Trajectory Conversion Algorithm Software User Manual

Version 2.2 Build 2

www.its.dot.gov/index.htm

**Draft Report — March 2014  
Publication Number: FHWA-JPO-XX-XXX**



Federal Highways Administration

| **Notice**  This document is disseminated under the sponsorship of the Department of Transportation in the interest of information exchange.  The United States Government assumes no liability for its contents or use thereof.  The U.S. Government is not endorsing any manufacturers, products, or services cited herein and any trade name that may appear in the work has been included only because it is essential to the contents of the work. |
| --- |

Produced by Noblis, Inc.

U.S. Department of Transportation

ITS Joint Program Office

Research and Innovative Technology Administration

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Technical Report Documentation Page** | | | | | | |
| **1. Report No.**  **FHWA-JPO-XX-XXX** | **2. Government Accession No.** | | | **3. Recipient’s Catalog No.** | | |
| **4. Title and Subtitle**  Trajectory Conversion Algorithm Software User Manual Version 2.2 Build 2 | | | | **5. Report Date**  March 24, 2014 | | |
| **6. Performing Organization Code** | | |
| **7. Author(s)**  Taylor Deurbrouck, Richard Glassco, Jim Larkin, and Dr. Karl Wunderlich | | | | **8. Performing Organization Report No.** | | |
| **9. Performing Organization Name And Address**  Noblis  600 Maryland Ave., SW, Suite 755  Washington, DC 20024 | | | | **10. Work Unit No. (TRAIS)** | | |
| **11. Contract or Grant No.** | | |
| **12. Sponsoring Agency Name and Address**  ITS-Joint Program Office  Research and Innovative Technology Administration  1200 New Jersey Avenue, S.E.  Washington, DC 20590 | | | | **13. Type of Report and Period Covered**  Draft Report | | |
| **14. Sponsoring Agency Code**  **HOIT-1** | | |
| **15. Supplementary Notes** | | | | | | |
| **16. Abstract**  The Trajectory Conversion Algorithm (TCA) Software is designed to test different strategies for producing, transmitting, and storing Connected Vehicle information. The TCA reads in and uses vehicle trajectory information or VISSIM output, Roadside Equipment (RSE) location information, cellular region information, event region information, and strategy information to produce a series of snapshots that the vehicle would produce. Vehicles can be equipped to generate and transmit Probe Data Messages (PDMs) and/or Basic Safety Messages (BSMs) which can be transmitted by either Dedicated Short Range Communication (DSRC),cellular or both. | | | | | | |
| **17. Key Words**  Connected Vehicles, Basic Safety Message, Probe Data Message, DSRC, Cellular, V2I, Mobility | | | **18. Distribution Statement** | | | |
| **19. Security Classif. (of this report)**  Unclassified | | **20. Security Classif. (of this page)**  Unclassified | | | **21. No. of Pages**  28 | **22. Price**  **N/A** |
| **Form DOT F 1700.7 (8-72) Reproduction of completed page authorized** | | | | | | |

Table of Contents

[1 Introduction 7](#_Toc383091359)

[1.1 New Features added in Build 2 7](#_Toc383091360)

[2 Installation and Running the TCA 8](#_Toc383091361)

[3 Input Files 9](#_Toc383091362)

[3.1 File Requirements to Run the TCA 9](#_Toc383091363)

[3.2 Control File 9](#_Toc383091364)

[3.2.1 Vehicle Equipage 9](#_Toc383091365)

[3.3 Vehicle Trajectory File 13](#_Toc383091366)

[3.4 VISSIM File 14](#_Toc383091368)

[3.5 RSE Location File 15](#_Toc383091370)

[3.6 Strategy File 16](#_Toc383091371)

[3.7 Regions File 20](#_Toc383091373)

[4 Output Files 26](#_Toc383091374)

[4.1 TCA Input Summary File 26](#_Toc383091375)

[4.2 Transmitted PDMs File 26](#_Toc383091377)

[4.3 All PDM Snapshots File 27](#_Toc383091379)

[4.4 Transmitted BSM File 29](#_Toc383091381)

List of Tables

[Table 3‑1. Symbol Key 7](#_Toc382813035)

[Table 3‑2: Control file fields 7](#_Toc382813036)

[Table 3‑3: Vehicle Trajectory file fields 12](#_Toc382813037)

[Table 3‑4. Required VISSIM file fields 12](#_Toc382813038)

[Table 3‑5: RSE location file fields 13](#_Toc382813039)

[Table 3‑6. Symbol Key 14](#_Toc382813040)

[Table 3‑7: Strategy file fields 14](#_Toc382813041)

[Table 3‑8: Regions file elements 19](#_Toc382813042)

[Table 4‑1. TCA Input Summary file fields 24](#_Toc382813043)

[Table 4‑2: Transmitted PDMs file fields 25](#_Toc382813044)

[Table 4‑3: All PDM snapshots file fields 26](#_Toc382813045)

[Table 4‑4. Transmitted BSM File Fields 27](#_Toc382813046)

List of Figures

[Figure 3‑1: Control input file example using market penetration 11](#_Toc383091384)

[Figure 3‑2. Trajectory file example 12](#_Toc383091385)

[Figure 3‑3: VISSIM input file example 13](#_Toc383091386)

[Figure 3‑4. RSE locations file example 14](#_Toc383091387)

[Figure 3‑5: Strategy input file example 18](#_Toc383091388)

[Figure 3‑6: Example of regions over a simple intersection network 19](#_Toc383091389)

[Figure 3‑7: Example regions input file with one cellular region 21](#_Toc383091390)

[Figure 3‑8: Example region input file using the standard deviation method 22](#_Toc383091391)

[Figure 3‑9: Example regions input file using probability and Poisson distribution methods 23](#_Toc383091392)

[Figure 4‑1. TCA Input Summary file example excerpt 24](#_Toc383091393)

[Figure 4‑2. Transmitted PDM snapshots file example excerpt 25](#_Toc383091394)

[Figure 4‑3. All PDM Snapshots file example excerpt 26](#_Toc383091395)

[Figure 4‑4: Transmitted BSM snapshots file example excerpt 28](#_Toc383091396)

# 

# Introduction

The Trajectory Conversion Algorithm (TCA) Software is designed to test different strategies for producing, transmitting, and storing Connected Vehicle information. The TCA reads in and uses vehicle trajectory information in CSV format or VISSIM output, Roadside Equipment (RSE) location information, cellular region information, event region information, and strategy information to produce a series of snapshots that the vehicle would produce. Vehicles can be equipped to generate and transmit Probe Data Messages (PDMs) and/or Basic Safety Messages (BSMs) which can be transmitted by either Dedicated Short Range Communication (DSRC), cellular or both. The TCA program version 2 Build 2 or 2.2 assumes perfect communication between vehicles and RSEs but future versions of the TCA 2 will include simulated communication disruptions. As soon as a vehicle equipped to transmit via DSRC is within range of an RSE, it will download all of its snapshot information directly without any loss of information. Similarly, if the vehicle is equipped to transmit via cellular, it will download all its snapshot information directly but those snapshots might be lost or delayed due to user-defined loss rate and latency. The TCA was programmed in open source Python programming language (<http://www.python.org>) and is protected under the Apache License Version 2 license agreement (<http://www.apache.org/licenses/LICENSE-2.0>).

## New Features added in Build 2

* Vehicles can be equipped to generate and transmit both PDMs and BSMs simultaneously
* Vehicles can be equipped to communicate via both DSRC and cellular simultaneously
* BSM output incorporates additional Part I and Part II elements which include heading and instantaneous acceleration
* A new input, the regions file, can be included to define cellular areas and/or event regions which allow the user to model probabilistic events such as wipers, external lights, traction control, etc.

# Installation and Running the TCA

To install the TCA you must have Python version 2.7.5 or later installed on your computer, but has not been tested with Python versions 3 and higher. Python is available at no charge from <http://www.python.org/download/releases/2.7.5>. Python runs in Windows, Linux and Mac operating environments and does not have any prerequisites to install.

The TCA also relies on the free external Python libraries Pandas, Numpy, Dateutil, and SciPy. These will need to be installed as well. These external Python libraries can be found at the following locations:

* Pandas - <http://pandas.pydata.org/>
* Numpy - <http://www.numpy.org/>
* Dateutil - <http://labix.org/python-dateutil>
* SciPy - <http://sourceforge.net/projects/scipy/files/scipy/>

Also many Python Windows binary files for the libraries can be downloaded from the Unofficial Windows Binaries for Python Extension Packages Website (<http://www.lfd.uci.edu/~gohlke/pythonlibs/>)

Once Python and the additional libraries are installed, run the TCA using the provided default input files or by placing all input files into the same directory as the TCA program python files. To run the TCA, type at the command prompt the Python executable name (include the path if the Python path is not included in the system PATH variable) and the TCA program file, followed by the name of the control file. Below shows the order of the items on the command prompt:

C:\>*<python directory>*python TCA2.py *<your control file>*

Example:

C:\>C:\python27\python TCA2.py myinput.xml

The TCA may also run without specifying a control file on the command line. If no control file is given, the software will default to TCAinput.xml as the control file in the same directory as the TCA2.py file. To run the TCA without specifying a control file, edit TCAinput.xml to contain the desired control parameters, and then type:

C:\>*<python directory>*python TCA2.py

Example

C:\>C:\python27\python TCA2.py

# Input Files

## File Requirements to Run the TCA

There are five maximum input files to run the TCA which are: a vehicle trajectory CSV or VISSIM vehicle record output file, a XML Control file, a XML Strategy file, a XML Regions file, and a CSV RSE file. At minimum, the TCA requires a vehicle trajectory CSV (VISSIM output file) and a Control file. In order to allow DSRC communication a RSE CSV file would need to be included. In order to change any of the default strategy values a XML Strategy file would need to be included. And lastly, in order to add cellular and/or event regions (wipers, exterior lights, etc.) a XML Region file would need to be included.

## Control File

The Control file is an XML format input file that contains all information about the names of all other input and output files for the TCA. Some elements in the Control file have default values and do not need to be included in the Control file unless the user wants to change the value. Required elements are marked with a symbol (see Table 3‑1) depending on the model type. The Control file has the structure outlined in Table 3‑2 and an example is shown in Figure 3‑1.

### Vehicle Equipage

Vehicles can only be equipped in TCA by using one and only one type of equipage method: penetration percentage, vehicle types, or IDs. First vehicles are assigned to one of three equipage scenarios and then assigned a communication method. The three equipage scenarios are: PDM only, BSM only, or Dual PDM-BSM. The communication methods are: DSRC, cellular, or Dual communication (transmit via DSRC if in range, else via Cellular). See Figure 3‑1 for an example of correct vehicle equipage using market penetration for BSM-only and Dual PDM-BSM equipped vehicles.

Table 3‑1. Symbol Key

| Symbol | Description |
| --- | --- |
| C:\Users\M29565\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\315YD8PJ\MC900441511[1].png | Required for CSV trajectory input files |
| C:\Users\M29565\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\TZ13Q51I\MC900441505[1].png | Required for DSRC communication |
| All | Required for every model |
| No Symbol | Optional element |

Table 3‑2: Control file fields

| Sym. | Enclosing Element | Element | Description | Values |
| --- | --- | --- | --- | --- |
|  | ControlFile | OutputLevel | Integer value determining how much information is output by the TCA:  0=None  1=Just major events (Default)  2=All Snapshot activities  3=Maximum output | Integer |
|  | ControlFile | Title | Title for the scenario | Character String |
|  | ControlFile | Seed | Number used to initialize the random number generator | Integer |
| All | InputFiles/  TrajectoryFile | FileType | Type of file to read into the TCA:  ‘vissim’=VISSIM output file  ‘csv’=Trajectory file | Character String |
| All | InputFiles/  TrajectoryFile | TrajectoryFileName | Name of the trajectory or VISSIM file that has all vehicle trajectory information | Character String |
| C:\Users\M29565\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\315YD8PJ\MC900441511[1].png | InputFiles/  TrajectoryFile/  CSVTrajectoryFileFields | XColumn, YColumn, TimeColumn, IDColumn, SpdColumn | For each of the six variables, specify the column name in the Vehicle Trajectory file where it is found. These fields are not required with VISSIM file input. | Character String |
|  | InputFiles/  TrajectoryFile/  CSVTrajectoryFileFields | AccelColumn | Optional variable to specify the column name where instantaneous acceleration is located. Average acceleration is computed by the TCA by default. Adding instantaneous acceleration allows for braking output to be included with BSMs | Character String |
|  | InputFiles/  TrajectoryFile/  CSVTrajectoryFileFields | TypeColumn | Optional variable to specify the column name in the Vehicle Trajectory file where vehicle type is found | Character String |
| C:\Users\M29565\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\TZ13Q51I\MC900441505[1].png | InputFiles | RSELocationFile | Name of the RSE location file that lists the name and coordinates of every RSE | Character String |
|  | InputFiles | StrategyFile | Name of the XML based strategy file that the TCA will use | Character String |
|  | InputFiles | RegionsFile | Name of the XML based regions file that contains all of the cellular regions and/or event region information | Character String |
|  | EquippedVehicles | PDMMarketPenetration, PDMVehicleTypes, or PDMVehicleIDs | Percentage of all vehicles (0-100), list of types, or list of Vehicle IDs that are equipped to generate and transmit only PDMs | Integer or Character String |
|  | EquippedVehicles | BSMMarketPenetration, BSMVehicleTypes, or BSMVehicleIDs | Percentage of all vehicles (0-100), list of types, or list of Vehicle IDs that are equipped to generate and transmit only BSMs | Integer or Character String |
|  | EquippedVehicles | DualPDMBSMMarketPenetration, DualPDMBSMVehicleTypes, or DualPDMBSMVehicleIDs | Percentage of all vehicles (0-100), list of types, or list of Vehicle IDs that are equipped to generate and transmit both PDMs and BSMs | Integer or Character String |
|  | EquippedVehicles/  PDMEquipped/DSRC | MarketPenetration, VehicleTypes, or VehicleIDs | Percentage of PDM vehicles (0-100), list of types, or list of Vehicle IDs that are equipped to generate and transmit PDMs via DSRC | Integer or Character String |
|  | EquippedVehicles/  PDMEquipped/Cellular | MarketPenetration, VehicleTypes, or VehicleIDs | Percentage of PDM vehicles (0-100), list of types, or list of Vehicle IDs that are equipped to generate and transmit PDMs via Cellular | Integer or Character String |
|  | EquippedVehicles/  PDMEquipped/  DualComm | MarketPenetration, VehicleTypes, or VehicleIDs | Percentage of PDM vehicles (0-100), list of types, or list of Vehicle IDs that are equipped to generate and transmit PDMs via DSRC or Cellular (DualComm) | Integer or Character String |
|  | EquippedVehicles/  BSMEquipped/DSRC | MarketPenetration, VehicleTypes, or VehicleIDs | Percentage of BSM vehicles (0-100), list of types, or list of Vehicle IDs that are equipped to generate and transmit BSMs via DSRC | Integer or Character String |
|  | EquippedVehicles/  BSMEquipped/Cellular | MarketPenetration, VehicleTypes, or VehicleIDs | Percentage of BSM vehicles (0-100), list of types, or list of Vehicle IDs that are equipped to generate and transmit BSMs via Cellular | Integer or Character String |
|  | EquippedVehicles/  BSMEquipped/  DualComm | MarketPenetration, VehicleTypes, or VehicleIDs | Percentage of BSM vehicles (0-100), list of types, or list of Vehicle IDs that are equipped to generate and transmit BSMs via DSRC or Cellular (DualComm) | Integer or Character String |
|  | EquippedVehicles/  PDMBSMEquipped/  DSRC | MarketPenetration, VehicleTypes, or VehicleIDs | Percentage of Dual PDM-BSM vehicles (0-100), list of types, or list of Vehicle IDs that are equipped to generate and transmit both PDMs and BSMs via DSRC | Integer or Character String |
|  | EquippedVehicles/  PDMBSMEquipped/  Cellular | MarketPenetration, VehicleTypes, or VehicleIDs | Percentage of Dual PDM-BSM vehicles (0-100), list of types, or list of Vehicle IDs that are equipped to generate and transmit both PDMs and BSMs via Cellular | Integer or Character String |
|  | EquippedVehicles/  PDMBSMEquipped/  DualComm | MarketPenetration, VehicleTypes, or VehicleIDs | Percentage of Dual PDM-BSM vehicles (0-100), list of types, or list of Vehicle IDs that are equipped to generate and transmit both PDMs and BSMs via DSRC or Cellular (DualComm) | Integer or Character String |
|  | OutputFiles | PDMAllFile | Name of the output file that will contain all of the PDMs generated by the TCA | Character String |
|  | OutputFiles | PDMTransFile | Name of the output file that will contain all of the transmitted PDM information. | Character String |
|  | OutputFiles | BSMTransFile | Name of the output file that will contain all of the transmitted BSM information | Character String |
|  | ControlFile | NumberEquippedVehicles | Maximum number of vehicles expected to be equipped with PDM or BSM capabilities | Integer |

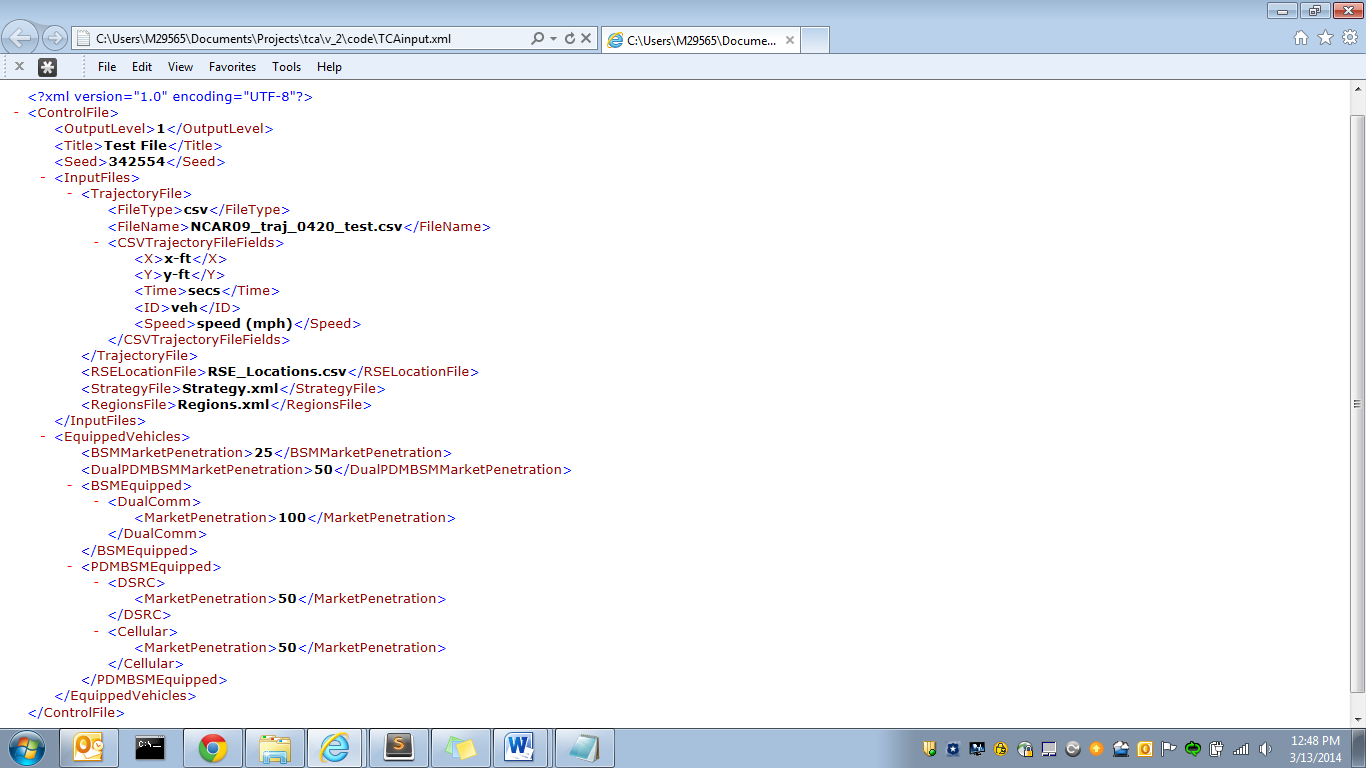


Figure 3‑1: Control input file example using market penetration

## Vehicle Trajectory File

The Vehicle Trajectory file is a comma-delimited file that stores all trajectory information for a number of vehicles. The trajectory points for each vehicle must be in order by time, although trajectories for different vehicles may be intermingled.

The trajectory file must have a column for each of the following variables in any order: vehicle ID, time in seconds from the beginning of the trajectory, speed of the vehicle, and finally the x and y values for the vehicle location. Optional column variables are instantaneous acceleration (average acceleration is calculated by the TCA) and vehicle type. Fields in the input control file specify the column where each of these variables may be found in the vehicle trajectory file. Other columns in the trajectory file will be ignored.

The first line of the vehicle trajectory file contains all header information for the file. The vehicle trajectory file can have the columns located in any order from Table 3‑3. An example of a trajectory file is shown in Figure 3‑2. Also, the sample control file in Figure 3‑1 shows how to correctly specify, for the TCA, where to locate each column of this particular file of trajectory data.

Table 3‑3: Vehicle Trajectory file fields

| Name | Description | Values |
| --- | --- | --- |
| Vehicle ID | Vehicle ID describing the vehicle | Character String |
| Time | Seconds from the beginning of the vehicle trajectory | Integer (seconds) |
| Speed | Speed of the vehicle in mph | Float (mph) |
| x value | X location of the vehicle. These values may be based on longitude | Integer (feet) |
| y value | Y location of the vehicle. These values may be based on latitude | Integer (feet) |
| Acceleration | Instantaneous acceleration of the vehicle (optional) | Float (ft/s2) |
| Type | Type of the vehicle (optional) | Integer |

## 

Figure 3‑2. Trajectory file example

## VISSIM File

Instead of a comma-delimited trajectory data file, the TCA can also read in a standard VISSIM (version 5.40) output file (.fzp file) which is created by VISSIM after running a simulation. This file can be obtained by choosing the option in VISSIM to produce the Vehicle Record Evaluation with the required VISSIM fields from Table 3‑4. For additional information about this file please read the VISSIM user guide. An excerpt from an example VISSIM file is shown in Figure 3‑3.

Table 3‑4. Required VISSIM file fields

| Parameter Name | Description | Values |
| --- | --- | --- |
| Vehicle Number (VehNr) | Number (ID) of the vehicle | Integer |
| Speed [mph] (v) | Speed [mph] at the end of the simulation step | Float (mph) |
| Acceleration [ft/s2] (a) | Acceleration [ft/s²] during the simulation step (optional) | Float (ft/s2) |
| Vehicle Type (Type) | Number of the vehicle type (optional) | Integer |
| Simulation Time (t) | Seconds from the beginning of the simulation | Float (seconds) |
| World Coordinate Front X (WorldX) | World coordinate x (vehicle front end at the end of the simulation step) | Float (feet) |
| World Coordinate Front Y (WorldY) | World coordinate y (vehicle front end at the end of the simulation step) | Float (feet) |

## 

Figure 3‑3: VISSIM input file example

## RSE Location File

The RSE Location file is a comma-delimited file that contains geographical location information for the RSEs. This file is only required if any vehicles will transmit via DSRC.

This file must have a header line and only the fields from Table 3‑5 in the exact order listed are required. An example of an RSE Location File is shown in Figure 3‑4.

Table 3‑5: RSE location file fields

| Column | Name | Description | Value |
| --- | --- | --- | --- |
| 1 | Name | String based name for RSE. This name is output in the TCA snapshots | Character String |
| 2 | X | X location of the RSE in feet | Integer (feet) |
| 3 | Y | Y location of the RSE in feet | Integer (feet) |



Figure 3‑4. RSE locations file example

## Strategy File

The Strategy file is an XML based file that stores all information for controlling how the TCA handles snapshot generation, RSE interaction, buffer management, gap management, and Probe Segment Number (PSN) generation. The Strategy file can be set to run the J2735 standard or several variations of the standard. Every element has a default value so a Strategy file is only necessary if the user wants to change the value of an element. For example, PSN gaps are turned off by default and a Strategy file must be used to turn them on and change the gap parameters if desired (see Figure 3‑5).

Elements that are required for different TCA model variations are noted with their respective symbol. (see). The Strategy Table 3‑6 file has the fields in Table 3‑7and an example is shown in Figure 3‑5.

Table 3‑6. Symbol Key

| Symbol | Description |
| --- | --- |
| C:\Users\M29565\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\315YD8PJ\MC900433804[1].png | Optional element for a DSRC model |
| C:\Users\M29565\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\OZ4HO43N\MC900441494[1].png | Optional element when PSN Gaps are enabled |
| No Symbol | Always Optional |

Table 3‑7: Strategy file fields

| Sym. | Root Element(s) | Element | Description | Value |
| --- | --- | --- | --- | --- |
|  | Strategy | Title | Title of the strategy | Character String |
|  | Inputs/PDM | TimeBeforePDMCollection | The time a vehicle must be active on the network before generating PDM snapshots | Integer (sec) |
|  | Inputs/PDM | DistanceBeforePDMCollection | The distance a vehicle must be active on the network before generating PDM snapshots | Integer (sec) |
| C:\Users\M29565\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\315YD8PJ\MC900433804[1].png  \* | Inputs/  PDM/  PSNStrategy | TimeBetweenPSNSwitches  \*Only required if the element below is not defined | Time between PSN changes | Integer (seconds) |
| C:\Users\M29565\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\315YD8PJ\MC900433804[1].png\*\* | Inputs/  PDM/  PSNStrategy | DistanceBetweenPSNSwitches  \*\*Only required if the element above is not defined | Distance between PSN changes | Integer (feet) |
| C:\Users\M29565\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\315YD8PJ\MC900433804[1].png | Inputs/  PDM/  PSNStrategy | Gap | Gap Setting for TCA:  0-No gaps  1-Gaps on | Integer |
| C:\Users\M29565\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\OZ4HO43N\MC900441494[1].png | Inputs/  PDM/  PSNStrategy/  GapInformation | MinTime | Min time in seconds for random generation of gap | Integer (seconds) |
| C:\Users\M29565\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\OZ4HO43N\MC900441494[1].png | Inputs/  PDM/  PSNStrategy/  GapInformation | MaxTime | Max time in seconds for random generation of gap | Integer (seconds) |
| C:\Users\M29565\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\OZ4HO43N\MC900441494[1].png | Inputs/  PDM/  PSNStrategy/  GapInformation | MinDistance | Min distance in feet for random generation of gap | Integer (feet) |
| C:\Users\M29565\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\OZ4HO43N\MC900441494[1].png | Inputs/  PDM/  PSNStrategy/  GapInformation | MaxDistance | Max distance in feet for random generation of gap | Integer (feet) |
| C:\Users\M29565\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\315YD8PJ\MC900433804[1].png | Inputs/  PDM/  StopStartStrategy | StrategyID | Stop/Start Strategy can be:  1 - Max time and speed (both time and speed trigger start/ stop snapshot)  2 - Max Distance or time (either distance traveled or time motionless triggers start/stop) | Integer |
| C:\Users\M29565\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\315YD8PJ\MC900433804[1].png | Inputs/  PDM/  StopStartStrategy | StopThreshold | Vehicle must be stopped at least this long to create a stop snapshot | Integer (seconds) |
| C:\Users\M29565\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\315YD8PJ\MC900433804[1].png | Inputs/  PDM/  StopStartStrategy | StopLag | Time in seconds that must pass before a 2nd stop snapshot can be taken | Integer (seconds) |
| C:\Users\M29565\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\315YD8PJ\MC900433804[1].png | Inputs/  PDM/  StopStartStrategy | StartThreshold | Speed in mph that a vehicle must have after a stop before a start snapshot can be taken | Integer (mph) |
| C:\Users\M29565\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\315YD8PJ\MC900433804[1].png | Inputs/  PDM/  StopStartStrategy | MultipleStops | Can more than one stop SS in a row be taken, 0-false 1-true | Integer |
| C:\Users\M29565\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\315YD8PJ\MC900433804[1].png | Inputs/  PDM/  PeriodicStrategy | StrategyID | Periodic Strategy can be:  1 - Speed interpolation (Periodic SS taken based on speed value) | Integer |
| C:\Users\M29565\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\315YD8PJ\MC900433804[1].png | Inputs/  PDM/  PeriodicStrategy | LowSpeedThreshold | The time to the next periodic snapshot uses these values. If the vehicle’s speed is below the LowSpeedThreshold, the ShortSpeedinterval is used as the time to the next periodic. If the vehicle’s speed is above the HighSpeedThreshold, the LongSpeedinterval is used. If the speed is between the thresholds, the time is interpolated. | Integer(mph) |
| C:\Users\M29565\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\315YD8PJ\MC900433804[1].png | Inputs/  PDM/  PeriodicStrategy | ShortSpeedInterval | See LowSpeedThreshold | Integer (seconds) |
| C:\Users\M29565\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\315YD8PJ\MC900433804[1].png | Inputs/PDM/ PeriodicStrategy | HighSpeedThreshold | See LowSpeedThreshold | Integer (mph) |
| C:\Users\M29565\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\315YD8PJ\MC900433804[1].png | Inputs/  PDM/  PeriodicStrategy | LongSpeedInterval | See LowSpeedThreshold | Integer (seconds) |
| C:\Users\M29565\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\315YD8PJ\MC900433804[1].png | Inputs/  PDM/  PeriodicStrategy | MaxDeltaSpeed | Percentage change in speed, periodic strategy 2 only | Float |
| C:\Users\M29565\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\315YD8PJ\MC900433804[1].png | Inputs/  PDM/  BufferStrategy | TotalCapacity | Snapshot capacity for the buffer | Integer |
| C:\Users\M29565\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\315YD8PJ\MC900433804[1].png | Inputs/  PDM/  BufferStrategy | SSRetention | Buffer Retention Strategy can be:  1 – FIFO  2 – Every other snapshot  3 – Every other plus keep the first and the last IDs  4 – Every other plus save the oldest SS | Integer |
| C:\Users\M29565\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\315YD8PJ\MC900433804[1].png | Inputs/  DSRC/  RSEInformation | MinRSERange | Minimum range in feet that vehicles can communicate to RSEs | Integer (feet) |
| C:\Users\M29565\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\315YD8PJ\MC900433804[1].png | Inputs/  DSRC/  RSEInformation | MaxRSERange | Maximum range in feet that vehicles can communicate to RSEs | Integer (feet) |
| C:\Users\M29565\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\315YD8PJ\MC900433804[1].png | Inputs/  DSRC/  RSEInformation | TimeoutRSE | Time in seconds that must pass before a vehicle can communicate with an RSE after just communicating with one | Integer (seconds) |
| C:\Users\M29565\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\315YD8PJ\MC900433804[1].png | Inputs/  DSRC/  RSEInformation | MinNumberofPDMtoTransmitViaDSRC | Minimum number of PDMs to transmit | Integer |
|  | Inputs/BSM | BrakeThreshold | Deceleration threshold when brakes are considered to be applied | Float (ft/s2) |

## 

Figure 3‑5: Strategy input file example

## Regions File

The Regions file is an optional XML based input file that describes cellular and/or event regions. By default, any PDMs or BSMs transmitted via cellular will have no latency and 0% loss rate. These defaults can be modified using the Regions file. The user can also add rectangular cellular regions to model areas that have different loss rates or latency.

Event regions could model weather or other events that might, for example, cause a driver to turn on their wipers or for a vehicle to enable their traction control. Event regions are rectangular and defined by the upper left and lower right coordinates. Event regions can contain multiple events and multiple time periods when the events are active.

Each event within a region requires a name, a probability of occurrence, and a recheck value. Event names are automatically included as columns in the BSM output data. Events are rechecked for occurrence as defined by the recheck value. The recheck value is always static except when the user defines a Poisson distribution. A Poisson is one of three possible types of user-defined probability listed below:

1. Standard deviation: user provides a mean value and a standard deviation value and the TCA returns values based on a normal distribution.(Example region event: Air Temp in Figure 3‑8)
2. Standard probability: user defines a probability of occurrence for an event, then TCA determines if the event happens based on the probability and returns either true or false (Example region event: traction control in Figure 3‑9)
3. Poisson distribution: user defines a probability of the event and the TCA uses a Poisson distribution for rechecks (Example region event: wipers in Figure 3‑9)

An example of event regions for a simple intersection network is pictured in Figure 3‑6. This example features three regions: two regions that use a standard deviation to determine the “Air Temp” and one region that contains two events, wipers and traction control, which use standard probability and a Poisson distribution, respectively.



Figure 3‑6: Example of regions over a simple intersection network

Table 3‑8: Regions file elements

| Root Element | Element(s) | Description | Value |
| --- | --- | --- | --- |
| Regions/Cell\_Regions | DefaultLossPercent | The default loss percentage of snapshots transmitted via cellular | Integer |
| Regions/Cell\_Regions | DefaultLatency | The default latency value between snapshot transmission and receive time | Integer (sec) |
| Regions/Cell\_Regions | MinPDMtoTransmit | Minimum number of PDMs in the vehicle buffer necessary for PDM transmission | Integer |
| Regions/Cell\_Regions/  Cell\_Region | Title | Title of the cellular region | Character String |
| Regions/Cell\_Regions/  Cell\_Region/  UpperLeftPoint | X | The x-coordinate of the upper left point of the cellular region rectangle | Float |
| Regions/Cell\_Regions/  Cell\_Region/  UpperLeftPoint | Y | The y-coordinate of the upper left point of the cellular region rectangle | Float |
| Regions/Cell\_Regions/  Cell\_Region/  LowerRightPoint | X | The x-coordinate of the lower right point of the cellular region rectangle | Float |
| Regions/Cell\_Regions/  Cell\_Region/  LowerRightPoint | Y | The y-coordinate of the upper left point of the cellular region rectangle | Float |
| Regions/Cell\_Regions/  Cell\_Region | LossPercent | Loss percentage of messages transmitted within the defined cellular region | Integer |
| Regions/Cell\_Regions/  Cell\_Region | Latency | The latency between snapshot transmission from vehicle and received time of cell tower in the defined region | Float (sec) |
| Regions/Event\_Regions/  Region | Title | Title of the event region | Character String |
| Regions/Event\_Regions/  Region/UpperLeftPoint | X | The x-coordinate of the upper left point of the event region rectangle | Float |
| Regions/Event\_Regions/  Region/UpperLeftPoint | Y | The y-coordinate of the upper left point of the event region rectangle | Float |
| Regions/Event\_Regions/  Region/LowerRightPoint | X | The x-coordinate of the lower right point of the event region rectangle | Float |
| Regions/Event\_Regions/  Region/LowerRightPoint | Y | The y-coordinate of the lower right point of the event region rectangle | Float |
| Regions/Event\_Regions/  Region/TimePeriods/  Period | StartTime | The start time of a defined active time period for the region | Float (sec) |
| Regions/Event\_Regions/  Region/TimePeriods/  Period | EndTime | The end time of a defined active time period for the region | Float (sec) |
| Regions/Event\_Regions/  Region/Events/Event | Title | Title of an event within the region, this title will be the name of the data element in the BSM output | Character String |
| Regions/Event\_Regions/  Region/Events/Event | Mean, SD, and Recheck | This probability option is to define a standard deviation and a mean. The recheck value for this probability type is static. | Integer |
| Regions/Event\_Regions/  Region/Events/Event | Probability and Recheck | This method of probability sets a static probability of occurrence and a static recheck value | Integer |
| Regions/Event\_Regions/  Region/Events/Event | Probability and RecheckPoisson | This method of probability uses a poisson distribution on a recheck value to determine the next recheck value, commonly used method for human interaction events | Integer |

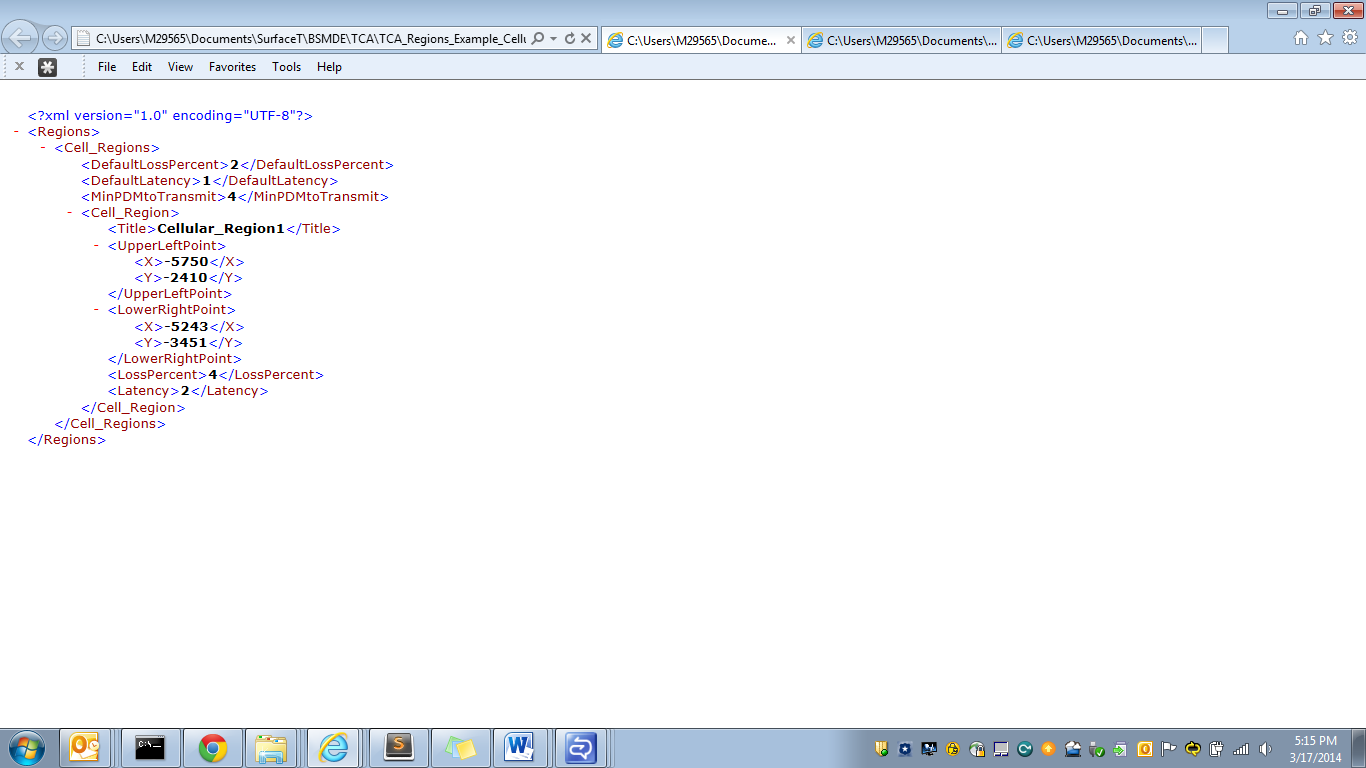


Figure 3‑7: Example regions input file with one cellular region

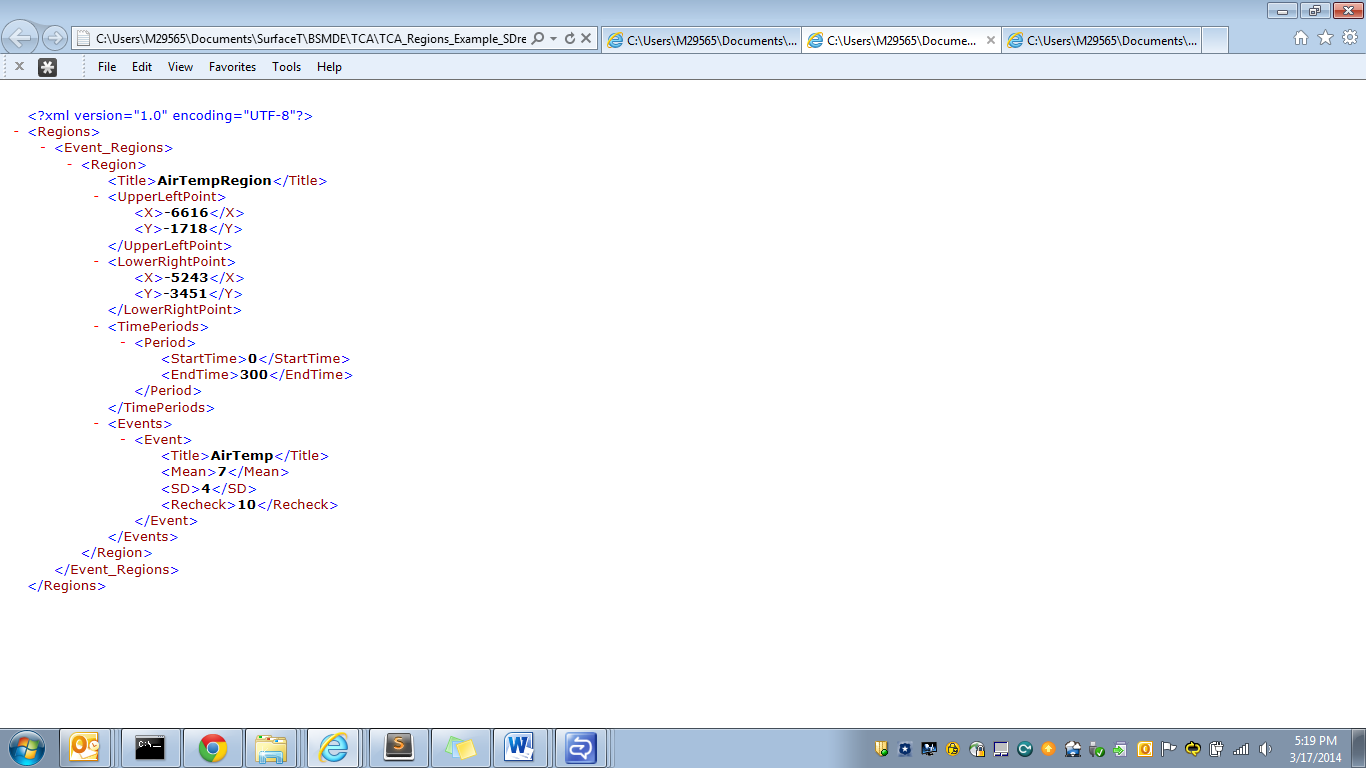


Figure 3‑8: Example region input file using the standard deviation method

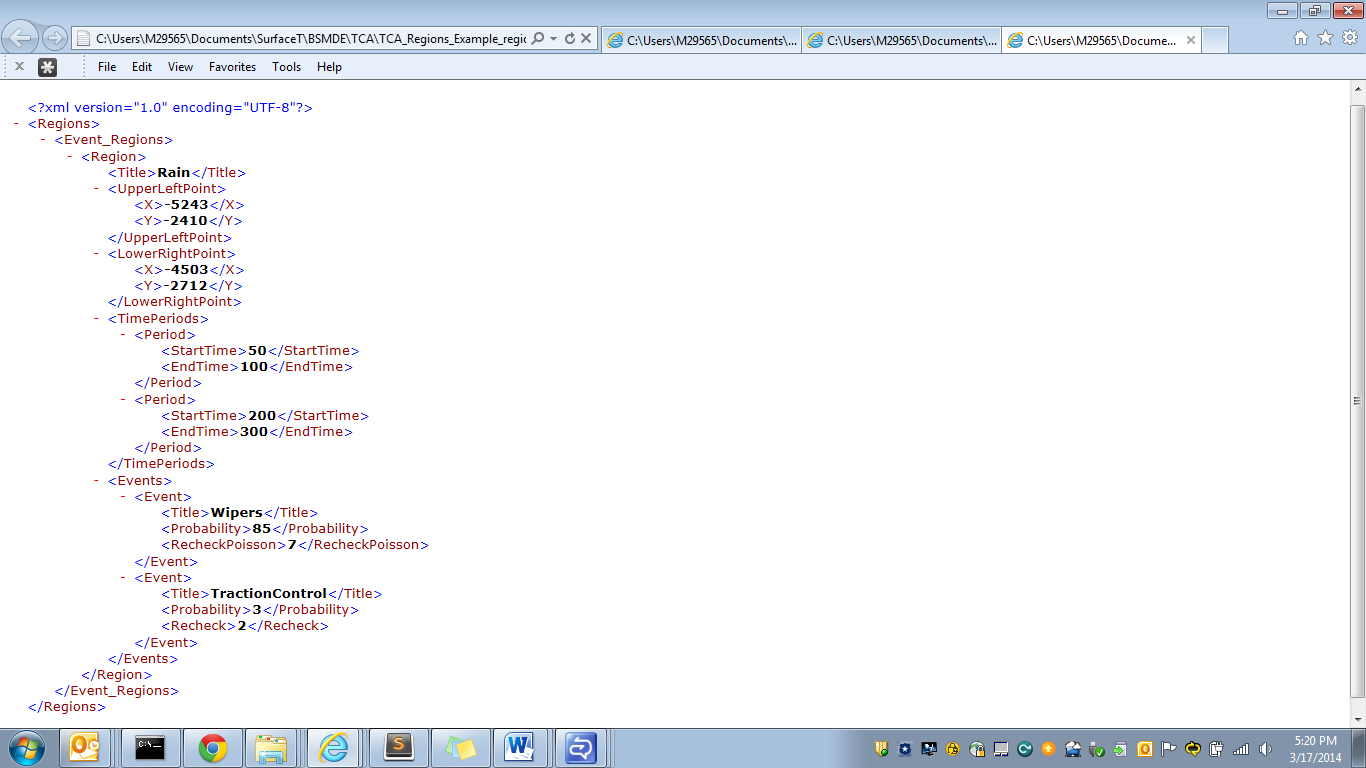


Figure 3‑9: Example regions input file using probability and Poisson distribution methods

# Output Files

## TCA Input Summary File

The TCA Input Summary File is a comma-delimited file that is always produced and lists every Control and Strategy element and their values. This file also includes an error message if an element is incorrectly defined. The file contains the items from Table 4‑1 for each element of the Control and Strategy files. An excerpt of an example TCA Input Summary file is shown in Figure 4‑1.

Table 4‑1. TCA Input Summary file fields

|  |  |
| --- | --- |
| Column | Description |
| FILE | The name of the file, either the name of the Control or Strategy file, from which the element is located |
| NAME | Name of the element |
| VALUE | Value of the element |
| XML\_TAG | The XML tag of the element |
| TYPE | Can be either Default or User-Defined |
| ERRORS | Error message, if any |

## 

Figure 4‑1. TCA Input Summary file example excerpt

## Transmitted PDMs File

The Transmitted PDM file is a comma delimited file that stores all PDM snapshot information that is transmitted to RSEs or via cellular, based on the RSE or cellular region locations and the strategy implemented in the TCA. The first line of the Transmitted Snapshot file is a header line describing all of the fields. The Transmitted Snapshot file contains the data elements from Table 4‑2 on each line. An excerpt from an example Transmitted PDM file is shown in Figure 4‑2.

Table 4‑2: Transmitted PDMs file fields

| Column | Name | Description | Value |
| --- | --- | --- | --- |
| 1 | Time Taken | Time that the snapshot was taken | Integer |
| 2 | PSN | The PSN number for the snapshot | Integer |
| 3 | Speed | Speed in mph that the vehicles were going then the snapshot was taken | Float (mph) |
| 4 | X | X value in feet for the location of the vehicle when the snapshot was taken | Integer (feet) |
| 5 | Y | Y value in feet for the location of the vehicle when the snapshot was taken | Integer (feet) |
| 6 | Transmit To | RSE or cellular region the snapshot was transmitted to | Character String |
| 7 | Transmit Time | The time the snapshot was transmitted | Integer |
| 8 | Message number | The order of the message within a transmission. Snapshots are transmitted in messages containing up to 4 snapshots | Integer |
| 9 | Snapshot number | Position of the snapshot within the message | Integer |
| 10 | Received Time | Time that the snapshot was received (accounts for latency if applicable) | Integer |

## 

Figure 4‑2. Transmitted PDM snapshots file example excerpt

## All PDM Snapshots File

The All PDM Snapshots file is a comma delimited file that stores all PDM snapshot information including PDM snapshots that were deleted and not transmitted. This file includes: type of snapshot taken, vehicle ID, whether the snapshot was deleted and why. The first line of the All Snapshots file is a header with all the field titles. The All PDM Snapshots file has the fields listed in Table 4‑3. An excerpt of an example All PDM Snapshots file is shown in Figure 4‑3.

Table 4‑3: All PDM snapshots file fields

| Column | Name | Description | Value |
| --- | --- | --- | --- |
| 1 | Vehicle ID | ID of the vehicles as stated in the vehicle trajectory file | Character String |
| 2 | SS Number | Total number of snapshots by all vehicles, starting at 1 | Integer |
| 3 | Time Taken | Time that the snapshot was taken | Integer |
| 4 | PSN | The PSN number for the snapshot | Integer |
| 5 | Speed | Speed in mph that the vehicles were going then the snapshot was taken | Float (mph) |
| 6 | X | X value in feet for the location of the vehicle when the snapshot was taken | Integer (feet) |
| 7 | Y | Y value in feet for the location of the vehicle when the snapshot was taken | Integer (feet) |
| 8 | Last Transmitted To | The name of the last RSE or cellular region that the vehicle transmitted to (only included if RSEFlag option is turned on in the Strategy file) | Character String |
| 9 | Type | Snapshot type (1-stop, 2-start, 3- periodic) | Integer |
| 10 | Transmit Time | The time the snapshot was transmitted to an RSE (-1 if not transmitted) | Integer |
| 11 | Transmit To | RSE or cellular region the snapshot was transmitted to (-1 if not transmitted) | Character String |
| 12 | Delete Time | Time the snapshot was deleted from the buffer. This value is 0 if the snapshot was not deleted | Integer |
| 13 | Delete Reason | Reason the snapshot was deleted (0-Not deleted, 1- Buffer overload, 2-Left in the buffer after the vehicle trajectory stopped, 3-PSN rollover gap, 4- RSE interaction, 5- Snapshot lost during cellular transmission) | Integer |

## 

Figure 4‑3. All PDM Snapshots file example excerpt

## Transmitted BSM File

The Transmitted BSM file is a comma delimited file that stores all BSM snapshot information that is transmitted via DSRC or cellular, based on the RSE or cellular region locations and the strategy implemented in the TCA. The first line of the Transmitted Snapshot file is a header line describing all of the fields. The Transmitted Snapshot file contains the data elements from Table 4‑4 on each line as well as any additional region elements defined in the Regions input file. An excerpt from an example file of transmitted BSMs is shown in Figure 4‑4.

Table 4‑4. Transmitted BSM File Fields

| Name | Description | Value |
| --- | --- | --- |
| Vehicle\_ID | ID of the vehicles as stated in the vehicle trajectory file | Character String |
| localtime | Time that the snapshot was taken | Integer |
| X | X value in feet for the location of the vehicle when the snapshot was taken | Integer (feet) |
| Y | Y value in feet for the location of the vehicle when the snapshot was taken | Integer (feet) |
| spd | Speed in mph that the vehicle was going when the snapshot was taken | Float (mph) |
| avg\_accel | Average acceleration that the vehicle was going between the previous snapshot and the current snapshot | Float (ft/s2) |
| VID | The PSN number | Character String |
| brakePressure | The deceleration value or zero if the vehicle is accelerating (vehicle instantaneous acceleration value required for this output) | Float (ft/s2) |
| brakeStatus | Notes the brakes applied status of each wheel independently. In the TCA Version 2.2, brakes are either off (0000) or on (1111) as determined by the brake threshold value in the Control input file (vehicle instantaneous acceleration value required for this output) | Character String |
| hardBraking | Is either true or false if the vehicle is hard braking (vehicle instantaneous acceleration value required for this output) | Character String |
| Heading | The heading of the vehicle (between 0 and 360 degrees) | Integer |
| transTo | The name of the RSE or Cellular region the BSM was transmitted to | Character String |
| Transmission\_received\_time | Time the BSM was received | Integer (sec) |
| Transtime | Time the BSM was transmitted | Integer (sec) |

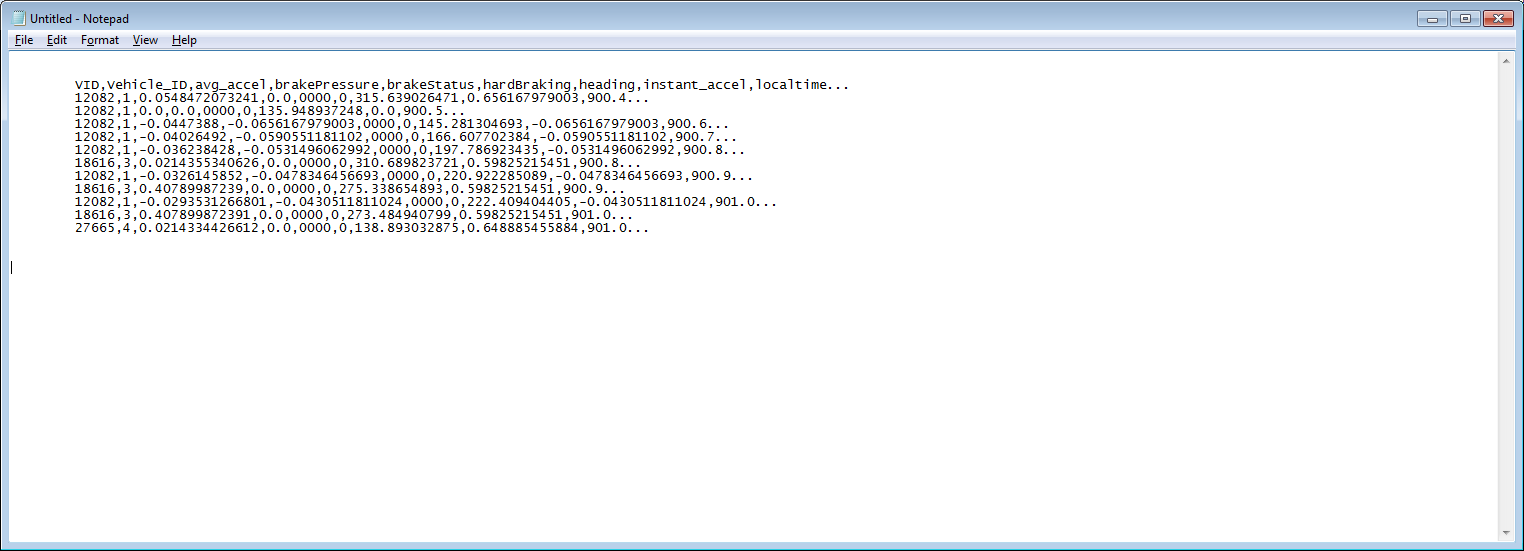


Figure 4‑4: Transmitted BSM snapshots file example excerpt

U.S. Department of Transportation  
ITS Joint Program Office-HOIT  
1200 New Jersey Avenue, SE  
Washington, DC 20590  
  
Toll-Free “Help Line” 866-367-7487  
[www.its.dot.gov](http://www.its.dot.gov)  
  
FHWA-JPO-XX-XXX



Federal Highways Administration