SHRP 2 Capacity Project C10B

Network Users Guide for the Dynamic, Integrated Model System: Sacramento-Area Application

SHRP 2
STRATEGIC HIGHWAY RESEARCH PROGRAM
Accelerating solutions for highway safety, renewal, reliability, and capacity

TRANSPORTATION RESEARCH BOARD
OF THE NATIONAL ACADEMIES

SHRP 2 Capacity Project C10B

Network Users Guide for the Dynamic, Integrated Model System: Sacramento-Area Application

TRANSPORTATION RESEARCH BOARD
Washington, D.C.
2014
www.TRB.org

ACKNOWLEDGMENT

This work was sponsored by the Federal Highway Administration in cooperation with the American Association of State Highway and Transportation Officials. It was conducted in the second Strategic Highway Research Program, which is administered by the Transportation Research Board of the National Academies.

NOTICE

The project that is the subject of this document was a part of the second Strategic Highway Research Program, conducted by the Transportation Research Board with the approval of the Governing Board of the National Research Council.

The Transportation Research Board of the National Academies, the National Research Council, and the sponsors of the second Strategic Highway Research Program do not endorse products or manufacturers. Trade or manufacturers' names appear herein solely because they are considered essential to the object of the report.

DISCLAIMER

The opinions and conclusions expressed or implied in this document are those of the researchers who performed the research. They are not necessarily those of the second Strategic Highway Research Program, the Transportation Research Board, the National Research Council, or the program sponsors. The information contained in this document was taken directly from the submission of the authors. This material has not been edited by the Transportation Research Board.

SPECIAL NOTE: This document IS NOT an official publication of the second Strategic Highway Research Program, the Transportation Research Board, the National Research Council, or the National Academies.

THE NATIONAL ACADEMIES

Advisers to the Nation on Science, Engineering, and Medicine

The **National Academy of Sciences** is a private, nonprofit, self-perpetuating society of distinguished scholars engaged in scientific and engineering research, dedicated to the furtherance of science and technology and to their use for the general welfare. On the authority of the charter granted to it by Congress in 1863, the Academy has a mandate that requires it to advise the federal government on scientific and technical matters. Dr. Ralph J. Cicerone is president of the National Academy of Sciences.

The **National Academy of Engineering** was established in 1964, under the charter of the National Academy of Sciences, as a parallel organization of outstanding engineers. It is autonomous in its administration and in the selection of its members, sharing with the National Academy of Sciences the responsibility for advising the federal government. The National Academy of Engineering also sponsors engineering programs aimed at meeting national needs, encourages education and research, and recognizes the superior achievements of engineers. Dr. C. D. (Dan) Mote, Jr., is president of the National Academy of Engineering.

The **Institute of Medicine** was established in 1970 by the National Academy of Sciences to secure the services of eminent members of appropriate professions in the examination of policy matters pertaining to the health of the public. The Institute acts under the responsibility given to the National Academy of Sciences by its congressional charter to be an adviser to the federal government and, upon its own initiative, to identify issues of medical care, research, and education. Dr. Victor J. Dzau is president of the Institute of Medicine.

The **National Research Council** was organized by the National Academy of Sciences in 1916 to associate the broad community of science and technology with the Academy's purposes of furthering knowledge and advising the federal government. Functioning in accordance with general policies determined by the Academy, the Council has become the principal operating agency of both the National Academy of Sciences and the National Academy of Engineering in providing services to the government, the public, and the scientific and engineering communities. The Council is administered jointly by both Academies and the Institute of Medicine. Dr. Ralph J. Cicerone and Dr. C.D. (Dan) Mote, Jr., are chair and vice chair, respectively, of the National Research Council.

The **Transportation Research Board** is one of six major divisions of the National Research Council. The mission of the Transportation Research Board is to provide leadership in transportation innovation and progress through research and information exchange, conducted within a setting that is objective, interdisciplinary, and multimodal. The Board's varied activities annually engage about 7,000 engineers, scientists, and other transportation researchers and practitioners from the public and private sectors and academia, all of whom contribute their expertise in the public interest. The program is supported by state transportation departments, federal agencies including the component administrations of the U.S. Department of Transportation, and other organizations and individuals interested in the development of transportation. **www.TRB.org**

www.national-academies.org

Author Acknowledgments

This work was sponsored by the Federal Highway Administration in cooperation with the American Association of State Highway and Transportation Officials. It was conducted in the second Strategic Highway Research Program (SHRP 2), which is administered by the Transportation Research Board of the National Academies. This project was managed by Stephen J. Andrle, Deputy Director for SHRP 2.

The research reported on herein was performed by Cambridge Systematics, Inc.; supported by the Sacramento Area Council of Governments (SACOG); University of Arizona; University of Illinois, Chicago; Sonoma Technology, Inc.; and Fehr and Peers. Thomas Rossi, Cambridge Systematics, Inc., was the Principal Investigator. The other authors of this report are Yi-Chang Chiu of the University of Arizona, Scott Meeks of Cambridge Systematics, and Bruce Griesenbeck of SACOG. The authors acknowledge the contributions to this research from Brenda Bustillos, Alireza Khani, Eric Nava, and Hyunsoo Noh of the University of Arizona; Michalis Xyntarakis, Vassili Alexiadis, Jason Lemp, Rich Margiotta, Eric Ziering, Gerard Vaio, Lawrence Liao, Tazeen Mahtab, and Moshe Ben-Akiva of Cambridge Systematics; Eric Petersen and Lin Zhang, formerly of Cambridge Systematics; Binu Abraham and Gordon Garry of SACOG; Mark Hickman, currently of the University of Queensland, who performed this work while with the University of Arizona; Song Bai and Doug Eisinger of Sonoma Technology, Inc.; Jane Lin and Suriya Vallamsundar of the University of Illinois, Chicago; and David Robinson and Kwasi Donkor of Fehr and Peers.

Contents

CHAPTER 1
Introduction

1.1 Overview

The second Strategic Highway Research Program (SHRP 2) Project C10B, Partnership to Develop an Integrated, Advanced Travel Demand Model with Fine-Grained, Time-Sensitive Networks: Sacramento-Area Application, is an important step in the evolution of travel modeling from an aggregate, trip-based approach to a completely dynamic, disaggregate methodology. In this project, an existing disaggregate activity-based model was integrated with an existing traffic simulation model to create a new, completely disaggregate model. Both models were implemented using open-source software.

The product of SHRP 2 C10B is an integrated model that simulates individuals' activity patterns and travel and their vehicle and transit trips as they move on a real-time basis through the transportation system. It produces a true regional simulation of the travel within a region, for the first time using individually simulated travel patterns as input rather than aggregate trip tables to which temporal and spatial distributions have been applied to create synthetic patterns. A unique feature of this model is the simulation of transit vehicles as well as individual person tours using transit.

The development of the new integrated model has been implemented for the entire Sacramento, California, region. The integrated model components include SACSIM, the regional travel model maintained by the Sacramento Area Council of Governments (SACOG)—the regional metropolitan planning organization—and DynusT, a mesoscopic traffic simulation model developed by the University of Arizona. SACSIM includes an activity-based demand model, DaySim. The transit simulation is performed by FAST-TrIPS, also developed by the University of Arizona. The integrated model also includes the ability to run MOVES, the air quality analysis program developed by the U.S. Environmental Protection Agency (EPA).

Project C10B is documented in a summary report (Cambridge Systematics et al., 2014a). Technical documentation of DynusT as used in Project C10B is provided in a separate report (Cambridge Systematics et al., 2014b) and user information for DynusT for the C10B integrated model is documented in this network user's guide. SACOG has documented the SACSIM model (Sacramento Area Council of Governments et al., 2008), and user documentation for DaySim has been provided by Bowman and Bradley (2006). EPA has documented MOVES (U.S. Environmental Protection Agency, 2012).

This report provides information for users who wish to run and to obtain in-depth understanding of the C10B integrated model. The model files to run the base scenario created for the C10B project for the Sacramento region can be downloaded from the Federal Highway Administration (FHWA) website. **It should be noted that this does not represent an official transportation planning scenario for the Sacramento region; the files are for testing purposes only.** This report instructs users how to install these files and how to run the C10B integrated model as well as all detailed input and output files associated with DynusT and FAST-TrIPs.

After this introductory chapter, Chapter 2 discusses the setup of the model on the user's hardware. Chapter 3 describes the input data files needed to run the model, and Chapter 4 provides instructions for running the model. Chapter 5 provide detailed descriptions for DynusT and FAST-TrIPs input and output files. Chapter 6 provides references, including web addresses, for the cited documents.

1.2 Summary of Model

The components of the integrated model developed in SHRP 2 C10B include SACSIM, DynusT, FAST-TrIPS, and MOVES. These components are summarized below; please refer to Cambridge Systematics et al., 2014a, for complete documentation of the C10B integrated model.

1.2.1 SACSIM

SACSIM is a complete travel demand model that is used for planning in the Sacramento region by SACOG. The demand for personal travel within the region is modeled by DaySim, an activity-based demand model. DaySim incorporates a variety of model features, including the following:

- Modeling each person in the Sacramento region separately through the use of a population synthesizer that creates a synthetic population representing each person and household in the region;
- Modeling of the complete daily activity pattern for each individual, including the number and sequencing of activities defined by seven purposes;
- A series of logit destination, mode, and time-of-day choice models at the tour and trip levels to simulate the choices for each individual;
- Estimation of the start and end times of all activities and trips to the half-hour level of resolution; and
- Parcel-level spatial resolution for home and activity locations.

Other components of SACSIM are used to model, at an aggregate level, the remaining components of regional travel, including travel into, out of, and through the region (external travel), truck travel, and travel to and from Sacramento International Airport.

1.2.2 DynusT

DynusT is a traffic simulation model that is used in a number of areas and lends itself well to integration with both SACSIM and MOVES. DynusT is a true disaggregate simulation model that can track individual vehicles and transit travelers through the network—consistent with tracking traveler activities in a travel demand model. Furthermore, DynusT is a true dynamic traffic assignment (DTA) model that takes into account both the spatial and temporal effects of congestion. Travelers departing at different times are assigned to routes calculated based on the traveler's actual experienced travel time, which is a critical capability for establishing a consistent and reliable traffic assignment outcome.

1.2.3 FAST-TrIPS

The Flexible Assignment and Simulation Tool for Transit and Intermodal Passengers (FAST-TrIPS) is a model that assigns transit passengers within the transportation network and loads those passengers in a dynamic (time-sensitive) simulation of actual travel. This system essentially serves as a plug-in to DynusT but is precompiled with DynusT and runs with the DynusT executable.

FAST-TrIPS is a regionwide dynamic transit assignment model that determines an individual-specific transit route for each transit traveler in the system, taking into account published transit schedules and transit vehicle run times that are congestion responsive and are provided by the traffic simulation component of DynusT. FAST-TrIPS deals with both transit-only and park-and-ride trips and is able to maintain multiple constraints associated with activity time-windows and the choice of modes in multimodal travel tours.

1.2.4 MOVES

The Motor Vehicle Emission Simulator (MOVES) is the next generation mobile source emission model developed by EPA. MOVES serves as a single comprehensive system for estimating emissions from both on-road and nonroad mobile sources and has replaced MOBILE as the approved model for state implementation plans and regional or project-level transportation conformity analyses. MOVES is designed to estimate emissions at scales ranging from individual roads and intersections to large regions. A significant break in MOVES design from EMFAC (the air quality analysis program used in California) and MOBILE is that MOVES is a database-driven model: inputs, outputs, default activities, base modal emission rates, and all intermediate calculation data are stored and managed in MySQL databases. MOVES model functions query and manipulate MySQL data pursuant to scenario parameters specified in a graphical user interface (GUI). This design also provides users with flexibility in constructing and storing their own databases under the unified framework in MySQL. MOVES incorporates input data that include vehicle fleet composition, traffic activities, and meteorology parameters at the macro-, meso-, or micro-scale and conducts modal-based emissions calculations using a set of model functions. The outputs of emissions inventories or emissions factors are functions of modal-based vehicle emission rates and detailed vehicle activities specified for the desired geographic scale.

1.3 Overview of Model Documentation

Because the C10B integrated model includes several existing models, which have been completely documented elsewhere, this start-up guide does not attempt to duplicate the existing user documentation for these models. Additional references include the following:

- The software to run SACSIM is documented by Bowman and Bradley (2006). It should be noted that this report documents the original version of SACSIM, and the C10B project made some modifications to the model, as documented by Cambridge Systematics et al. (2014a). However, the modified program to run SACSIM operates very similarly to the original Daysim05 program, with the same input files and the same file formats. Key information from the SACSIM user documentation is included in this report, and any differences from the original model operation are noted in this report.
- General DynusT user documentation is available at www.dynust.net. The University of Arizona has developed detailed user documentation for the DynusT/FAST-TrIPS integrated model system that is included in the C10B integrated model; the key points from the detailed user documentation are included in this network user's guide.
- MOVES has been documented by the EPA (U.S. Environmental Protection Agency, 2012).

The executable versions of SACSIM (including DaySim) and DynusT that are part of the C10B integrated model are included in the file downloads from the FHWA website downloads. SACSIM and DynusT are available under open source licenses, and the National Academies of Science is the owner of all software produced initially for the C10B project (which is also included in the FHWA downloads). While the tests of the model performed as part of the C10B project used some input data from SACSIM that were developed using a proprietary modeling software package licensed to SACOG and Cambridge Systematics, Inc., the operation of the integrated model does not require any commercial travel demand modeling or simulation software. The files generated by the proprietary modeling software package have been converted to flat files and are included in the FHWA website downloads.
MOVES can be downloaded from the EPA website at http://www.epa.gov/otaq/models/moves/#downloading.

CHAPTER 2
Model Setup

This chapter describes setting up the model on the user's hardware system. The minimum hardware requirements are shown, and instructions are provided on the files that need to be installed and the folder structure for those files.

2.1 Hardware Requirements

This integrated model runs on computers using the standard Windows operating system. The hardware requirements are as follows:

- 16 GB RAM;
- 128 GB drive (SSD preferred); and
- Intel Core i7 processor or equivalent (4 cores/8 threads).

2.2 Model File Installation

All files needed to run the Sacramento regional model for the base test scenario can be downloaded from the FHWA website. Generally, the folder structure on the FHWA site follows the structure used in running the model although some of the files need to be unzipped before the model is run.

A main folder where the C10B integrated model will be run should be created. Two main subfolders should be created: *DaysimRegional* and *DynusTRegional*.

2.2.1 DaySim

The *DaysimRegional* folder includes the following files related to running DaySim as part of the C10B integrated model:

- The DaySim executable (*DaySim_v7_Nov2012.exe*) and control file (*Daysim.ctl*) applicable to the C10B integrated model;
- The DaySim model coefficient file, *coeffs.txt;*
- The internal-external trip table file, *ixximat.txt;*
- Various Python scripts used to convert data between DaySim and DynusT:
    - *createPathFileS.py,*
    - *createVehFileS.py,*
    - *DynusT2DaySim_skim.py,*
    - *popReqDictS.py,*
    - *PT2VT.py,and*
    - *TransitSkimsUpdate.py; and*

- Three template files, *pfiletemplate.dbf*, *sfiletemplate.dbf*, and *tfiletemplate.dbf*. These files will be populated with DaySim output for person-level, trip-level, and tour-level data respectively.

A flat file with the initial skims for the first iteration of DaySim, named *skims.dat*, is also located here. The file format is dictated by the requirements of DaySim. Text files with the original skim data for the base year SACSIM model are located in a subfolder named *InitialSkims*. The file names have the format *skXXYY.txt*, where *XX* represents the mode for the skims (*au* for auto, *tw* for transit with walk access, and *td* for transit with drive access) and *YY* represents the time period from SACSIM (*am* for a.m. peak period, *md* for midday, *pm* for p.m. peak period, *ev* for evening, and *em* for early morning). These files were created by SACOG using a proprietary modeling software package licensed to SACOG; this proprietary package is not needed to run the C10B integrated model. The script *ExpandRegionalSkims.bat* creates the initial skim inputs for DaySim at the half-hour level from the more aggregate time periods.

There are two other subfolders to the *DaysimRegional* folder. The *data* subfolder includes various data files related to the synthetic population and other socioeconomic/land use input data to DaySim, which in the initial download are located in the zipped file *data.zip*. These include

- *parc.dbf* – Parcel data;
- *pop.dbf* – Synthetic population;
- *marg_130.dbf* – Census Transportation Planning Products (CTPP) data used in developing the synthetic population;
- *pumssrt_130_mod.dbf* – Public Use Microdata Sample (PUMS) sample used in developing the synthetic population; and
- *taz.dbf* – Miscellaneous zone level data.

See the DaySim documentation (Bowman and Bradley, 2006) for more details on these files.

2.2.2 DynusT and FAST-TrIPS

The *DynusTRegional* folder includes the files related to running DynusT and FAST-TrIPS as part of the C10B integrated model. A large number of files are required to run DynusT and FAST-TrIPS. Table 2.1 shows these files by type.

In addition to the files shown in Table 2.1, the files representing "exogenous travel" must be included in the DynusT vehicle simulation. Some vehicle demand in SACSIM is not generated by DaySim, including truck, external, and airport travel. Truck travel data are located in the *demand_truck.dat* file while the airport and external travel data are included in the *demand.dat* file. These files are zipped (e.g., *demand.zip*) on the FHWA website and should be unzipped before the model is run.

Table 2.1 Files Required in the *DynusTRegional* Folder

Executable Files	
CalculateDynamicSkimsv3_64bit.exe	ft_PassSim30_32.exe
dbfpy-2.2.5.win32.exe	ft_PassSim30_64.exe
DynusTv3bx64.exe	ft_Simulation.exe
FAST_TrIPs.exe	MIVA2012.exe
ft_intermodal.exe	vcredist_x64.exe
Data Files	
AltPath_transit.dat	moves_input-RC-EXAMP.dat
AltTime_InterModal.dat	network.dat
AltTime_transit.dat	origin.dat
bg_demand_adjust.dat	output_option.dat
bitmap.dat	parameter.dat
bus.dat	pricing.dat
BusTrajectory.dat	ramp.dat
CheckTrafficFlowModel.dat	reliability_input.dat
control.dat	scenario.dat
convergence.dat	sensor.dat
demand_HOV.dat	sim_terminate.dat
demand_superzone.dat	SpaceTimePar.dat
destination.dat	StopCap2Way.dat
DivertedVeh.dat	StopCap4Way.dat
DST-FSTRPConfig.dat	subarea.dat
epoch.dat	superdemand.dat
ft_input_accessLinks.dat	superzone.dat
ft_input_centroids.dat	super_space_split.dat
ft_input_parameters.dat	super_time_split.dat
ft_input_Park-n-Rides.dat	system.dat
ft_input_routes.dat	TAZ_mapping.dat
ft_input_stops.dat	toll.dat
ft_input_stopTimes.dat	TollRevenue.dat
ft_input_system.dat	TrafficFlowModel.dat
ft_input_transfers.dat	TransitDwellTime.dat
ft_input_transitVehicles.dat	TransitDwellTime_original.dat
ft_input_trips.dat	TransitRouteSchedule.dat
GradeLengthPCE.dat	VehPos.dat
hazmat.dat	version.dat
incident.dat	vms.dat
incident_Do_not_change.dat	WorkZone.dat
leftcap.dat	xy.dat
linkname.dat	YieldCap.dat
linkxy.dat	zone.dat
movement.dat	

Table 2.1 Files Required in the *DynusTRegional* Folder (continued)

Library Files	
DLL_ramp.dll	mfc42.dll
leftcap.dat	openDTA.dll
libexpat.dll	pdbx.dll
libguide40.dll	Ramp_Meter_Feedback_CDLL.dll
libifcoremd.dll	Ramp_Meter_Feedback_FDLL.dll
libiomp5md.dll	Ramp_Meter_Fixed_CDLL.dll
libmmd.dll	Ramp_Meter_Fixed_FDLL.dll
Python Scripts	
DST_FT_module.py	ft_BST.py
FAST_TrIPs.py	ft_demandConv.py
Other Files	
node.csv	Regional Model-Transitline03262012-postETG.dws
PathInput.txt	SimPeriod.opt
pathInputNAME.txt	sktwam.txt
queue.csv	sktwmd.txt
queue_summary.csv	workingdir.ini
Regional Model-Transitline-postETG-AddTransitlines-04302012.dws	

The *VehicleDemandGen* subfolder resides in the *DynusTRegional* folder. This folder includes four Python scripts needed for the model:

- *createPathFileS.py;*
- *createVehFileS.py;*
- *DaySim2DynusTMAINprog.py; and*
- *popReqDictS2.py.*

InputFile is a subfolder of the *VehicleDemandGen* subfolder. It contains the following necessary network and parcel input files:

- *DSTtoDYSzones.txt;*
- *Node-LatLong.geo;*
- *Parcel-LatLongZone.geo;*
- *parcelMOD.txt;*
- *ParcelnewXY.py;*
- *PathInput.txt; and*
- *pathInputNAME.txt.*

2.2.3 Moves

The C10B integrated model can be run with or without MOVES. If MOVES is to be run, the user should download the MOVES installer from the EPA website: http://www.epa.gov/otaq/models/moves/index.htm. The installer can install all requirements of the MOVES software, including MySQL and Java.

MOVES and MySQL do not have to be on the same server with DaySim and DynusT. However, once DynusT is run, it will be necessary to transfer DynusT output to a directory that MOVES can access.

CHAPTER 3
Input Data

This chapter briefly describes the input data for the model. The user should refer to the user documentation for SACSIM/DaySim (Bowman and Bradley, 2006) and for DynusT/FAST-TrIPS (Chapter 5 of this guide) for detailed information about these inputs.

3.1 Inputs To Daysim

3.1.1 Zone/Parcel Information (Including Socioeconomic/Land Use Data)

The *data* subfolder of the *DaysimRegional* folder contains the input files for land use, parcel, and miscellaneous zone data. These files include

- *parc.dbf* – Parcel data. This file includes information about each land use parcel in the region, including location, activity in and around the parcel, transit access and parking information, and other information. Complete file format information can be found in Bowman and Bradley, 2006.
- *taz.dbf* – Miscellaneous zone level data. The file structure is shown in Table 3.1.

Table 3.1 Zone Data File Format

Label	Definition
TAZ	Zone number
AUTACC	Auto access time (minutes × 100)[a]
AUTEGR	Auto egress time (minutes × 100)[a]
PRKCOST	Parking cost in zone (cents/hour)[a]
DAVIS	Davis dummy (0/1)
PEDENV	Pedestrian environment score[a]
PUMA	PUMA code for zone
RAD	RAD code for zone
XCORD	X coordinate of zone centroid (state plane feet)
YCORD	Y coordinate of zone centroid (state plane feet)
PKNRCOST	Park and ride lot cost in zone (cents)
SQFT_Z	Area of zone (square feet)

[a] Not used in models.
Source: Bowman and Bradley (2006), pg 5.

3.1.2 Synthetic Population

The file *pop.dbf* contains the synthetic population data for the region. While the DaySim model includes a synthetic population generator, for the C10B project the same generated synthetic population was used for all testing; this is the information contained in *pop.dbf*. [For information about how to generate different synthetic populations in DaySim (for example, for a different forecast year or different land use scenario), the user should consult the DaySim documentation.]

The files *marg_130.dbf* and *pumssrt_130_mod.dbf* include CTPP data and PUMS sample data, respectively, used in developing the synthetic population. These should not be changed by the user.

3.1.3 Travel Time/Cost Information (Skims)

For the initial iteration of DaySim, before DynusT has been run for the first time, an initial set of travel time/cost information (skims) is needed. For the testing done for the C10B integrated model, SACOG created these initial skims from SACSIM, using the network and skimming procedures associated with its proprietary modeling software package. SACSIM uses two peak periods (a.m. and p.m.) and three off-peak periods (midday, evening, and early morning) for skimming and trip assignment. The file names for the skims files created by SACOG have the format *skXXYY.txt*, where *XX* represents the mode for the skims (*au* for auto, *tw* for transit with walk access, and *td* for transit with drive access) and *YY* represents the time period from SACSIM (*am* for a.m. peak period, *md* for midday, *pm* for p.m. peak period, *ev* for evening, and *em* for early morning).

DaySim requires that the skim inputs be text files with different formats depending on transportation mode. The formats for the skim files for the walk, auto peak, auto off-peak, and transit walk access, and transit drive access modes are shown in Tables 3.2, 3.3, 3.4, 3.5, and 3.6, respectively. For transit, the reverse directions of the a.m. peak paths are used for the p.m. peak.

Table 3.2 Walk Skim File Format

Label	Definition
ORIG	Origin zone
DEST	Destination zone
WALKDIST	Walk distance (miles × 100)

Source: Bowman and Bradley (2006), pg. 6.

Table 3.3 Peak Auto Skim File Format

Label	Definition
ORIG	Origin zone
DEST	Destination zone
D1TIME	SOV time (minutes × 100)
D1DIST	SOV distance (miles × 100)
D1EXTT	SOV congested time 1 (minutes × 100)
D1EXTT2	SOV congested time 2 (minutes × 100)
D1TOLL	SOV toll (cents)
D2TIME	HOV + time (minutes × 100)
D2DIST	HOV distance (miles × 100)
D2EXTT	HOV congested time 1 (minutes × 100)
D2EXTT2	HOV congested time 2 (minutes × 100)
D2TOLL	HOV toll (cents)

Source: Bowman and Bradley (2006), pg. 6.
Note: SOV = single-occupancy vehicle; HOV = high-occupancy vehicle.

Table 3.4 Off-Peak Auto Skim File Format

Label	Definition
ORIG	Origin zone
DEST	Destination zone
D1TIME	SOV time (minutes × 100)
D1DIST	SOV distance (miles × 100)
D1EXTT	SOV congested time 1 (minutes × 100)
D1EXTT2	SOV congested time 2 (minutes × 100)
D1TOLL	SOV toll (cents)

Source: Bowman and Bradley (2006), pg. 6.

Table 3.5 Transit Walk Access Skim File Format

Label	Definition
ORIG	Origin zone
DEST	Destination zone
XFNUMW	Number of transfers
XFTIMW	Transfer time (minutes × 100)
FWTIMW	First wait time (minutes × 100)
FAREW	Fare (cents)
TRDISW	In-vehicle distance (miles ×100)[a]
WATIMW	Walk time (minutes × 100)[a]
TRTIMW	In-vehicle time (minutes × 100)

[a] Not used in models.
Source: Bowman and Bradley (2006), pg. 6.

Table 3.6 Transit Drive Access Skim File Format

Label	Definition
ORIG	Origin zone (Drive end)
DEST	Destination zone (Walk end)
PKTAZD	Park and ride lot zone number
XFTIMD	Transfer time (minutes × 100)
FWTIMD	First wait time (minutes × 100)
DRTIMD	Drive access time (minutes × 100)
FARED	Fare (cents)
DRDISD	Drive access distance (miles × 100)
TRDISD	In-vehicle distance (miles × 100) [a]
WATIMD	Walk egress time (minutes × 100) [a]
XFNUMD	Number of transfers
TRTIMD	In-vehicle time (minutes × 100)

[a] Not used in models.
Source: Bowman and Bradley (2006), pg. 6.

The initial skims for the first iteration of DaySim are contained in a combined file named *skims.dat*. DaySim uses skim inputs at the half-hour level of temporal resolution. The file *skims.dat* is created from the *skXXYY.txt* files by the program *ExpandRegionalSkims.bat*.

3.2 Exogenous Travel

Exogenous travel comprises external, truck, and airport/special generator travel. Estimates of exogenous travel can come from any source and are provided as inputs in zonal origin–destination (O-D) format. In the tests performed as part of the C10B project, exogenous travel trip tables were developed using SACSIM model runs, which used a proprietary software package licensed to SACOG. These files for exogenous demand were previously created by SACOG to reflect the base scenario and are not changed by the C10B integrated model. These versions of the files are included in the downloads from the FHWA website.

The files that include the exogenous demand are *demand.dat* for the airport and external travel data and *demand_truck.dat* for truck travel data. [There is also an input file named *demand_HOV.dat* for high-occupancy vehicle (HOV) travel, but since there are no HOV facilities in the base Sacramento network, this file is empty in the downloads.]

The *demand.dat* and *demand_truck.dat* files have the same format. The data represent time-dependent origin–destination matrices, with each matrix containing number of trips to be generated for a specified time period. The format is described in Table 3.7, with an example shown in Figure 3.1. These files are free form with spaces between the data items in each line.

Table 3.7 File Format of *demand.dat* and *demand_truck.dat*

Parameter				
Number of O-D matrices listed in this file		Multiplier for uniform increase/decrease		
Start Time of Each O-D Matrix				
Timestamp for (first) demand period	Timestamp for demand period	Timestamp for demand period	…….	Timestamp for (final) demand period
O-D trips				
Start time = *xx*				
Demand for period	Demand for period	Demand for period	Demand for period	Demand for period

```
                                      24 1.000
   0.0   60.0  120.0  180.0  240.0  300.0  360.0  420.0  480.0  540.0  600.0  660.0  720.0
 780.0  840.0  900.0  960.0 1020.0 1080.0 1140.0 1200.0 1260.0 1320.0 1380.0 1440.0
                               Start Time =    0.0
              1.7086    9.7574    1.5430    0.7619    0.9307    0.7198
              0.6268    0.0513    1.0952    0.3282    0.2562    0.7801
              0.3030    0.0000    0.0000    0.0531    0.7311    1.3299
              0.6478    0.0000    0.6603    0.2535    0.5571    0.2960
              0.5207    0.2428    1.0720    0.5804    0.0424    0.2338
              0.1187    0.2167    0.2844    0.1305    0.4086    0.5285
              0.1864    0.0676    0.3907    0.0461    0.1670    0.0586
              0.0822    0.0939    0.0693    0.1591    0.1285    0.1256
              0.2113    0.4016    0.4969    0.3038    0.4302    0.3100
              0.4551    0.1548    0.3968    0.1090    0.1274    0.1169
              0.1318    0.1482    0.0081    0.2376    0.2025    0.3813
              0.4495    0.8524    0.4559    0.1384    0.0604    0.9844
              0.2398    0.1311    0.1809    0.1502    0.0585    0.4383



                               Start Time =   60.0
              1.4646    8.3635    1.3225    0.6530    0.7977    0.6170
              0.5372    0.0440    0.9387    0.2813    0.2196    0.6686
              0.2597    0.0000    0.0000    0.0455    0.6267    1.1399
              0.5552    0.0000    0.5660    0.2173    0.4775    0.2537
              0.4463    0.2081    0.9189    0.4975    0.0363    0.2004
              0.1017    0.1857    0.2438    0.1118    0.3502    0.4530
              0.1598    0.0579    0.3349    0.0395    0.1432    0.0502
              0.0704    0.0805    0.0594    0.1364    0.1102    0.1076
              0.1811    0.3442    0.4259    0.2604    0.3688    0.2657
              0.3901    0.1327    0.3401    0.0934    0.1092    0.1002
             0.1130    0.1271    0.0069    0.2036    0.1736    0.3268
```

Figure 3.1 Example of *demand.dat* file.

If a user wished to create a new scenario and have exogenous travel demand estimates that are consistent with the new scenario, the parts of SACSIM related to truck, airport, and external travel would have to be run with the appropriate scenario assumptions. SACSIM outside of DaySim requires the use of the proprietary software package, which is not part of the C10B integrated model. Other means of creating appropriate exogenous travel demand estimates consistent with the new scenario could be used, as long as the information is stored in the files *demand.dat* and *demand_truck.dat* in the correct formats.

3.3 DynusT Highway Network

For the tests done as part of Project C10B, a DynusT network was created to represent the base scenario. The files associated with this network are included as part of the FHWA website downloads. This section briefly describes some of these files.

The DynusT network input file is named *network.dat*. This file describes the roadway network configurations including node IDs and link characteristics. The file format and fields are described in Section 5.2 and in Tables 5.18 and 5.19.

The file *movement.dat* describes the available movements of every link at the connecting downstream node. The file format and fields are described in Section 5.2 and in Tables 5.20 and 5.21.

The file *origin.dat* describes the "generation links" for each zone. A generation link is the location in which a vehicle will enter the network and begin travel toward its destination. Every zone (parcel) must have at least one generation link. A link can be a generation link for more than one zone when that link borders or intersects with more than one zone. The file format and fields are described in Section 5.2 and in Tables 5.22 and 5.23.

The file *destination.dat* describes the destination nodes for each zone. A destination node is the location in which a vehicle will exit the network at the completion of its trip. Every zone (parcel) must have at least one destination node. A node can be a destination node for more than one zone when that node borders or intersects with more than one zone. The file format and fields are described in Section 5.2 and in Tables 5.24 and 5.25.

The remainder of Chapter 5 describes a variety of other files with settings and parameters required by DynusT. These files are all included in the FHWA website downloads.

3.4 FAST-TrIPS Transit Network

For the tests done as part of Project C10B, a FAST-TrIPS transit network was created based on data from the General Transit Feed Specification (GTFS) to represent the transit system in the base scenario. The input files for FAST-TrIPS are described in Chapter 5. This section provides a brief description of the network files. The details for each of the following files can be found in Subsection 5.12:

The file *ft_input_stops.dat* contains the information on the name, location, type, capacity, and other properties of transit stops. No information is stored in this file about the transit service at each stop.

The file *ft_input_routes.dat* contains the information about transit routes. In FAST-TrIPS, a route is the group of trips displayed to riders as a single service. This file includes the general information about routes, such as the name and type of service, and does not provide any information about the service itself.

The file *ft_input_trips.dat* contains the list of trips that are served by each route and the associated start time of each trip at a terminal. A trip is a sequence of two or more stops that occurs at specific times. More information is also provided in this file about the type of service and the capacity of vehicles.

The file *ft_input_stopstimes.dat* is the main file describing transit service and is in the form of a schedule. It contains the times that each transit vehicle arrives at and departs from individual stops for each trip.

The walk access/egress links are stored in the file *ft_input_accesslinks.dat*, which includes the estimate of walking distance and time at each zone to the stops within that zone. The estimate is based on the zone area and the density of stops. The generated links can be used both for access and egress.

Similar to the access links, transfer walking links are stored in the file *ft_input_transfers.dat*, which includes the distances and times between pairs of stops at which a transfer is possible. The generated transfer links can be used in both directions even though they are stored as one-way links.

For the purpose of transit passenger simulation and to generate transit vehicle objects, the combination of information about transit service is used to generate the file *ft_input_transitVehicles.dat*, in which each line represents a transit vehicle. The file includes some of the details about each vehicle's movements in the network.

The file *ft_input_Park-n-Rides.dat* shows the information of park-and-ride facilities in the network, where travelers can drive, park their cars, and transfer to transit systems. The park-and-ride facilities must be connected to some nodes in the auto network and some stops in the transit network using walking links.

There are many other files associated with running FAST-TrIPS that are included in the downloads from the FHWA website. These include files with parameters and settings as well as files related to the transfer of demand and simulation information between FAST-TrIPS and both DaySim and DynusT. These files are described in the remainder of Section 5.1.

CHAPTER 4
Running the Integrated Model

4.1 Overview

This chapter explains how to set up DaySim and DynusT and to run the C10B integrated model, including MOVES if desired. This includes feeding the DaySim output into DynusT and generating updated skim files to feed back into DaySim. These directions assume some familiarity with DaySim (see Bowman and Bradley, 2006) but not as much with DynusT; however, they are still fairly explicit about the DaySim steps, in particular as they relate to the modifications in the skim files and processing the output for DynusT. The objective is that the user will be able to fairly easily set up new scenarios by copying files, modifying configuration parameters, and then running scripts and executables.

Section 4.2 describes setting up the initial directories for DaySim and DynusT. Then to run the programs, the user will first need to make copies of certain files from certain directories, replacing any input data files as needed such as the DynusT network file. The user will then need to set various values in various configuration files appropriately.

Sections 4.3 and 4.4 describe how to run the integrated model. First, DaySim will be run, possibly with additional command line arguments. Once DaySim completes its run, a Python script will be run on the output and then the results copied into a particular folder in DynusT. At that point the user can run the master DynusT Python script, which takes care of preparing inputs for DynusT, running DynusT, and then running the components of the FAST-TrIPS transit simulation. Once DynusT completes its run, if another loop back to DaySim is to be run, an auxiliary program will be run that generates a raw skim file from the DynusT outputs. (DynusT's built-in process for producing the skim file is not used.) The DynusT skim file is then moved to the DaySim folder, and a script that converts the DynusT skim file into the corresponding DaySim skim files is run. At that point the user can start the process of the next loop by running DaySim again with the new skim files.

If it is desired to run MOVES as part of the integrated model, the process for doing so is described in Section 4.5.

4.2 Setup

To simplify understanding and keep track of the various files, DaySim and DynusT files are kept in separate folders. These folders contain various subfolders. To prevent output files from overwriting needed input files or previous outputs, it will be necessary at various times to move or copy certain files. There are several Python scripts used to run various components or convert data, and so it is necessary to have Python installed as well as the dbf Python library.

4.3 DaySim

The version of DaySim05 being used in the C10B integrated model has a few minor modifications compared with the original DaySim05 implementation used by SACOG:

- DaySim generates and uses variable value of time information.
- The a.m. and p.m. peak period skims (4- and 3-hour aggregations respectively) have been broken out into individual half-hour aggregations, and the midday 6-hour aggregation has been broken down into individual 1-hour aggregations.

To install the DaySim regional model, the following steps should be performed:

1. Copy the *DaysimRegional* folder from the FHWA site to the destination machine.
2. Extract the zipped files from *data.zip* and *daysim.zip*.
3. Run *ExpandReginalSkims.bat*.

In addition to the standard DaySim files, there will be an additional folder (*InitialSkims*), which contains the versions of the skim files that should be used whenever DaySim is run for the first time when starting a new scenario. An additional script, *PT2VT.py*, pre-processes the DaySim output for DynusT, and an additional executable, *CalculateDynamicSkimsv3_64bit.exe*, converts the DynusT skim file into the DaySim skims. The main DaySim control file is *daysim.ctl*, though all the parameters in that file can be overridden on the command line. More details on the DaySim files can be found in Bowman and Bradley (2006).

The installation can be tested by running *DaySimCode_v7_Nov2012.exe*. This will perform a 1% sample run of DaySim using the default parameters. DaySim will display output on the console as well as generating a *daysim.txt* output file. In addition, if the simulation completes successfully, the files *pout.dbf*, *tout.dbf* and *sout.dbf* will be generated. If the *PT2VT.py* script is run, providing a random seed as the argument (e.g., *PT2VT.py 123*), this should produce the file *veh_sout.dbf*. This file is the primary input from DaySim to DynusT.

4.4 DynusT

DynusT has a very large number of both input and output files, as well as various executables and software libraries (see Section 2.2 and Table 2.1 in this report). The input files primarily include parameter files to control individual scenario runs and highway and transit network configuration files. The output files include both logging files and data files. There are a number of files that are necessary for DynusT but not directly relevant to the C10B integrated model, and so those files will be sparsely documented. In addition, DynusT uses a number of Python scripts during its execution.

To install the DynusT regional model, the following steps should be followed:

1. Copy the *DynusTRegional* folder to the destination machine.
2. Extract *demand.zip* and *demand_truck.zip*.
   Note: These files for exogenous demand were previously created by SACOG to reflect the base scenario and are not changed by the C10B integrated model. If a user wished to create a new scenario and have exogenous travel demand estimates that are consistent with the new scenario, the parts of SACSIM related to truck, airport, and external travel would have to be run with the appropriate scenario assumptions. SACSIM outside of DaySim requires the use of a proprietary software package that is not part of the C10B integrated model. Other means of creating appropriate exogenous travel demand estimates consistent with the new scenario could be used, as long as the information is stored in the files *demand.dat* and *demand_truck.dat* in the correct formats (see Section 3.2).
3. If this is the initial install of DynusT on this machine, the user may need to run *vcredist_x64.exe* to install some Visual C++ redistributables that DynusT depends on.
4. Copy the *veh_sout.dbf* file produced from running DaySim and then the *PT2VT.py* script into the *VehicleDemandGen\InputFile* folder.

The user should now be able to run the main DynusT script *DST_FT_module.py*. With the default settings, this script will print out information to the console, run some programs to set up for the transit simulation, and then loop between DynusT and the transit simulation a couple of times (DynusT itself will run two iterations internally).

There are a number of files that must be in place before running the master script. There are a few of these files that are updated when DynusT runs, and so if DynusT quits unexpectedly, these files could be left in an inconsistent state. Because of that and also because of the many output files that DynusT produces, it is recommended to create a copy of the files in a source directory and then copy those files into a new directory for doing the actual run.

Four main configuration files must be modified for different scenarios:

- *DST-FSTRPConfig.dat* —This is a simple file consisting of two numbers on two lines. The first number indicates the number of DynusT to FAST-TrIPS to DynusT iterations that should take place. The second number indicates whether or not FAST-TrIPS should run.
- *parameter.dat*—This file contains many parameters that control the running of DynusT. The four that the user needs to be concerned about are
  - Line 37—Number of Threads. This should generally match the number of CPUs on the server machine.
  - Line 40—Read Transit Flag. Set this to 0 only if running a nontransit scenario.
  - Line 41—Generate Transit from *TransitRouteSchedule.dat*. Set this to 0 only if running a nontransit scenario.

- Line 54—Skim output switch. Since the *skims.dat* file from DynusT is not used to generate skims for feedback to DaySim, set this to 0.

- *scenario.dat*—This file controls a number of factors related to the way DynusT runs a scenario, but the only part that is relevant is Line 7, which controls the distributions of drivers among the various responsiveness classes. This turns out not to be a major factor in the sensitivity testing or validation, and so this file likely would not be modified.
- *system.dat*—This file controls three important scenario parameters.
  - The first number of the second line controls the maximum number of iterations within DynusT that will occur if DynusT does not converge first.
  - The second number of the second line must be set depending on whether or not exogenous trip demand should be included in the simulation. A “0” indicates exclude exogenous demand, and a “3” indicates include exogenous demand.
  - The fourth number on the third line sets the factor (the relative gap) used to determine whether or not the simulation has converged. The value in the file is an integer >0; multiply the value in the file by 0.0001 to get the actual relative gap threshold used by DynusT.

There are several script, executable, and library files that are needed:

- *DST_FT_module.py*—Master python control file for running the combined DynusT and transit simulation. Controls all the other scripts and executables that have to be run for a full scenario. If everything is configured correctly and no errors occur, running this script is the only manual action needed until the DynusT simulation completes.
- *ft_BST.py*—Script for converting *veh_sout.dbf* into *veh_sout.dat*, then running *ft_intermodal.exe*.
- *DynusTv3bx64.exe*—The main DynusT executable.
- *FAST_TrIPs.exe*—The main transit simulation executable.
- *FAST_TrIPs.py*—Primary control script for the transit simulation executables.
- *ft_Assignment.exe*—Simulation transit network assignment.
- *ft_demandConv.py*—Converts *veh_sout.dbf* to *sout.txt*.
- *ft_intermodal.exe*—Executable that creates intermodal trip pieces.
- *ft_PassSim30_64.exe*—64-bit version of the passenger simulator.
- *ft_Simulation.exe*—Simulation of transit vehicle movements.
- *DLL_ramp.dll*, *libexpat.dll*, *libguide40.dll*, *libifcoremd.dll*, *libiomp5md.dll*, *libmmd.dll, Ramp_Meter_Feedback_FDLL.dll*, *Ramp_Meter_Fixed_CDLL.dll, Ramp_Meter_Feedback_CDLL.dll*, *Ramp_Meter_Fixed_FDLL.dll,* and *mfc42.dll – .dll* files required for running the DynusT executable.

4.5 Running DaySim

Once DaySim has been configured, it is run by executing the file *DaySimCode_v7_Nov2012.exe*. This run should produce the *pout.dbf*, *tout.dbf*, and *sout.dbf* output files. Next, the *PT2VT.py* script is run, providing a random seed as the argument (e.g., *PT2VT.py 123*). This should produce the file *veh_sout.dbf*. This file should be copied to the *VehicleDemandGen\InputFile* folder.

4.6 Running DynusT and FAST-TrIPS

Copy the *veh_sout.dbf* file from DaySim into the *VehicleDemandGen\InputFile* folder. Make any desired changes to the configuration files. Run *DST_FT_module.py*.

The following output files are created:

- *Convergence.dat*—Output file of iteration convergence statistics.
- *OutMUC.dat*—Output file of iteration vehicle status and summary statistics.
- *ft_convergence.dat*—Transit iteration convergence stats.
- *SummaryStat.dat*—Global summary statistics.
- *VehTrajectory.dat*—Postprocessed to generate data for display in the C10B application and to generate feedback skim files.

The following programs may be used to postprocess data from the application:

- *C10postProcessing.py*—Processes *VehTrajectory.dat*, produces *vehTripDataOut.dat*, processes *SummaryStat.dat*, and produces *networkStatDataOut.dat*.
- *routeBasedTravelTime_FullRoute.py*—Processes *AltPath.dat* and *AltTime.dat* and produces *routeBasedTravelTime_FullRoute_output.csv*.
- *CalculateDynamicSkimsv3_64bit.exe*—Processes *VehTrajectory.dat* and produces skim files for feeding into DaySim.

DynusT produces many gigabytes of output files. It is recommended to delete or compress and archive these files, particularly if running on an SSD with limited space.

To feed the updated skims back to DaySim, the program *CalculateDynamicSkimsv3_64bit.exe* is run after running DynusT, and the resulting *skims.dat* files must be copied to the DaySim input directory. Next, the *DynusT2DaySim_skim.py* script is run to convert the single *skims.dat file* into the multiple *sk*.txt* files that DaySim uses.

4.7 MOVES

As previously noted, MOVES and MySQL do not have to be on the same server with DaySim and DynusT. However, once DynusT is run, it will be necessary to transfer DynusT output to a directory that MOVES can access.

If MOVES is to be run, there are specific settings that are needed in DynusT. In the file *parameter.dat*, set the value of “Switch for MOVES outputs” to 1. This will cause DynusT to generate three MOVES output files for each hour of simulation:

- *MovesCO2_Links_Hour_<hour>.csv*
- *MovesCO2_LinkSourceTypes_Hour_<hour>.csv*
- *MovesCO2_opmodedistribution_Hour_<hour>.csv*

The specific network links for which DynusT provides outputs for MOVES are configured in the file *moves_input.dat*. Other settings include

- *MovesMode*—While MOVES can model other pollutants, for now DynusT only generates input for CO_2 modeling.
- *DayID*—This should be set to weekday since DaySim models a “typical” weekday.
- *HourID*—This will usually have a value of 0 for modeling a full day, but a specific start time could be chosen for modeling a shorter period.

The next section in the file specifies the links to be modeled. The number of links appears on a line by itself, then each subsequent line specifies the following:

“from node” _> “to node” _> MOVES link type _> County ID _> Zone ID

MOVES works on an hour-by-hour basis. It is necessary to create separate configurations for every hour of simulation to be run and then create a batch file to run all the configurations.

- *Importer.xml*—This file is a template for the files that describe how the data for 1 hour is to be imported from the DynusT output files. This template file includes the keyword “FILLIN” at every point that needs to be filled in for a particular scenario. The sections that need to be updated are
    - *timespan*
    - *databaseselection*
    - *link*
    - *linksourcetypehour*
    - *linkopmodedistribution*
    - *agedistribution*
    - *zonemonthhour*
- *RunSpec.mrs*—This file is a template for the configuration file for running the MOVES simulation for a particular hour. This template file includes the keyword “FILLIN” at every point that needs to be filled in for a particular scenario. The sections that need to be updated are

    - *timespan*
    - *outputdatabase*
    - *scaleinputdatabase*
- *RunSpecList.txt*—This file lists the *.mrs* files for each hour that the MOVES simulation should be run. Each *.mrs* file is listed on a separate line. This file is an example file for a full 24-hour scenario.
- *Moves.bat*—This is a Windows batch file that runs all the pieces for a complete scenario. This file is an example file for a full 24-hour scenario. Once all the configuration files have been updated as needed, execute this batch file in order to run the complete MOVES scenario.
    - The first line calls the *setenv.bat* file from the MOVES installation, which ensures that all the environment variables are set up correctly.
    - The following lines except for the last line invoke the MOVES Java command line processor with each of the configured *Importer*.xml* files in order to import the DynusT output data into the MOVES database.

CHAPTER 5
Programmer’s Guide Overview

This chapter provides information about the models’ structure and the main files and structure of the codes in DynusT and FAST-TrIPS. It has been provided mainly for programmers and model developers, so it gives information on how the models have been designed and coded. To better understand the contents of this report, the reader is strongly advised to read about the theoretical aspects of the integrated model and its framework. This report starts with FAST-TrIPs, which contains assignment of transit passengers, assignment of intermodal passengers, and the simulation of both types of travelers. After that, DynusT, which models dynamic traffic assignment and simulation, as well as simulation of transit vehicles in the traffic network, is discussed.

The DynusT/Fast-TrIPS integrated system is designed to be a “loosely coupled” system. This design has allowed the DynusT and Fast-TrIPS teams to develop components in parallel with separate software development cycles and progress. This design does not provide for the most computationally efficient system, but it permitted the robust system development for a complex system design given a rather tight development schedule.

As shown in Figure 5.1, DynusT and FAST-TrIPS communicate via flat text files. The inputs to the entire system include the DaySim output files containing trip and tour rosters. Such files are needed for both DynusT and FAST-TrIPS. Other important input files into FAST-TrIPS and DynusT are those containing the transit route and demand information. The transit route file information is converted from the Google General Transit Feed Specification (GTFS) file.

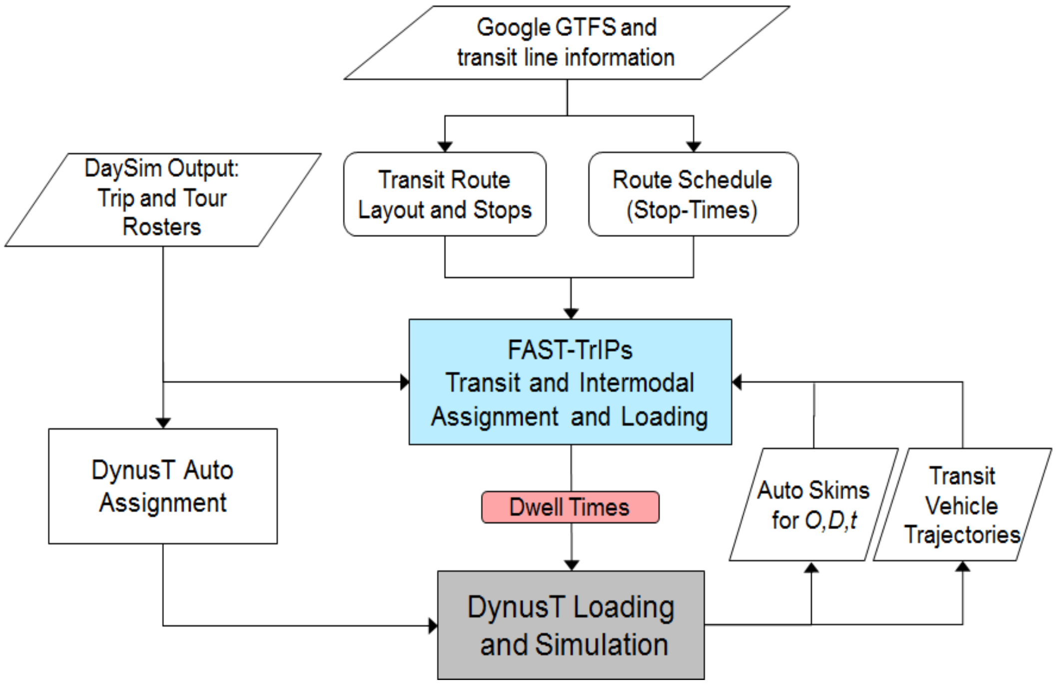

Figure 5.1 DynusT/FAST-TrIPS integration framework (*O* = origin, *D* = destination, *t* = travel time).

The overall FAST-TrIPS model structure is discussed in detail in Volume 2 of *Dynamic, Integrated Model System: Sacramento-Area Application* (Cambridge Systematics et al., 2014b). Figure 5.2 shows the detailed integration framework, including all the files needed for DynusT/FAST-TrIPS communication.

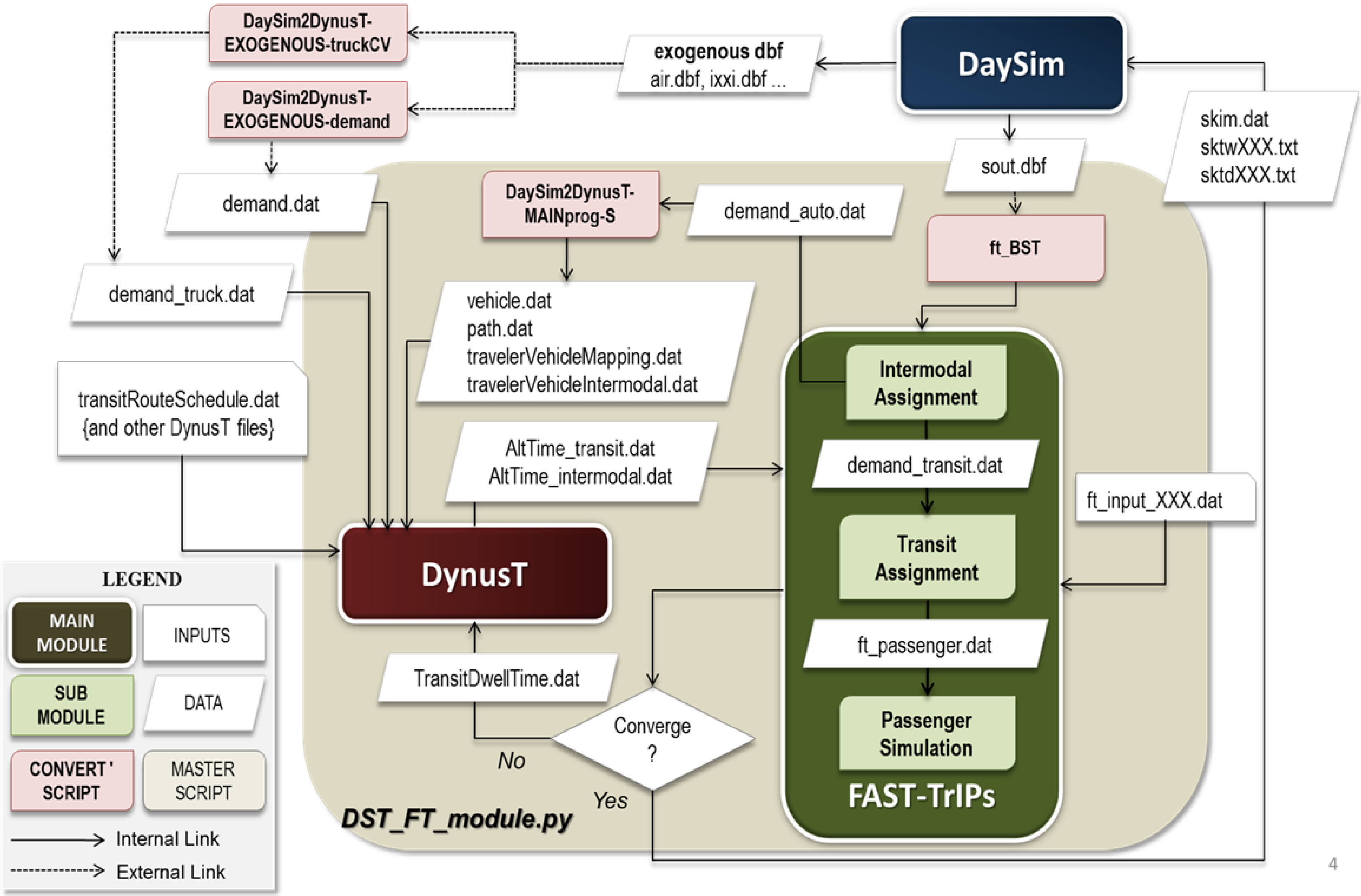

Figure 5.2 Interfacing files in DynusT/FAST-TrIPS integrated system.

5.1 FAST-TrIPS Software

5.1.1 Model Structure

The FAST-TrIPS model structure is documented in Chapter 4 of Volume 2 of the *Dynamic, Integrated Model System: Sacramento-Area Application* (Cambridge Systematics et al., 2014b) report. Readers are encouraged to refer to that chapter for details.

- **Transit Stop Class**
  This class is designed to represent transit stop objects, both for finding a transit path and for simulating transit passengers. In addition to the basic characteristics of a stop, several attributes and variables are included in the transit stop class. In particular, for transit assignment and also intermodal assignment, each stop has a set of labels for the vehicle arrival time, departure time, cost, predecessors along routes, and so on. For the purpose of simulation, a container is also designed for each stop with a given capacity from the input file (the capacity may be infinite). This container is a representation of a passenger queue for transit vehicles at the stop. When a transit vehicle arrives in the simulation, passengers are removed from the container and are loaded into the transit vehicle if the vehicle's route is among their attractive transit route set. This boarding procedure has the first in, first out (FIFO) property, and passengers cannot board if the vehicle is full.
- **Transit Vehicle Class**
  This class is defined to model the transit vehicles and their movements. In this class, each object represents a GTFS trip (vehicle movement from one terminal to another), and the schedule data for that trip are included. In other words, for each transit vehicle departure from a terminal, an object is generated which has many attributes, including the route number, the sequence of stops to be visited, the scheduled time at each stop, and so forth. Transit vehicle objects are simulated by moving them between stops based on the scheduled times. In the integration with DynusT, the schedule data are replaced by the vehicle trajectories from the DynusT simulation. The primary function of the transit vehicles is to carry transit passengers from their boarding stops to their alighting stops. The passengers are removed from the container at the boarding stop and are added to a container representing their being on board the vehicle. Obviously, this container has a limit in size equal to capacity of the vehicle. At the time of passenger alighting, the passengers are removed from the transit vehicle container and are added to the transit stop container to continue their trips toward the destinations. Then a spot is opened for the next passenger to get on board. A transit vehicle object has a set of vectors to store the results of the simulation, including the number of boarding and alighting passengers at each stop and the value of dwell time at each stop, which is calculated based on the number of passengers boarding and alighting and is also based on the vehicle type. These vehicle results are provided in the output, and the dwell times are used as feedback into the DynusT vehicle simulation.

- **Passenger Class**

  Finally, each passenger in the transit network is an object based on the class of passengers. Originally, a passenger has the basic attributes provided from the demand model, most importantly the ID, origin, destination, mode, preferred departure time (PDT) and preferred arrival time (PAT) The passenger is assigned a transit path in the assignment model, including a set of vectors for the boarding stop(s), the alighting stop(s), the vehicle trip(s), and walking (access and egress) times. During the simulation, passenger objects are generated at their start time and are moved to the first boarding stop. There, they wait for the transit vehicle and get on as soon as a vehicle from their attractive route set arrives. As mentioned before, passengers are loaded into the stops and are kept in the queue, and they are moved onto the vehicle at the time of vehicle arrival at the stop. A similar methodology is applied for the passengers at the designated alighting stop. When a passenger object gets off and walks to the next point (either to the next boarding stop for a transfer or to the destination), that passenger object is added to the global set of passengers. Passengers also have a variable for their status, which can be walking, waiting, on board, or arrived, which is updated during the simulation. Each time the passenger status changes, this time is added to a set of vectors in the passenger object and is recorded as the experienced path (or the trajectory) of the passenger.
- **Intermodal Assignment Function**

  The first part of FAST-TrIPs to be run in the integrated model is the intermodal assignment function. The reason this is first is because the intermodal demand has to be assigned at the level of the tour to ensure the traveler returns to the same location where he or she changed modes (parked his or her car). Once the intermodal tour is assigned, the auto part and the transit part of the tour, both connected to the optimal park-and-ride location, can be modeled simultaneously with the auto-only and transit-only trips. For this reason, the intermodal assignment has to be run separately, even before starting the auto or transit assignment, and the intermodal demand (trip) file should be prepared in advance.

In this context, the intermodal assignment function takes the traveler’s tour information and selects the optimal park-and-ride location by calculating the total tour cost. To do so, for each passenger tour, first the anchor activity in the list of daily activities is found; other activities are assumed to be performed as the discretionary or minor activities within the primary tour. Then the PAT to, and the PDT from, the anchor activity are set as the main time constraints in the algorithm. In addition, a time window is used for the trip toward the anchor activity based on the PDT from the previous activity. Then the algorithm is run to find the total tour cost:

1. A forward auto shortest path tree is found from the origin using the PDT from home or from the activity before the anchor activity, and a label is set for each park-and-ride lot showing the auto travel cost (Label 1).
2. A backward transit shortest path is found from the destination using the PAT to the anchor activity, and a label is set for each park-and-ride showing the transit travel cost (Label 2).
3. The sum of Labels 1 and 2 will be the travel cost from the origin to the destination through each park-and-ride if the intermodal trip is feasible according to the time windows for the PDT and the PAT.
4. A forward transit shortest path is found from the destination (the anchor activity) using the PDT from the anchor activity and a label is set for each park-and-ride showing the transit travel cost (Label 3).
5. A backward auto shortest path is found to the origin or the next activity after the anchor activity using its PAT, and a label is set for each park-and-ride lot showing the auto travel cost (Label 4).
6. The sum of Labels 3 and 4 is the travel cost from the destination to the origin through each park-and-ride if the travel is feasible according to the time windows for the PDT and the PAT.
7. The sum of the travel costs calculated in Steps 3 and 6 is the total tour cost through each park-and-ride. So, the optimal park-and-ride is found and assigned to the tour.
8. Finally, the intermodal tour result is divided into separate legs for the auto and transit assignment, and the park-and-ride is considered as a dummy destination for the intermodal passengers. Therefore, each intermodal trip from a given origin to a given destination is broken into two trips: an auto trip from the origin to the park-and-ride and a transit trip from the park-and-ride to the destination. A similar decomposition happens for the second half of the tour.

- **Transit Assignment Function**
  The transit assignment function finds the optimal path for each passenger connecting the origin to the destination at the given time. The given time is important as the service schedule changes over time. Thus, the optimal path might use different transit routes at different times of the day. The path assignment is done in a two-step procedure which is described in the following.
- **Trip-Based Shortest Path (one-to-all)**
  The assignment takes advantage of a transit shortest path algorithm called trip-based shortest path (TBSP) which works for schedule-based transit systems. The reason it is called trip based is that it uses transit vehicle trips as opposed to links as in traditional shortest path algorithms. This approach makes the algorithm computationally more efficient. The TBSP has two versions in the code:

- ***Backward TBSP***: This version of the algorithm gets the destination traffic analysis zone (TAZ) and the PAT as the inputs and finds the shortest path tree from all other TAZs to the destination TAZ, so that the arrival time is in a window within 30 minutes before the PAT.
- ***Forward TBSP***: This version of the algorithm gets the origin TAZ and the PDT as the inputs and finds the shortest path tree from the origin TAZ to all other TAZs, so that the departure time is in a window within 30 minutes after the PDT.

  The backward TBSP is used in the case that the passenger is traveling in the first half of the tour (i.e., to the anchor activity), and the forward TBSP is used in the second half of the tour (i.e., from the anchor activity). This means that passengers leaving home to an anchor activity want to be at the destination on or before the activity start time, and passengers leaving this activity toward home or another activity can leave the origin on or after finishing the anchor activity.

  The algorithms can also be run using a generalized cost function that applies higher weight to the walking and waiting times and a penalty cost for transfers. While running the algorithms, the travel cost to/from the stop, the predecessor/successor, and either the arrival or the departure time is stored for each stop, and also for each TAZ centroid. A priority queue is also used for the scan-eligible (SE) list of stops. The SE is based on a heap structure, and the first element in the queue always has the minimum label (travel cost). Therefore, the algorithms run in a label-setting format.

o **Path Extracting Function**

After finding the shortest path tree for each origin-time or destination-time (for which there is demand), the elementary path has to be found and assigned to the passengers based on the other end of the trip. For example, after finding the shortest path tree to a destination, the path from the passenger's origin should be extracted from the tree. To do this, corresponding functions are called to get either the forward paths or the backward paths and to assign them to the passenger objects.

The output of both functions (getForwardPath and getBackwardPath) is a string containing the start time from the origin, the boarding stop(s), the alighting stop(s), the transit vehicle trip(s), and the walking times (access, transfers, and egress). Using this set of information, passengers can be simulated in the transit simulation model.

- **Transit Simulation Function**

After providing path assignments to the passengers, the whole transit system is simulated to model the interaction between the network (transit vehicles) and the demand (transit passengers). During this procedure, first, all of the objects are created, including transit stops, transit vehicles, and passengers, and these objects are initiated for the simulation model. Then, the trajectories of the transit vehicles from the DynusT simulation are read (the vehicle's arrival at each stop), processed, and stored in a list, sorted in increasing order of time. This list is called eventList. An event is the combination of transit vehicle, transit stop, and the arrival time of the vehicle to the stop. In the main simulation function, each transit event is processed in that the boarding and alighting of passengers are determined based on several factors. In addition, during the time between events, the walking passengers are scanned and loaded to the next location in their paths.

- **Transferring Passengers**
  After a passenger object is generated at the start time of its trip, the pointer to the object is added to the list of walking passengers called pass2transfer. The same thing happens when a passenger gets off a vehicle and walks toward the next boarding stop or to the destination. Therefore, at a single time, pass2transfer contains all the passengers who are walking in the network. These passengers have to be scanned periodically to check for their arrivals to the next locations in their paths. This scanning procedure is performed every second, and the passengers who have arrived to the next boarding stops are loaded into those stops, and they are removed from the list of walking passengers. Those passengers arriving to their destinations after the last walking segment are removed from the simulation.
- **Boarding and Alighting Passengers**
  The most important part of the simulation model is where a transit vehicle arrives at a stop and passengers try to board the vehicle. The model here has to ensure two properties as below:
  - Capacity Constraint—This is a critical aspect of transit systems and the source of complexity in the analytical transit assignment models. However, with the simulation model, this property is ensured easily, as transit vehicles can load only a limited number of passengers. In fact, this approach models the hard capacity constraint, but even a soft capacity constraint can be modeled with minor modification in the simulation code.
  - Passenger Priority—The priority in transit systems applies to (1) onboard passengers versus waiting passengers, and (2) FIFO behavior within the queue of waiting passengers. Fortunately, the simulation model can capture both of these behaviors without any problem, since the onboard passengers remain on board until their arrival, and the remaining capacity in the vehicle is provided to the waiting passengers to board in a FIFO discipline.

    So, briefly, when a transit event happens, all the onboard passengers are checked if they have arrived at their alighting stop, and they get off if needed. Then the waiting passengers are checked to see if the transit vehicle is the one they should take. In this case, the passengers board the vehicle until all passengers have boarded or there is no more capacity on the vehicle. If a passenger cannot get on the vehicle because of the capacity limitation, it is reported as a capacity violation, and the assigned path has to be adjusted for the passenger object. Finally, based on the number of boardings and alightings, and also the type of transit vehicle, the dwell time is calculated for the event and stored for further processing and for feedback to the next transit vehicle simulation within DynusT. Additionally, the passenger load is stored for each vehicle at each stop, which is used as output for evaluating the performance of the system.

- **Postprocessing Functions**
  Like any simulation tool, the passenger simulation model generates data to be processed for generating useful output measures. There are several postprocessing functions in which the results of the simulation are either converted or printed in the output files. Here we mention the main functions and the reason they are needed.

  First, the new dwell times—calculated based on passenger boardings and alightings—are assigned to the transit vehicles and are reported in the output files. More specifically, the transit network input files to DynusT are modified using the new dwell times to be used in the next iteration of simulation in DynusT (the outer loop of the loosely coupled model). Furthermore, a gap value is calculated based on the relative amount of change in the dwell times, which is one measure of how much the transit passengers change their paths compared with the last iteration. The dwell time gap is used for evaluating the convergence of the outer loop of the DynusT/FAST-TrIPs system; the outer loop terminates if the dwell time gap value is lower than a predefined value. In the case that the result does not satisfy the convergence criterion, termination may occur based on the maximum allowed number of iterations.

  The second set of postprocessing functions is composed of the skim generators. There are two functions for generating transit and intermodal skims to prepare measures of the level of service, which may be intended either for feedback to DaySim or as final output. Using these functions, the averages of travel time, in-vehicle time, number of transfers, walking distance, and so on, are generated for each origin–destination at each time of day (AM, MD, PM, NT—that is, morning, midday, evening, and nighttime) and are printed in the output files for feedback to the DaySim demand model.

  Finally, some statistical measures are generated from the transit vehicles and the transit passengers and are printed in the output files. Detailed information about the output files is provided later in this report in Sections 5.1.6 and 5.2.2.

5.1.2 Input Files

The following are transit network and schedule files.

- **ft_input_stops.dat**
  This file contains the information on the name, location, type, capacity, and other properties of transit stops. However, no information is stored in this file on the transit service at each stop. Table 5.1 shows the contents of the file with a description of each field, and Figure 5.3 gives an example of this file.

Table 5.1 Content of ft_input_stops.dat

Field	Data Type	Description
ID	String	The identification number (ID) of the stop, which is unique over stop set.
Name	String	The name or description of the stop, usually street or intersection name where it is located.
Lat	Real	Latitude of the stop location.
Lon	Real	Longitude of the stop location.
FromNode	String	The ID of the upstream node of the link on which the stop is located.
ToNode	String	The ID of the downstream node of the link on which the stop is located.
Dist	Integer	The distance from the stop to the downstream node.
PulloutCapacity	Integer	A number showing the number of buses that can stop off the street.
TimePoint	Binary	Indicates whether or not buses should wait for scheduled time if they are ahead of schedule.
PassengerCapacity	Integer	Number of passengers that can wait on the platform, typically a large number for regular bus stops.

```
ID Name Lat Lon FromNode ToNode Dist PulloutCapacity TimePoint PassengerCapacity
319 MATHER_LRS_&_BAY_2_(NB) 38.584172 -121.311236 0 0 0 0 1 100
323 MATHER_LRS_&_BAY_3_(NB) 38.584315 -121.311572 0 0 0 0 1 100
336 MATHER_LRS_&_BAY_4_(SB) 38.584011 -121.311641 0 0 0 0 1 100
370 MATHER_LRS_&_BAY_5_(SB) 38.583851 -121.311524 0 0 0 0 1 100
```

Figure 5.3 Example of ft_input_stops.dat.

- **ft_input_routes.dat**
  This file contains the information about transit routes, where a route is the group of trips displayed to riders as a single service. It includes the general information about routes, like the name and type of service, and does not provide any information about the service itself. Details of the file can be found in the list in Table 5.2 and Figure 5.4.

Table 5.2 Content of ft_input_routes.dat

Field	Data Type	Description
ID	String	The identification number of the route, which is unique over route set.
ShortName	String	The short name that riders use to identify a route.
LongName	String	Full name of a route, which is more descriptive and includes the route's destination or stop.
Type	Integer	The type of service, e.g., bus, light rail transit (LRT), ferry.

```
ID        ShortName          LongName Type
21        21        SUNRISE_-_CITRUS_HEIGHTS  3
28        28        FAIR_OAKS_-_FOLSOM_BLV    3
72        72        ROSEMONT_-_LINCOLN_VILLAGE          3
74        74        INTERNATIONAL     3
75        75        MATHER_FIELD      3
```

Figure 5.4 Example of ft_input_routes.dat.

- **ft_input_trips.dat**
  As explained in Table 5.3 and illustrated in Figure 5.5, this file contains the list of trips that are served by each route and the associated start time of each trip at a terminal. A trip is a sequence of two or more stops that occurs at specific times. More information is also provided in this file on the type of service and the capacity of vehicles.

Table 5.3 Content of ft_input_trips.dat

Field	Data Type	Description
ID	String	The identification number of the trip, which is unique over trip set.
Route	String	The route ID in which the trip is provided.
Type	Integer	The type of service, e.g., bus, LRT, ferry.
StartTime	Integer	The departure time of the transit vehicle at the first stop. The time is in hours-minutes-seconds (hh-mm-ss) format.
Capacity	Integer	The maximum number of passengers who can board the vehicle.
Shape	Integer	The ID of the shape that defines how a line should be drawn on the map to represent a trip.

```
ID       ShortName          LongName Type
21       21       SUNRISE_-_CITRUS_HEIGHTS   3
28       28       FAIR_OAKS_-_FOLSOM_BLV     3
72       72       ROSEMONT_-_LINCOLN_VILLAGE          3
74       74       INTERNATIONAL      3
75       75       MATHER_FIELD       3
```

Figure 5.5 Example of ft_input_trips.dat.

- **ft_input_stopsTimes.dat**
  This file is basically the main file providing transit service, in the form of a schedule. It contains the times that a vehicle arrives at and departs from individual stops for each trip. The format of the file is described in Table 5.4. The example is given in Figure 5.6.

Table 5.4 Content of ft_input_stopsTimes.dat

Field	Data Type	Description
Trip	String	The identification number of a trip, defined in ft_input_trips.dat.
SchDeparture	Integer	The scheduled arrival time at a specific stop for a specific trip on a route.
SchArrival	Integer	The scheduled departure time at a specific stop for a specific trip on a route.
Stop	String	The ID that uniquely identifies a stop, initiated in ft_input_stops.dat. Multiple routes may use the same stop.
Sequence	Integer	It identifies the order of the stops served in a particular trip.

```
ID       ShortName          LongName Type
21       21       SUNRISE_-_CITRUS_HEIGHTS   3
28       28       FAIR_OAKS_-_FOLSOM_BLV     3
72       72       ROSEMONT_-_LINCOLN_VILLAGE          3
74       74       INTERNATIONAL      3
75       75       MATHER_FIELD       3
```

Figure 5.6 Example of ft_input_stopsTimes.dat.

- **ft_input_accessLinks.dat**
  The walking access/egress links are stored in this file. The file includes the estimate of walking distance and time at each TAZ to the stops within that TAZ. The estimate is based on the TAZ area size and the density of stops in it. The generated links can be used both for access and egress. Table 5.5 explains the format, and Figure 5.7 illustrates this file.

Table 5.5 Content of ft_input_accessLinks.dat

Field	Data Type	Description
TAZ	String	The TAZ ID in which the access link is generated.
Stop	String	The stop ID that is accessible to the given TAZ.
Dist	Float	The estimate of distance between TAZ and stop in miles.
Time	Float	The estimate of walking time in minutes.

```
TAZ  Stop Dist Time
70   12147    0.391079
     7
70   12148    0.297618
     5
```

Figure 5.7 Example of ft_input_accessLinks.dat.

- ft_input_transfers.dat
  Similar to the access links, transfer walking links are also stored in this file, including the distance and time between pairs of stops for which a transfer is possible. The generated transfer links can be used in both directions even though they are stored as one-way links only. Table 5.6 explains the format, and Figure 5.8 illustrates the contents.

Table 5.6 Content of ft_input_transfers.dat

Field	Data Type	Description
FromStop	String	The ID of the first stop to generate transfer link.
ToStop	String	The stop ID that is accessible to the given TAZ.
Dist	Float	The distance between the two stops in miles.
Time	Float	The walking time between the two stops in minutes.

FromStop	ToStop	Dist
319	323	0.0292051
319	336	0.0346856
319	370	0.0382895
323	319	0.0292051
323	336	0.0301518
323	370	0.0454614
336	319	0.0346856

Figure 5.8 Example of ft_input_transfers.dat.

- **ft_input_transitVehicles.dat**
  For the purpose of transit passenger simulation, and to generate transit vehicle objects, the combination of information on transit service is used to generate this file, in which each line is the representation of a transit vehicle and includes some details about its movement in the network. Table 5.7 explains the format, and Figure 5.9 shows the content of this file.

Table 5.7 Content of ft_input_transitVehicles.dat

Column	Content	Data Type	Description
1	Trip ID	String	The trip ID that the vehicle is generated for. Usually called vehicle ID.
2	Route ID	String	The route ID of the given trip ID.
3	Shape ID	String	The shape ID of the given trip ID.
4	Start Time	Integer	The start time of the vehicle in minutes after midnight. This is the departure time from the first stop.
5	Node IDs	String	The list of the node IDs in the roadway network, showing the transit vehicle's path, separated by commas.
6	Stop IDs	String	The list of the stop IDs that the transit vehicle visits during its trip, separated by commas.
7	Scheduled Times	String	The list of the scheduled departure times at each stop in minutes after midnight, separated by commas.
8	Travel Distances	String	The cumulative distances of stops from the start point of the trip in feet, separated by commas.

```
341538  1       13197   308
        454,2638,3079,2626,3078,8205,3077,3076,3075,3074,6156,3058,3057,3056,3395,3055,2342
,3101,8202,4474,4476,3352,4316,4473,3396,3054,3041,3053,3052,5408,9391,5409,14166,3330,7253
,8235,7238,4448,3332,3019,3333,1892,1887
        1184,3301,1187,1188,1189,1190,1191,1192,1193,1195,1196,1197,1198,3348,3349,3350,335
1,3352,3353,3354,3355,3356,3357,563,564,565,566,567,568,569,570,571,572,573,3097,3098,9800,
3099,3100,3101,3102,5366,520,2945
        308,308,310,310,310,312,312,312,314,315,317,317,317,319,319,321,321,323,324,324,326
,326,326,328,328,330,330,330,333,333,333,333,335,335,335,337,337,339,339,339,339,341,342,34
4
```

Figure 5.9 Example of ft_input_transitVehicles.dat.

- **ft_input_park-n-rides.dat**
  This file shows the information concerning park-and-ride facilities in the network, where travelers can drive, park their car, and transfer to transit systems. The park-and-rides have to be connected to some nodes in the auto network and some stops in the transit network using walking links. Table 5.8 shows the format of the files containing park-and-ride facilities, and Figure 5.10 shows the contents of this file.

Table 5.8 Content of ft_input_park-n-rides.dat

Field	Data Type	Description
ID	String	The park-and-ride ID that is unique over the set of park-and-rides.
Name	String	The name of park-and-ride lot that is known by users.
LAT	Integer	The latitude of the centroid of park-and-ride.
LON	Integer	The longitude of the centroid of park-and-ride.
Capacity	Integer	The maximum number of car parking spaces.
Fee	Integer	The monetary cost of parking a car in the park-and-ride lot.
Nodes	String	The list of node IDs in the auto network that are directly connected to the park-and-ride.
DriveTimes	String	The estimate of drive time from the given nodes to the park-and-ride and time to park the car.
Stops	String	The list of stop IDs that are accessible by walking from the park-and-ride.
WalkTimes	String	The estimate of walking time from the park-and-ride to the given stops.

```
ID   Name LAT  LON  Capacity  Fee  Nodes     DriveTimes     Stops
     WalkTimes
6384 Florin_LRT     -121 -121 -121 -121 693  2    8049 3
4606 Butterfield_LRT     -121 -121 -121 -121 905  2    8007 3
4194 Watt/Manlove_LRT    -121 -121 -121 -121 907  2    8006 3
```

Figure 5.10 Example of ft_input_park-n-rides.dat.

5.1.3 Demand File

- **veh_sout.dat**

  One of the outputs of the activity-based model is the disaggregate trip table showing travelers' trips with origin, destination, mode, time, and some information about their tours. This output is in dbf format, and a script is used in the integration model to convert it to a text file. The result is the *veh_sout.dat* file and is read by FAST-TrIPs. The content of this file, which is exactly the same as the dbf file, is shown in Table 5.9. Some of the information in this file is not used directly in FAST-TrIPs.

Table 5.9 Content of veh_sout.dat

Column	Content	Data Type	Description
1	Household ID	Integer	The ID of the household in the generated population.
2	Person ID	Integer	The ID of the person in the household.
3	Tour Number	Integer	The tour number for the person.
4	Half-Tour Number	Integer	An indicator showing whether the record is for the first or the second half of the tour (1 = outbound; 2 = return).
5	Trip Number	Integer	The trip number within the half-tour.
6	Origin TAZ ID	Integer	TAZ ID showing where the trip is generated from.
7	Origin Parcel ID	Integer	Parcel ID showing where the trip is generated from.
8	Destination TAZ ID	Integer	TAZ ID showing where the trip is destined to.
9	Destination Parcel ID	Integer	Parcel ID showing where the trip is destined to.
10	Travel Mode	Integer	An integer showing the mode of transportation used for making the trip
11	Origin Activity Purpose	Integer	Trip origin activity purpose.[a]
12	Destination Activity Purpose	Integer	Trip destination activity purpose.[a]
13	Departure Time	Integer	The departure time from the origin in hours-minutes (hhmm) format.
14	Arrival Time	Integer	The arrival time to the destination in hhmm format.
15	Travel Time	Float	The estimated travel time in minutes.
16	Travel Distance	Float	The estimated travel distance in miles.
17	Expansion Factor	Float	Fixed to 1.0.

[a] 1 (work), 2 (school), 3(escort), 4 (personal business), 5 (shopping), 6 (meal), 7 (social/recreation), and 8 (home).

5.1.4 FAST-TrIPs Internal Files

- **ft_demand_auto.dat**

  After running the intermodal assignment, the auto, transit, and intermodal trips are separated and are given to either DynusT or the transit assignment function in FAST-

TrIPs. This file contains the auto trips and the drive part of the intermodal trips with the optimal park-and-ride as the origin or the destination. The format of this file is the same as *veh_sout.dat* (see Table 5.10) but with four additional columns for intermodal trips showing the adjusted origin, destination, and time of travel. This file is read by DynusT.

Table 5.10 Content of ft_demand_auto.dat

Column	Content	Data Type	Description
1-17			The same as veh_sout.dat.
18	New Origin TAZ ID	string	The new origin TAZ after the intermodal assignment. It differs from the main origin TAZ if the trip is a drive-to-transit and the half-tour is 2.
19	New Destination TAZ ID	string	The new destination TAZ after the intermodal assignment. It differs from the main destination TAZ if the trip is a drive-to-transit and the half-tour is 1.
20	New Departure Time	Integer	The new departure time after the intermodal assignment. It differs from the main departure time if the trip is a drive-to-transit and the half-tour is 2.
21	New Arrival Time	Integer	The new arrival time after the intermodal assignment. It differs from the main arrival time if the trip is a drive-to-transit and the half-tour is 1.

- **ft_demand_transit.dat**
  As explained in Table 5.11, this file contains the transit trips and the transit part of the intermodal trips with the optimal park-and-ride as the origin or the destination. The format of this file is the same as *veh_sout.dat* but with four additional columns for intermodal trips showing the adjusted origin, destination, and times of travel. This file is read by FAST-TrIPs for transit assignment and simulation.

Table 5.11 Content of ft_demand_transit.dat

Column	Content	Data Type	Description
1–17			The same as veh_sout.dat.
18	New Origin TAZ ID	string	The new origin TAZ after the intermodal assignment. It differs from the main origin TAZ if the trip is a drive-to-transit and the half-tour is 1.
19	New Destination TAZ ID	string	The new destination TAZ after the intermodal assignment. It differs from the main destination TAZ if the trip is a drive-to-transit and the half-tour is 2.
20	New Departure Time	Integer	The new departure time after the intermodal assignment. It differs from the main departure time if the trip is a drive-to-transit and the half-tour is 1.
21	New Arrival Time	Integer	The new arrival time after the intermodal assignment. It differs from the main arrival time if the trip is a drive-to-transit and the half-tour is 2.

- **ft_passengers.dat**
  As explained in Table 5.12, the result of transit passenger assignment, which is the optimal path for each passenger trip, is stored in this file and passed to the passenger simulation model. This file contains not only the information on the passenger, but also the details of the paths and, thus, can be considered as an output as well. This file is illustrated in Figure 5.11.

Table 5.12 Content of ft_passengers.dat

Column	Content	Data Type	Description
1	Passenger ID	string	The unique ID of each passenger, which is generated inside FAST-TrIPs based on the input information.
2	Mode	Integer	The mode of transportation, which is either 3 for transit only or 1 or 2 for intermodals.
3	Origin TAZ ID	string	TAZ ID showing where the trip is generated from.
4	Destination TAZ ID	string	TAZ ID showing where the trip is destined to.
5	Start Time	Integer	The start time of the passenger from the origin TAZ.
6	Boarding Stop IDs	String	The list of stop IDs where the passenger gets on transit vehicle, separated by a comma if more than one.
7	Trip IDs	String	The list of trip IDs the passenger boards, separated by a comma if more than one.
8	Alighting Stop IDs	String	The list of stop IDs where the passenger gets off transit vehicle, separated by a comma if more than one.
9	Walking Times	String	The list of walking times the passenger should use for access, transfer, and egress, separated by a comma.

```
1001.1111    3   1161 812  886  2243 343649   522  4,2
1001.1121    3   812  1161 1167 519,224,380
```

Figure 5.11 Example of ft_passengers.dat.

5.1.5 Integration Files

- **network.dat**
  This file can be found in the discussions in Section 5.2.1.
- **AltTime_transit.dat**
  This file takes the identical information as that in the table for *AltTime.dat* in Section 5.2.2.
- **AltTime_interModal.dat**
  This file takes the identical information as that in the table for *AltTime.dat* in Section 5.2.2.

- **travelerVehicleMapping.dat**
  This file is used to make the connection between travelers as the outputs of the activity-based model, on the one hand, and vehicles that are generated in DynusT, on the other hand. Each traveler in the demand file may have several trips in a day. Because every time a trip is simulated in DynusT a new vehicle ID is assigned to it, the mapping between traveler ID and all the associated vehicle IDs is kept in this file. Table 5.13 shows how the mapping is stored. Figure 5.12 illustrates the content of this file.

Table 5.13 Content of travelerVehicleMapping.dat

Column	Content	Data Type	Description
1	sampn	String	Showing the household ID for a traveler, directly from sout.dbf.
2	persn	Integer	Showing the person ID in a household for a traveler, directly from sout.dbf.
3	numVeh	Integer	The number of vehicles generated for the traveler, equal to the number of trips for the traveler in sout.dbf.
>3	veh	Integer	The DynusT vehicle IDs generated for the traveler ID.

```
Trav ID (sampn, persn) numVeh veh1, veh2,...
     18940    5    3   2018705   2168786   2558322
    308525    1    4    759698   1226578   2147747
2401207
    130373    4    2   3685198   4864081
    295643    1    5    928825   2684792   2728078
3328695   3530870
```

Figure 5.12 Example of travelerVehicleMapping.dat.

- **travelerVehIntermodal.dat**
  For the purpose of intermodal tours assignment and simulation, a separate file but similar to *travelerVehicleMapping.dat* is used to store more information about linkage between travelers and their vehicles. The main reason for preparing this file is to provide the information about intermodal vehicles' arrival time to park-and-rides and to store the intermodal vehicles separately from other vehicles for more efficiency. The format of the file is shown in Table 5.14**,** and an example is illustrated in Figure 5.13.

Table 5.14 Contents of travelerVehIntermodal.dat

Column	Content	Data Type	Description
1	sampn	Integer	Showing the household ID for a traveler, directly from sout.dbf.
2	persn	Integer	Showing the person ID in a household for a traveler, directly from sout.dbf.
3	tour	Integer	The tour number for the given traveler in the day.
4	halfTour	Integer	Indicator showing if the trip is in the first half or the second half of the tour.
5	trip	Integer	The trip sequence in the given half-tour.
6	veh	Integer	The DynusT vehicle IDs generated for the traveler ID.

```
Trav ID (sampn, persn, tour, halfTour, trip) veh
743081   1   1   1   1   100030
52896    1   1   2   1   4133861
357197   1   1   2   1   3058043
204954   1   1   1   3   176149
664574   1   1   2   2   4125660
```

Figure 5.13 Example of travelerVehIntermodal.dat.

- **TransitRouteSchedule.dat**
  This file contains the information on transit network as input to DynusT for simulating transit vehicles in a congested traffic network. It basically provides transit vehicles as specific vehicles with given routes and scheduled times at stops. The file contains a header that describes the content of the file. A brief explanation is also given here. After the header, at Line 17, there are two numbers: the first is the number of routes in the transit network, and the second is the start time of simulation. For example, if the number is 2.00, all the scheduled times have to be added to 3:00 and converted to minutes after midnight. Then there are blocks of data for transit routes, separated by a separator line. For each route, there are two blocks of data: first is the path that vehicles operate on, and second is the schedule time for all the vehicles in the route. In the first block, the first line has four numbers referring to

  1. DynusT Route ID, a combination of GTFS route ID and shape ID.
  2. An indicator for frequency-based (1) versus schedule-based (0) transit route.
  3. Transit system type (0: Bus, 1:Rail)
  4. Number of links in the path of the route.
     Then the path is generated as below:

Link Upstream Node, Link Downstream Node, Number of Stops on the Link, Triples Showing (stop ID, distance to downstream node, pull-out capacity)

There might be several triples for several stops on a link.

The next block of data for each route stop includes a line containing the number of trips in the route and the number of stops in each trip. Then there is a line for each trip starting with trip ID and continuing with scheduled departure time for the stops, all separated by a tab or four spaces. This file is illustrated in Figure 5.14.

```
! TransitRoute.dat header - leave these 14 lines unchanged - YC Chiu, Univ of Arizona
! number of routes, start clock of simulation (note that all time are in decimal format)
! for each route
    ! route number, schedule-based(0), number of route nodes
    ! route nodes
    ! route node type - 0. withbay bus stop, 2. w/o bay bus stop, 0. non-stop
    ! capacity - if none zero -> PnR facility
    ! Number of schedules/trips
    ! For each schedule
        ! departure time at stops
        ! boardings at stops
        ! alightings at stops

=============================================
!FromNode      ToNode  numOfStops      Stops(4)        Distances(4)    Capacity(4)
=============================================
5       4.00
21_13247        0       0       17
32      33      0       -1      -1      -1      -1      -1      -1      -1      -1      0       0       0       0
33      117     0       -1      -1      -1      -1      -1      -1      -1      -1      0       0       0       0
117     550     1       2722    -1      -1      -1      167     -1      -1      -1      0       0       0       0
550     116     1       2723    -1      -1      -1      366     -1      -1      -1      0       0       0       0
116     127     1       2724    -1      -1      -1      377     -1      -1      -1      0       0       0       0
127     111     3       2726    2727    2728    -1      1315    384     255     -1      0       0       0       0
111     275     2       2729    2730    -1      -1      969     22      -1      -1      0       0       0       0
275     120     2       2731    2732    -1      -1      604     70      -1      -1      0       0       0       0
120     114     1       2733    -1      -1      -1      545     -1      -1      -1      0       0       0       0
114     274     1       2734    -1      -1      -1      495     -1      -1      -1      0       0       0       0
274     99      1       2735    -1      -1      -1      263     -1      -1      -1      0       0       0       0
99      263     0       -1      -1      -1      -1      -1      -1      -1      -1      0       0       0       0
263     100     2       2736    2737    -1      -1      201     160     -1      -1      0       0       0       0
100     262     1       2738    -1      -1      -1      72      -1      -1      -1      0       0       0       0
262     98      0       -1      -1      -1      -1      -1      -1      -1      -1      0       0       0       0
98      147     0       -1      -1      -1      -1      -1      -1      -1      -1      0       0       0       0
147     77      1       319     -1      -1      -1      370     -1      -1      -1      0       0       0       0
4       17
342176  4.53    4.53    4.55    4.56    4.56    4.57    4.58    4.59    5       5       5.01    5.02    5.03    5.04    5.05    5.05    5.
342149  5.23    5.23    5.25    5.26    5.26    5.27    5.28    5.29    5.3     5.3     5.31    5.32    5.33    5.34    5.35    5.35    5.
342155  5.53    5.53    5.55    5.56    5.56    5.57    5.58    5.59    6       6       6.01    6.02    6.03    6.04    6.05    6.05    6.
342150  6.22    6.22    6.24    6.25    6.25    6.26    6.27    6.28    6.29    6.3     6.31    6.32    6.33    6.34    6.35    6.35    6.
=============================================
21_13248        0       0       17
32      33      0       -1      -1      -1      -1      -1      -1      -1      -1      0       0       0       0
33      117     0       -1      -1      -1      -1      -1      -1      -1      -1      0       0       0       0
117     550     1       2722    -1      -1      -1      167     -1      -1      -1      0       0       0       0
550     116     1       2723    -1      -1      -1      366     -1      -1      -1      0       0       0       0
116     127     1       2724    -1      -1      -1      377     -1      -1      -1      0       0       0       0
127     111     3       2726    2727    2728    -1      1315    384     255     -1      0       0       0       0
111     275     2       2729    2730    -1      -1      969     22      -1      -1      0       0       0       0
```

Figure 5.14 Example of TransitRouteSchedule.dat.

- **TransitDwellTime.dat**

  The other file for transit input data into DynusT contained information about dwell times in a format similar to that in *TransitRouteSchedule.dat*. Each route in this file has one block of data starting with the DynusT route ID in the first line and the number of trips and number of stops in the second line. Then there is a line for each trip starting with trip ID and containing dwell time at stops of the trip. The route information is separated by a separator line from other routes. Figure 5.15 illustrates this file.

```
1_13197
46   13
341538   0.30 0.30 0.30 0.30 0.30 0.30 0.30 0.30 0.30 0.30 0.30 0.30 0.30
341539   0.30 0.30 0.30 0.30 0.30 0.30 0.30 0.30 0.30 0.30 0.30 0.30 0.30
341540   0.30 0.30 0.30 0.30 0.30 0.30 0.30 0.30 0.30 0.30 0.30 0.30 0.30
341541   0.30 0.30 0.30 0.30 0.30 0.30 0.30 0.30 0.30 0.30 0.30 0.30 0.30
341542   0.30 0.30 0.30 0.30 0.30 0.30 0.30 0.30 0.30 0.30 0.30 0.30 0.30
```

Figure 5.15 TransitDwellTime.dat.

5.1.6 Output Files

- **ft_output_convergence.dat**
  This file measures the stability of the FAST-TrIPs–DynusT solution based on the gap function with respect to dwell time. The first line in the file shows an indicator that is 1 if the system is converged and is 0 otherwise. Any additional comment is given in Line 2, and the value of dwell time gap for the whole system is given in Line 3. The rest of the content of the file is the same information in route level. In each line in the rest of the file, the route ID, sum of dwell time in the given route, and the gap value for the route are given for more accurate output. Figure 5.16 illustrates this file.

```
0
System has not converged!
Relative gap on dwell time
= 0.428816
342147   18   54
342148   28   128
```

Figure 5.16 Example of ft_output_convergence.dat.

- **ft_output_loadProfile.dat**
  The main result of the transit simulation is the passenger load on each transit vehicle along a route. The format of information in this file is similar to that in *ft_input_stopTimes.dat*, but with some additional information that is the result of the simulation. Table 5.15 shows the format of this file, and Figure 5.17 illustrates this file.

Table 5.15 Content of ft_output_loadProfile.dat

Field	Data Type	Description
vehRouteId	String	–
vehShapeId	String	–
eventVehicle	String	This is the same as the vehicle ID defined in the input file. Basically vehicle ID is based on trip ID.
cumDist	String	The cumulative distance that the transit vehicle has traveled from the start point in the route.
scheduledStop	String	The stop ID that has to be served in the given sequence based on the schedule.
eventStop	String	The stop ID that is served in the given sequence based on the simulation in DynusT.
scheduledTime	Real	Scheduled departure time from the stop based on the schedule.
eventTime	Real	The realized departure time from the stop based on simulation results.
vehLoad	Integer	The number of passengers being on board at the time of departing from the stop.

```
vehRouteId    vehShapeId   eventVehicle
cumDist   scheduledStop    eventStop
scheduledTime     eventTime    vehLoad
507  13431    345193   0    7093 7093 229  229  0
```

Figure 5.17 Example of ft_output_loadProfile.dat.

- **ft_output_passengers.dat**
  The other disaggregate output of the simulation consists of the passenger trajectories in the form of main time points during their trips. The file provides the information on the departure and arrival times as well. Table 5.16 explains the format of this file, and Figure 5.18 illustrates this file.

Table 5.16 Content of ft_output_passengers.dat

Content	Data Type	Description
passenger ID	String	The unique ID of the passenger, which is generated inside FAST-TrIPs based on the input information.
startTime	Real	The start time of the passenger from the origin TAZ based on the simulation.
endTime	Real	The arrival time of the passenger trip at the destination TAZ based on the simulation.
arrivalTimes	String	The list of arrival times to each stop that the passenger experienced in the simulation, separated by commas.
boardingTimes	String	The list of boarding times for each transit vehicle that the passenger experienced in the simulation, separated by commas.
alightingTimes	String	The list of alighting times from each transit vehicle that the passenger experienced in the simulation, separated by commas.

```
passengerId  startTime    endTime  arrivalTimes  boardingTimes    alightingTimes
2613.3311    496  493  36400,37020, 36580,37210, 36920,37360,
12246.2111   677  806  36000,   36060,   37690,
12895.1111   915  768  36700,37531, 36780,37808, 37530,38008,
12553.1111   690  786  30700,33216,35841,    30800,33570,36600,    33016,35840,37980,
12894.1111   915  768  36700.37761. 36780.38260. 37760.38610.
```

Figure 5.18 Example of ft_output_passengers.dat.

- **ft_output_capacityViolation.dat**
  This is a self-explanatory file showing the passengers who cannot finish their trips to their destinations for any reason. It lists the passenger ID, the target vehicle ID that passenger failed to board, the stop ID, and time. The last number is the missing case, which is 1 if the passenger is late for the vehicle or is 4 if there is no capacity on the vehicle for the passenger to board. Figure 5.19 illustrates this file.

```
Passenger 31981.1111      missed bus 254816 at stop 23513    at time 382.15    MC: 1
Passenger 16302.1221      missed bus 254943 at stop 23399    at time 385.133   MC: 1
Passenger 1972.2211       missed bus 342696 at stop 529      at time 385.15    MC: 4
Passenger 2753.1111       missed bus 343022 at stop 1828     at time 406.733   MC: 4
Passenger 5799.2111       missed bus 341779 at stop 327      at time 411.717   MC: 4
```

Figure 5.19 Example of ft_output_capacityViolation.dat.

- **ft_output_transitPassengersStat.dat**
  Again, this file is a self-explanatory file reporting the network-level measures of effectiveness, including average in-vehicle time, waiting time, walking time, number of transfers, and so on. The first column of the numbers is the total value, and the second is the average value. Figure 5.20 illustrates this file.

```
totalPassengers =     635
totalInVehDist = 4205.27
    6.62248
totalInVehTime = 18324.4
    28.8573
```

Figure 5.20 Example of ft_output_transitPassengersStat.dat.

- **ft_output_transitVehiclesStat.dat**
  Similar to the output for passengers, the simulation results are provided in this file for the transit route. This file can be important in the application of the model. The data description is provided in Table 5.17. Figure 5.21 illustrates this file.

Table 5.17 Content of ft_output_transitVehiclesStat.dat

Content	Data Type	Description
routeId	String	–
shapeId	String	–
vehicleId	String	This is the same as the vehicle ID defined in the input file. Basically vehicle ID is based on trip ID.
totalTraveledDistance (mile)	Real	The distance that vehicles travel in the route.
scheduledTravelTime (minutes)	Integer	The average travel time of the vehicles in the route, based on schedule.
realizedTravelTime (minutes)	Integer	The average travel time of the vehicles in the route, based on simulation.
averageSpeed (mph)	Real	The average operating speed of the vehicles in the route, based on simulation.
averageStopDelay (minutes)	Real	The average time a vehicle can be late to stop in the route, based on simulation.
totalBoardings	Integer	The total number of passengers boarding the vehicle along the route.
totalAlightings	Integer	The total number of passengers alighting the vehicle along the route.
averageLoad	Integer	Average number of onboard passengers on the vehicles in the route.
maxLoad	Integer	Maximum number of onboard passengers on the vehicles in the route.
averageDwell	Integer	The average dwell time of the vehicles in the route, based on simulation.
maxDwell	Integer	The maximum dwell time of the vehicles in the route, based on simulation.

```
routeId   shapeId  vehicleId    totalTraveledDistance(mile)   scheduledTravelTime(min)  realizedTravelTime(min)
averageSpeed(mph) averageStopDelay(min)     totalBoardings   totalAlightings  averageLoad  maxLoad  averageDwell
maxDwell
472  34   14786    43.2938  65   65.1667  39.8613  0.166667 1    1    0    1    1    8
472  34   14787    43.2938  65   65.1667  39.8613  0.166667 0    0    0    0    0    0
472  34   14788    43.2938  79   79   32.8813  0    0    0    0    0    0    0
```

Figure 5.21 Example of ft_output_transitVehiclesStat.dat.

5.2 DynusT Software Documentation

DynusT (Dynamic Urban Systems in Transportation) is a mesoscopic, simulation-based, dynamic traffic assignment (DTA) system. The documentation details the major input and output files for the DynusT simulation model. The following files are described as input files necessary to execute the DynusT simulation model.

All files described are not in fixed-location format, meaning the locations of values are not set in a specific spacing location of a line. As long as there is spacing between values, the DynusT simulation model will be able to read these input files. However, the NEXTA graphical user interface (GUI) will format the files in a fixed-location format when a network change is saved in the GUI. Also, the DynusT simulation model for output files will be formatted in a fixed-location format.

More information on the DynusT simulation model can be found in the online user's manual (Chiu, 2013).

5.2.1 Input Files

- **Network Data (network.dat)**
  The network input file is found in the data set folder under the title "network.dat." This file describes the roadway network configurations, including node IDs and link characteristics, as shown in Table 5.18.

Table 5.18 File Format of network.dat

Basic Data												
# Zones	# Nodes	# Links	# K-shortest paths	Use super zones								
Node Data (for all nodes)												
Node ID	Zone ID											
Link Data (for all links)												
From_ID	To_ID	# Left-turn bays	# Right-turn bays	Length	# Lanes	Traffic flow model	Speed adjustment factor	Speed limit	Max service flow rate	Saturation flow rate	Link type	% Grade

Table 5.19 describes each value of *network.dat*, and Figure 5.22 illustrates this file.

Table 5.19 Value Descriptions of network.dat

Basic Data	
# Zones	Number of zones in the network.
# Nodes	Number of nodes in the network.
# Links	Number of links in the network.
# K-shortest paths	Number of K-shortest paths calculated (default = 1).
Use super zones	Binary switch to use superzone structure; typically used for evacuation modeling (0 = no, 1 = yes).
Node Data (for all nodes)	
Node ID	Node ID.
Zone ID	Zone ID that the Node ID is found in or associated with.
Link Data (for all links)	
From_ID	Upstream Node ID of link.
To_ID	Downstream Node ID of link.
# Left-turn bays	Number of left-turn bays on link; length of turning bay is default value for all turning bays.
# Right-turn bays	Number of right-turn bays on link; length of turning bay is default value for all turning bays.
Length	Link length in feet.
# Lanes	Number of lanes of link.
Traffic flow model	Traffic flow model type that is associated to trafficflowmodel.dat. Default types: 1. 2-Regime (uninterrupted flow) 2. 1-Regime (interrupted flow)
Speed adjustment factor	Speed limit increase in free-flow conditions.
Speed limit	Speed limit of link.
Max service flow rate	Maximum capacity of moving vehicles along the link (vehicle/hour/lane).
Saturation service flow rate	Maximum discharging rate from a queue at the downstream node of the link. Applied to node intersection with signal (vehicle/hour/lane).
Link type	Link functional class in DynusT: 1. Freeway 2. Freeway Segment with Detector (for Ramp Metering) 3. On-Ramp 4. Off-Ramp 5. Arterial 6. HOT 7. Highway

	8. HOV 9. Freeway HOT 10. Freeway HOV
% Grade	Percentage grade incline on link.

Note: HOT = high-occupancy toll; HOV = high-occupancy vehicle.

```
 13  180  445   1   0
 1  3
 2  4
 3  3
 4  4
 5  6
170  13
171  13
172  13
173  13
174  13
175  11
176  8
177  10
198  10
199  1
200  2
 1   2  0  0  420 2   2 +0 40 1800 1800 5 +0
 1   20  0  0  1190 2   2 +0 40 1800 1800 5 +0
 1   81  0  0  1640 2   2 +0 40 1800 1800 5 +0
 2   1  0  0  420 2   2 +0 40 1800 1800 5 +0
 2  116  0  0  1410 2   2 +0 40 1800 1800 5 +0
 2  139  0  0  1200 2   2 +0 40 1800 1800 5 +0
 3   4  0  0  450 2   2 +0 40 1800 1800 5 +0
 3   24  0  0  1350 2   2 +0 40 1800 1800 5 +0
 3   82  1  0  1280 2   2 +0 40 1800 1800 5 +0
```

Figure 5.22 Example of network.dat.

- **Link Movement Data (movement.dat)**
  This file describes the available movements of every link at the connecting downstream node. Therefore, for every link it is known what movements can be made, including left-turn, through, right-turn, two additional turns, and U-turn, as indicated in Table 5.20.

Table 5.20 File Format of movement.dat

Movement Data (for all links)							
From_ID	To_ID	Left-turn	Through	Right-turn	Other-turn 1	Other-turn 2	U-turn

Table 5.21 describes each value of movement.dat.

Table 5.21 Value Descriptions of movement.dat

Movement Data (for all links)	
From_ID	Upstream Node ID of link.
To_ID	Downstream Node ID of link.
Left-turn	Downstream node from To_ID representing left-turn movement.
Through	Downstream node from To_ID representing through movement.
Right-turn	Downstream node from To_ID representing right-turn movement.
Other-turn 1	Downstream node from To_ID representing a turning movement that is not left, through, or right movement.
Other-turn 2	Downstream node from To_ID representing a turning movement that is not left, through, or right movement.
U-turn	Binary switch to allow U-turn at To_ID (0 = no, 1 = yes).

Other-turn 1 and Other-turn 2 are additional movements for situations where there exists additional connectivity beyond left, through, and right turns.

File example is illustrated in Figure 5.23.

```
1      2    116    139      0      0      0      1
1     20      0      3      0      0      0      0
1     81     82    130    116      0      0      1
2      1     20     81      0      0      0      1
2    116     19    199    100     98     81      0
2    139      0     98     97      0      0      1
3      4     22     89      0      0      0      1
3     24      0      5      0      0      0      0
3     82     83    131     81      0      0      1
4      3     24     82      0      0      0      1
4     22      0      2      0     21      0      0
4     89      0     97     90      0      0      1
5      6     26     90      0      0      0      1
5     27      0     65      0     28      0      0
5     83     84     80     82      0      0      1
6      5     27     83      0      0      0      1
6     26      0      4      0     25      0      0
6     90     89     91      0      0      0      1
7      8     66     92      0      0      0      1
7     31      0     36      0     32      0      0
7     85     86     71     84      0      0      1
8      7     31     85      0      0      0      1
```

Figure 5.23 Example of movement.dat.

- **Vehicle Generation Data (Origin.dat)**
  This file describes the generation links for each zone. A generation link is the location at which a vehicle will enter the network and begin travel toward its destination. Every zone must have at least one generation link. A link can be a generation link for more than one

zone in the instance that the link borders or intersects with more than one zone. This vehicle generation is shown in Table 5.22 and Table 5.23.

Table 5.22 File Format of origin.dat

Zone Data (for all zones)		
Zone ID	# Generation links	Vehicle generation weight switch
Generation Link Data		
From ID	To_ID	Vehicle generation weight

Table 5.23 describes each value of *origin.dat*

Table 5.23 Value Descriptions of origin.dat

Zone Data (for all zones)	
Zone ID	Zone ID number.
# Generation links	Number of links assigned as generation links for this zone. Every zone must have at least one generation link.
Vehicle generation weight switch	Binary switch that allows user to specify a loading weight to each generation link in the zone. Default switch (value = 0) means vehicle generation will be evenly distributed among all generation links. User-defined switch (value = 1) means vehicle generation will be weighted toward links with higher weights, as described below for generation links.
Generation Link Data	
From_ID	Upstream node of link.
To_ID	Downstream node of link.
Vehicle generation weight	By default, the vehicle generation weight is 0.0000. The user may specify loading weights to each generation link in the zone. If weights are user defined, all weights for generation links in the zone must sum up to 1.0000.

File example is illustrated in Figure 5.24.

```
1    1    0
199    116  0.000
2   18    0
 17     61  0.000
 17     79  0.000
 61    117  0.000
 78     79  0.000
 79     17  0.000
 79     78  0.000
 79    117  0.000
 79    129  0.000
115    117  0.000
115    173  0.000
115    174  0.000
117     63  0.000
        117     79  0.000
```

Figure 5.24 Example of origin.dat.

- **Vehicle Exit Data (destination.dat)**
  This file describes the destination nodes for each zone, as shown in Table 5.24 and Table 5.25. A destination node is the location at which a vehicle will exit the network in completion of its trip. Every zone must have at least one destination node. A node can be a destination node for more than one zone in the instance that the node borders or intersects with more than one zone.

Table 5.24 File Format of destination.dat

Zone Data (for all zones)		
Zone ID	# Destination nodes	Node ID of destination node

Table 5.25 describes each value of *destination.dat*.

Table 5.25 Value Descriptions of destination.dat

Zone Data (for all zones)	
Zone ID	Zone ID number.
# Destination nodes	Number of nodes assigned as destination nodes for this zone. Every zone must have at least one destination node.
Node ID of destination node	Node ID of destination node. The number of nodes found after the field “# Destination nodes” must be equal to the number specified in the field “# Destination nodes.”

File example is illustrated in Figure 5.25.

```
 1    1    199
 2    1    117
 3    6      3     20     23     67    119    131
 4    3      2     97     98
 5    2    103    158
 6    1     71
 7    5      8     96    140    141    146
 8    6     99    104    105    107    162    163
 9    6     11     72     73     74     75    125
10    4     95    109    110    168
11    4     13     76     77    175
12    5     78     87     88    128    135
       13    2    113    170
```

Figure 5.25 Example of destination.dat.

- **Demand Origin–Destination Matrices (demand.dat, demand_truck.dat, and demand_hov.dat)**
  As explained in Table 5.26, these three files describe the number of trips to be produced for each O-D pair for each vehicle type. Auto passenger vehicles will be produced by *demand.dat*, trucks produced by *demand_truck.dat*, and HOV vehicles by *demand_hov.dat*, as indicated in Table 5.26. Each file contains time-dependent O-D matrices with each matrix containing the number of trips to be generated for a specified time period. In the event of encountering fractions or decimal numbers, a rounding process is implemented to generate only an integer number of vehicles. However, mass balance is also enforced to ensure that the total number of vehicles generated for each O-D pair across all times is consistent with the ones specified in these files.

Table 5.26 File Format of demand.dat, demand_truck.dat, and demand_hov.dat

Parameter						
Number of O-D matrices listed in this file			Multiplier for uniform increase/decrease			
Start time of each O-D matrix						
Timestamp Link 1 Queue	Timestamp Queue	Timestamp Queue	…….			Timestamp for end of demand generation
O-D trips						
Start time – place holder						
# of trips	# of trips	# of trips	# of trips	# of trips	# of trips	
. . .						

File example is illustrated in Figure 5.26.

```
   24 1.000
    0.0   60.0  120.0  180.0  240.0  300.0  360.0  420.0  480.0  540.0  600.0  660.0  720.0
780.0  840.0  900.0  960.0 1020.0 1080.0 1140.0 1200.0 1260.0 1320.0 1380.0 1440.0
Start Time =    0.0
    1.7086    9.7574    1.5430    0.7619    0.9307    0.7198
    0.6268    0.0513    1.0952    0.3282    0.2562    0.7801
    0.3030    0.0000    0.0000    0.0531    0.7311    1.3299
    0.6478    0.0000    0.6603    0.2535    0.5571    0.2960
    0.5207    0.2428    1.0720    0.5804    0.0424    0.2338
    0.1187    0.2167    0.2844    0.1305    0.4086    0.5285
    0.1864    0.0676    0.3907    0.0461    0.1670    0.0586
    0.0822    0.0939    0.0693    0.1591    0.1285    0.1256
    0.2113    0.4016    0.4969    0.3038    0.4302    0.3100
    0.4551    0.1548    0.3968    0.1090    0.1274    0.1169
    0.1318    0.1482    0.0081    0.2376    0.2025    0.3813
    0.4495    0.8524    0.4559    0.1384    0.0604    0.9844
    0.2398    0.1311    0.1809    0.1502    0.0585    0.4383

    .
    .
    .
    .
Start Time =   60.0
    1.4646    8.3635    1.3225    0.6530    0.7977    0.6170
    0.5372    0.0440    0.9387    0.2813    0.2196    0.6686
    0.2597    0.0000    0.0000    0.0455    0.6267    1.1399
    0.5552    0.0000    0.5660    0.2173    0.4775    0.2537
    0.4463    0.2081    0.9189    0.4975    0.0363    0.2004
    0.1017    0.1857    0.2438    0.1118    0.3502    0.4530
    0.1598    0.0579    0.3349    0.0395    0.1432    0.0502
    0.0704    0.0805    0.0594    0.1364    0.1102    0.1076
    0.1811    0.3442    0.4259    0.2604    0.3688    0.2657
    0.3901    0.1327    0.3401    0.0934    0.1092    0.1002
           0.1130    0.1271    0.0069    0.2036    0.1736    0.3268
```

Figure 5.26 Example of demand.dat, demand_hov.dat, or demand_truck.dat.

- **Simulation and Assignment Configuration Data (system.dat)**
  As described in Table 5.27 and Table 5.28 and illustrated in Figure 5.27, this file describes the assignment and simulation settings when executing the DynusT simulation model.

Table 5.27 File Format of system.dat

Simulation Data			
Sim Time			
Vehicle Generation Mode Data			
# Iterations	Vehicle Generation Mode		
Aggregation Interval and Convergence Data			
Dummy	Aggregation interval	Dummy	Convergence Threshold
Dummy			
Default = 0			

Dummy
Default = 0

Table 5.28 Value Descriptions of system.dat

Simulation Data	
Sim Time	Simulation time in minutes.
# Iterations	Number of iterations specified. If running one-shot (short-term analysis), the value would equal 0. This means that DynusT will run through simulation *only* and will not continue through traffic assignment. This resembles the immediate impact of network conditions without the learning mechanism of DTA. If value is greater than 0, this will run through iterative mode (long-term analysis). This resembles the learning mechanism of DTA in response to network conditions.
Vehicle Generation Mode	There are three vehicle generation modes: - 0 = Vehicle file only - 1 = O-D demand file - 2 = Vehicle + path files "Vehicle file only" means a trip roster is present in the network with given traveler population. This will not include assigned paths for travelers to travel, so paths for each traveler will be determined by running the DynusT simulation model. "O-D demand file" means the traveler population and paths will be determined from the O-D demand matrices as input. "Vehicle + path files" means a trip roster is present, as well as each traveler's most updated paths from a previous DynusT simulation run.
Aggregation Interval and Convergence Data	
Aggregation Interval	The aggregation interval is the interval in which network conditions are updated for the time-dependent shortest path to use for updating the shortest path tree. The value is determined by the total number of simulation intervals per aggregation interval. A simulation interval is described in scenario.dat. By default, this value is 50, meaning there are 50 simulation intervals in one aggregation interval. This equals a 5-minute aggregation interval.
Convergence Threshold	DynusT uses the relative gap measurement to determine network solution convergence. This value refers to the % error of measuring the experienced path travel time versus the shortest path calculated travel time. The least convergence threshold allowable in DynusT is 0.0001. By default, the convergence threshold is 100, which means 0.0001 ×100 = 0.01 or 1% error relative gap threshold.

File example is illustrated in Figure 5.27.

```
180
20 1
100 100 1 10
0
0
```

Figure 5.27 Example of system.dat.

- **Model Scenario Data (scenario.dat)**
  For this file, Table 5.29 describes the general settings of the DynusT assignment procedure. Table 5.30 explains the value in *scenario.dat*. An example of the file is illustrated in Figure 5.28.

Table 5.29 File Format of scenario.dat

General Settings Data									
Relative indifference band		Threshold bound		Random number seed		Path index		Variable message sign mode preemption	
Dummy = 1									
Simulation interval									
TDSPinterval		Dummy = 5							
Stat Start Time		Stat End Time							
Multiple User Class (MUC) Information									
# Veh Types									
Veh type	Demand mode switch	Demand fraction	MUC proportion switch	% class 1	% class 2	% class 3	% class 4	% class 5	

Note: TDSP = time-dependent shortest path.

Table 5.30 Value Descriptions of scenario.dat

General Settings Data	
Relative indifference band	Percentage improvement for which traveler will switch his or her route mid-trip. This applies to Class 4 only (en route info).
Threshold bound	Travel time savings threshold at which traveler will switch to a better route. This applies to Class 4 only (en route info).
Random number seed	Random number generation seed. This can be any positive value. By default, this value is 0, which means the system will reset the random number each run.
Simulation interval	The time interval in which the simulation updates network conditions. By default this value is 6 seconds.
TDSP interval	The time interval in which the time-dependent shortest path algorithm is executed. This value is in terms of simulation intervals, therefore, by default, the value is 30. This means there are 30 simulation intervals that make one TDSP interval, or 3 minutes.
Stat Start Time	Start time in which all simulation statistics will be recorded. By default, the start time is at minute 0.
Stat End Time	End time in which all simulation statistics will end recording. By default, the end time is the end of simulation time in system.dat.
Multiple User Class Information	
# Veh Types	Number of vehicle types being simulated: - 1 = SOV; 2 = SOV, truck; 3 = HOV
Veh Type Format = Integer Width = 5	Vehicle type ID: - 1 = SOV; 2 = truck; 3 = HOV
Demand Mode switch Format = Integer Width = 5	Binary switch on if using a separate O-D demand file for this vehicle type, or using a percentage from SOV O-D demand file. (0 = separate demand file, 1 = % of SOV O-D demand file). By default, SOV must be switched to 0.
Demand proportion Format = Float Width = 6 (3 decimal precision)	If demand mode is switched to use percentage of SOV O-D demand file, this specifies what percentage will be used. If using a separate O-D demand file, this value is 0.000. By default, SOV O-D demand fraction is 1.000.
MUC proportion switch Format = Integer Width = 3	Binary switch used to determine if vehicle type will use the same proportion of MUC distribution as SOV. (0 = use own distribution, 1 = use SOV distribution). By default, SOV is set to 0.
% Class 1 Format = Float Width = 6 (3 decimal precision)	Class 1 = Historical Information. This can only be used when trip roster and path information are present (“Vehicle generation mode = 2”). Sum of percentages of all classes must equal 1.00.
% Class 2 Format = Float Width = 6 (3 decimal	Class 2 = System Optimal. This class minimizes total system travel time rather than individual traveler minimal travel time. Sum of percentages of all classes must equal 1.00.

precision)	
% Class 3 Format = Float Width = 6 (3 decimal precision)	Class 3 = User Equilibrium. This class minimizes individual traveler travel time. Sum of percentages of all classes must equal 1.00.
% Class 4 Format = Float Width = 6 (3 decimal precision)	Class 4 = En Route Info. This class represents those who use GPS unit or radio information with up-to-date network conditions. This class will switch route mid-trip based on network information. Sum of percentages of all classes must equal 1.00.
% class 5 Format = Float Width = 6 (3 decimal precision)	Class 5 = Pretrip Info. This class represents the informed traveler who will choose a route based on network conditions at the time of departure. Sum of percentages of all classes must equal 1.00.

Note: SOV = single-occupancy vehicle; HOV = high-occupancy vehicle; GPS = Global Positioning System.

File example is illustrated in Figure 5.28.

```
0.2 1 123 1 0
1
6
50 10
0 180
2
    1    0 1.000  0 0.000 0.000 1.000 0.000 0.000
           2    1 0.000  0
```

Figure 5.28 Example of scenario.dat.

- **Traffic Control Data (control.dat)**
  This file describes the settings of node control information, including four-way stop signs, two-way stop signs, actuated control, pretimed control, and yield sign. Format explanations are contained in Table 5.31.

Table 5.31 File Format of control.dat

General Control Timing Plan Data									
# Timing plans									
Plan Start Time									
Node Data (for all nodes)									
Node ID	Control Type	# Phases	Cycle Time						
Node Control Data (for all nodes with control type 4 or 5)									
Node ID	Phase #	Max Green / Offset	Min Green / green	Yellow Time	# inbound links	Upstream Node 1	Upstream Node 2	Upstream Node 3	Upstream Node 4
From_ID	To_ID	Phase #	# outbound nodes	Downstream Node 1- #					

Table 5.32 describes each value of *control.dat*.

Table 5.32 Value Descriptions of control.dat

General Control Timing Plan Data	
# Timing plans	Number of timing plans. There may be timing plans for a.m., p.m., and off-peak periods.
Plan Start Time	The start time of the timing plan.
Node Data (for all nodes)	
Node ID	Node ID.
Control Type	DynusT has 6 control types: 1. No control 2. Yield sign 3. 4-way stop sign 4. Pretimed signal control 5. Actuated signal control 6. 2-way stop sign
# Phases	If node is control type 4 or 5, specify the number of phases for the signal control. Otherwise, the value = 0.
Cycle time	If node is control type 4 or 5, specify the cycle time length in seconds. Otherwise, the value = 0.
Node Control Data (for all nodes with control type 4 or 5)	
Node ID	Node ID of node with control.
Phase #	The phase number in which the following signal timing information applies.
Max Green/offset	If control type = 5, specify maximum green time; if control type = 4, specify the offset time.
Min Green/Green	If control type = 5, specify minimum green time; if control type = 4, specify the green time.
Yellow Time	Yellow time of phase.
# inbound links	Number of inbound links into intersection of this phase. .
Upstream Node 1 - 4	Upstream node of the inbound link. Otherwise, this value is 0.
From_ID	Upstream node of inbound link.
To_ID	Downstream node of inbound links (intersection node ID).
Phase #	The phase number in which this information applies.
# outbound nodes	Number of outbound nodes from this inbound link.
Outbound Node 1-#	Node ID of outbound ID of inbound link movement (1 node through # of outbound node specified).

File example is illustrated in Figure 5.29.

```
  1
      0.00
      1 1 0   0
      2 1 0   0
      3 1 0   0
      4 1 0   0
      5 1 0   0
      6 1 0   0
      7 1 0   0
   5381 1 0   0
   5387 1 0   0
   5695 3 0   0
   5749 1 0   0
   5750 2 0   0
   5751 1 0   0
     75      1     50      8      3      1     74      0      0      0
     74     75      1      3   3365     76   3307
     75      2     50      8      3      1     76      0      0      0
     76     75      2      3   3307     74   3365
     75      3     50      8      3      1   3307      0      0      0
   3307     75      3      3     74   3365     76
     75      4     50      8      3      1   3365      0      0      0
   3365     75      4      3     76   3307     74
   1352      1     40     10      3      1   4504      0      0      0
   4504   1352      1      2   4507   4506
   1352      2     40     10      3      1   4506      0      0      0
   4506   1352      2      2   4505   4504
   1352      3     30     10      3      1   4505      0      0      0
   4505   1352      3      3   4504   4507   4506
   1352      4     30     10      3      1   4507      0      0      0
   4507   1352      4      2   4506   4505
=======Two Way Stop Signs/Yield Signs Below =======
     76      2      1
     75     76   1851     76
     77     76
   1312      2      1
   1313   1312   3511   1312
   3544   1312
   1313      2      1
   1312   1313   1314   1313
   2514   1313
   1314      2      2
   1313   1314   1315   1314
          1338   1314   1886   1314
```

Figure 5.29 Example of control.dat.

- **Work Zone Data (workzone.dat)**
  In this file, Table 5.33 and Table 5.34 describe the format and the value information of a work zone on a link.

Table 5.33 File Format of workzone.dat

Number of Work Zones						
# work zones						
Work Zone Description						
From_ID	To_ID	StartTime	EndTime	CapReduc	SpdLimit	QDischarge

Table 5.34 Value Descriptions of workzone.dat

Number of Work Zones	
# work zones	Number of work zones applied to links in the network.
Work Zone Description	
From_ID	Upstream Node ID of work zone link.
To_ID	Downstream Node ID of work zone link.
StartTime	Start time of work zone on link.
EndTime	End time of work zone on link.
CapReduc	Percentage of capacity lost due to work zone.
SpdLimit	Posted speed limit for link under work zone.
QDischarge	Queue discharge rate in which vehicles travel through the work zone link (vehicle/hour/lane). This is lower than normal capacity.

File example is illustrated in Figure 5.30.

```
166
14071 13366 0 600 0.25 45 1500
14018 14071 0 600 0.25 45 1500
14200 14018 0 600 0.25 45 1500
13572 14200 0 600 0.25 45 1500
12164 13572 0 600 0.25 45 1500
14072 12164 0 600 0.25 45 1500
14024 14072 0 600 0.25 45 1500
14075 14024 0 600 0.25 45 1500
14026 14075 0 600 0.25 45 1500
13581 14026 0 600 0.25 45 1500
13580 13581 0 600 0.25 45 1500
13583 13580 0 600 0.25 45 1500
13582 13583 0 600 0.25 45 1500
14029 13582 0 600 0.25 45 1500
14079 14029 0 600 0.25 45 1500
       14031 14079 0 600 0.00 60 1500
```

Figure 5.30 Example of workzone.dat.

- **Vehicle Trip Schedule Data (vehicle.dat)**
  This file describes the trip roster of individual vehicles and their traveler characteristics. Format of *vehicle.dat* is explained in Table 5.35.

Table 5.35 File Format of vehicle.dat

Header														
Total vehicle	Max destinations	Header information (default)												
Header Information (default)														
Vehicle Data (for all vehicles)														
Veh ID	Usec	Dsec	Stime	Vehcls	Vehtype	Ioc	#ONode	#IntDe	Info	Ribf	Comp	Izone	Evac	InitPos
DZone	waitTime													

Table 5.36 below describes each value of *vehicle.dat*.

Table 5.36 Value Descriptions of vehicle.dat

Header	
Total Vehicle	Total number of vehicles in trip roster.
Max Destinations	Maximum number of possible intermediate destinations of trip chain.
Vehicle Data (for all vehicles)	
VehID	Vehicle unique ID.
Usec	Upstream node ID of generation link that vehicle appears in network.
Dsec	Downstream node ID of generation links that vehicle appears in network.
Stime	Departure time of vehicle.
Vehcls	Vehicle MUC class (1–5).
Vehtype	Vehicle type (SOV, truck, HOV).
Ioc	Occupancy level of vehicle (SOV = 1, HOV = 2).
#ONode	If vehicle generation mode is "Veh + path file," then this is the number of nodes in route. If vehicle generation mode is "Veh only file,, then this is a value of 1.
#IntDe	Number of intermediate stops for vehicle's travel, including final destination.
Info	En route info vehicle binary switch (0 = no info, 1 = info). This applies to MUC Class 4 (en route info).
Ribf	Indifference band for switching the vehicle's route.
Comp	Compliance rate of vehicle.
Izone	Origin Zone ID.
Evac	Evacuation vehicle binary switch (0 = not evac, 1 = evac).
InitPos	Initial position along the generation link.
DZone	Destination zone of intermediate destination or final destination.
waitTime	Activity duration at the destination, otherwise 0 if final destination.

File example is illustrated in Figure 5.31.

```
5264446          1    # of vehicles in the file, Max # of stops
   #   usecdsecstimevehclsvehtypeioc #ONode #IntDe info ribf    comp   izoneEvacInitPosVoTtFlagpArrTimeTPIniGas
   1  13625   6288    0.00     3     1     1     1     1     0  0.0000  1.0000 1470    0    0.76377136    9.36    2  13.00    7  0.0
    431  12.90
   2   4586   3233    0.00     3     1     1     1     1     0  0.0000  1.0000  227    0    0.93079982    5.06    2  18.00    7  0.0
    444  17.97
   3   3289  14510    0.00     3     3     2     1     1     0  0.0000  1.0000  402    0    0.37531041    7.91    2   2.00    6  0.0
    402   2.23
   4   3103   3102    0.00     3     1     1     1     1     0  0.0000  1.0000  438    0    0.61575363    1.09    2  17.00    7  0.0
    292  17.33
   5   2028   3418    0.00     3     3     2     1     1     0  0.0000  1.0000  769    0    0.88036160    0.90    2  11.00    6  0.0
    288  11.23
   6  12835  12782    0.00     3     1     1     1     1     0  0.0000  1.0000 1331    0    0.99511410    7.26    2  58.00    7  0.0
    907  58.18
               7  14553   8098    0.00     3     3     2     1     1     0  0.0000  1.0000  913    0    0.42548162    5.50    2   9.00    6  0.0
```

Figure 5.31 Example of vehicle.dat.

- **Vehicle Route Data (path.dat)**
  This file describes the node-to-node path of the trip roster, as shown in Table 5.37 and, as shown for each value of *path.dat*, in Table 5.38.

Table 5.37 File Format of path.dat

Path Data (for all vehicles)		
Node 1	…	Last node

Table 5.38 Value Descriptions of path.dat

Path Data (for all vehicles)	
Node 1	Start node for vehicle's route.
…	
Last Node	Last node of vehicle's route.

File example is illustrated in Figure 5.32.

```
 199   116    19    23    28    32    35    39    44    45    49    55    58    62   117
  17    79   117    64    60    53    52    48    41    37    34    30    25    21   116
  61   117    64    60    53    52    48    41    37    34    30    25    21   116
  78    79   117    64    60    53    52    48    41    37    34    30    25    21   116
  79    17    61   117    64    60    53    52    48    41    37    34    30    25    21   116
  79   117    64    60    53    52    48    41    37    34    30    25    21   116
 115   117    64    60    53    52    48    41    37    34    30    25    21   116
 115   173    18    59    16    54    53    52    48    41    37    34    30    25    21   116
```

Figure 5.32 Example of path.dat.

- **Advanced Parameter Data (parameter.dat)**
  This file, as indicated in Table 5.39, describes advanced parameters pertaining to both the simulation and assignment. This is a unique file in which advanced settings can be adjusted. It is important to note this file is required, but all values are set at default.

Table 5.39 Value Descriptions of parameter.dat

Advanced Parameter Data	
800	SIR Freeway Length.
500	SIR Arterial Length.
500	SIR On-Ramp Length.
500	SIR Off-Ramp Length.
1.00	(basepce) Reduce PCE during 1st half of iterations. Default: 1.0. Reducing the parameter allows the model to be less likely to be gridlocked in the initial interactions in a highly congested network.
8000	(entrymx) Maximum entry volume rate per lane-mile.
0	(vehjo) Interval for vehicle position info. 0 to number of iteration. Obsolete.
1.00	(multf) Factor to adjust multifactor in demand.dat. Default: 1.0. This is for a quick uniform adjustment of demand.
1	(i18) 1: vehtrajectory.dat written at the VehJO interval, 0: no.
1	(amsflag) 0: original meso; 1: AMS.
0	(asgflag) 0: keeps the initial path, 1: discards initial path.
0.99	(entrymeter) 1: maximum percentage of capacity filled by newly generated vehicles. Obsolete.
0	(netcheck) 1: check network connectivity, 0: no.
1.00	(vhpcereturn) Ratio of number ite reduced mtnum will be restored.
1	(jamswitch) 1: activate jam switch rule; 0: no.
0	(jamswbar) Threshold for switch (% of jam density).
0	(jampar) Jam switch participation rate.
1000000	(itersub).
1.05	(gencheck).
0	(keepcls). Keep vehicle class in vehicle.dat. 0: new class, 1: original.
100000	(tmparysize) Size of the temp arrays used in read vehicle subroutine.
1.05	(nv_vebuffer) Ratio of allocated vehicle array size to the estimated # of vehicles.
0	(logout) Flag to write debugging info to runlog.dat. 1: write.
0	(trajalt) Flag to write AltVeh.dat, AltPath.dat, AltEnQ.dat, AltTime.dat.
0	(IncClsFlag) 0: capacity reduction, 1: closure upstream node.
200	Average bay length (feet).
1700	Left-turn saturation flow rate.
1700	Right-turn saturation flow rate.
5.0	Critical gap for left-turn vehicles (seconds).
2.0	Scaling factor for GFV assignment approach.
0.5	Maximum factor for each iteration.
1	0: MSA; 1: GFV.
0	1: write UE Traveltime.
0	1: read UE Traveltime.
0	0: no shelter.dat to be used; 1: with shelter.
4	Thread.
0	Write fuel consumption.
0	Demand format. 0: old 6-column format.
0	Read Transit flag. 0: no read.

0	0: do nothing, 1: generate transit from TransitRouteSchedule.dat.
0	Traffic flow model switch. 0: old format, 1: new format with 3 number in 1st line of each model.
0	Read habitual path node arrival time. This activates the diversion if vehicle is delayed when comparing with the baseline case.
0	DLLFlagRM. 0: Use static library; 1: Fortran DLL; 2: CDLL.
15.0	Delay tolerance mean (minutes).
2.0	Delay tolerance std (minutes).
1000	# of simulated intervals without vehicle existing before simulation stops.
300	(maxnu_pa) Size of temporary vehicle path array.
0	Capacity aware.
120	Start time of pretrip and en route information.
1.00	Percentage (100) willing to divert to transit.
0	Peak spread modeling switch. 0: no departure time choice, 1: with departure time choice.
0	# of departure time choice iterations.
0	Skim output switch. 0: no skim table, 1: with skim table.
30	Time period (minutes) over which skim is reported.
0	Output switch for mixed vehicle generation id mapping.
0	Switch for MOVES outputs. 0: no output, 1: output MOVES related outputs.
1	(VeFormat) 0: old format for NEXTA, 1: new format for DynuStudio.
300	Vehicle array increase step size.
0	Reliability flag.

Note: SIR = Speed Influence Region; PCE = parameter control entry; GFV = gap function vehicle; MSA = method of successive averages; UE = user equilibrium.

File example is illustrated in Figure 5.33**.**

800 =   Freeway
500 =   Arterials
500 =   On ramp
500 =   off ramp
Reduce PCE during 1st half of iterations. DF: 1.0
80000 =   Maximum entry volume rate per lane-mile
0 =   Interval for vehicle position info. 0 to number of iteration
1.0 :   Factor to adjust multifactor in demand.dat. Default: 1.0
1 =   1 - vehtrajectory.dat written at the VehJO interval, 0: no
1 =   0 - Original meso: 1- AMS
0 =   0 - Keeps the initial path, 1- discard initial path
0.99 :   1 – Max Percent of capacity filled by newly gen vehicles.
0 =   1 - Check network connectivity. 1: check, 0: no
1.0   Ratio of number ite reduced mtnum will be restored.
0 =   1 - Activate jam switch rule; 0: no
0.95 :   Threshold for switch (% of jam density)
1.00 :   Jam switch participation rate
1 =   IterSub
0 =   GenCheck
1 =   keepcls. keepvehcls in vehicle.dat. 0: new class, 1: original
1 =   keeptype. keepvehcls in vehicle.dat. 0: new class, 1: original
1000000 = tmparysize - size of the temp arrays
1.05 :   nv_vebuffer
0 =   logout - flag to write to runlog.dat. 1: write
0 =   trajalt - flag to write AltVeh.dat, AltPath.dat, AltEnQ.dat, AltTime.dat
0 =   IncClsFlag. 0: capacity reduction, 1: closure upstream node
200 =   Average bay length
1800 =   Left-turn saturation flow rate
1800 =   Right-turn saturation flow rate
5.00 :   critical gap for left-turn vehicles (sec)
2.00 :   Scaling factor for GFV assignment approach
0.50 :   max factor for each iteration
1 =   0 = MSA,1 = GFV
0 =   1: write UE Traveltime
0 =   1: read UE Traveltime
0 =   0: no shelter.dat to be used. 1: with shelter
8 =   Thread
0 =   Write fuel consumption
0 =   Demand format. 0: old 6-column format
0 =   Read Transit flag. 0: no read
0 =   1: Generate Transit from TransitRouteSchedule.dat
0 =   Traffic flow model switch. 0: old format. 1: new format with 3 numbers in 1st line of each model
0 =   Read habitual path node arrival time.
0 =   DLLFlagRM. 0: Use static library; 1: Fortran DLL; 2: CDLL
15.00 :   Delay tolerance mean (min)

2.00 :	Delay tolerance std (min)
1000 =	# of sim intervals without vehicle existing before sim stops
300 =	(maxnu_pa) size of temporary veh path array
0 =	Capacity Aware
122.00 :	Start Time of Pretrip and en route information
1.00 :	percent (100) willing to divert to transit
0 =	Peak spread modeling switch 0: no departure time choice, 1: with departure time choice
0 =	# of departure time choice iterations
0 =	skim output switch: 0: no skim table, 1: with skim table
30 =	Time period (min) over which skim is reported
0 =	output switch for mixed vehicle generation id mapping
0 =	Switch for MOVES outputs. 0: no output, 1: output MOVES related outputs
1 =	VehFormat
200 =	VehArray increase Size
0 =	Reliability Flag

Figure 5.33 Example of system.dat.

5.2.2 Output Files

- **Link-Based Queue Data (fort.600)**
  This file describes for each link in the network and for every simulation minute the average fraction of the link length having a queue. The file format for this data is shown in Table 5.40.

Table 5.40 File Format of fort.600

Time Stamp				
Simulation Minute				
Average Density Data (for all links)				
Link 1 Queue	Link 2 Queue	Link 3 Queue	…….	Link 10 Queue
.				
.				
.				
.				
Last Link Queue				

The same file format (Table 5.40) is repeated at every simulation minute completed. Please note, the number of simulated minutes reported may not equal the total simulation time specified in Section 5.2.1 concerning the Simulation and

Assignment Configuration Data, but this would occur only if all vehicles exit the network before the specified simulation time.

The average queue data is reported for every link. The format of the file reports the link data at 10 links per row in sequential order. The link order corresponds to the link ordering in the Network Data file discussed in Section 5.2.1.

File example is illustrated in Figure 5.34.

```
                                                  1.0
0.01006   0.00000   0.00000   0.00000   0.00000   0.00000   0.00235   0.00000   0.00000   0.00469
0.00000   0.00000   0.00000   0.00000   0.00146   0.00000   0.00000   0.00000   0.00305   0.00000
0.00000   0.00990   0.00000   0.00364   0.00660   0.00000   0.00000   0.00000   0.00000   0.00000
0.00220   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000
0.00000   0.00000   0.00182   0.00000   0.00078   0.00000   0.00000   0.02933   0.01650   0.00000
0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000
0.00000   0.00000   0.00235   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000
0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000
0.00503   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000
0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000
0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000
0.00000   0.00000   0.00313   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000
0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00114   0.00000
0.00000   0.00000   0.00109   0.00000   0.00000   0.00000   0.00821   0.00000   0.00000   0.00000
0.00183   0.00000   0.00000   0.00122   0.00000   0.00394   0.00000   0.00000   0.00000   0.00000
0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000
0.00078   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00048   0.00000
0.00000   0.00120   0.00084   0.00000   0.00000   0.00125   0.00210   0.00000   0.00000   0.00000
0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000
0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000
0.00000   0.00061   0.00000   0.00000   0.00000   0.00075   0.00000   0.00132   0.00000   0.00000
0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000
0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000
0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000
0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000
0.00000   0.00000   0.00000   0.00000   0.00660   0.00000   0.00000   0.00000   0.01242   0.00000
0.00000   0.00207   0.00000   0.00000   0.00000   0.00000   0.00000   0.00253   0.00000   0.00000
0.00000   0.00152   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000
0.00000   0.00000   0.00033   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000
0.00000   0.00000   0.00000   0.00000   0.00000   0.00029   0.00000   0.00000   0.00000   0.00000
```

Figure 5.34 Example of fort.600.

- **Link-Based Density Data (fort.700)**

  This file describes the average density [passenger car (pc)/mile/lane] of each link in the network for every simulation minute. The format is shown in Table 5.41.

Table 5.41 File Format of fort.700

Time Stamp				
Simulation Minute				
Average Density Data (for all links)				
Link 1 Density	Link 2 Density	Link 3 Density	…….	Link 10 Density
.				
.				
.				
.				
Last Link Density				

The same file format (Table 5.41) is repeated at every simulation minute completed. Please note that the number of simulated minutes reported may not be equal to the total simulation time specified in Section 5.2.1 concerning the Simulation and Assignment Configuration Data, which would occur only if all vehicles exit the network before the specified simulation time.

The average density data are reported for every link. The format of the file reports the link data at 10 links per row in sequential order. The link order corresponds to the link ordering in the Network Data file discussed in Section 5.2.1.

File example is illustrated in Figure 5.35.

```
                                                     1.0
3.77143    0.00000    0.00000    0.00000    2.43404    0.66000    1.17333    1.76000    0.00000    2.34667
0.00000    2.70769    0.82500    0.50286    1.09241    0.00000    0.00000    0.69474    1.52284    1.32000
0.00000    2.47500    0.00000    2.73103    3.30000    0.10173    0.00000    0.00000    1.18209    0.00000
1.10000    0.00000    0.37676    0.00000    1.11158    0.55000    1.65000    0.97059    0.27863    1.52284
2.09000    0.33000    5.00690    0.16000    0.19556    0.00000    1.10000   10.47619    6.60000    0.40615
2.03680    0.00000    2.35978    2.40000    0.00000    2.64000    0.00000    0.00000    0.00000    0.00000
0.00000    0.49759    1.76000    0.00000    0.00000    0.00000    0.00000    0.00000    0.00000    0.00000
0.00000    0.00000    1.50857    1.34118    0.11379    0.50286    0.00000    0.00000    0.00000    0.00000
2.51429    0.00000    0.62118    1.68960    0.00000    0.00000    0.46588    0.00000    0.00000    0.00000
0.96000    0.00000    0.00000    0.00000    0.00000    0.00000    0.50761    0.46588    0.00000    0.00000
0.00000    0.00000    0.91277    2.34667    0.00000    0.00000    0.00000    0.35967    1.60000    0.00000
0.00000    0.00000    1.95556    0.00000    0.00000    0.61395    0.16500    0.00000    1.66737    3.20262
0.21120    2.03077    1.12000    0.00000    0.35200    0.74894    0.00000    0.00000    0.42581    0.00000
0.00000    0.00000    0.95751    0.00000    2.64000    1.64145    3.52000    1.82069    0.00000    1.17333
1.06821    0.00000    0.52800    1.67861    0.37358    3.93540    0.00000    0.18679    0.30520    1.02857
0.00000    0.21529    0.76301    0.00000    0.00000    0.80734    0.00000    0.41250    0.00000    1.44375
0.39111    0.00000    0.00000    1.81108    0.78222    0.45517    0.00000    1.36552    0.47891    0.00000
0.00000    1.07755    3.05020    1.10000    0.00000    2.82018    2.10359    1.93731    0.00000    0.45517
1.33109    0.27358    1.76000    0.48293    1.32663    0.00000    0.00000    1.23750    0.79598    0.00000
0.57391    0.00000    1.72174    0.00000    0.22957    0.00000    0.00000    0.00000    0.00000    0.00000
0.00000    0.00000    0.41358    0.00000    0.00000    0.00000    0.00000    0.00000    0.00000    0.32000
1.65000    0.00000    1.44000    0.36414    0.68870    0.00000    0.00000    0.00000    0.00000    0.00000
0.00000    0.82715    0.00000    0.00000    0.00000    0.00000    0.80000    0.00000    0.00000    1.33333
0.00000    0.00000    0.00000    0.00000    0.00000    0.00000    0.00000    0.00000    0.00000    0.00000
0.00000    0.00000    1.11158    2.50105    0.00000    0.00000    0.00000    9.60000    0.00000    2.46799
2.00000    0.00000    1.21846    1.52308    0.00000    0.68870    0.00000    0.00000    0.00000    0.00000
```

Figure 5.35 Example of fort.700.

- **Link-Based Speed Data (fort.900)**

  This file describes the average speed (mph) of each link in the network for every simulation minute (see Table 5.42).

Table 5.42 File Format of fort.900

Time Stamp				
Simulation Minute				
Average Speed Data (for all links)				
Link 1 Speed	Link 2 Speed	Link 3 Speed	…….	Link 10 Speed
.				
.				
.				
.				
Last Link Speed				

The same file format in Table 5.42 is repeated at every simulation minute completed. Please note that the number of simulated minutes reported may not be equal to the total simulation time specified in Section 5.2.1 concerning the Simulation and Assignment Configuration Data, which would occur only if all vehicles exit the network before the specified simulation time.

The average speed data is reported for every link. The format of the file reports the link data at 10 links per row in sequential order. The link order corresponds to the link ordering in the Network Data file discussed in Section 5.2.1 above.

File example is illustrated in Figure 5.36.

```
                                                       1.0
31.01205   40.00000   40.00000   40.00000   39.67887   40.00000   36.50000   39.89902   40.00000   33.00000
40.00000   40.00000   40.00000   40.00000   32.99999   40.00000   40.00000   40.00000   37.97157   40.00000
40.00000   33.91470   40.00000   36.50000   37.97157   40.00000   40.00000   40.00000   39.95392   40.00000
37.97157   40.00000   39.99999   40.00000   40.00000   40.00000   39.76263   39.82583   40.00000   40.00000
39.75499   30.00000   38.54696   40.00000   40.00000   40.00000   39.74342   34.92892   38.57190   40.00000
27.46606   40.00000   39.90800   39.88504   40.00000   64.99999   40.00000   64.99999   40.00000   40.00000
40.00000   64.99999   36.50000   64.99999   40.00000   40.00000   40.00000   40.00000   64.99999   40.00000
64.99999   40.00000   40.00000   39.74231   64.99999   40.00000   64.99999   40.00000   40.00000   64.99999
34.56585   64.99999   39.99999   40.00000   40.00000   64.99999   36.50000   40.00000   64.99999   40.00000
40.00000   40.00000   40.00000   64.99999   40.00000   64.99999   40.00000   40.00000   64.99999   40.00000
40.00000   64.99999   39.79282   39.99999   64.99999   40.00000   64.99999   39.99999   40.00000   40.00000
64.99999   40.00000   33.00000   40.00000   64.99999   39.83086   64.99999   40.00000   39.99999   38.85715
64.99999   39.90784   64.99999   40.00000   40.00000   40.00000   40.00000   40.00000   33.00000   30.00000
40.00000   40.00000   39.89330   30.00000   40.00000   39.53972   27.87574   40.00000   30.00000   39.90784
```

Figure 5.36 Example of fort.900.

- **Time-Based Network Statistics Data (fort.800)**

  This file as shown in Table 5.43 describes the average density (pc/mile/lane) of each link in the network for every simulation minute.

Table 5.43 File Format of fort.800

Network Statistics (for every simulation minute)
Time Stamp
# of vehicles generated
# of vehicles out of network
# of vehicles in the network
Average travel time for all vehicles
Average travel time for vehicles out of network
Average speed for freeway links
Average speed for arterial links

The same file format, shown in Table 5.43, is repeated at every simulation minute completed. Please note that the number of simulated minutes reported may not be equal to the total simulation time specified in Section 5.2.1 concerning the Simulation and Assignment Configuration Data, which would occur only if all vehicles exit the network before the specified simulation time.

File example is illustrated in Figure 5.37.

```
==============================
Current Time               :       1.0
Total # of Gen Vehs        :      2132
Total # of Out Vehs        :      1748
Total # of In Vehs         :       384
Avg Travel Time All Vehs   :       0.0
Avg Travel Time for out Veh:       0.0
Avg speed for all links    :      38.8
Avg speed for freeways     :      65.0
Avg speed for arterials    :      37.0
==============================
Current Time               :       2.0
Total # of Gen Vehs        :      2132
Total # of Out Vehs        :      1350
Total # of In Vehs         :       782
Avg Travel Time All Vehs   :       0.0
Avg Travel Time for out Veh:       0.0
Avg speed for all links    :      37.4
Avg speed for freeways     :      64.9
Avg speed for arterials    :      35.5
==============================
Current Time               :       3.0
Total # of Gen Vehs        :      2132
Total # of Out Vehs        :       949
Total # of In Vehs         :      1183
Avg Travel Time All Vehs   :       0.0
Avg Travel Time for out Veh:       0.1
Avg speed for all links    :      37.4
Avg speed for freeways     :      64.9
                   Avg speed for arterials    :      35.6
```

Figure 5.37 Example of fort.800 (gen = generated).

- **Network-Level Summary Statistics (SummaryStat.dat)**
  This file describes the general network and simulation information generated by the DynusT run. The summary file can be categorized in three major segments, which are shown in Table 5.44.

Table 5.44 File Segments of SummaryStat.dat

Network, Simulation, Assignment Settings
Network Data
Control Data
Ramp Data
Solution Mode
Time Periods
Congestion Pricing
Vehicle Loading Mode
Traffic Management Strategies
Capacity Reduction
Simulation Loading Information
Time Stamp
Accumulated Generated Vehicles
Generated Vehicles
Nontagged Vehicles That Exit the Network
Tagged Vehicles That Exit the Network
Vehicles Still in Network
Network Simulation Statistics Report (Total and Average)
Travel Time
Trip Time
Entry Queue Time
Stop Time
Trip Distance

File example is illustrated in Figure 5.38.

```
==========================================================
==========================================================
H                        DynusT                          H
H                                                        H
H       Dynamic Urban Systems for Transportation         H
H                                                        H
H                    Version (2012)                      H
H                     Build    419                       H
H                                                        H
H                                                        H
H           Released by: University of Arizona           H
H             Copyright: Yi-Chang Chiu, Ph.D.            H
H                                                        H
H               Release Date: November, 2012             H
H                                                        H
==========================================================
==========================================================
```

```
          ****************************************
          *      Basic Information               *
          ****************************************

                       NETWORK DATA
                       ------------
      Number of Nodes                          :     180
      Number of Links                          :     445
      Number of Zones                          :      13
          ***************************************

                 INTERSECTION CONTROL DATA
                 -------------------------
      Number of No Control                     :      87
      Number of Yield Signs                    :       0
      Number of 4-Way STOP Signs               :      31
      Number of 2-Way STOP Signs               :       1
      Number of Pretimed Control               :       0
      Number of Actuated Control               :      61
          ***************************************

                        RAMP DATA
                        ---------
      Number of Ramp Control                   :       0
      Number of VMS Control                    :       0
          ***************************************

                      SOLUTION MODE
                      -------------
     Execute Iterative Consistency Algorithm(Equlilbrium)
      Max. Number of Iterations                :      20
      Current Iteration                        :       0
          ***************************************

                       TIME PERIODS
                       ------------
     Planning Horizon(min)                    :     180.0
       Aggregation Interval(# of SimInt)       :      50
       Assignment Interval(# of SimInt)        :      50
      Max # of Iterations                      :      20
   mtc Threshold (# of Vehicles)             :      1.0
        Convergence Threshold - Gap Function %   :  1.00
          ***************************************

        FOR CONGESTION PRICING DETAILS CHECK TOLL.DAT
                    ------------------

                   VEHICLE LOADING MODE
                   --------------------
                      O-D Demand Table

 Tripmakers perceive freeway travel time to be 20.0 % lower than arterials

          ***************************************

                 TRAFFIC MANAGEMENT STRATEGIES
                 -----------------------------
           No Traffic Management Strategy Was Specified
          ***************************************

                    CAPACITY REDUCTION
                    ------------------
           No Capacity Reduction Scenario Was Specified

          ****************************************
          *      Loading Information             *
          ****************************************

T:   5.0 Tot Veh:      2132 Gen:    2132 Out_n:       0 Out_t:    227 In_v:   1905
T:  10.0 Tot Veh:      4346 Gen:    2214 Out_n:       0 Out_t:   1195 In_v:   2924
T:  15.0 Tot Veh:      6594 Gen:    2248 Out_n:       0 Out_t:   1604 In_v:   3568
```

```
T:  20.0 Tot Veh:      8838 Gen:    2244 Out_n:       0 Out_t:   1617 In_v:   4195
T:  25.0 Tot Veh:     11128 Gen:    2290 Out_n:       0 Out_t:   1653 In_v:   4832
T:  30.0 Tot Veh:     13415 Gen:    2287 Out_n:       0 Out_t:   1763 In_v:   5356
T:  35.0 Tot Veh:     15715 Gen:    2300 Out_n:       0 Out_t:   1850 In_v:   5806
T:  40.0 Tot Veh:     18080 Gen:    2365 Out_n:       0 Out_t:   1848 In_v:   6323
T:  45.0 Tot Veh:     20436 Gen:    2356 Out_n:       0 Out_t:   1886 In_v:   6793
T:  50.0 Tot Veh:     22811 Gen:    2375 Out_n:       0 Out_t:   1867 In_v:   7301
T:  55.0 Tot Veh:     25219 Gen:    2408 Out_n:       0 Out_t:   1880 In_v:   7829
T:  60.0 Tot Veh:     27571 Gen:    2352 Out_n:       0 Out_t:   1925 In_v:   8256
T:  65.0 Tot Veh:     30054 Gen:    2483 Out_n:       0 Out_t:   1897 In_v:   8842
T:  70.0 Tot Veh:     32555 Gen:    2501 Out_n:       0 Out_t:   2030 In_v:   9313
T:  75.0 Tot Veh:     35115 Gen:    2560 Out_n:       0 Out_t:   2132 In_v:   9741
T:  80.0 Tot Veh:     37556 Gen:    2441 Out_n:       0 Out_t:   2115 In_v:  10067
T:  85.0 Tot Veh:     40118 Gen:    2562 Out_n:       0 Out_t:   2097 In_v:  10532
T:  90.0 Tot Veh:     42640 Gen:    2522 Out_n:       0 Out_t:   1942 In_v:  11112
T:  95.0 Tot Veh:     45235 Gen:    2595 Out_n:       0 Out_t:   2008 In_v:  11699
T: 100.0 Tot Veh:     47853 Gen:    2618 Out_n:       0 Out_t:   1907 In_v:  12410
T: 105.0 Tot Veh:     50504 Gen:    2651 Out_n:       0 Out_t:   1978 In_v:  13083
T: 110.0 Tot Veh:     53149 Gen:    2645 Out_n:       0 Out_t:   1974 In_v:  13754
T: 115.0 Tot Veh:     55832 Gen:    2683 Out_n:       0 Out_t:   1889 In_v:  14548
T: 120.0 Tot Veh:     58610 Gen:    2778 Out_n:       0 Out_t:   1851 In_v:  15475
                      ***************************************

                          VEHICLE TYPE PERCENTAGES
                          ------------------------
           PC                                    :      47127    80.41 %
           TRUCK                                 :      11483    19.59 %
           HOV                                   :          0     0.00 %
           BUS                                   :          0     0.00 %
                      ***************************************

                           MTC CLASS PERCENTAGES
                           ---------------------
         Historical/Habitual                   :          0     0.00 %
           System Optimal                        :          0     0.00 %
           User Optimal                          :      58610   100.00 %
           Enroute Information                   :          0     0.00 %
           Pre-Trip Information                  :          0     0.00 %
        NOTE : There are        4078   target vehicles still in the network

                        ******* VEHICLE INFORMATION *******
                          TOTAL VEHICLES        :        58610
                          NON-TAGGED VEHICLES   :            0
                          TAGGED VEHICLES (IN)  :         4078
                          TAGGED VEHICLES (OUT) :        58610
                          OTHERS                :            0

           ******* HOT LANE(S) INFORMATION SEE TOLLREVENUE.DAT **********

                      ***************************************
                      *  OVERALL STATISTICS REPORT          *
                      ***************************************
               Iteration Number                        :          0
               Max Simulation Time (min)               :      180.0
                Actual Sim. Intervals                   :     1800
               Simulation Time     (min)               :      180.0
               Start Time in Which Veh Stat are Collected :        0.0
               End   Time in Which Veh Stat are Collected :      180.0
                Total Number of Vehicles of Interest      :    58610
                                 With    Info           :        0
                                 Without Info           :    58610
                  ------------------------------------------------
                          TOTAL TRAVEL TIMES (HRS)
                          OVERALL    :     16556.9883
                          NOINFO     :     16556.9883
                          1 STOP     :     16556.9883
                          2 STOPs    :         0.0000
                          3 STOPs    :         0.0000
                          INFO       :         0.0000
                          1 STOP     :         0.0000
                          2 STOPs    :         0.0000
```

```
     3 STOPs    :         0.0000

   AVERAGE TRAVEL TIMES (MINS)
     OVERALL    :        16.9497
     NOINFO     :        16.9497
     1 STOP     :        16.9497
     2 STOPs    :         0.0000
     3 STOPs    :         0.0000
     INFO       :         0.0000
     1 STOP     :         0.0000
     2 STOPs    :         0.0000
     3 STOPs    :         0.0000

 ------------------------------------------------
TOTAL TRIP TIMES (INCLUDING ENTRY QUEUE TIME) (HRS)
     OVERALL    :     24675.3379
     NOINFO     :     24675.3379
     1 STOP     :     24675.3379
     2 STOPs    :         0.0000
     3 STOPs    :         0.0000
     INFO       :         0.0000
     1 STOP     :         0.0000
     2 STOPs    :         0.0000
     3 STOPs    :         0.0000
AVERAGE TRIP TIMES (INCLUDING ENTRY QUEUE TIME) (MINS)
     OVERALL    :        25.2605
     NOINFO     :        25.2605
     1 STOP     :        25.2605
     2 STOPs    :         0.0000
     3 STOPs    :         0.0000
     INFO       :         0.0000
     1 STOP     :         0.0000
     2 STOPs    :         0.0000
     3 STOPs    :         0.0000

 ------------------------------------------------
   TOTAL ENTRY QUEUE TIMES (HRS)
     OVERALL    :      8118.3286
     NOINFO     :      8118.3286
     1 STOP     :      8118.3286
     2 STOPs    :         0.0000
     3 STOPs    :         0.0000
     INFO       :         0.0000
     1 STOP     :         0.0000
     2 STOPs    :         0.0000
     3 STOPs    :         0.0000
  AVERAGE ENTRY QUEUE TIMES (MINS)
     OVERALL    :         8.3109
     NOINFO     :         8.3109
     1 STOP     :         8.3109
     2 STOPs    :         0.0000
     3 STOPs    :         0.0000
     INFO       :         0.0000
     1 STOP     :         0.0000
     2 STOPs    :         0.0000
     3 STOPs    :         0.0000

 -------------------------------------------------
     TOTAL STOP TIME ( HRS )
     OVERALL    :     13519.8711
     NOINFO     :     13519.8711
     1 STOP     :     13519.8711
     2 STOPs    :         0.0000
     3 STOPs    :         0.0000
     INFO       :         0.0000
     1 STOP     :         0.0000
     2 STOPs    :         0.0000
     3 STOPs    :         0.0000
    AVERAGE STOP TIME ( MINS )
     OVERALL    :        13.8405
     NOINFO     :        13.8405
```

```
           1 STOP     :        13.8405
           2 STOPs    :         0.0000
           3 STOPs    :         0.0000
           INFO       :         0.0000
           1 STOP     :         0.0000
           2 STOPs    :         0.0000
           3 STOPs    :         0.0000

-------------------------------------------------
          TOTAL TRIP DISTANCE ( MILES )
           OVERALL    :    280844.8125
           NOINFO     :    280844.8125
           1 STOP     :    280844.8125
           2 STOPs    :         0.0000
           3 STOPs    :         0.0000
           INFO       :         0.0000
           1 STOP     :         0.0000
           2 STOPs    :         0.0000
           3 STOPs    :         0.0000
         AVERAGE TRIP DISTANCE ( MILES )
           OVERALL    :         4.7918
           NOINFO     :         4.7918
           1 STOP     :         4.7918
           2 STOPs    :         0.0000
           3 STOPs    :         0.0000
           INFO       :         0.0000
           1 STOP     :         0.0000
           2 STOPs    :         0.0000
           3 STOPs    :         0.0000

-------------------------------------------------
```

Figure 5.38 Example of SummaryStat.dat (MTC = multiple traveler classes.

The first major segment in Table 5.44 concerns network, simulation, and assignment settings; these provide information on how the particular DynusT run is configured. Network data describe number of nodes, links, and zones. Control data describe intersection control data for each of the 6 node-control classifications. Ramp data detail the ramp metering configurations for each link coded with ramp metering in the network. Solution mode describes the solution method of the current run as either one-shot or iterative. Time periods provide the various settings of time intervals within DynusT, as well as the number of iterations and the convergence threshold. Congestion pricing details the costs of traveling in single-occupancy vehicles (SOVs) and HOV vehicles traveling high-occupancy toll (HOT) links, as well as value of time. Vehicle loading mode provides information on the type of vehicle generation from O-D table, vehicle + path file, or vehicle file only. Traffic management strategies present details of the locations of variable message signs (VMSs) and configurations for each link coded with VMSs in the network. Capacity reduction refers to links coded with incident and work zone circumstances, including all capacity reduction configurations.

The second major segment is simulation loading information; this segment provides the information of vehicle generation and exiting throughout the simulation time. The time stamp is the time in the simulation in which the simulation loading information is reported. Accumulated generated vehicles provide the number of vehicles generated in the network thus far, while generated vehicles provide the number of vehicles generated at the current time stamp. Nontagged vehicles are vehicles that do not have recorded characteristics from simulation, while tagged vehicles have recorded characteristics. Vehicles still in the network are those that are still traveling in the network and have not reached their destination. Last, the third segment is the network simulation statistics report, which provides the various traffic measures for both the total network value and the network average.

- **Vehicle Trajectory Data (VehTrajectory.dat)**
  This file describes each vehicle simulated in the network and their traveling characteristics, including their traveled route and trajectory toward their final destination. Table 5.45 and Table 5.46 contain details concerning these routes and trajectories.

Table 5.45 File Format of VehTrajectory.dat

Vehicle Information												
Veh #	Tag	OrigZ	DestZ	Class	UstmN	DownN	DestN	STime	Total Travel Time	# of Nodes	VehType	LOO
Node Route												
Node-to-Node Route												
Node Exit Travel Time												
Node-to-Node Accumulated Route Travel Time												
Node Travel Time												
Node-to-Node Travel Time												
Node Delay Time												
Node-to-Node Delay Time												

Table 5.46 Value Descriptions of VehTrajectory.dat

Vehicle Information	
Veh #	Vehicle ID
Tag	Vehicle Tag for traffic characteristics 0 = No Tag 1 = Tag, did not exit network 2 = Tag, exited network
OrigZ	Origin TAZ
DestZ	Destination TAZ
Class	Vehicle Class (refer to Section 5.2.1concerning Model Scenario Data)
UstmN	Upstream node of generation link
DownN	Downstream node of generation link
DestN	Destination node
STime	Starting departure time
Total Travel Time	Total travel time (minutes)
# of Nodes	Number of nodes in experienced route
VehType	Vehicle Type: 1 = SOV 2 = Truck 3 = HOV
LOO	Level of Occupancy
Node Route	
Node IDs of experienced route (must match # of nodes)	
Node Exit Travel Time	
Travel time experienced when exiting each node equaling total travel time	
Node Travel Time	
Node-to-node travel time experienced	
Node Delay Time	
Node-to-node delay time experienced	

File example is illustrated in Figure 5.39.

```
 ****  Output file for vehicles trajectories: FormatCode B ****
       =================================================
        This file provides all the vehicles trajectories

          #############################################
          ######Vehicles still in the network   #######
          #############################################
Veh #     4961 Tag= 1 OrigZ=    2 DestZ=    1 Class= 3 UstmN=    117 DownN=     79 DestN=    199 STime=   11.30
 Total Travel Time=  168.70 # of Nodes=  22 VehType 1 EVAC 0 VOT 20.00 tFlag 0 PrefArrTime    0.0 TripPur   0
                                         IniGas  0.0
       79      17      18      59      16     114      16      54      51      14      47      12      42
                    41      37      34      30      25      21     116       1       2
     1.62    2.80    3.73    4.30    8.94    9.12   15.40   15.59   16.73   16.88   17.70   18.05   18.96
                      19.73   20.18   20.34   20.72   21.34   21.55   23.87  137.05
     1.62    1.18    0.93    0.57    4.64    0.18    6.28    0.19    1.14    0.15    0.82    0.35    0.91
                       0.77    0.45    0.16    0.38    0.62    0.21    2.32  113.18
     0.00    0.00    4.30    4.30   15.47   15.87   24.57   24.57   24.57   24.57   24.57   24.57   24.65
                      24.75   24.75   24.75   24.75   24.75   24.75   25.66  146.96
Veh #     5005 Tag= 1 OrigZ=    2 DestZ=    1 Class= 3 UstmN=    117 DownN=     79 DestN=    199 STime=   11.40
 Total Travel Time=  168.60 # of Nodes=  22 VehType 1 EVAC 0 VOT 20.00 tFlag 0 PrefArrTime    0.0 TripPur   0
                                         IniGas  0.0
       79      17      18      59      16     114      16      54      51      14      47      12      42
                    41      37      34      30      25      21     116       1       2
     1.46    2.62    5.43    6.04    7.61    8.06   11.02   11.19   12.29   12.43   13.21   13.86   14.69
                      15.43   15.90   16.06   16.44   17.06   17.27   19.53  116.97
     1.46    1.16    2.81    0.61    1.57    0.45    2.96    0.17    1.10    0.14    0.78    0.65    0.83
                       0.74    0.47    0.16    0.38    0.62    0.21    2.26   97.44
     0.00    0.00    4.20    4.20   15.40   15.90   24.30   24.30   24.30   24.30   24.30   24.64   24.64
                      24.64   24.64   24.64   24.64   24.64   24.64   25.55  145.87
```

Figure 5.39 Example of VehTrajectory.dat.

- **Vehicle Path Time Data (AltTime.dat, AlTime_Transit.dat, AltTime_Intermodal.dat)**
  As explained in Table 5.47, this file describes the nodal exit time for each path for each vehicle. The same file format provided in Table 5.47 is repeated for each vehicle taking one line. The actual time stamp for each node is the exit time + start time.

Table 5.47 File Format of AltTime.dat

Average Density Data (for all links)					
Vehicle ID	# of nodes – 1	Start time	Exit time for 1st node	Exit time for 2nd node …	Exit time for final node

File example is illustrated in Figure 5.40.

```
     1      1     0.10     1.51
    10      4     1.40     0.01     0.27     0.46     0.65
     3      6     0.70     0.99     1.63     1.99     2.19     2.40     2.95
    24      1     5.20     0.07
    17      6     3.40     1.06     1.45     1.81     2.42     2.90     3.60
    13     10     2.10     0.74     1.21     1.61     1.79     2.64     2.99     3.20     3.40     4.11     5.07
    27      3     6.40     0.13     0.40     0.80
```

Figure 5.40 Example of AltTime.dat.

References

Bowman, J., and M. Bradley. 2006. *SACSIM/05, Activity-Based Travel Forecasting Model for SACOG, Featuring DaySim—the Person Day Activity and Travel Simulator*. Technical Memo Number 10, DaySim05 Documentation. Prepared for Sacramento Area Council of Governments, September 25, 2006. https://www.yumpu.com/en/document/view/18810722/technical-memo-10-daysim05-documentation-jbowmannet. Retrieved May 28, 2014.

Cambridge Systematics, Inc.; Sacramento Area Council of Governments; University of Arizona; University of Illinois, Chicago; Sonoma Technology, Inc., and Fehr and Peers. 2014a. *S2-C10B-RW-1: Dynamic, Integrated Model System: Sacramento-Area Application. Volume 1: Summary Report*. Transportation Research Board of the National Academies, Washington, D.C. http://www.trb.org/Main/Blurbs/170759.aspx. Accessed May 28, 2014.

Cambridge Systematics, Inc.; Sacramento Area Council of Governments; University of Arizona; University of Illinois, Chicago; Sonoma Technology, Inc., and Fehr and Peers. 2014b. *S2-C10B-RW-2: Dynamic, Integrated Model System: Sacramento-Area Application. Volume 2: Network Report*. Transportation Research Board of the National Academies, Washington, D.C. http://www.trb.org/Main/Blurbs/170877.aspx. Accessed May 28, 2014.

Chiu, Y.-C. 2013. DynusT Official Website (http://dynust.net).

Sacramento Area Council of Governments, DKS Associates, Bradley Research and Consulting, and Transportation Systems and Decision Sciences. 2008. *Sacramento Activity-Based Travel Simulation Model (SACSIM07): Model Reference Report* (Draft). https://www.yumpu.com/en/document/view/2327410/sacramento-activity-based-travel-simulation-model-sacsim07. Retrieved May 28, 2014.

U.S. Environmental Protection Agency. 2012. *Motor Vehicle Emission Simulator (MOVES) User Guide for 2010b* , Office of Transportation and Air Quality. http://www.epa.gov/otaq/models/moves/documents/420b12001b.pdf. Retrieved May 28, 2014.