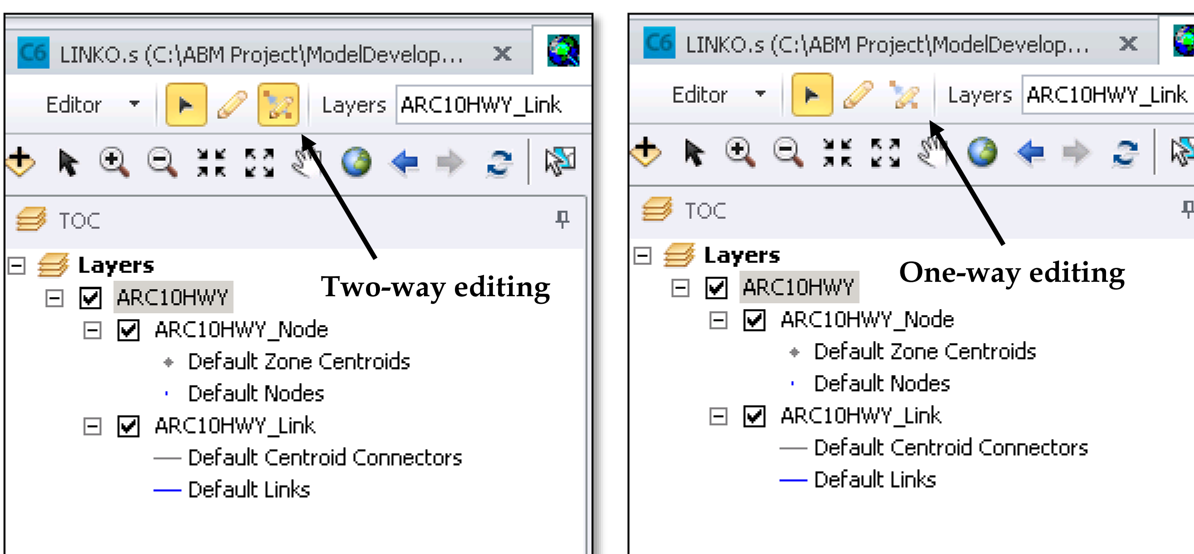


Figure 42: Feature Template



When adding new links in edit mode, it is important to understand the difference between one-way and two-editing. Once start editing has been enabled, the user can toggle off/on the two-way edit option (program defaults to two-way editing). If the user wants to add a two-way facility, then the two-way edit should be toggled to on. Conversely, if the user wishes to add a one-way facility, the two-way edit should be toggled to off, otherwise the program will inherently add in the link as two-way even if attributes are only coded in one direction. The two-editing toggle is provided in **Figure 43**.

Figure 43: Two-Way vs. One-Way Editing



When coding new projects, the user will be required to determine a project's facility type (FACTYPE). The available facility types for vehicle use and their definitions are provided below:

1. Interstate / Freeway = Limited access highway mainline serving trips traveling longer distances. These facilities do not provide direct access to land use activities. Access is limited to fully grade-separated interchanges.
   - Examples: I-75, I-85, I-20, GA-400
2. Expressway = High speed controlled access highway serving trips traveling longer distances but typically not as long as freeways. These facilities are not designed to provide direct access to land use activities. Traffic is typically separated by concrete barriers.
   - Examples: Langford Parkway, Monroe Bypass
3. Parkway = these facilities are typically not designed to freeway / expressway standards (sharper curves, steeper grades, and narrower lanes). In rural areas, a limited number of traffic signals can provide some access to land use activities. Traffic is typically separated by grass medians.
   - Examples: Buford Spring Connector, Rockmart Highway
4. Freeway HOV Buffer Separated = same as interstate / freeway but for HOV. Separated from general purpose lanes by striping.
   - Examples: I-75 HOV lanes, I-85 HOV lanes
5. Freeway HOV Barrier Separated = same as interstate / freeway but for HOV. Separated from general purpose lanes by concrete barrier.
6. Freeway Truck Only = same as interstate / freeway but for truck only.
7. System to System Ramp = Freeway to freeway ramps.
   - Examples: I-85 / I-285 Interchange, I-75 / Canton Road Connector Interchange
8. Exit Ramp = Off-ramp to a controlled intersection.
   - Examples: I-75 off ramps to Windy Hill Road, Pleasant Hill Road off ramps to Buford Highway
9. Entrance Ramp = On-ramp from a controlled intersection.
   - Examples: I-75 on ramps from Windy Hill Road, Pleasant Hill Road on ramps from Buford Highway
10. Principal Arterial = Major road with a higher emphasis on serving thru trips with less access to adjacent property. Minimum of four lanes. Common characteristics include fewer curb cuts, raised medians, and limited signal density. Turn lanes provided at most intersections.
    - Examples: Cobb Parkway, Camp Creek Parkway
11. Minor Arterial = Medium road with a balance of serving thru trips and providing access to adjacent property. Often provide movement between collectors and principal arterials. Typical cross sections of two to four lanes. Typically does not include on-street parking.
    - Examples: Howell Mill Road, Briarcliff Road
12. Arterial HOV = same as minor arterial but for HOV.
13. Arterial Truck Only = same as minor arterial but truck only.
14. Collector = Minor road with a primary purpose of providing connectivity to/from arterials and to serve adjacent properties. Does not typically include a median. May include on-street parking.
    - Examples: Joseph E Boone Boulevard, Lullwater Road, Ferst Drive

The ABM includes some new link attributes which are important to highway network coding. The traffic assignment is not an operational assignment; therefore, the action of vehicles crossing multiple lanes of traffic to access the HOV lanes from general purpose (GP) lanes is not explicitly modeled. To reflect that there is some motivation to not perform this maneuver in certain conditions (short trips, uncongested conditions), a flag was attached to the "slip-links" that connect the GP lanes with the HOV lanes called HOVMERGE where a '1' indicates the link is an HOV merge link. For those links with the flag, an additional 5 seconds is added to the travel time.

To reflect that ramps with small curve radii (i.e. loop ramps) have observed lower free-flow speeds, an attribute was added to the network called RAMPFLAG where a '1' indicates the ramp operates at lower free-flow speeds than other ramps. Ramps with this attribute will be assigned a 35 MPH free-flow speed.

When performing traffic assignments and comparing observed and estimated model speeds, it was noted that the model was generally overestimated travel speeds on freeway links near major system-to-system interchanges. These interchanges typically have high volumes of traffic transitioning from one freeway to another which causes delay associated with the impacts of operational movements such as weaving through multiple lanes of traffic. As mentioned previously, the ABM traffic assignment is not an operational assignment which limits its ability to accurately reflect the operational issues that cause delay. To mitigate these issues, a link attribute was added to the network to identify the freeway approach and departure links at these major interchanges called WEAVEFLAG where a '1' indicates the link as a weaving section. The volume-delay relationships on links with this flag are applied with a different curve than normal freeway sections where the modified curve results in more speed reduction at lower volume to capacity ratios than a normal freeway section.

Figure 44: HOV Merge Coding

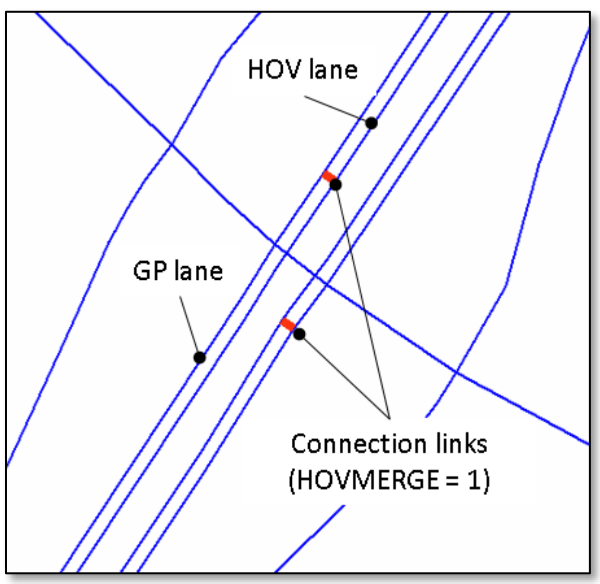


Figure 45: Loop Ramp Coding

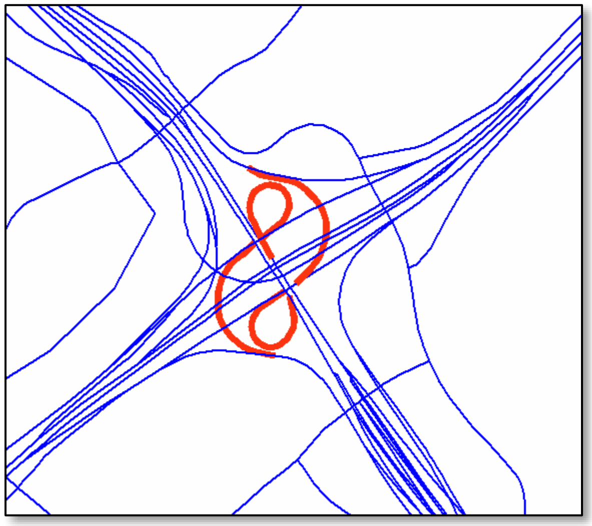
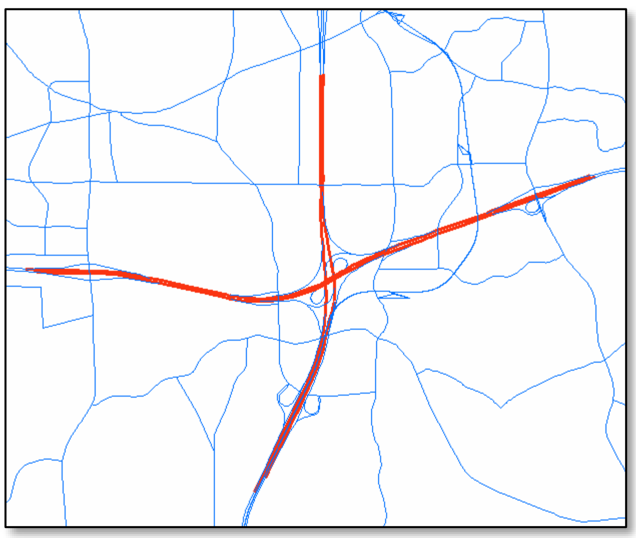


Figure 46: Weave Flag



8.4 Notes on Link Attributes

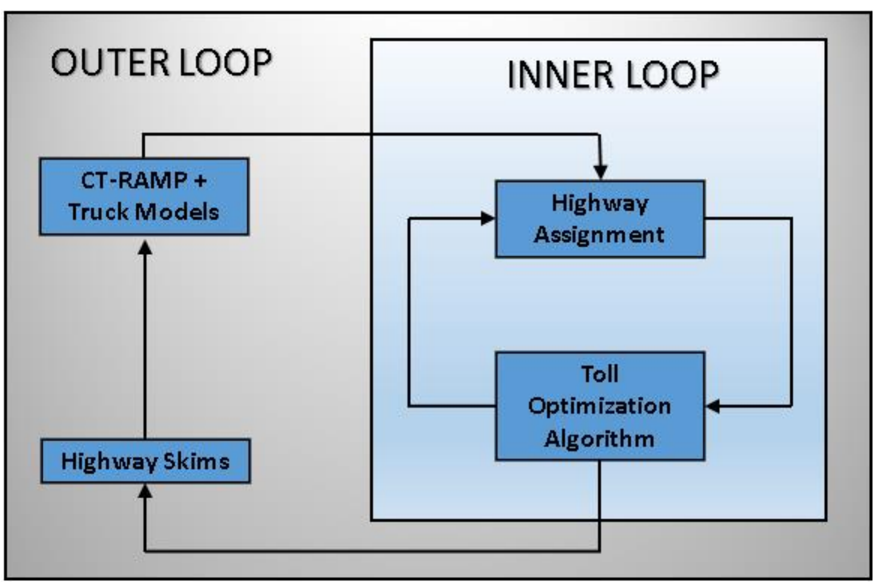
Users must pay special attention to certain link attributes, especially when using existing link templates to create new links (i.e. copying and pasting link attributes). These attributes are potentially unnoticed when copying templates because they are perhaps not considered as important as other attributes when coding; however, they can have direct impacts to forecast results. For example, if a user were to copy data from a link with an existing TOLLID and not update that attribute appropriately, incorrect tolls will be applied in the model which will have a substantial impact in the results. These attributes are as follows:

- TOLLID / GPID: These attributes are related to how tolls are applied at the link level, and in the case of GPID, the application of the toll optimization routine.
- HOVMERGE: When flagged, this attribute results in additional travel time on a link to reflect that vehicles must typically cross lanes of traffic to access HOV lanes.
- RAMPFLAG: When flagged, this attribute indicates the ramp is a loop-type ramp (i.e. small radius) and reduces the free-flow speed to reflect the geometry.
- WEAVEFLAG: When flagged, this attribute indicates that the link is an approach or departure from a major system-to-system interchange which is treated with a different volume-delay function than a normal freeway segment.
- EASPD: This attribute represents the observed early AM period speeds from the NPMRDS. Where coded, this speed is used along with a lookup table of speeds to compute the free-flow speed used in the model (average of observed and lookup value). As such, these values should not be used on new roadways, as there is no observed speed available.

8.5 Toll Optimization

As the ABM includes toll and non-toll in both tour and trip mode choice, the toll prices have an effect on the toll eligible demand. When introducing a new toll lane that is managed with pricing, the user will likely need to perform an iterative process in which the ABM is run to generate toll demands which are then run through a series of highway assignments where the tolls are optimized. The new toll prices are then fed into another ABM run to estimate new demands and the process continues until the demand and toll prices stabilize. The full process is shown in **Figure 47**. As the process can take time, the inner loop of the methodology has been automated for ARC.

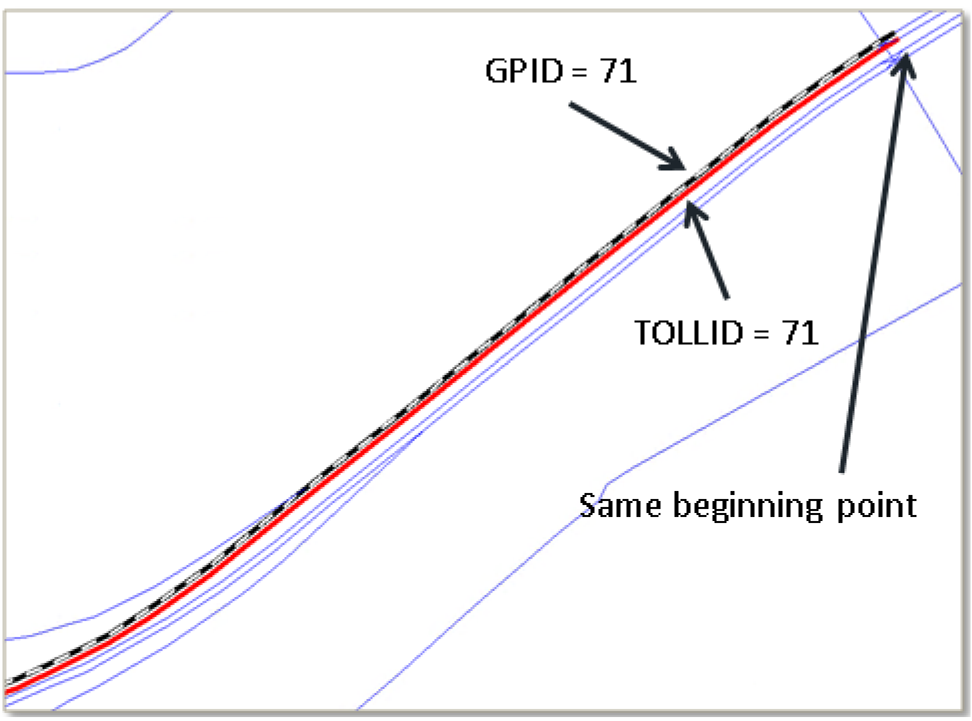
Figure 47: Toll Optimization Process



The toll optimization process is working towards an equilibrium between the generalized costs between the managed lanes and the general purpose lanes. This optimization might be different from maximizing revenue. For managed lane corridor studies, it is the user's responsibility to determine the most appropriate toll rates for those studies. This process is intended to assist in finding a logical balance between travel time savings and tolls.

The first step in the process is to ensure that the general purpose lanes and managed lanes have a unique attribute for comparing the generalized costs. This is handled using two attributes in the network, GPID and TOLLID. The GPID and TOLLID for a parallel segment must be coded with the same ID number to inform the optimization routine which segments are to be compared against one another. The segments should be treated uniquely for all managed lane entry/exit points. A managed lane facility could have numerous segments across the length of the corridor. As the optimization routine is comparing generalized costs, it is important to ensure the segment coding between the general purpose lanes and managed lanes have similar break points and therefore, similar segment distances. An example of the coding of the two attributes is provided in **Figure 48**.

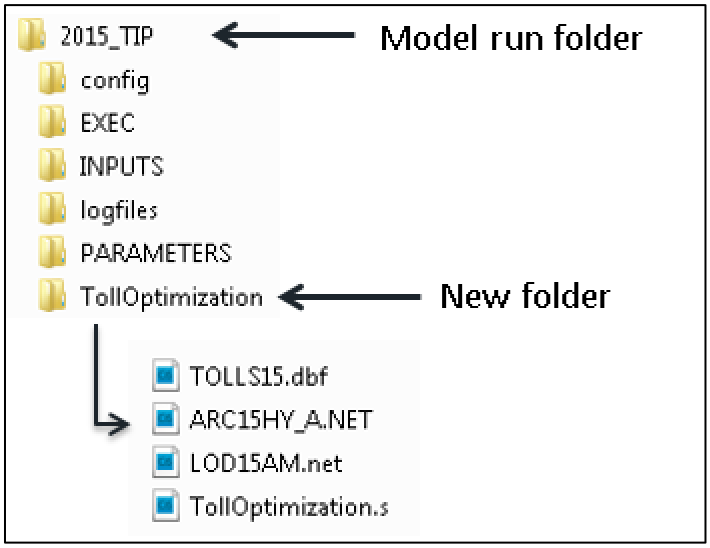
Figure 48: Network Coding for Toll Optimization



As toll demand is generated by the CT-RAMP routine, the toll optimization process uses input from a full model run of the scenario year. The toll optimization routine has an expected directory structure which is provided in **Figure 49**. The user should create a new folder entitled TollOptimization in the root of the model directory and copy the following files from the model run:

- TOLLS{year}.DBF
- ARC{year}HY_A.NET
- LOD{year}@PER@.NET (EA, AM, MD, PM, EV)
- TollOptimization.S

Figure 49: Toll Optimization Directory Structure



As there are five time periods with varying levels of congestion, the tolls must be optimized for each time period separately. The current automated routine requires the user to select which period is being optimized. This was done because certain periods will likely require more iterations than others. The Voyager script includes the five periods and the user simply selects which period to optimize by commenting out the other periods. Note that in application, the user maybe required to manually adjust tolls particularly for uncongested conditions where the travel time savings is negligible. In ARC's testing, this was done to ensure some level of traffic (even minimal) in the managed lanes during uncongested conditions to reflect that some vehicles do use the lanes. The code showing how the user selects a period is provided in **Figure 50**. The user can also select the number of optimization loops for the selected period by modifying the token {iters} in the code as shown in **Figure 51**. As some periods are more congested than others, those periods may require more loops through the toll adjustment to reach equilibrium. Note that if the user runs a few loops but realizes more loops are needed, the process would be started from the previous final loop results (i.e. not starting over from the beginning). This allows the user to run a few loops and then analyze the results in a more efficient manner.

Figure 50: Period for Optimization

```
;=====================================================
;SELECT PERIOD FOR OPTIMIZATION (COMMENT OUT OTHERS)
;PERNO=1   ;EA
;PERNO=2   ;AM
PERNO=3   ;MD
;PERNO=4   ;PM
;PERNO=5   ;EV
;=====================================================
```

Figure 51: Optimization Loops

```
;{label2,Note,10,"General Settings"}
;{year,editbox,"Year (Two-digit)",N,"15"}
;{iters,editbox,"Optimization Loops",N,"3"}
```

The user is also required to set up the loops for the number of segments being optimized. Here, the user can select which segments are desired for optimization. Any segments not entered will not be optimized during the routine. The user must change the number of loops defined by IDS and then set the appropriate TLID for the IDS loop as shown in **Figure 52**. In this example, the user added two additional segments for optimization which required the IDS loop changing from 10 to 12 and also adding the TLID for 101 and 102 for loops 11 and 12.

Figure 52: Toll IDs for Optimization

Current set up:

```
;===========================================================================
;Loop through tollids to be optimized - user must set loop number and tollids
LOOP IDS=1,10
 IF(IDS=1)  TLID=51
 IF(IDS=2)  TLID=52
 IF(IDS=3)  TLID=61
 IF(IDS=4)  TLID=62
 IF(IDS=5)  TLID=71
 IF(IDS=6)  TLID=72
 IF(IDS=7)  TLID=81
 IF(IDS=8)  TLID=82
 IF(IDS=9)  TLID=91
 IF(IDS=10) TLID=92
 ;===========================================================================
```

Revised set up:

```
;===========================================================================
;Loop through tollids to be optimized - user must set loop number and tollids
LOOP IDS=1,12
 IF(IDS=1)  TLID=51
 IF(IDS=2)  TLID=52
 IF(IDS=3)  TLID=61
 IF(IDS=4)  TLID=62
 IF(IDS=5)  TLID=71
 IF(IDS=6)  TLID=72
 IF(IDS=7)  TLID=81
 IF(IDS=8)  TLID=82
 IF(IDS=9)  TLID=91
 IF(IDS=10) TLID=92
 IF(IDS=11) TLID=101
 IF(IDS=12) TLID=102
 ;===========================================================================
```

User must change loop number

User must change if statements to include new loops and appropriate toll ids

The optimization routine includes several outputs:

- TOLL_DATA@PER@@ITERS@.DBF
- NEWTOLLS@PER@@ITERS@.DBF
- TOLLS{year}.DBF

TOLL_DATA@PER@@ITERS@.DBF – This file provides the information the optimization routine is using for each segment and is generated for each period and iteration loop that is run (note that if the user runs a few loops for a period and then determines the need to run more loops, these files will be overwritten the second time through the iteration process). A sample output is provided in **Figure 53** where the columns are as follows:

- TOLLID – segment ID
- EXTPMILE – existing per mile toll (cents) for the current iteration
- GPTIME - general purpose lane travel time (minutes)
- GPDIST – general purpose lane travel distance (miles)
- MLTIME – managed lane travel time (minutes)
- MLDIST – managed lane travel distance (miles)
- MLTOLL – managed lane segment toll total (dollars)
- GPGC – general purpose lane generalized cost
- MLGC – managed lane generalized cost
- GCDIF – generalized cost difference (MLGC – GPGC)
- NEWTOLL – revised segment toll total (dollars)
- NEWPMILE – revised per mile toll (cents)
- GPSOV – general purpose lane SOV volume
- GPHOV2 – general purpose lane HOV2 volume
- GPHOV 3– general purpose lane HOV3 volume
- MLSOV – managed lane SOV volume
- MLHOV2 – managed lane HOV2 volume
- MLHOV 3– managed lane HOV3 volume
- MLSOVSHR – managed lane share of SOV trips
- MLHOV2SHR – managed lane share of HOV2 trips
- MLHOV3SHR – managed lane share of HOV3 trips

NEWTOLLS@PER@@ITERS@.DBF – This file provides the TOLLID and revised TOLL for the segments selected for optimization. A sample output is provided in **Figure 54**.

TOLLS{year}.DBF – This file is the final output that would be used to begin a new full model run. Note that this file is updated for each period and iteration that the user runs. A sample output is provided in **Figure 55**. The file includes the TOLLID, the per-mile toll rates (in cents) for each time period, and whether the toll is fixed or applied on a per mile basis.

The toll optimization output also includes the loaded networks for each period and iteration that the user runs. These are the same as the standard model run loaded networks and allow the user to manually inspect the loadings as desired.

Figure 53: Sample Optimization Output File

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
TOLLID	EXTPMILE	GPTIME	GPDIST	MLTIME	MLDIST	MLTOLL	GPGC	MLGC	GCDIF	NEWTOLL	NEWPMILE	GPSOV	GPHOV2	GPHOV3	MLSOV	MLHOV2	MLHOV3	MLSOVSHR	MLHOV2SHR	MLHOV3SHR
71	6.30	3.20	3.02	2.72	3.02	0.19	1.75	1.74	0.01	0.20	6.50	16300	1495	171	1281	132	346	0.07	0.08	0.67
72	6.53	3.22	3.02	2.72	3.02	0.20	1.76	1.75	0.01	0.20	6.77	14514	1493	36	1848	145	284	0.11	0.09	0.89
81	7.52	2.78	2.52	2.28	2.53	0.19	1.51	1.49	0.02	0.20	7.84	14679	1270	60	1281	112	384	0.08	0.08	0.86
82	8.52	2.81	2.53	2.28	2.52	0.22	1.52	1.52	0.01	0.22	8.62	12503	1288	30	2094	142	269	0.14	0.10	0.90

Figure 54: New Tolls Sample File (Optimized Only)

A	B
TOLLID	TOLL
51.00	2.00
52.00	2.00
61.00	2.00
62.00	2.04
71.00	6.50
72.00	6.77
81.00	7.84
82.00	8.62
91.00	10.31
92.00	2.00

Figure 55: New Toll Input File

A	B	C	D	E	F	G
TOLLID	TOLLEA	TOLLAM	TOLLMD	TOLLPM	TOLLEV	FIXED
11.00	50.00	50.00	0.00	0.00	0.00	0.00
12.00	0.00	0.00	10.00	80.00	2.00	0.00
31.00	50.00	50.00	0.00	0.00	0.00	0.00
32.00	0.00	0.00	10.00	80.00	2.00	0.00
41.00	15.00	15.00	5.00	30.00	5.00	0.00
42.00	5.00	5.00	5.00	44.00	5.00	0.00

As noted previously, the ABM includes toll / non-toll as an alternative mode choice. Therefore, after optimizing the tolls, the user must re-run the ABM to generate new toll / non-toll demand tables. It is possible that the new demand tables will require further optimization of the per-mile toll rates in the assignment where the user runs through the optimization routine again. In that case, the user must copy all files from the model re-run that were listed as input into the TollOptimization folder (i.e. overwrite the ones that were previously used). This process should continue until the user determines the toll demand and toll rates have reached a level of equilibrium.

9 Transit Network Coding

This section provides guidance for coding transit routes in the new model. As discussed previously, the transit component has been transitioned from TRNBULD to PT. While much of the transit coding remains similar to previous versions, there are also some major revisions discussed below.

9.1 Non-Transit and Transit Modes

With the addition of KNR, the non-transit modes have been modified. The model includes five non-transit modes which are shown in **Table 75**. The mode 1 walk connectors represent both access and egress. The KNR and PNR connectors are also generated in both directions; however, either the access or egress mode has to be walk (i.e. PNR access – transit – walk egress). The directionality is also period specific as previously discussed. The station / feeder mode is generated using the FACTYPE 53 links coded in the network to ensure connectivity between feeder buses and rail stations. The mode 5 transfer links are other transfer opportunities between transit stops that are not explicitly coded in the network (e.g. bus stops in close proximity to one another). These transfer links are generated up to a tenth of mile between transit stops. All connectors are created utilizing PT's GENERATE command.

Real transit modes are similar to previous model versions and are provided in **Table 76**. There are five modes available for each operator: 1) Local bus 2) Heavy rail 3) Express bus 4) Light rail and 5) Premium BRT. In addition, there is one commuter rail mode. The model also includes place holders for new projects. This is primarily intended for transit studies to allow detailed extraction of trip information for project trips.

Table 75: Non-Transit Modes

Mode	Description
1	Walk
2	Kiss-and-ride
3	Park-and-ride
4	Station / feeder link
5	Transfer link

Table 76: Transit Modes

Mode	Description
14	MARTA Local Bus
15	MARTA Heavy Rail
16	MARTA Express Bus
17	MARTA Light Rail
18	MARTA Bus Rapid Transit
24	CCT Local Bus
25	CCT Heavy Rail
26	CCT Express Bus
27	CCT Light Rail
28	CCT Bus Rapid Transit
34	CTRAN Local Bus
35	CTRAN Heavy Rail
36	CTRAN Express Bus
37	CTRAN Light Rail
38	CTRAN Bus Rapid Transit
44	GCT Local Bus
45	GCT Heavy Rail
46	GCT Express Bus
47	GCT Light Rail
48	GCT Bus Rapid Transit
54	GRTA Local Bus
55	GRTA Heavy Rail
56	GRTA Express Bus
57	GRTA Light Rail
58	GRTA Bus Rapid Transit
59	Commuter Rail
64	CATS Local Bus
65	CATS Heavy Rail
66	CATS Express Bus
67	CATS Light Rail
68	CATS Bus Rapid Transit
74	HAT Local Bus
75	HAT Heavy Rail
76	HAT Express Bus
77	HAT Light Rail
78	HAT Bus Rapid Transit
84	Project Local Bus
85	Project Heavy Rail
86	Project Express Bus
87	Project Light Rail
88	Project Bus Rapid Transit
100	Shuttle buses

9.2 Transit Route File Structure

While there are similarities in the route structure within the geodatabase framework, there are also many differences which will be discussed in more detail in this section. However, the line files can be exported into their native text format and in that regard, remain much the same as in previous model versions. Once in the text file, the structure of the transit route files in PT is basically the same as TRNBUILD. The file is still in text format consisting of individual lines with nodes corresponding to the highway network. An example route is shown in **Figure 56**. The primary differences are the attributes that can be coded. For instance, HEADWAY replaces FREQ.

PT also allows for the coding of short names and long names of transit routes via NAME and LONGNAME. For ARC, NAME has been designated for operator and route number while LONGNAME is the actual route name. The NAME attribute can be up to 28 characters while LONGNAME can be up to 50 characters.

The attributes that are utilized in coding are as follows:

- NAME – operator name and route number
- MODE – mode number of route
- OPERATOR – route operator (corresponds to operator in system data file)
- LONGNAME – route name (e.g. Coronet)
- ONEWAY – true / false indicator if route operates in only one direction
- HEADWAY[1] – early AM headway in minutes
- HEADWAY[2] – AM peak headway in minutes
- HEADWAY[3] – mid-day period headway in minutes
- HEADWAY[4] – PM peak headway in minutes
- HEADWAY[5] – evening / late night headway in minutes
- CIRCULAR – true / false indicator if route operates in circular fashion. Not necessary for routes that do not operate in this manner.
- DWELL_C – dwell time in minutes to be assessed at each transit stop

Figure 56: Route Structure Text Format

```
;;<<PT>><<LINE>>;;
LINE NAME="MARTA 1", MODE=14, OPERATOR=1, LONGNAME="CORONET",
     ONEWAY=T, XYSPEED=20, HEADWAY[1]=18, HEADWAY[2]=30, CIRCULAR=T,
     N=19843, DWELL_C=0.5, N=-64138, -37042, -64147, -19929, 79278,
     -19930, -19931, -64109, 20082, -64111, 20241, 79276, 20242,
     19974, -20048, 20049, 20360, 20050, 20051, 20052, -20132,
     -79257, 20223, -64030, -20131, 64029, -20199, -20130, 64024,
```

As was the case in the previous model, the transit routes are split into two files to distinguish between premium and non-premium routes. Premium routes consist of heavy rail, express bus, light rail, BRT, and commuter rail. Non-premium routes include local buses and express buses. When choosing the appropriate mode, the user should consider how the route operates. The followings are guidelines for mode definitions:

Non-Premium:

- Local bus – routes operating on surface streets with numerous stops. Operate in mixed use traffic.
- Shuttle bus – short distance routes distributing passengers back and forth at activity centers or university campuses.

Premium:

- Express bus – routes operating on high speed facilities (e.g. interstates) with few intermediate stops. Routes typically serve park-and-ride lots in the suburbs and distribute passengers around an activity center.
- BRT – routes operating in high speed access controlled facilities (managed lanes) or within exclusive bus-ways. Intermediate rail-type stations provided including passenger amenities such as covered stations, platform boarding, etc.
- Heavy / Light rail – routes operating on tracks in exclusive right-of-way. In some cases, light rail may also operate in the street as well.
- Commuter rail – routes operating on main line railroad track carrying passengers to and from work around major activity centers.

9.3 Transit Route Coding

Route coding in PT via the geodatabase has similarities to TRNBUILD but also some notable differences. In the Data Manager window shown in **Figure 57**, the user can expand the transit files and double click any of them to open the layers. Note that since the transit networks overlay the highway network, this will also add the highway network into the TOC and GIS window as illustrated by **Figure 58**.

Figure 57: Opening Route File

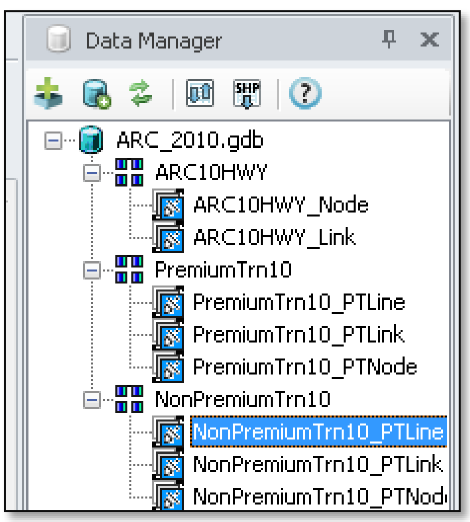
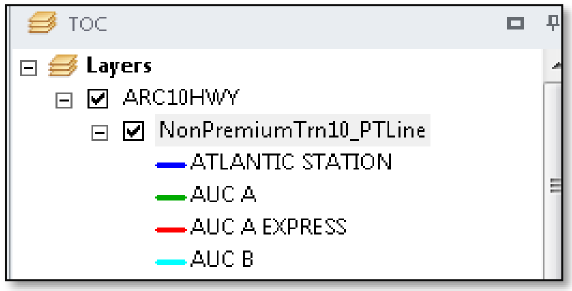


Figure 58: Route File TOC



Once a transit layer is open, displaying individual or multiple lines is similar to previous versions of Cube. To select a line or lines, click the Display Lines icon (**Figure 59**) in the toolbar. A new window will open containing all the routes currently available in the network as shown in **Figure 60**. By scrolling through the list, the user can select an individual route or multiple routes by left clicking on the desired routes. Once the routes are selected, click OK and the GIS window will open with the route(s) selected.

Figure 59: Displaying Transit Lines

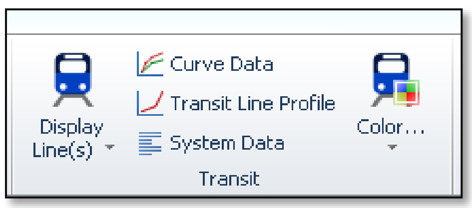
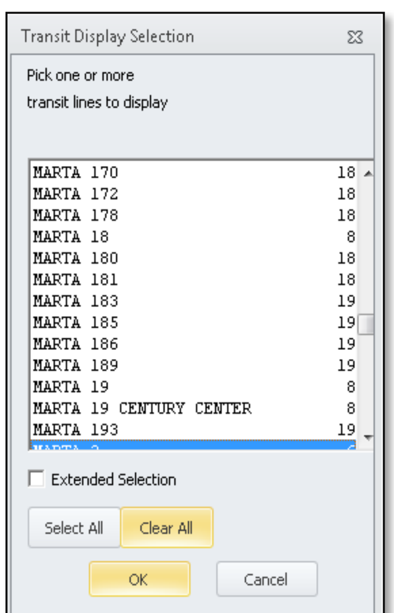


Figure 60: Selecting Transit Line for Display



To edit a transit route, editing must be enabled in the same manner as for highway coding (make transit layer active and then click Editor – Start Editing). To edit route attributes, the user must select the Edit Feature pointer (same as in highway edit mode). Using the pointer, click on a route to open the Feature Explorer which will also highlight the route as shown in **Figure 61**.

In the Feature Explorer window, the route attributes are stored in the Line tab (**Figure 62**). This tab includes all route attributes that are available; however, most are not used. To change a route attribute, click in the appropriate field and manually type the change.

Figure 61: Selected Transit Line for Editing

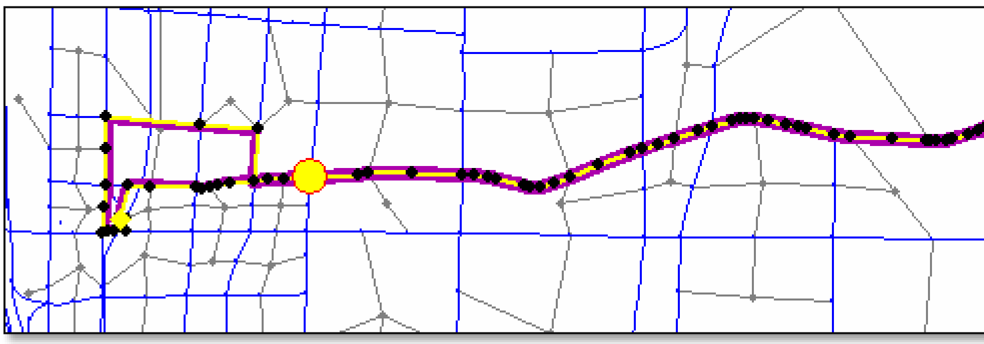
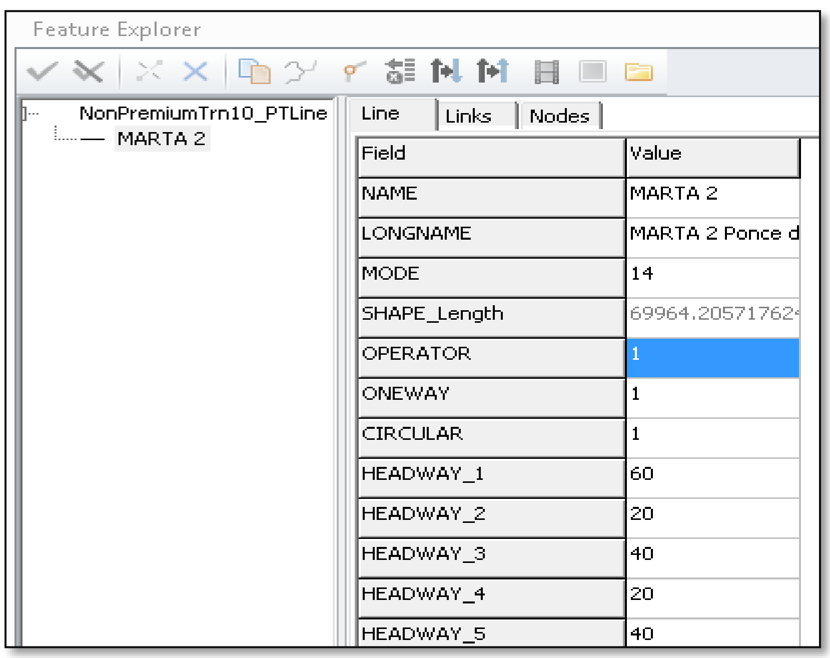
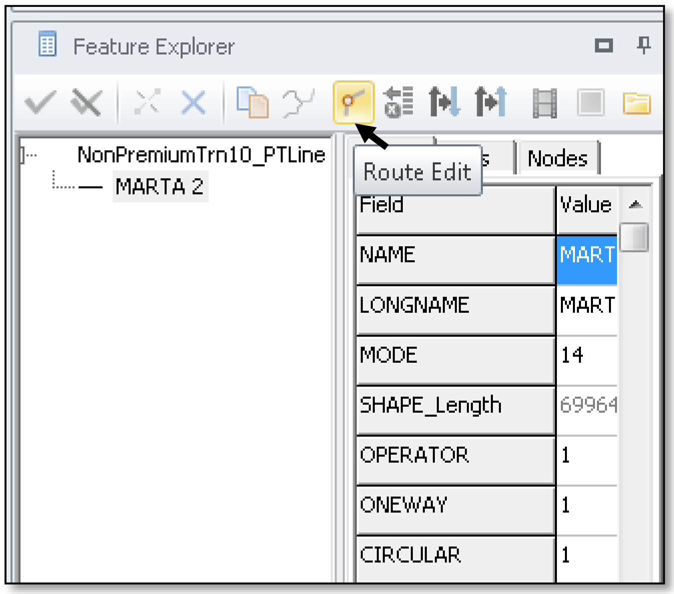


Figure 62: Route Attributes



The Feature Explorer contains a Route Edit function that is used for modifying the alignment of an existing route and is depicted in **Figure 63**. Using the route editor in the GIS window is similar to the standard Cube editor. When in edit mode, the pointer will appear and allow the user to select a beginning node where the alignment is to change. Then, then the user left clicks on desired stop locations and the program will route the transit line through the network to the stop.

Figure 63: Route Edit



For example, if the user would like to change the alignment of the route shown in **Figure 64** to follow the black links with stops at the nodes highlighted, with the route selected, click Route Edit. The user should click on the node where the alignment is changing which in the case of **Figure 64** is node 19718 (note that the directionality of how the route is coded is important). To code stops at the highlighted nodes, the user should use the pointer to left click on those nodes. Cube will automatically add the non-stops nodes that occur between the highlighted nodes. When modifying the route alignment, Cube displays both the old and new alignment as shown in **Figure 65**. To tie the alignment together, the user needs to click on the first node where the route is back on the original alignment (node 19675 in **Figure 64**). When the modifications are complete, the user should press the ESC key which ends the route editing. However, the route modifications are not changed until the user clicks the green checkmark to save the edits. Conversely, if the user is unsatisfied with the edits, the user can click the crossed-out checkmark to cancel the edits.

Figure 64: Route Alignment

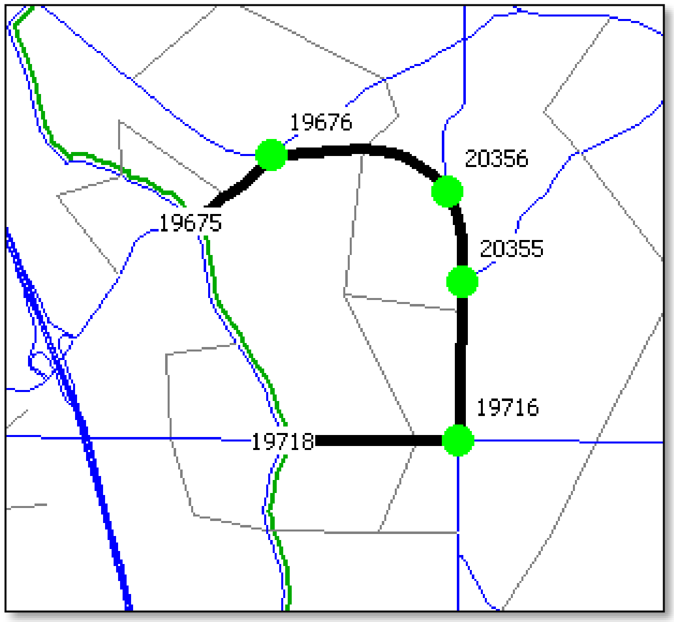
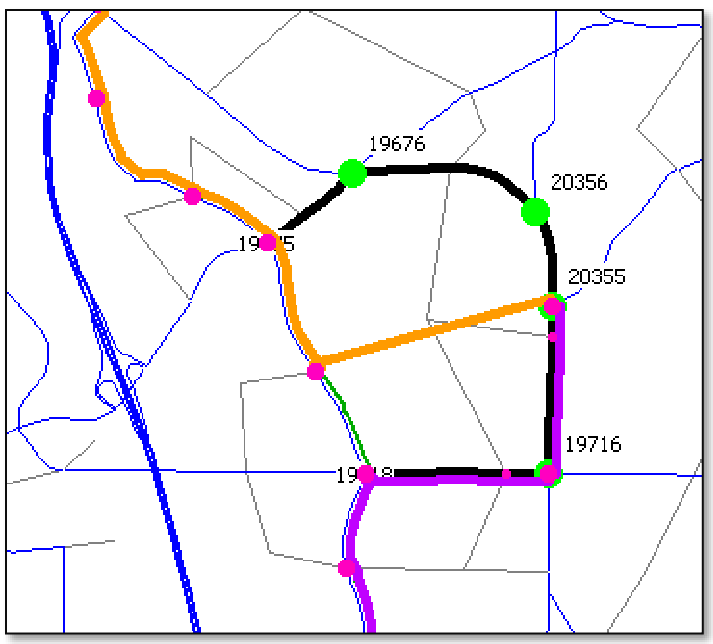


Figure 65: Route Modification



To create a new route, click on the Create Feature which looks like a pencil in the editor tool bar. To start a new route, left click on the desired starting point node. To code the route alignment, left click on each preferred stop location until the full alignment is coded. Prior to saving the edits, the user must give the route a name and mode. It is helpful to have the transit system data file available to ensure proper attribute coding. For example, to add a new MARTA bus, the mode would be 14 and the operator set to 1.

Alternatively, if the user prefers using the previous coding environment, the transit files can be exported from the geodatabase to a *.lin file. This is accomplished by right-clicking on a transit line layer in the Data Manager window and then selecting Export as shown in **Figure 66** which produces the Transit Line Layer Export GUI as shown in **Figure 67**. This allows the user to open the text files and change frequencies, dwell times, etc. which can be more efficient in a text editor than clicking individual routes in the GIS window. Once the changes are complete, the transit file needs to be imported back into the geodatabase as shown in **Figure 68** which produces the Transit Line Layer Import GUI as shown in **Figure 69**. Prior to importing, the transit layer in the geodatabase being updated needs to be deleted. This is done by right-clicking the layer and selecting Delete from the Data Manager window. To import the revised transit line file, left-click the Import/Export Data button in the Data Manager as shown in **Figure 69**.

Figure 66: Exporting from Data Manager

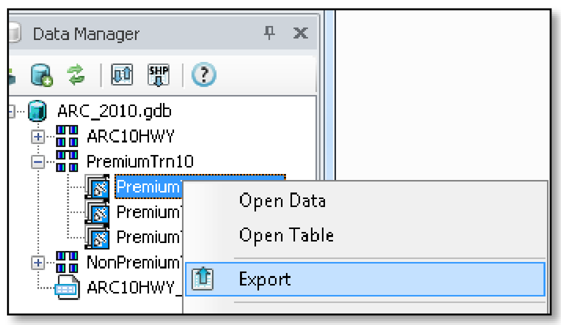


Figure 67: Transit Line Layer Export

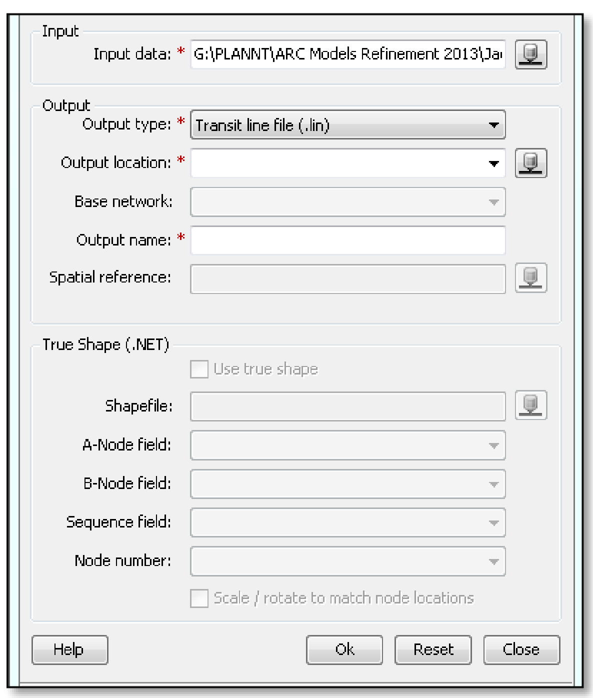


Figure 68: Importing from Data Manager

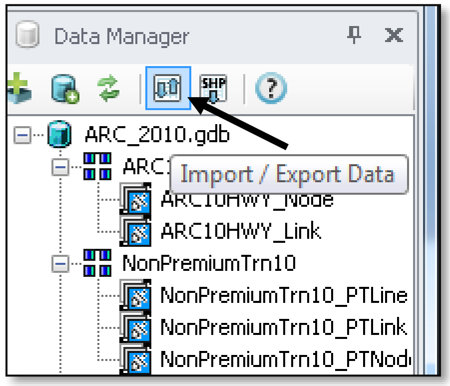
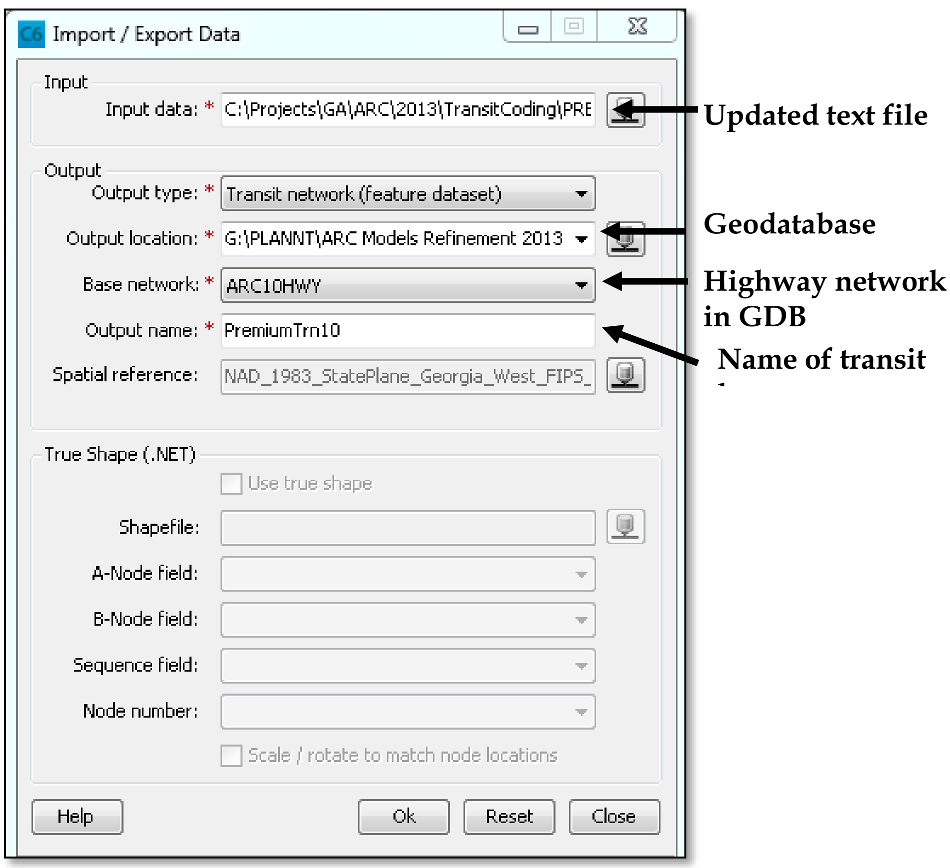


Figure 69: Transit Line Layer Import



9.3.1 Dwell Time Coding

Previously, bus speeds were calculated as function of the highway speeds by applying factors based on facility types and area types. This process was limited because two routes operating in the same alignment would have the same travel speeds even if one route provided limited stops while the other included numerous stops. In the new model, the bus speeds are still based on the highway speeds. However, with the conversion from TRNBUILD to PT, dwell times can be added at each coded stop to more accurately reflect the speeds. The dwell times are coded for each route by use of the variable DWELL or DWELL_C. If viewing the transit network in text format, the variable should be placed after the first node and is coded in minutes. The coded dwell time will be applied to the previous node and all subsequent stop nodes. If operating in the GIS window, to update the dwell time, the user must first be in edit mode. Then, once a route is selected for editing, click the Nodes tab as shown in **Figure 70**. This contains node specific attributes for routes, including the dwell time. Right-click the column labeled DWELL and select Multi Cell Edit. This will open another window as shown in **Figure 71**. In this window, the user can select the appropriate nodes for updating the dwell time or apply a dwell time to the entire route. This is done by selecting the first node in the From Row and the last node in the To Row and then setting the cell value to the desired number.

In validation of the bus speeds, the dwell times for local buses were set to 0.5 while express buses were set to 1.0. This is reasonable given that express buses have fewer stops and generally more people boarding/alighting per stop. The coded MARTA rail routes do not include dwell times in the route file. This is because the coded MARTA rail link speeds include the dwell times. When coding fixed guide-way, it is important the user be aware of how the dwell times are represented.

Figure 70: Dwell Time Coding 1

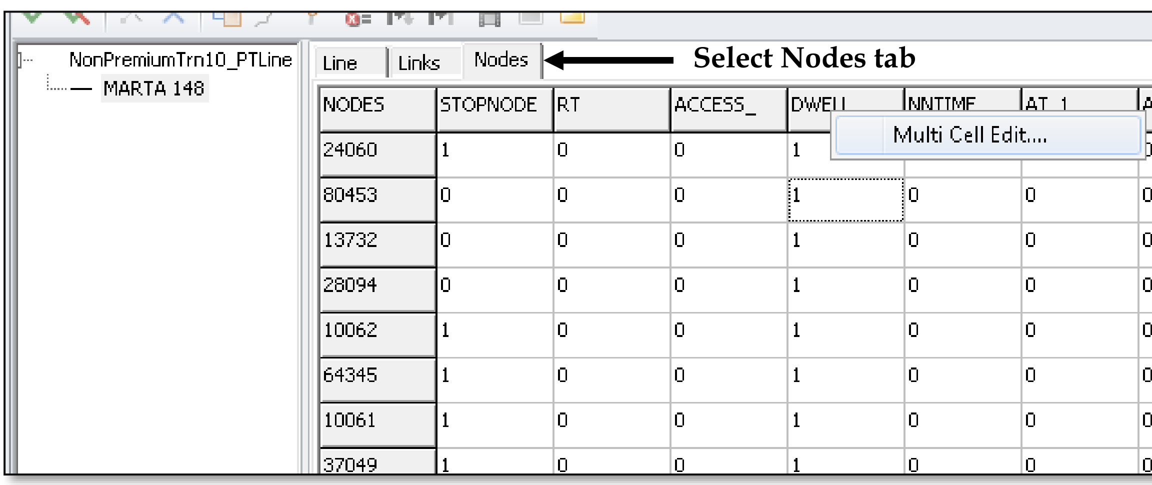
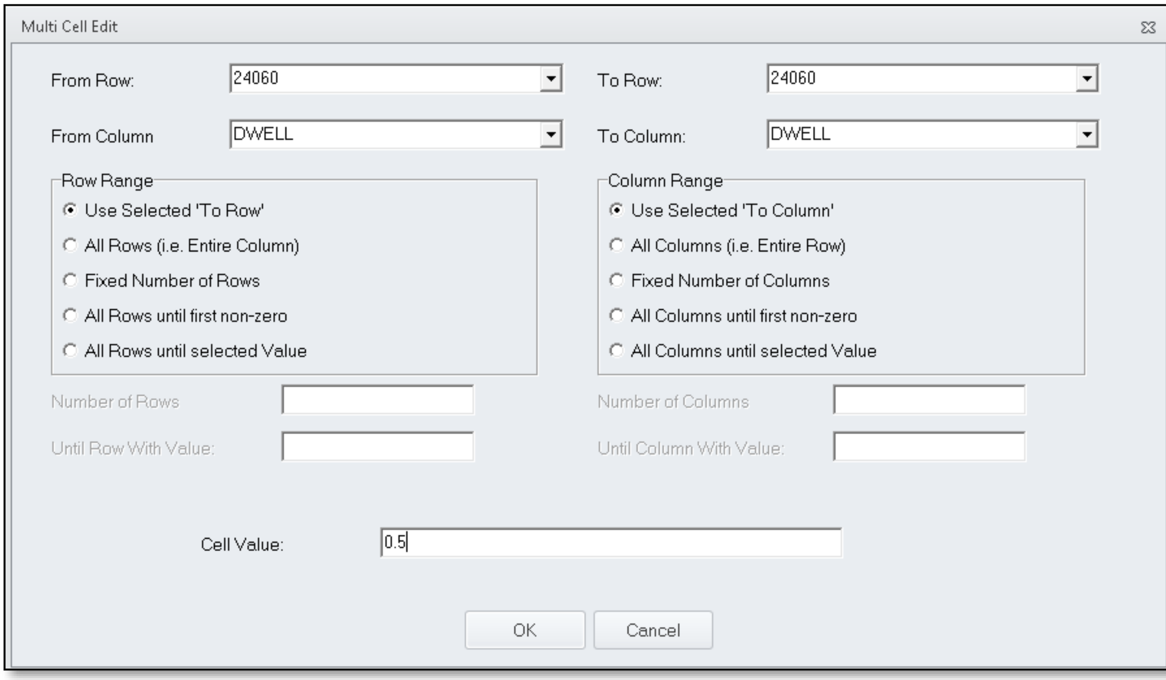


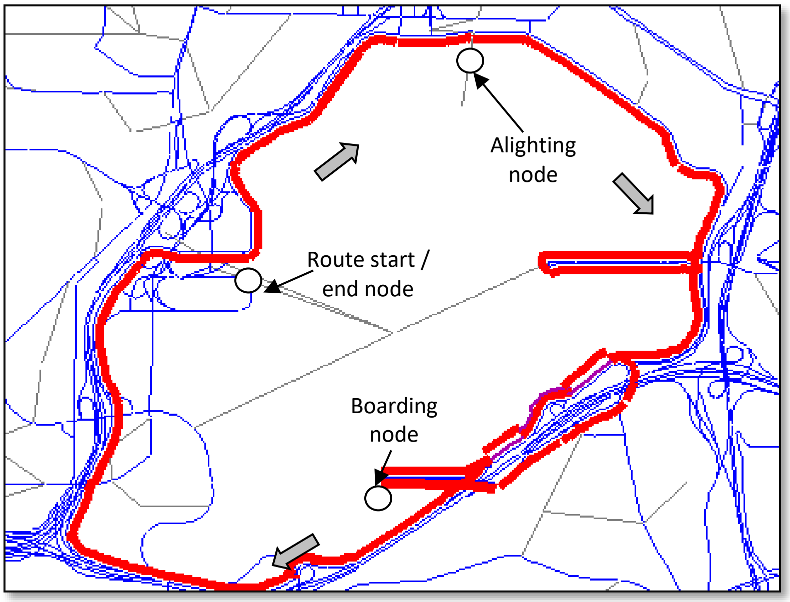
Figure 71: Dwell Time Coding 2



9.3.2 Circular Route Coding

The conversion from TRNBUILD to PT also allows for more accurate coding of route alignments by use of circular coding logic. The CIRCULAR variable is a route specific attribute that informs the program whether or not to invoke circular logic in path building. If CIRCULAR is coded as ‘1’, the program treats the route in a circular manner. The default is linear so for non-circular routes, the user can exclude the CIRCULAR variable from the route. The circular logic allows for a route to have the same beginning and ending node without forcing a transfer at that node. In **Figure 72**, the route starts at the airport and then runs clockwise. In TRNBUILD, if the boarding and alighting nodes were as labeled, the program would force a transfer at the route start/end node. However, in reality passengers would stay on the bus which is reflected in PT using the CIRCULAR variable.

Figure 72: Circular Coding Example 1



The circular logic is also helpful when routes operate on one-way facilities and/or have deviating alignments by direction. For example, there are numerous buses operating in the downtown area where this is the case. One example is provided in **Figure 73.** The route operates on one-way streets in downtown and has a different alignment on the eastern portion of the route. Previously, this route would have been coded as two separate routes to reflect this pattern. With the circular logic, the entire alignment can be coded as one route. Where the route doubles back on itself the coding can be a bit difficult because Cube doesn't allow the backtracking without the user clicking on each individual node and holding down either the Shift key or the Alt key. The Shift key will make the node a stop node while the Alt key will result in a non-stop node.

Alternatively, the user could initially code the alignment as two separate one-way routes and then open the route text file in order to merge the routes together. The user would copy/paste the reverse direction nodes at the end of the other route and then delete the reverse route. A simple example of how the copy/paste would work is shown in **Figure 74.** In this example, once the user pasted the backtracking nodes to 'TestRoute', the route called 'TestRoute2' would be deleted from the text file.

Figure 73: Circular Coding Example 2

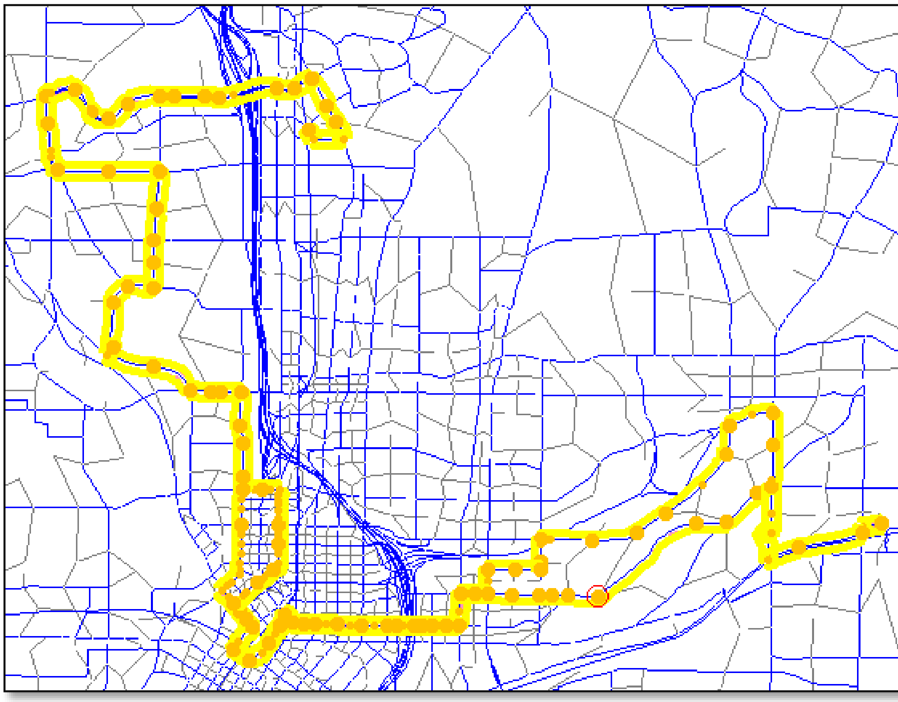
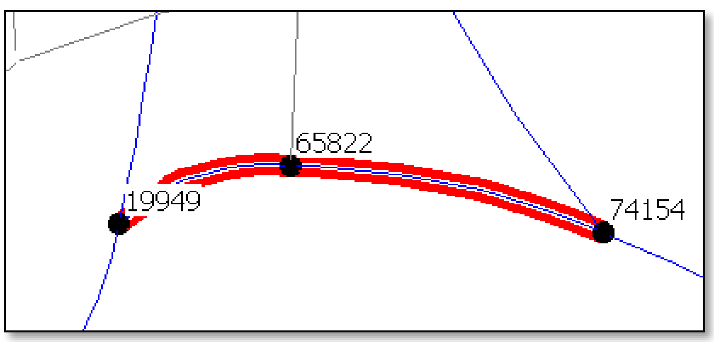


Figure 74: Circular Coding Example 3




```
LINE NAME="TestRoute", MODE=14, HEADWAY[1]=15, HEADWAY[2]=30,
     OPERATOR=1, ONEWAY=T, CIRCULAR=T, N=19949, 65822, 74154

LINE NAME="TestRoute2", MODE=14, HEADWAY[1]=15, HEADWAY[2]=30,
     OPERATOR=1, ONEWAY=T, CIRCULAR=T, N=74154, 65822, 19949
```

Copy nodes for backtracking

```
LINE NAME="TestRoute", MODE=14, HEADWAY[1]=15, HEADWAY[2]=30,
     OPERATOR=1, ONEWAY=T, CIRCULAR=T, N=19949, 65822, 74154, 65822, 19949
```

9.4 Transit Related Highway Network Coding

A number of modifications have been made in the representation of transit in the highway network in relation to station coding, park-and-ride coding, and fixed guide-way transit.

9.4.1 Station Coding

The link based coding of stations is similar to the previous version in that transit only links are recommended to represent stations, feeder buses, park-and-ride lots, and transfer links. However, there have been additional link types added along with node level coding. The facility types for coding a station with descriptions are:

- FACTYPE 50: Feeder bus access to station.
- FACTYPE 52: PNR access to parking lot
- FACTYPE 53: Transfer between feeder buses and station
- FACTYPE 55 (new): Connection between PNR lot and transit station/feeder bus nodes
- FACTYPE 98 (new): Pedestrian access to station
- FACTYPE 99 (new): Fixed guide-way links for transit (e.g. MARTA rail)

The station coding can become pretty complex as shown in the example in **Figure 75**. This example is of the MARTA Oakland City Station. The station includes a feeder bus node, PNR lot, and special pedestrian access. The FACTYPE 55 links were added due to the conversion from TRNBUILD to PT. For generating PNR access links, PT requires a direct connection to the transit stop. For example, if the FACTYPE 55 link were not coded between the PNR node and feeder bus node, drive access paths would not be available to the feeder buses.

The FACTYPE 98 links were added to represent direct pedestrian access that is not currently reflected in the model network by the road system. There are numerous examples of this throughout the region including at the Oakland City station. An aerial photograph of this station is provided in **Figure 76**. As shown in the aerial, the station is bordered on the east side by railroad tracks. There is a portal located east of the railroad tracks which allows transit patrons access to the station under the railroad tracks. The FACTYPE 98 link is coded to represent this access. Otherwise, the transit paths from the east would require a longer walk using the coded road system.

Figure 75: Station Coding Example

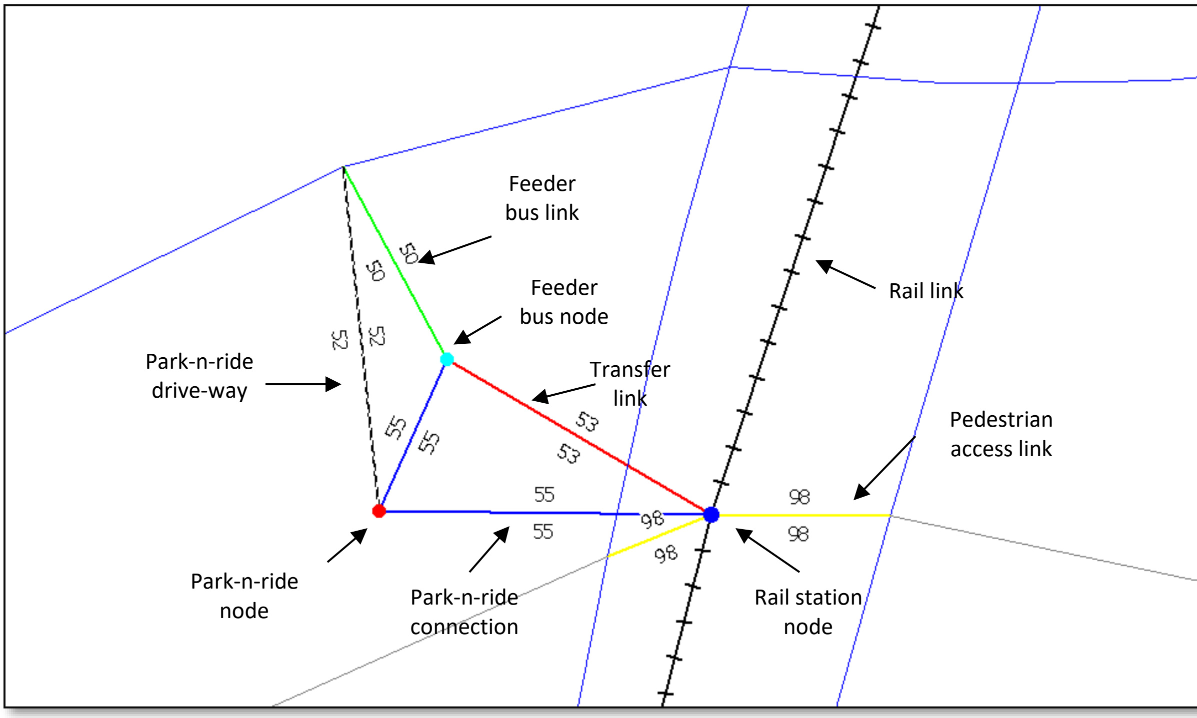
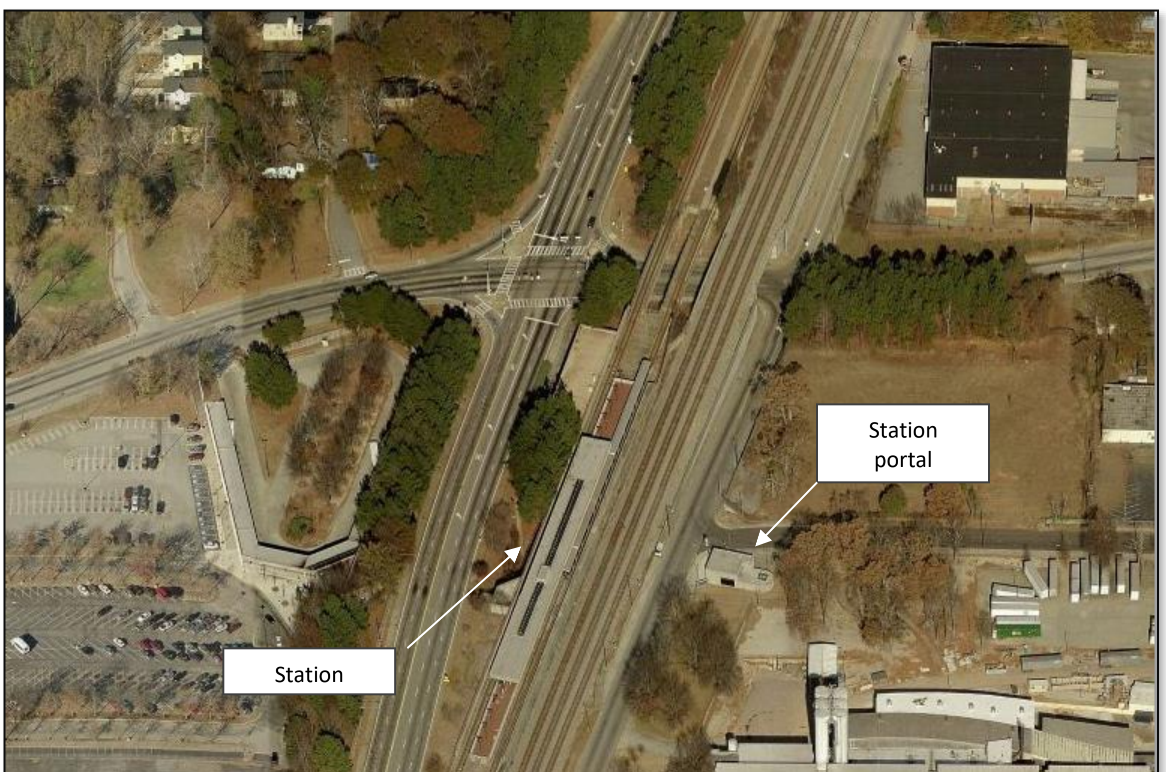


Figure 76: Station Aerial

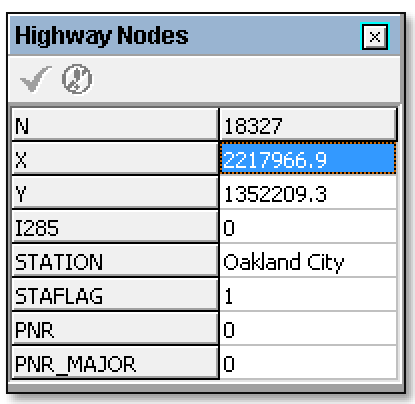


9.4.2 Transit Coding Highway Node Attributes

Several node attributes have been added to the highway network to facilitate transit coding for rail stations and PNR lots. For station coding, the new attributes are STATION and STAFLAG. The STATION attribute is a text field allowing the user to include the station name. The STAFLAG attribute is a numeric field which should be coded as a '1' to indicate a station. This allows the user to manually view station locations. It is used in Cube Voyager scripts to post-process model results in order to summarize station boardings. The model also uses STAFLAG as an indicator to allow for longer KNR access connectors to nodes designated as stations as the transit on-board survey data revealed the tendency of longer trips to the MARTA rail stations than ordinary transit stops. **Figure 77** is an example of how the Oakland City station is coded in the node attributes.

Previously, PNR lot coding was handled using text files where the user was required to code the PNR lot nodes in two separate files to represent major and minor lots. This has been streamlined by adding two PNR flags to the node attributes. To designate a node as a PNR lot, the user sets the PNR attribute to '1'. To designate the lot as a major lot, the user sets the PNR_MAJOR attribute to '1'. The model then reads these attributes to in determining the allowable drive access time to the lot. Currently, the major lots allow for a 40 minute drive time while the minor lots allow for a 10 minute drive time. A major lot should generally include extensive parking spaces (approximately 1,000 spaces or more). However, there are some cases where smaller lots could be designated as major lots when the expected market would exceed the 10 minute threshold. These would include end-of-line locations. In the case of the base year, the suburban PNR lots serving express buses are a good example of where the 10 minute threshold is not sufficient. The new PNR attributes make coding more efficient by allowing the user to manually review lot locations and the major lot designation without having to open multiple files as was previously the case.

Figure 77: Node Attributes



9.4.3 Fixed Guideway Coding

The fixed guide-way coding has been substantially modified. Previously, the MARTA rail links were coded in a separate text file. In the new model version, a new facility type (FACTYPE=99) has been included providing the user the ability to code fixed guide-way links directly in the highway network. Additionally, the user is required to code several link attributes to represent the fixed guide-way speeds, distances, and travel times. The attributes are:

- TRNDIST = station to station distance (miles)
- TRNSPD = station to station average speed (MPH)
- TRNTIME = station to station travel time (minutes)

For the speed and time, it is recommended the average speeds and travel times include the dwell times associated with passenger boarding and alighting. In the base year, the fixed guide-way links only include MARTA rail. However, the new facility type could be utilized for any new fixed guide-way projects. With the attributes part of the input highway network, it allows the user to manually review the fixed guide-way coding without opening multiple files.

10 Shadow Pricing

In the ARC ABM, the Usual Work and School Location models are shadow priced. Shadow pricing is a technique used to doubly-constrain destination choice models. In a doubly-constrained model the total number of workers assigned to a workplace TAZ is proportional to the input employment in the TAZ. Typically there are more jobs than workers in the region, to account for workers that hold multiple jobs. However at this time the ARC ABM assigns only one work location per worker. Shadow prices cannot be estimated; instead, they are calculated iteratively within the Usual Work and School Location (UWSL) Model. Calculating new shadow prices is very time-consuming, therefore the shadow prices are written to a file and can be used in subsequent model runs to cut down computational time.

When setting up a new model run, the user has to choose between using a previously calculated set of shadow prices (each baseline scenario includes shadow prices), or updating the shadow prices for this model run. The shadow prices must be updated whenever the synthetic population or the input employment data are modified, for example when preparing a model run for a new scenario year. The shadow prices may also need to be updated when the modeled transportation improvements are significant enough to modify the usual workplace distribution. Note however that in many instances the baseline shadow prices are sufficient for the build alternatives.

Shadow prices follow the same stratification as the UWSL size terms – that is, the terms in the model that quantify the number of jobs and student enrollment in each destination TAZ. The size terms vary according to worker occupations and student types, to reflect the different types of jobs and schools that are likely to attract different workers and students. As such, shadow prices are calculated for each worker occupation and student type subgroup. The procedure works as follows:

i. At the first iteration, the UWSL models are applied, so that work and school locations are predicted for each eligible person in the model.
ii. The model sums, by occupation, the total number of workers assigned to each workplace TAZ. The following occupation groups are recognized: white collar jobs, health services jobs, retail and food service jobs, other service jobs, and blue collar jobs). Similarly it sums the total number of students of each type (university, pre-driving and driving) assigned to each school TAZ.
iii. The estimated total workers and students by TAZ are compared against the size term calculated for each occupation and student type. The shadow price is calculated as the ratio of the size term and the estimated workers, for each TAZ.

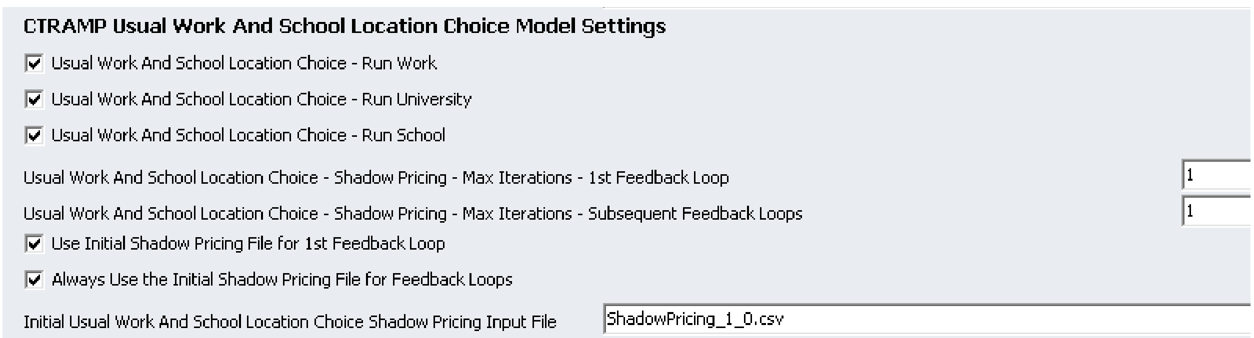
iv. At each subsequent iteration, the UWSL model is applied again, with the addition of the natural log of the shadow price added to the utility function. The steps are repeated until the shadow prices do not change anymore.

Note that since the shadow price acts as a destination-specific constant, it will tend to dampen the effect of level of service on work or school location choice. Some dampening is appropriate since this is a long-term decision. However, very large shadow prices will tend to make the model insensitive to level of service. For this reason it is critical that any user attempting to update the baseline shadow prices review carefully the output from this procedure prior to proceeding with the model run.

10.1 Running Shadow Pricing

There are several options for running shadow pricing, which were previously discussed in this document.

By default, a baseline set of shadow prices is provided with each model run. The default procedure is to use these shadow prices for both the first and subsequent feedback loops, without modifications. This is accomplished by setting the shadow pricing GUI options as follows:



This default option results in the fastest model execution, since the UWSL models are applied only once per feedback iteration.

There will be instances where a user will need to revise the shadow prices. For example, when the scenario under study may use an updated employment projection, it will be necessary to update the shadow prices so that the model fully reflects the revised employment projection.

To recalculate shadow prices starting from scratch, proceed as follows:

i. Set the number of minimum feedback loops to 1 (see **Figure 7**).
ii. Run only the UWSL models (see **Figure 13**).
iii. Set the household sampling rate to 100% (see **Figure 14**).
    a. Population synthesizer sample percent = 1.0
iv. Uncheck the following two options, which ensures that the initial iteration is performed without shadow prices:
    a. Use Initial Shadow Pricing File for 1st Feedback Loop.
    b. Always use the Initial Shadow Pricing File for Feedback Loops.
v. Set the USWL max iterations for the first feedback loop to 10 or a similar value.

With these settings the model will perform 10 iterations of shadow pricing for both the work and school models, based on a full population sample. At the end of each iteration the model will write the calculated shadow prices to an output file, and it will write a summary of the convergence (i.e., a comparison to the size terms) to the standard log file (event.log). If more shadow pricing iterations are desired, this output file can be used as the starting point for additional iterations.

To perform additional shadow pricing iterations, proceed as follows:

i. Set the number of minimum feedback loops to 1 (see **Figure 7**).
ii. Run only the UWSL models (see **Figure 13**).
iii. Set the household sampling rate to 100% (see **Figure 14**).
    a. Population synthesizer sample percent = 1.0
iv. Set the following two options as indicated, which ensures that the initial iteration is performed with the previously calculated shadow prices:
    a. Use Initial Shadow Pricing File for 1st Feedback Loop [check].
    b. Always use the Initial Shadow Pricing File for Feedback Loops [uncheck].
v. Copy the last output shadow pricing file to the input folder, and specify its name in the GUI.
    a. Initial usual work and school location choice shadow pricing input file = [filename]
vi. Set the USWL max iterations for the first feedback loop to 10 or a similar value.

It is essential to use a full population sample when calculating shadow prices. Only advanced users should attempt to update shadow prices previously created with a different scenario. The internal calculations use scaling factors based on the number of workers and size terms which, if incorrectly applied, can erroneously scale the calculated shadow prices, rendering the model insensitive to level of service.

10.2 Issue with School Location Choice

The non-university school location choice model is driven by a size term that is mostly based on educational employment. Educational employment includes the teaching jobs located at the schools, but also many jobs which are not located at the schools. For example all the employees of Atlanta Public Schools (APS), the administrative arm of the school district, are education jobs located at the APS headquarters in downtown Atlanta. A mechanic application of many school shadow pricing iterations, blindly attempting to converge to the current size terms, can lead to many trips attracted to zones that in fact do not have schools, and will result in very long school trips, since the majority of students live outside of the downtown core. To minimize this problem, the model has been calibrated to keep school trips short, rather than forcing shadow price convergence.

Note that it is not possible to update only the work shadow prices. The school shadow prices will be automatically updated whenever the user updates the work location model. To keep the baseline school prices, the user can manually combine the revised work shadow prices with the previous school prices, by manipulating the files in Excel or any other program that reads CSV files.

11 Accessibility Calculator

The accessibility calculator computes for each LUZ pair the logsum-based accessibilities for 2 time-of-day periods (off-peak, peak), 4 income categories (low, medium, high, very high), and 3 auto sufficiency categories (zero autos, number of autos greater than number of workers, number of autos less than or equal to number of workers). Thus, a total of 24 origin-based accessibilities are produced for each LUZ.

11.1 Running the Calculator

The accessibility calculator is included in the ARC CT-RAMP Java archive file, *arc.jar*. It is currently set up to run on a single machine using 8 threads, but can be configured to utilize additional computational resources through the "*accessibility.properties*" file under the project directory.

To run the calculator:

1. Set **Project.Directory** in "*accessibility.properties*" to the current project directory.
2. Run the batch file "*Config/RunAccessibility.bat*" under DOS command.

11.2 Required Files

Table 77 lists the files required to run the accessibility calculator. All the input *.csv* and *.xls* files are specified through *accessibility.properties*.

Table 77: Files Needed to Run the Accessibility Calculator

File Name and Path (in relation to the project directory)	Description
accessibilities.properties	File containing property settings for the accessibility calculator
accessibility.csv	TAZ-level accessibilities computed by *ARCTourBasedModel.S*
zoneData.csv	TAZ-level zonal data produced by *ARCTourBasedModel.S*
ctrampModels/geographicCwalk.csv	TAZ to LUZ lookup table
ctrampModels/accessibility_utility_constants.xls	Constants to be added to the utility calculation for different auto sufficiency levels, modes, and purposes
ctrampModels/accessibility_utility.xls	Mode choice utility equations for calculating logsums
ctrampModels/mandatory_accessibility_alts.csv	Accessibilities defined for mandatory purpose
ctrampModels/nonMandatory_accessibility_alts.csv	Accessibilities defined for non-mandatory purpose
Skims for AM, PM and MD periods	Skims computed by *ARCTourBasedModel.S*

11.3 Output Files

The calculator produces 24 output files, one for each combination of time-of-day periods, income category, and auto sufficiency category. **Table 78** lists the names of the accessibility 24 files and **Table 79** describes the fields in each of these files. The calculator also produces a column-wise merge of all 24 accessibilities in "*all_accessibilities.csv*". All output files are written to the "*accessibility*" sub-directory under the project folder.

Table 78: Accessibility Files Produced by the Accessibility Calculator

Time Period	Output Files
Off-peak Period	Op_lowInc_0_autos
	Op_medInc_0_autos
	Op_highInc_0_autos
	Op_veryHighInc_0_autos
	Op_lowInc_autos_lt_workers
	Op_medInc_autos_lt_workers
	Op_highInc_autos_lt_workers
	Op_veryHighInc_autos_lt_workers
	Op_lowInc_autos_ge_workers
	Op_medInc_autos_ge_workers
	Op_highInc_autos_ge_workers
	Op_veryHighInc_autos_ge_workers
Peak Period	Pk_lowInc_0_autos
	Pk_medInc_0_autos
	Pk_highInc_0_autos
	Pk_veryHighInc_0_autos
	Pk_lowInc_autos_lt_workers
	Pk_medInc_autos_lt_workers
	Pk_highInc_autos_lt_workers
	Pk_veryHighInc_autos_lt_workers
	Pk_lowInc_autos_ge_workers
	Pk_medInc_autos_ge_workers
	Pk_highInc_autos_ge_workers
	Pk_veryHighInc_autos_ge_workers

Table 79: Content of Accessibility Files

Field	Description
origLUZ	Origin LUZ
destLUZ	Destination LUZ
accessibility	Accessibility between origLUZ and destLUZ
zone_pairs	Number of TAZ pairs nested within the LUZ pair

12 Transit Project Level Forecast

When applying the model for the Regional Transportation Plan (RTP), all steps are applied from start to finish in what is called a “full-model application”. In this application, the model includes a feedback loop where the resulting travel times from highway assignment are “fed back” into the demand models. This is done to ensure consistency between travel times used for the demand choices and the travel times computed after assignment of these trips to the network. It is well documented in the modeling community that the feedback loop process introduces some level of variability into the results. This variability is often negligible when viewing aggregate regional results; however, when analyzing a single project, this variability can make it difficult to isolate the differences solely based on the project.

To mitigate this effect, the model is only partially run for a Build Scenario where a number of inputs into the mode choice models are consistent with the No Build Scenario. Previous guidance from the Federal Transit Administration (FTA) for New Starts funding involved transit project sponsors using the SUMMIT application to demonstrate that project benefits were being isolated to the actual project and that the variability associated with feedback loops and assignment weren’t being accumulated. This was done by using as many consistent inputs into the mode choice model as possible between the No Build and Build scenarios, and while the new guidance no longer requires the SUMMIT model, the lessons learned from its use still remain.

In the previous trip-based model (TBM), this approach was straightforward as the demand tables from the No Build scenario could be copied into the Build directory and then certain components of the model could be executed separately. Therefore, in the TBM, the person demand in this application would be identical. For a transit project, the only difference in the Build Scenario from the No Build Scenario would be the transit level-of-service skims.

However, in the activity-based model (ABM), the project level forecast is applied in a slightly different manner due to the nature of ABMs. The ABM includes two levels of mode choice – one at the tour level and one at the trip level which occurs after tour modes have been chosen. Trip mode choice is highly dependent on tour level mode choice because the availability of modes at the trip level is dependent on the mode chosen for the tour. For example, if a person’s tour mode choice to/from work is transit, they do not have a car available to them to make drive alone trips throughout the work day. Because of this, the ABM project level forecast does involve the use of more model components than the previous TBM and the application of these components must be done in a specific manner.

Important note: This should not be considered official guidance from FTA. It is the project sponsor’s responsibility to engage FTA in the methods that will be used for developing transit forecasts as part of a New Starts application.

12.1 No Build Scenario

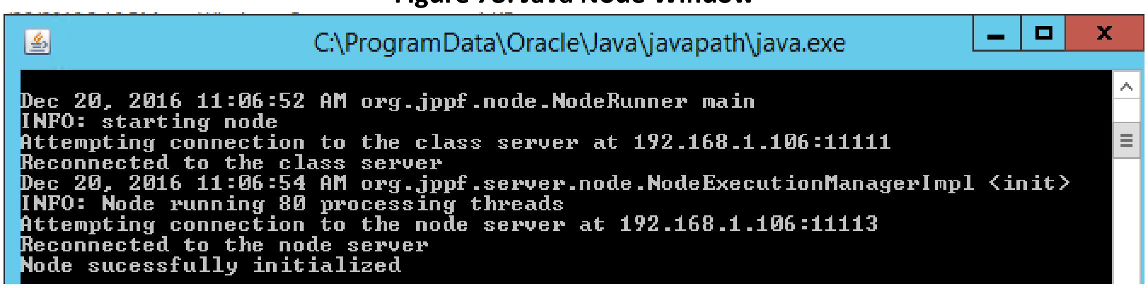
The No Build Scenario is actually run in the same manner as is used for the RTP which is running the full model from start to finish. However, because the model holds choices in memory, the user must keep the household manager open at the end of the No Build run while closing the nodes. In some applications of the model, the Cube script has been modified to launch and close the nodes from script. It is imperative that the household manager not be closed after the No Build Scenario finishes; therefore, the automated closing of the Java nodes must be removed in this application.

How to handle the file management is at the user's discretion, but there are a number of files from the No Build run that the user must either copy to another location or modify the file names. The approach used in testing for ARC was to create a No Build directory and copy the CT-RAMP specific output and travel time skims into that directory. These specific files are as follows:

- hhData.CSV
- indivTourData.CSV
- indivTripData.CSV
- jointTourData.CSV
- jointTripData.CSV
- personData.CSV
- tripData.CSV
- All highway and transit travel skims

If not copied or renamed, these CT-RAMP generated files will be overwritten in the Build scenario application. To shut down the Java node, the user can simply open the Java window and click the 'X' to close the window. It is recommended that the user pay attention to which Java window is associated with the node upon launching the node initially. An example of the Java node upon launching is provided below.

Figure 78: Java Node Window



12.2 Build Scenario

The Build scenario needs to be re-run in the same directory as the original No Build scenario where the household manager is still open and the nodes should be restarted. The user will need to re-build the transit skims to reflect the Build Scenario transit service. Preferably, this is done using the same background highway network as previously used to develop the No Build scenario skims. This means that the user should give thought to how the Build Scenario will be coded prior to running the No Build scenario. For example, if highway links require splitting to incorporate the Build project, it is recommended that those network splits are made in the no build as well. Ideally, both the No Build and Build transit routes would work seamlessly with the same highway network to ensure as little variation as possible.

Upon creation of the Build Scenario transit skims, the user must replace the No Build skims in the model directory. The last step prior to running CT-RAMP is updating the properties file (arcTourBased.properties) to only apply certain components of CT-RAMP. Basically, all models through the generation of tours are not re-run, but all models from tour mode choice and onward are run. This is done through use of the True/False keywords in the properties file (each component and keyword is detailed in the next subsection). Once the Build scenario completes CT-RAMP, the user must again determine how to manage the output files for later use. In testing for ARC, similar to the No Build scenario, the CT-RAMP specific files and skims were copied into a different directory for the Build scenario.

12.3 Step by Step Application

The following outlines the steps:

1. Regular run of ARC Model – call this result the No Build scenario
2. Close the nodes and restart the nodes for the next run
3. Household manager should remain open after the No Build run
4. Update the input transit skims to run the Build scenario.
5. Change the properties file as follows
    a. RunModel.RestartWithHhServer= immc
    b. RunModel.UsualWorkAndSchoolLocationChoice= False
    c. RunModel.AutoOwnership= False
    d. RunModel.FreeParking= False
    e. RunModel.CoordinatedDailyActivityPattern= False
    f. RunModel.IndividualMandatoryTourFrequency= False
    g. RunModel.MandatoryTourDepartureTimeAndDuration= False
    h. RunModel.JointTourFrequency= False
    i. RunModel.JointTourLocationChoice= False
    j. RunModel.JointTourDepartureTimeAndDuration= False
    k. RunModel.IndividualNonMandatoryTourFrequency= False
    l. RunModel.IndividualNonMandatoryTourLocationChoice= False
    m. RunModel.IndividualNonMandatoryTourDepartureTimeAndDuration= False
    n. RunModel.AtWorkSubTourFrequency= False
    o. RunModel.AtWorkSubTourLocationChoice= False
    p. RunModel.AtWorkSubTourDepartureTimeAndDuration= False

Following models are rerun (no change in their setting):

RunModel.MandatoryTourModeChoice= True
RunModel.JointTourModeChoice= True
RunModel.IndividualNonMandatoryTourModeChoice= True
RunModel.AtWorkSubTourModeChoice= True
RunModel.StopFrequency= True
RunModel.StopLocation= True
RunModel.StopTiming= True
RunModel.TripModeChoice= True

Note: The files wsLocResults.csv, cdapResults.csv, and aoResults.csv are not generated again as there will be no change in these files. All other inputs are recreated using the new inputs.

13 Activity-Based Model Visualization Tool (ABMVIZ)

13.1 Introduction to ABMVIZ

ABMVIZ is a visualization tool designed to graphically display the forecasting results produced by the activity-based model. The tool is hosted on the website GitHub and is publicly accessible. The tool provides graphics for forecasting scenarios modeled by the ARC ABM.

13.1.1 ABMVIZ v.2

The original version of ABMVIZ (version 1) was developed as a desktop-based application built in Adobe AIR (Flash). The current version (version 2) is a web-based tool, hosted on GitHub and scripted using HTML5.

13.1.2 Definitions and Acronyms

Table 80: ABMVIZ Definitions and Acronyms

CODE	DEFINITION
AUTO	All automobile modes
BIKE	Bicycle
DRIVEALONE	Private automobile with a single occupant (the driver only), using either general purpose or toll lanes
DRIVEALONEFREE	Private automobile with a single occupant (the driver only) using general purpose lanes only
DRIVEALONEPAY	Private automobile with a single occupant (the driver only) using toll lanes, including HOT lanes
KNR	Kiss-and-ride
KNR_ALLTRN	Premium and Non-premium transit by kiss-and-ride access/egress
KNR_PRMTRN	Premium transit by kiss-and-ride access/egress
PNR	Park-and-ride
PNR_ALLTRN	Premium and Non-premium transit by park-and-ride access/egress
PRN_PRMTRN	Premium transit by park-and-ride access/egress
SCHOOL_BUS	School district-operated school bus
SHARED2	Private automobile with two occupants (the driver and a passenger) using either general purpose or toll lanes
SHARED2FREE	Private automobile with two occupants (the driver and a passenger) using general purpose lanes only
SHARED2PAY	Private automobile with two occupants (the driver and a passenger) using toll lanes, including HOT lanes and 2 passenger HOV lanes
SHARED3	Private automobile with three or more occupants (the driver and two or more passengers) using either general purpose or toll lanes
SHARED3FREE	Private automobile with three or more occupants (the driver and two or more passengers) using general purpose lanes only

SHARED3PAY	Private automobile with three or more occupants (the driver and two or more passengers) using toll lanes, including HOT lanes and 3+ passenger HOV lanes
TRANSIT	All modes of public transit, free and paid
WALK	Walking
WALK_ALLTRN	Premium and Non-premium transit by walking access/egress
WALK_PRMTRN	Premium transit by walking access/egress
WALK_TRANSIT	Any transit by walking access/egress

13.1.3 Scenario Explanations

Table 81: Scenario Explanations

SCENARIO	DEFINITION
BASE2010	Base year 2010 Estimate
RP2015	Regional Plan 2015 Forecast
RP2017	Regional Plan 2017 Forecast
RP2020	Regional Plan 2020 Forecast
RP2024	Regional Plan 2024 Forecast
RP2030	Regional Plan 2030 Forecast
RP2040	Regional Plan 2040 Forecast
NB2040	No-Build 2040 Forecast

13.2 Visuals

In the current version, there are six visuals available for each scenario. These visuals show the different forecasted conditions of each scenario by presenting the data as a map, chart, or graph. Each of the descriptions below addresses one of the visuals and provides instructions for interacting with the visual and methods of interpreting the visual.

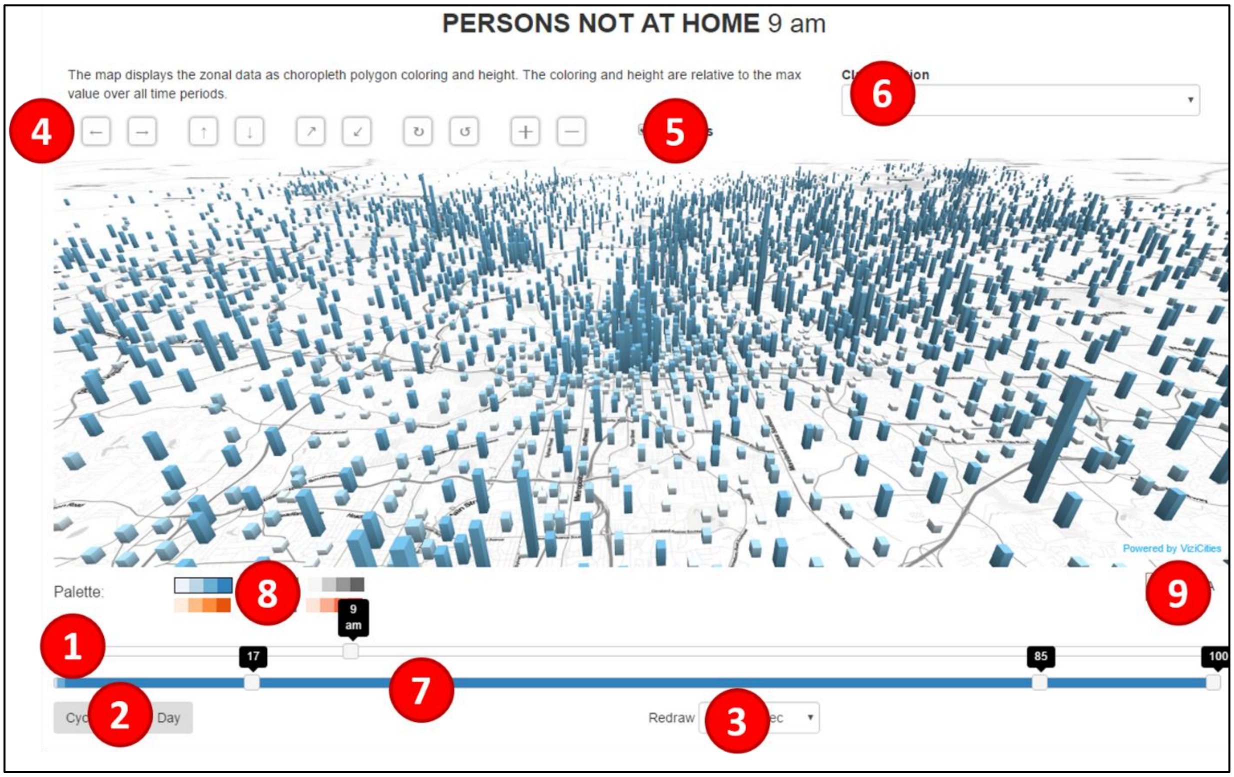
The terms defined here will be used frequently in the descriptions below:

- "Mode" refers to how a person is traveling, for example by car or bus or train. The complete list of modes used in these visuals can be found in Section 13.1.2 Definitions and Acronyms.
- "TAZ" stands for Traffic Analysis Zone and refers to a unit of geography common in transportation planning. The entire Atlanta metropolitan area is divided into 5,873 TAZs.

13.2.1 Persons Not At Home

The Persons Not At Home visual is a 3D map showing how many people in a given TAZ at a given time are away from their home. Figure 79 below displays the locations of the interactive features of the Persons Not At Home visual. The interactive features are discussed below.

Figure 79: Locations of the interactive features of the Persons Not At Home visual



Interactive Features List:

1) Time slider
2) Cycle Through Day button
3) Redraw menu
4) View adjustment controls
5) Centroid box
6) Classification menu
7) Classification bar
8) Classification color palette
9) Null color palette

The primary function of this visual is to demonstrate how people are moving around the city throughout the day. To activate the cycle, either manually adjust the time slider (found at Location 1), or click the "Cycle Through Day" button (found at Location 2) to automate the 24-hour progression beginning at midnight. If you would like to adjust the rate at which the hours pass in the automated progression, select a new time from the "Redraw" menu (found at Location 3). The default view displayed in the map panel can be adjusted using the controls found at Location 4.

By default, the map is set to display a single column for each TAZ; these columns are located on the centroid of each TAZ. To display a choropleth color block over the entire TAZ area, instead of just the centroid, uncheck the Centroid box found at Location 5.

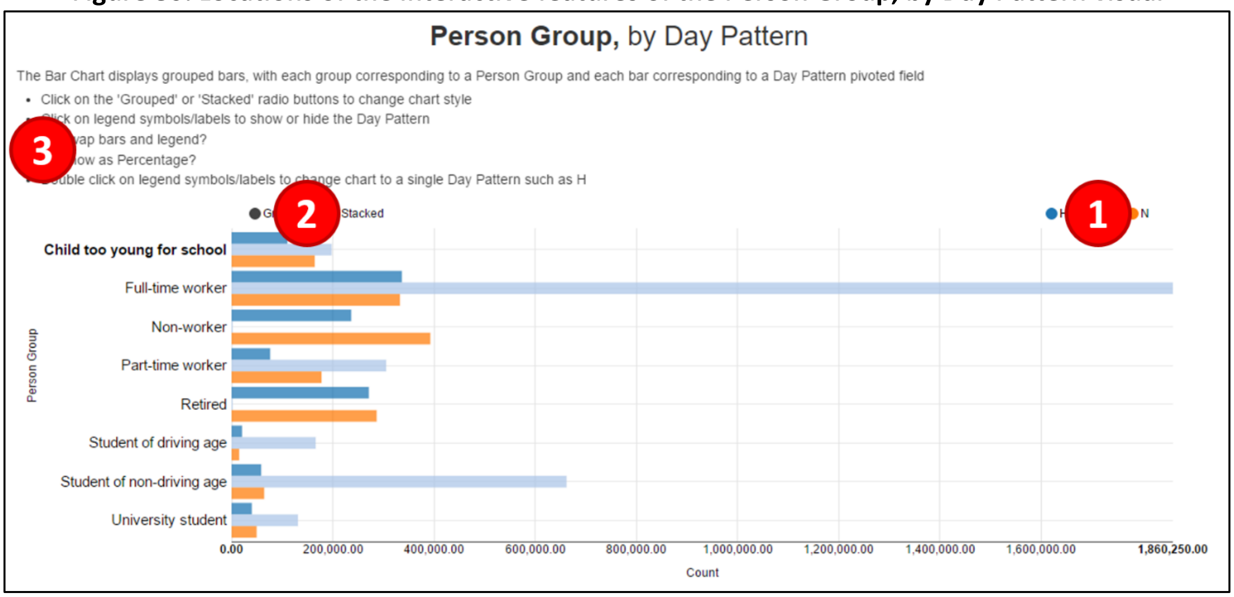
The saturation of color in each column represents the classification bracket that the TAZ is currently in. The Classification menu (found at Location 6) allows the user to set the type of classification system displayed in the visual. By default, a Quartiles classification system is used, but an Even Interval or a Custom system can be selected. Once a Classification system is selected at Location 6, the corresponding Classification Bar (found at Location 7) will adjust and can be manipulated directly to create a Custom setting. The color palette of the Classification system can be adjusted by selecting one of the six alternatives given at Location 8. To highlight a TAZ with no persons not at home, use the color selector found at Location 9; the default color is white.

In the Persons Not At Home visual, the relative height of the columns indicates how many people in that TAZ are away from home. Therefore, the taller the column, the more people are away from home. When the cycle is automated, the user will notice that during the earliest morning hours and latest evening hours, the columns are almost flat across the map, indicating that most people are at home. As the day progresses, the columns grow much taller and cluster around job-rich areas such as Midtown and Cumberland. The tallest columns are typically seen around the airport and the Emory University/CDC area. These tall columns represent a large number of people concentrated into a small area.

13.2.2 Person Group, by Day Pattern

The Person Group, by Day Pattern visual is a bar chart showing a summary of the projected population's typically daily travel pattern. Figure 80 below displays the locations of the interactive features of the Person Group, by Day Pattern visual. The interactive features are discussed below.

Figure 80: Locations of the interactive features of the Person Group, by Day Pattern visual



Interactive Features List:

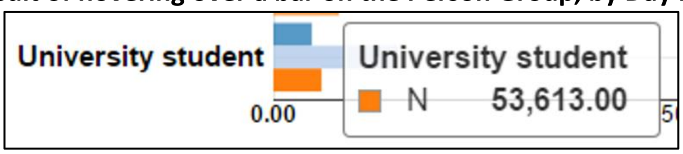
1) Display toggle
2) Day Pattern toggle (H = Home, M = Mandatory, N = Non-mandatory)
3) Data choice toggle

The primary function of this visual is to describe how the individuals in a given "person group" are expected to routinely travel. For example, in the scenario pictured in Figure 80, it is expected that 1,860,250 people classified as "Full-time Workers" make "Mandatory" trips to work on a daily basis.

By default, the visual displays all three types of trips (home, mandatory, and non-mandatory) at the same time. To choose which types of trip to display, toggle between the options found at Location 1 in Figure 80. To see a single type of trip, double click on its toggle button. The visual can either show the different trip types as individual bars grouped next to each other, or as a single bar with all trip types stacked together. Use the buttons found at Location 2 to choose the display method.

The data displayed in the visual can be manipulated with the toggles found at Location 3. The first option is to switch the person type and day pattern rows. By default, the visual displays each person group on the y-axis and displays three bars (representing the three day patterns) per person group. If the "Swap bars and legend" button is toggled, the visual will display the three day patterns on the y-axis and will display a bar for each of the eight person groups per day pattern. Additionally, the visual can display a percentage for each bar (instead of a count) by toggling the "Showing as Percentage" button. To determine the exact count or percentage of a particular bar, hover the mouse over the bar, as seen in Figure 81 below.

Figure 81: Result of hovering over a bar on the Person Group, by Day Pattern visual

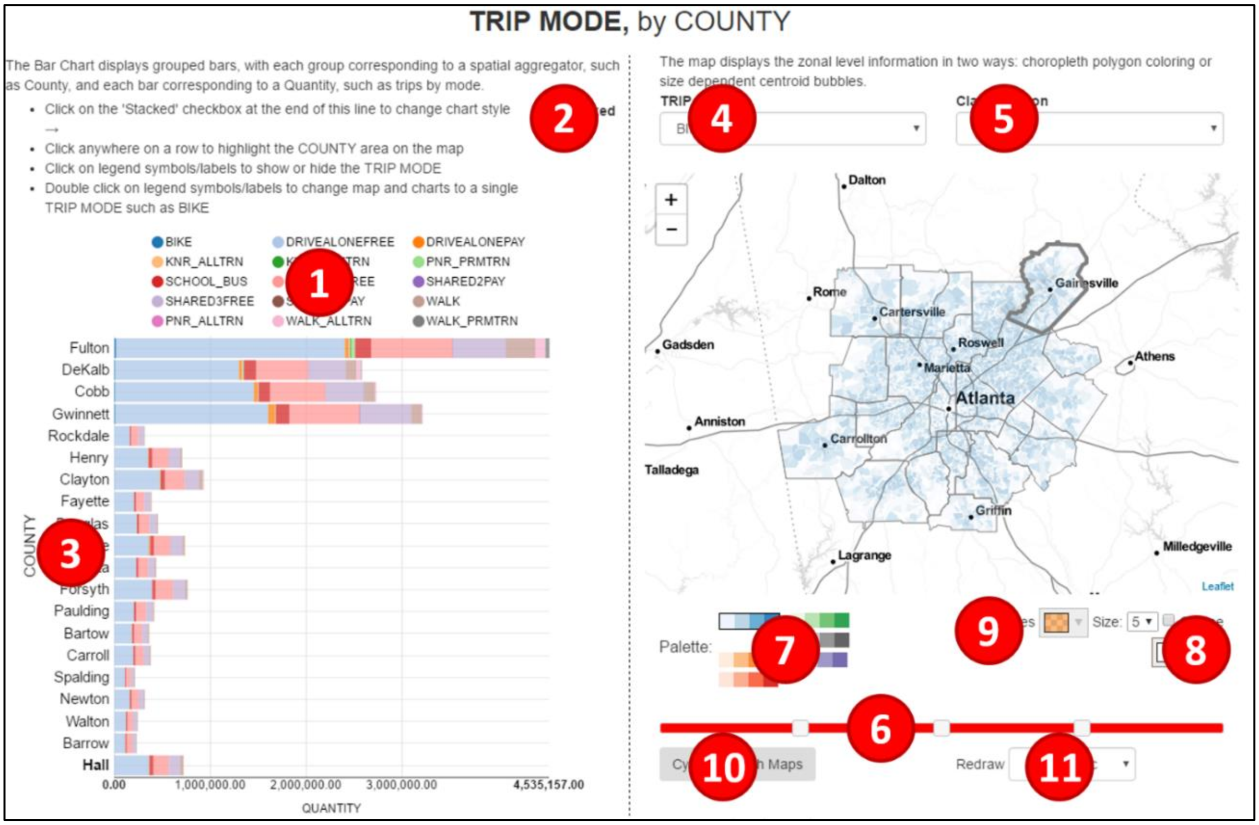


In the Person Group, by Day Pattern visual, "H" represents home activities, "M" represents mandatory activities, and "N" represents non-mandatory activities.

13.2.3 Trip Mode, by County

The Trip Mode, by County visual is a combination bar group and map showing the breakdown of modes used by each county in the region. Figure 82 below displays the locations of the interactive features of the Trip Mode, by County visual. The interactive features are discussed below.

Figure 82: Locations of the interactive features of the Trip Mode, by County visual



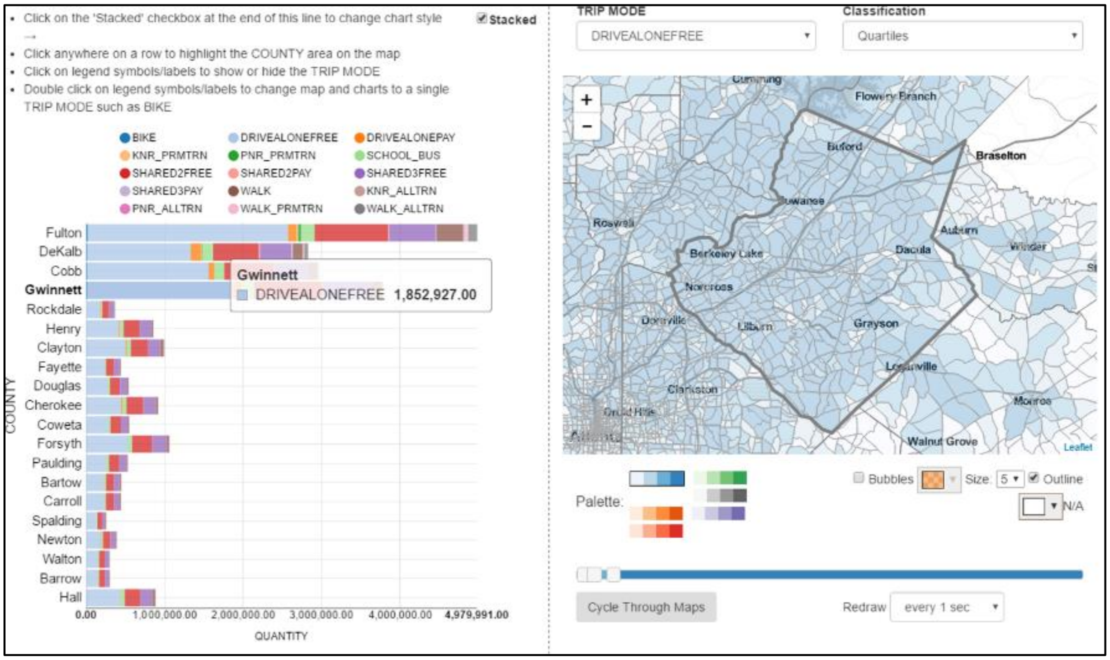
Interactive Features List:

1) Mode selection list
2) Stacked toggle button
3) County List
4) Trip mode selection menu
5) Classification selection menu
6) Classification bar
7) Color palette selection list
8) Null color palette selector
9) Bubble toggle
10) "Cycle Through Maps" button
11) "Redraw" menu

The left side of the Trip Mode, by County visual is a bar graph showing the 20 counties composing the Atlanta region. Each county is given a single bar with each of the 15 possible modes represented in a different color and stacked together. The toggle buttons found at Location 1 on Figure 82 allow the user to choose which modes are displayed on the bar graph. Multiple modes can be selected, or, to display a single mode, double click that mode's toggle button. To see each mode as a separate bar, instead of all modes stacked into one bar, toggle the button found at Location 2. Hovering the mouse over any county name found at Location 3 will thickly outline that county on the map seen on the right side of the visual.

Hovering the mouse over any of the bars will thickly outline that county and display an exact count of the mode hovered over. Figure 83 below gives an example of these hovering motions.

Figure 83: Result of hovering over a bar on the Trip Mode, by County visual



The map displays the intensity of use of a selected mode by TAZ across the 20 county region. The mode to be displayed can be selected from the drop-down menu found at Location 4. The default classification system for the map is Quartiles, and the user can select a different classification system from the drop-down menu found at Location 5. When a classification system is selected, the classification bar (found at Location 6) will automatically adjust and can be manually manipulated to a custom setting.

The default color palette is blue, but a different color palette can be chosen from the options found at Location 7. To highlight the TAZs in which the selected mode is not used by any person, adjust the color displayed in the selector found at Location 8. To display bubbles overs the TAZs (instead of the default choropleth polygons), toggle the “Bubbles” button found at Location 9.

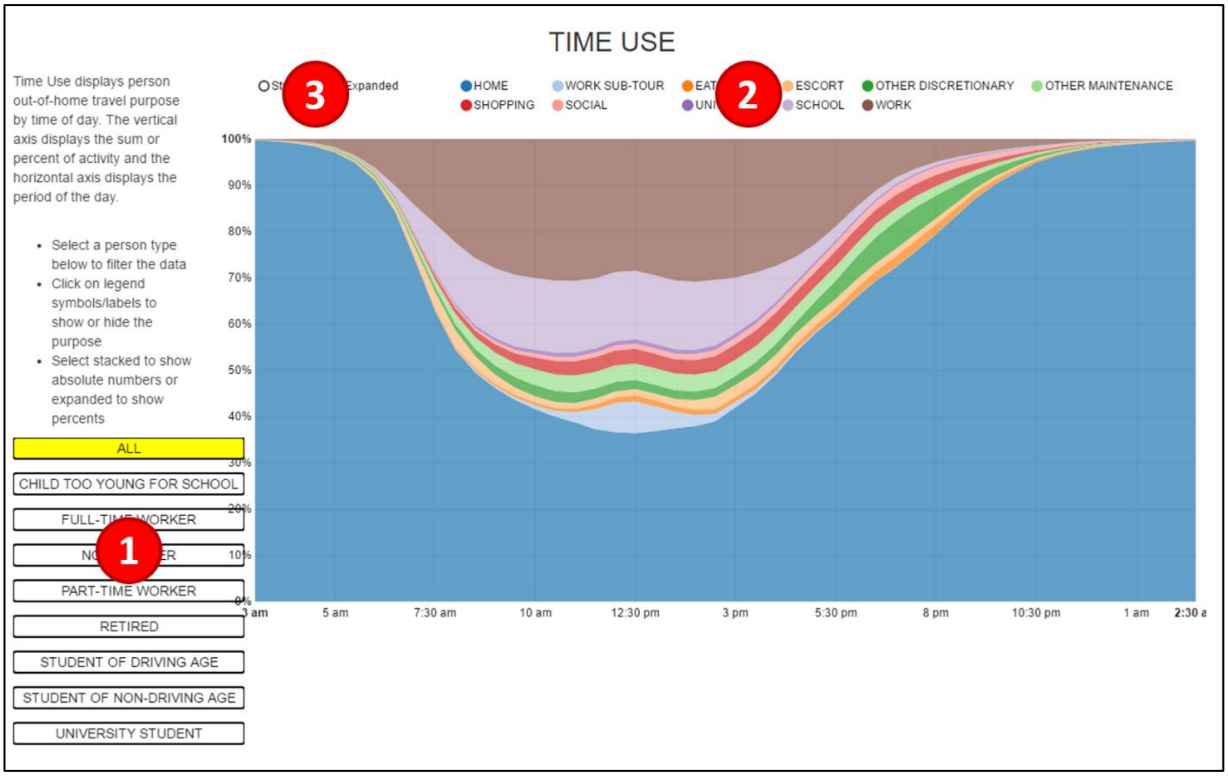
The map can be automated to cycle through all of the available mode choices by clicking the “Cycle Through Maps” button found at Location 10. The rate at which the modes shift can be adjusted by selecting a new rate from the “Redraw” menu found at Location 11.

The Trip Mode, by County visual is useful for comparing the volume of trips undertaken by different counties and TAZs in the region. It is obvious that driving alone will remain the dominant trip mode for years to come, however, there is significant interest in alternative modes throughout the region. The map portion illustrates the intensity of use of a particular mode by more heavily saturating the TAZs where use is higher. Because of the outline and zoom functionality of the map, it is ease to examine individual TAZs for specific mode choice patterns.

13.2.4 Time Use

The Time Use visual is a graphic showing the breakdown of how people in different Person Groups are likely to spend their time. Figure 84 below displays the locations of the interactive features of the Time Use visual. The interactive features are discussed below.

Figure 84: Locations of the interactive features of the Time Use visual



Interactive Features List:

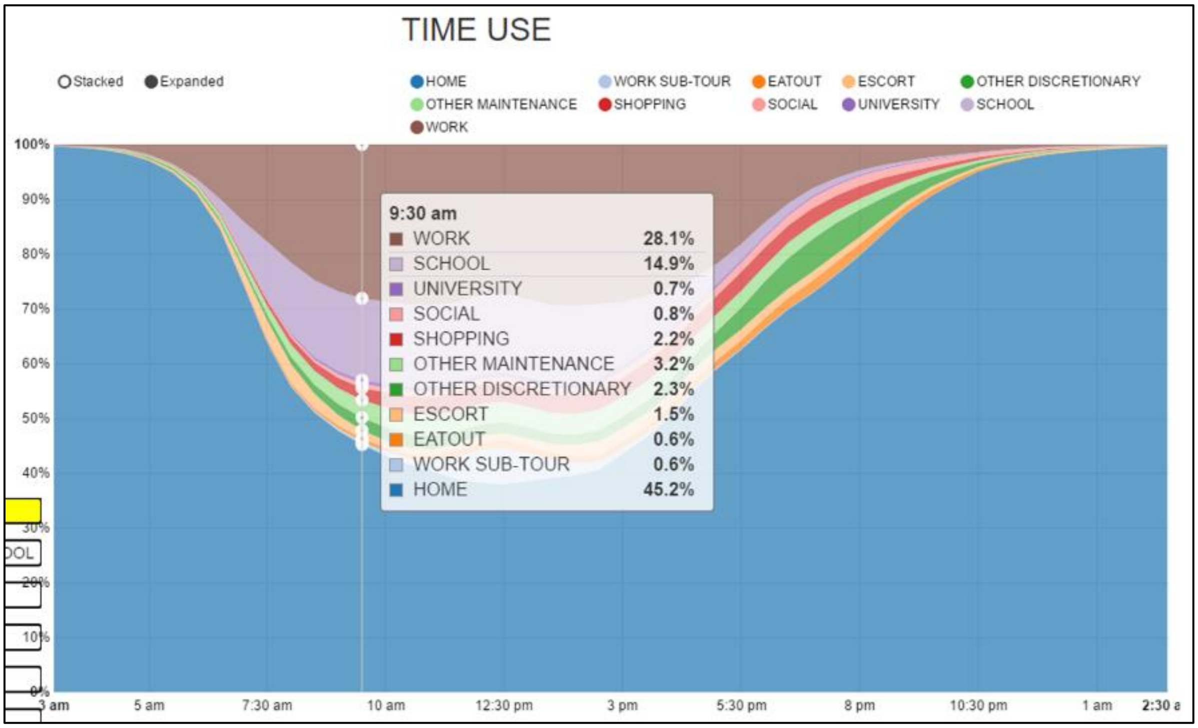
1. Person Group selection menu
2. Trip purpose buttons
3. Display type buttons

The Time Use visual summarizes the different types of trips each Person Group is expected to make throughout an average day. By default, the visual displays information about all Person Groups averaged together. To select a single Person Group, use the menu found at Location 1. To choose which trip purposes are displayed in the visual, toggle the color-coded buttons found at Location 2.

The data in the Time Use visual is given as a percentage by default. To change the presentation method to count instead, use the buttons found at Location 3.

Once the desired inputs are selected, the user can hover over the visual to see exact percentages or counts for any given hour (see Figure 85 below for an example).

Figure 85: Results of hovering over the Time Use visual

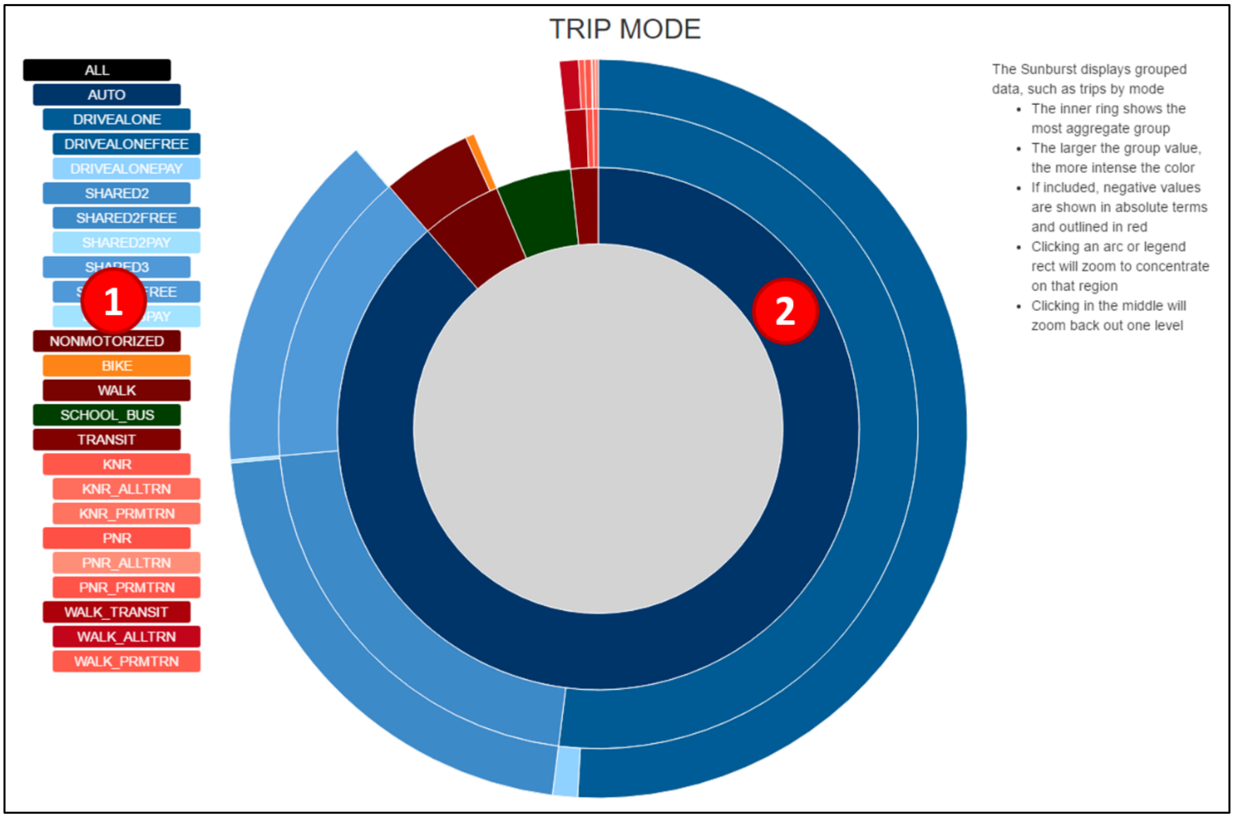


The Time Use visual is important for understanding how different Person Groups typically spend their time. Because the visual uses percentages, it can be difficult to interpret. The overwhelming blue section in the default view represents the percentage of people who are at home, which is many people for the majority of the day. From the morning rush hour to the evening rush hour, the well in the middle of the visual grows, meaning that more people are away from home.

13.2.5 Trip Mode

The Trip Mode visual is a sunburst chart showing the breakdown of modes used in a given scenario. Figure 86 below displays the locations of the interactives features of the Trip Mode visual. The interactive features are discussed below.

Figure 86: Locations of the interactives features of the Trip Mode visual



Interactive Features List:

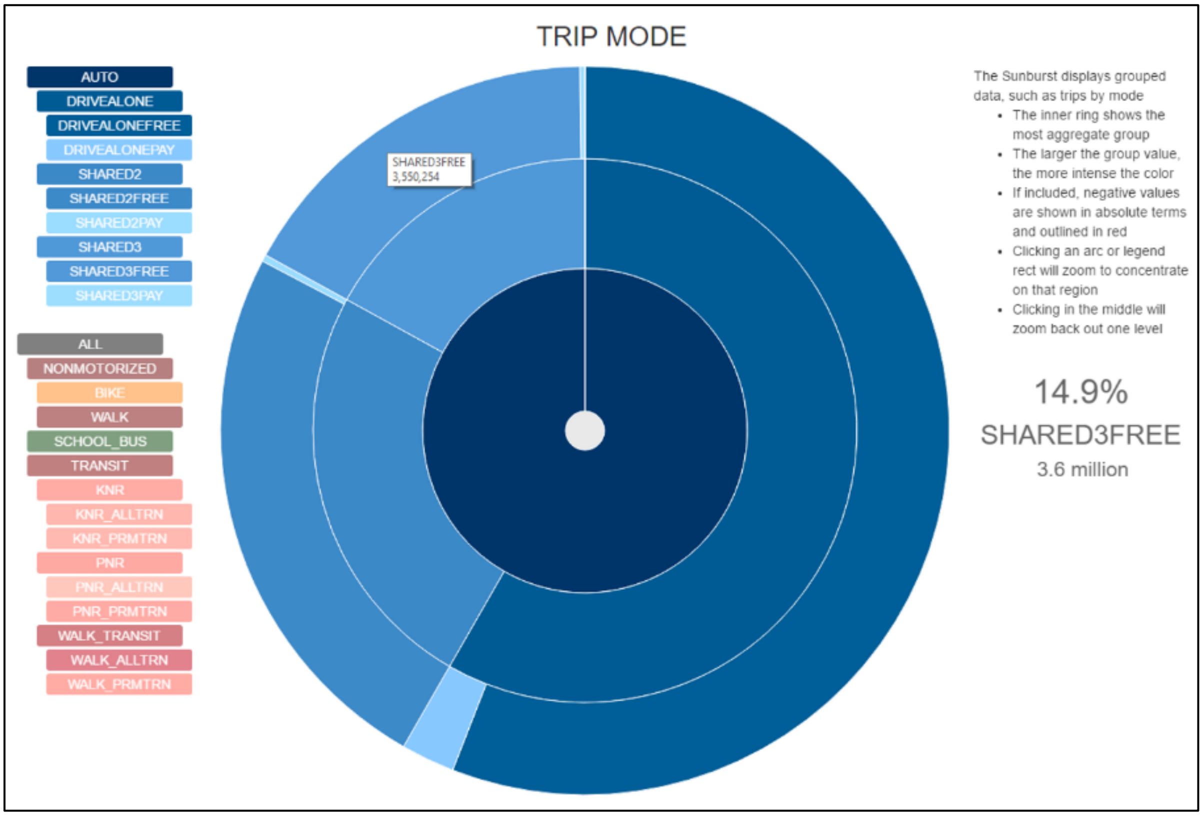
1. Mode selection menu
2. Mode selection rings

By default, the Trip Mode visual displays a summary of all modes used on an average day. To narrow down the modes displays, choose from the mode selection menu found at Location 1. The list is both color-coded and hierarchically organized. Those modes indented in the list are more specific categories of the mode under which it is nested. For example, under "All" (all modes) is "Transit" (transit modes) and under "Transit" is "KNR" (kiss-and-ride access/egress to transit) and under "KNR" is "KNR_PRMTRN" (kiss-and-ride access/egress to premium transit). Figure 87 below represents the user filtering out all modes expect Auto modes.

Alternatively, the user can narrow the modes displayed by clicking on the color-coded rings found at Location 2. The gray core is inert, but each subsequent ring represents a different level in the hierarchy of modes. The innermost ring corresponds to the first-level indented modes "Auto," "Nonmotorized," "School_Bus," and "Transit." The second ring corresponds to the second-level indented modes "DriveAlone," "Shared2," "Shared3," "Bike," "Walk," "KNR," "PNR," and "Walk_Transit." The third ring corresponds to the third-level modes and so on. Clicking on any of these rings will focus on that mode and level within the hierarchy, and filter out all other modes.

Hovering the mouse over any of the rings at any level of focus will display both the percentage and count of that particular mode on the right side of the visual (see Figure 87 below for an example).

Figure 87: Result of hovering over a ring on the Trip Mode visual

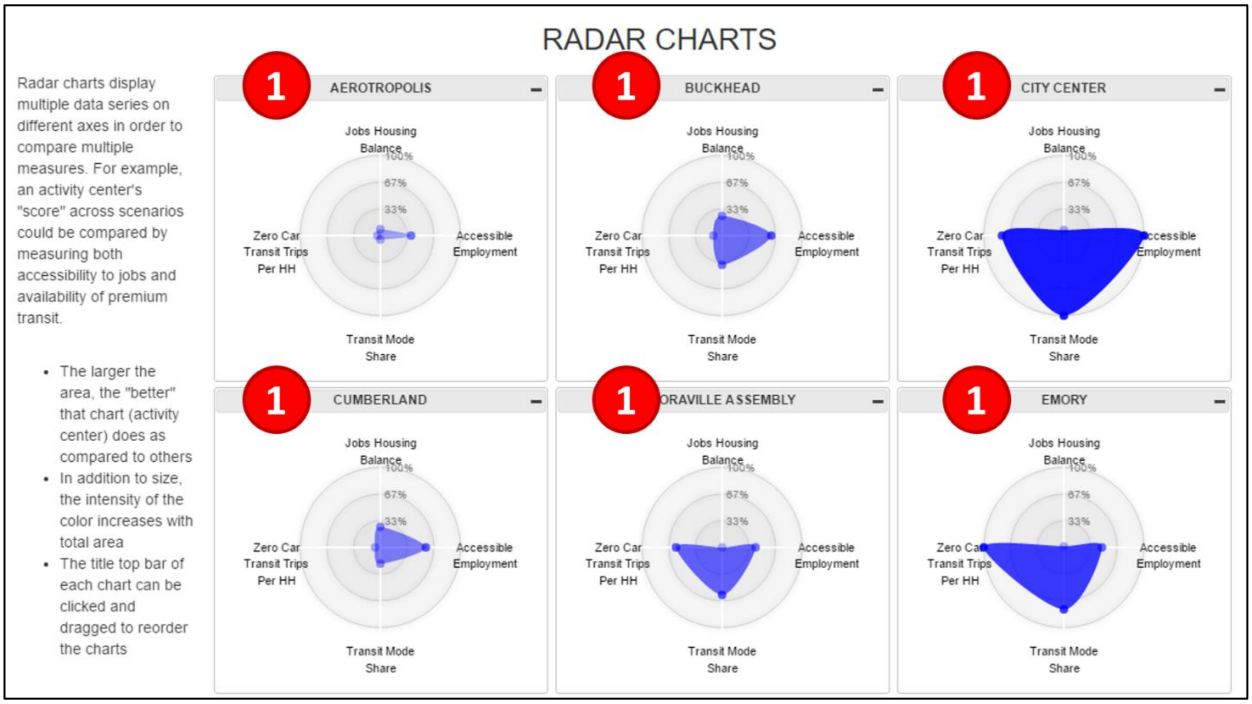


The center-most ring represents the broadest categorization of modes, as general as "Auto," and the user can see that the dark blue ring representing "Auto" is the first, and therefore the most common, mode chosen. The second and third rings show the various subsets of the broader classifications, and the entirety of the subset is given only as much space as the larger classification needed. For example, in the first ring, the maroon segment representing "Nonmotorized" modes takes up a small portion of the ring. The subsets of "Nonmotorized" are "Walk" and "Bike," which together take up portions of the second ring equal to that of the "Nonmotorized" portion of the first ring. The system of concentric rings makes it easy to compare the prevalence of various modes.

13.2.6 Radar Charts

The Radar Charts visual is a collection of radar charts showing the combinational effect of four measures for a given location. Figure 88 displays the locations of the interactive features of the Radar Charts visual, and the features are discussed below.

Figure 88: Locations of the interactive features of the Radar Charts visual

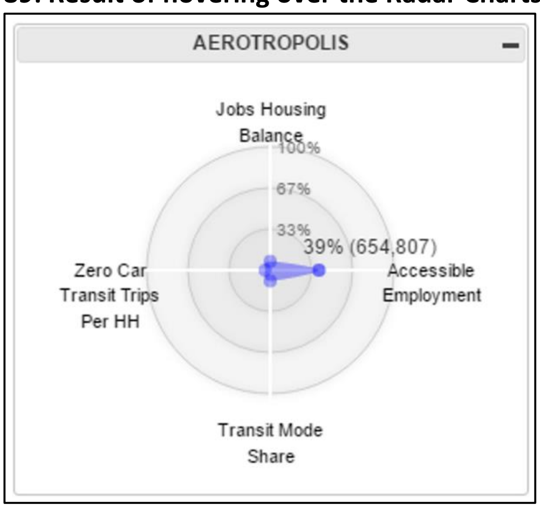


Interactive Features List:

1. Title bars

Radar charts represent 21 major activity centers in metro Atlana. By default, they are displayed in alphabetical order. To reorder the plots to better compare specific regions, click and drag the title bars (found at Locations 1) into the appropriate order. Exact percentages and numbers for each of the four metrics can be found by hovering the mouse over the terminal point of any of the metrics (see Figure 89 below).

Figure 89: Result of hovering over the Radar Charts visual



The Radar Charts are best for quick visual comparisons of the activity centers around the region. Each of the metrics is configured to cover a larger area of the circle if the region performs "better" at that particular metric. Additionally, the saturation of color increases proportionally with the size of the area covered.

Appendix

A1. DATABASE CONTROL TABLES

Table 82: MAZ (ARC TAZ) Level Control Totals

control_totals_maz		
FIELD	**Description**	**Type (H=HH/P=Per/G=Geography)**
MAZ	MAZ (ARC TAZ) Identifier	G
TAZ	TAZ (ARC PECAS Zone) Identifier	G
DISTRICT2	District (ARC County group) Identifier	G
PUMA	PUMA Identifier	G
HH	Total households in MAZ	H
HHSIZE1	Household size 1 person	H
HHSIZE2	Household size 2 person	H
HHSIZE3	Household size 3 person	H
HHSIZE4	Household size 4 person	H
HHSIZE5	Household size 5 person	H
HHSIZE6	Household size 6+ person	H
HHINC1	Household income <$25k	H
HHINC2	Household income $25k-$60k	H
HHINC3	Household income $60k-$120k	H
HHINC4	Household income $120k+	H
HWORK0	Household workers 0	H
HWORK1	Household workers 1	H
HWORK2	Household workers 2	H
HWORK3	Household workers 3+	H

Table 83: TAZ (ARC PECAS Zone) Level Control Totals

control_totals_taz		
FIELD	**Description**	**Type (P=Per /G=Geography)**
TAZ	TAZ (ARC PECAS Zone) Identifier	G
DISTRICT2	District (ARC County group) Identifier	G
CL23WhiteCollar	Total persons by PECAS occupation category	P
CL24Services		P
CL25Health		P
CL26Retail		P
CL27BlueCollar		P

Table 84: Meta-Control (ARC County) Totals

control_totals_meta		
FIELD	**Description**	**Type (P=Per /G=Geography)**
REGION2	Region (ARC County group) Name	G
DISTRICT2	District (ARC County group) Identifier	G
AGE0014	Persons aged between 0 and 14	P
AGE1524	Persons aged between 15 and 24	P
AGE2534	Persons aged between 25 and 34	P
AGE3544	Persons aged between 35 and 44	P
AGE4554	Persons aged between 45 and 54	P
AGE5564	Persons aged between 55 and 64	P
AGE6574	Persons aged between 65 and 74	P
AGE7584	Persons aged between 75 and 84	P
AGE85	Persons aged between 85 and above	P

A2. DATABASE PUMS TABLES

Table 85: PUMS Processed Household table

hhtable		
FIELD	**Description**	**Source**
HHNUM	User defined unique HH ID	User-defined variable
NWRKRS_ESR	Number of workers in the household - derived from `ESR’	User-defined variable
SERIALNO	Unique housing PUMS record identifier	PUMS HH Dataset
All other PUMS HH fields	All other PUMS HH fields	PUMS HH Dataset

Table 86: PUMS Processed Person Table

perstable		
FIELD	**Description**	**Source**
HHNUM	User defined unique HH ID	User-defined variable
EMPLOYED	Is person employed (derived from `ESR`)	User-defined variable
PECAS_OCC	PECAS Occupation code	User-defined variable
SERIALNO	Unique housing PUMS record identifier	PUMS HH Dataset
SPORDER	Unique person PUMS record identifier	PUMS Person Dataset
All other PUMS person fields	All other PUMS person fields	PUMS Person Dataset

A3. DATABASE OUTPUT TABLES

Table 87: Synthetic Population Household Table in Unexpanded Form

synpop_hh	
FIELD	**Description**
TEMPID	Household ID
DISTRICT	District code (ARC County) in which HH is located
PUMA	PUMA code for HH record
TAZ	TAZ (ARC PECAS zone) code in which HH is located
MAZ	MAZ (ARC TAZ) code in which HH is located
WGTP	Housing Weight
FINALPUMSID	HH ID generated during population synthesis
FINALWEIGHT	HH weight generated during population synthesis
SERIALNO	Unique housing PUMS record identifier
NWRKRS_ESR	Number of workers in the household
*	Other PUMS HH fields

Table 88: Synthetic Population Person Table in Unexpanded Form

synpop_person	
FIELD	**Description**
TEMPID	Household ID
DISTRICT	District code (ARC County) in which HH is located
PUMA	PUMA code for HH record
TAZ	TAZ (ARC PECAS zone) code in which HH is located
MAZ	MAZ (ARC TAZ) code in which HH is located
WGTP	Housing Weight
FINALPUMSID	Person ID generated during population synthesis
FINALWEIGHT	Person weight generated during population synthesis
SPORDER	Person number in HH
EMPLOYED	Is person employed
PECAS_OCC	PECAS Occupation code for this person
*	Other PUMS Person fields

* The definitions for other PUMS variables can be found in the 2007-2011 ACS PUMS data dictionary:
http://www2.census.gov/programs-surveys/acs/tech_docs/pums/data_dict/PUMS_Data_Dictionary_2007-2011.pdf

Table 89: Synthetic Population Household Table in Expanded Form

households	
FIELD	**Description**
HHID	Unique household ID
TEMPID	Unexpanded household ID
DISTRICT	District code (ARC County) in which HH is located
PUMA	PUMA code for HH record
TAZ	TAZ (ARC PECAS zone) code in which HH is located
MAZ	MAZ (ARC TAZ) code in which HH is located
WGTP	Housing Weight
FINALPUMSID	HH ID generated during population synthesis
FINALWEIGHT	HH weight generated during population synthesis
SERIALNO	Unique housing PUMS record identifier
NWRKRS_ESR	Number of workers in the household
*	Other PUMS HH fields

Table 90: Synthetic Population Person Table in Expanded Form

persons	
FIELD	**Description**
HHID	Unique household ID
PERID	Unique person ID
TEMPID	Unexpanded household ID
DISTRICT	District code (ARC County) in which HH is located
PUMA	PUMA code for HH record
TAZ	TAZ (ARC PECAS zone) code in which HH is located
MAZ	MAZ (ARC TAZ) code in which HH is located
WGTP	Housing Weight
FINALPUMSID	Person ID generated during population synthesis
FINALWEIGHT	Person weight generated during population synthesis
SPORDER	Person number in HH
EMPLOYED	Is person employed
PECAS_OCC	PECAS Occupation code for this person
*	Other PUMS Person fields