Diagnostics How-To

This document provides basic information on how to resolve some of the different types of errors that can be encountered when attempting to run the suite of TRANSIMS tools. There are 69 tools available within the TRANSIMS toolbox. Each TRANSIMS utility has unique error messaging specifically related to the calculations and processes that a particular tool is used to perform. This How-To will not provide a detailed listing of all the possible error messages that can be produced. However the document does outline a strategy using illustrative examples that should allow a user to debug and correct their setups, scripts and control files accordingly.

The remainder of the How-To will focus on the Problem files that are created by the Router and Microsimulator, and will provide some strategies for resolving these problems so the final model results reflect the total regional travel demand being simulated.

Revision History

5/2/2010 Created by RSG, Inc.

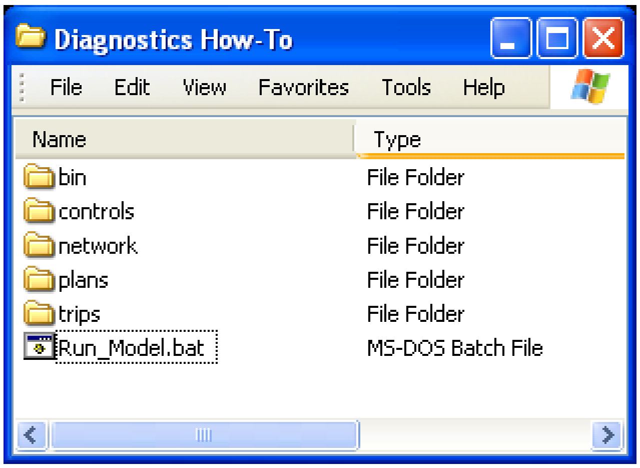
Table of Contents

1.0 Execution Errors

The core TRANSIMS technology includes an array of console-based programs. A console-based program runs in a terminal window (i.e., TRANSIMS does not include a graphical user interface, or GUI). To execute the program, the user typically types the name of the program (e.g. Router.exe) at the command prompt followed by an application control file. The program provides text messages about its execution status to the screen and to an output report file (*.prn). At the end of the execution the output files specified in the input control file will have been generated. As demonstrated in the Alexandria and TestNet test case data Windows DOS batch files can also be used to control the execution of TRANSIMS functions. The Parallelization & Partitioning How-To describes how other scripting languages such as Python or Cygwin can be used to control model flow either in partitioned or non-partitioned mode.

As demonstrated in the Alexandria and TestNet test case data, a model directory with a series of subdirectories where various inputs will be read from and outputs written to is typical. The figure below illustrates what a typical TRANSIMS model directory might look like. In this very simplified example, the subdirectories would contain the following:

\bin\	- compiled TRANSIMS executbables (e.g. Router.exe, PlanSum.exe)
\controls\	- control text files for each TRANSIMS function that will be utilized
\network\	- TRANSIMS network files (e.g. link, node, activity locations)
\plans\	- travel plan files generated by running the Router
\trips\	- trip list containing the regional travel demand



This is a simplified case for illustrative purposes. The user of course has complete freedom to specify any model directory and/or sub-directory structure that will suit their needs. All files could be placed in a single master directory, although maintenance and management of the model would likely be a challenge given the significant number of files associated with an iterative TRANSIMS model simulation.

In this example, the TRANSIMS executables are located in the **\bin** directory and the control files are located in the **\controls** directory. A sample router control file is shown below.

```
TITLE                            Route Travelers

#---- Input Files ----

NET_NODE_TABLE                   network\Node
NET_LINK_TABLE                   network\Link
NET_LANE_CONNECTIVITY_TABLE      network\Lane_Connectivity
NET_PARKING_TABLE                network\Parking
NET_ACTIVITY_LOCATION_TABLE      network\Activity_Location
NET_PROCESS_LINK_TABLE           network\Process_Link

TRIP_FILE                        trips\Trip
VEHICLE_FILE                     trips\Vehicle

#---- Output Files ----
```

```
NEW_PROBLEM_FILE                     plans\Problems
NEW_PLAN_FILE                        plans\Plans
```

When executed this control file will instruct the Router to read the network files (Node, Link, Lane Connectivity, Parking, Activity Location, and Process Link) from the **\network** subdirectory. The Trip and Vehicle file will be read from the **\trips** subdirectory and the outputs from the Router execution, the Plans and Problems files, will be saved to the **\plans** subdirectory. To run this sample router control file with the Router, the following syntax in the Run_Model.bat would be needed.

bin\Router.exe -K controls\Router.ctl

This syntax in the Run_Model.bat batch file specifies that the Router executable is located in the **\bin** directory and the control file is located in the **\controls** directory.

1.1 Directory and Pathname Specification

Pointing to the incorrect directories when developing the syntax for the batch file (or other scripts in Python used to control model flow) is a typical error encountered by new users. For instance, the syntax below fails because there is no Router.ctl file in the main directory where the Run_Mode.bat file is located. The Router.ctl file resides in the **\controls** directory.

bin\Router.exe -K Router.ctl

The following error message would be returned by the Router in this case using the syntax above in the Run_Model.bat. There was an error reading the control file because the control file could not be found.

```
********************************************
|                                          |
|         Router - Version 4.0.63          |
|   Copyright (c) 2009 by AECOM Consult    |
|         Thu May 13 11:48:37 2010         |
|                                          |
********************************************

Syntax is Router [-flag] [control_file] [partition]

Error: Reading Control File: "Router.ctl"

Press Enter to Continue
```

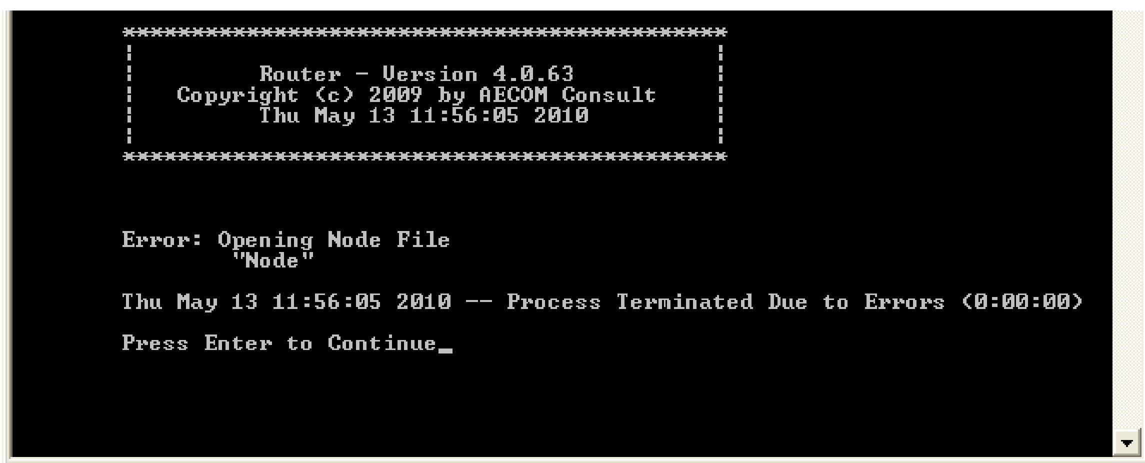
The directories specified within the control file are also of critical importance. For instance let's test the same Router.ctl file but this time the TRANSIMS network files have had the path **network** removed from each control key variable specification. The modified Router.ctl file is illustrated below without the **network** path prefix for the TRANSIMS network files (e.g. Node, Link, Parking).

```
TITLE                               Route Selected Travelers

#---- Input Files ----

NET_NODE_TABLE                      Node
NET_LINK_TABLE                      Link
NET_LANE_CONNECTIVITY_TABLE         Lane_Connectivity
NET_PARKING_TABLE                   Parking
NET_ACTIVITY_LOCATION_TABLE         Activity_Location
NET_PROCESS_LINK_TABLE              Process_Link

TRIP_FILE                           trips\Trip
VEHICLE_FILE                        trips\Vehicle

#---- Output Files ----

NEW_PROBLEM_FILE                    plans\Problems
NEW_PLAN_FILE                       plans\Plans
```

When the Router is executed by running the Run_Model.bat file with the proper syntax:

bin\Router.exe -K controls\Router.ctl

it will fail because the TRANSIMS network files will not be able to be found. Indeed, the error message returned by the Router is that an error occurred when attempting to open the “Node” file, which is the first network file listed in the Router.ctl file above.



The information written to the DOS screen will also be saved to the output print file, in this case Router.prn which will be written to the **\controls** directory. Reviewing the *.prn files is a very good way to identify what if any errors and/or warnings were encountered during the execution of a TRANSIMS utility.

Because of the importance of the pathname and directory specifications, some users find it useful to specify full path and directory names throughout the model system. In this way, the developer always knows precisely where files are being read and/or written and where the execution of the function is occurring. For example, instead of specifying:

NET_NODE_TABLE network\Node

a user could elect to specify the control key variable as:

NET_NODE_TABLE C: \Transims\My_Model\network\Node

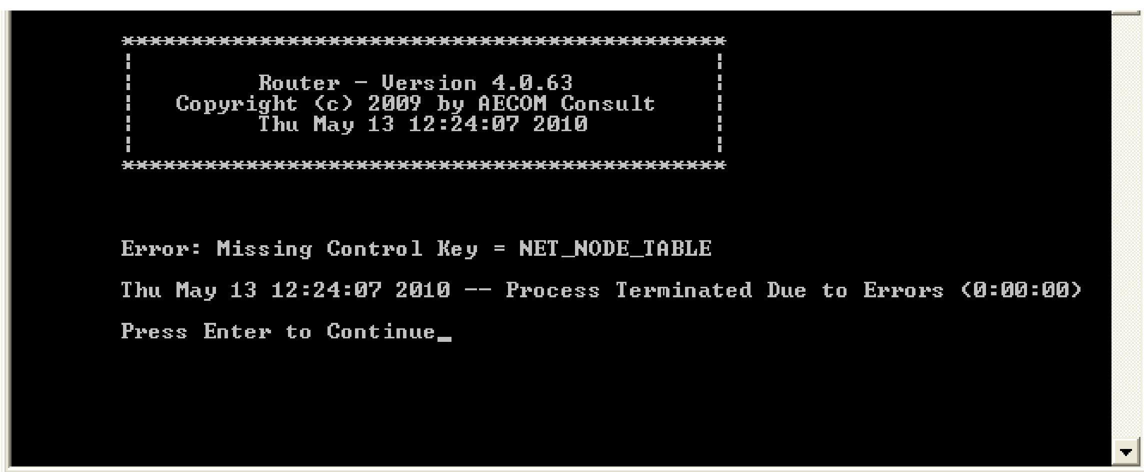
The syntax in the Run_Model.bat file could also be changed to utilize full pathnames as well.

C:\Transims\My_Model\bin\Router.exe –K C:\Transims\My_Model\controls\Router.ctl

It is very easy to setup a model which runs successfully and yet does not save results to the intended location. This can lead to a lot of confusion. Therefore, the main point is that a user must take good care to ensure that files are being read, written, and saved in the intended locations and further that the TRANSIMS executables and batch files are being executed from the proper directory locations as well.

1.2 Control Key Specification

TRANSIMS will also throw an error if a required input file is not specified in the control file. In addition, the control keys must be spelled correctly to be recognized by the TRANSIMS functions. Take the case where the NET_NODE_TABLE key option is simply omitted from the Router control file.



The NET_NODE_TABLE is a required key that must be present to run the Router, so an error message above is produced when the Router is executed. An error is also generated if the control file mistakenly contains the following line with the letter ‘E’ omitted from the control key variable:

NET_NODE_TABL network\Node

```
*********************************************
|                                           |
|          Router - Version 4.0.63          |
|    Copyright (c) 2009 by AECOM Consult    |
|          Thu May 13 12:27:32 2010         |
|                                           |
*********************************************

Warning: Control Key "NET_NODE_TABL" was Not Used

Error: Missing Control Key = NET_NODE_TABLE

Thu May 13 12:27:32 2010 -- Process Terminated Due to Errors (0:00:00)

Press Enter to Continue_
```

Since misspelling a control key variable (which is also a required key) will cause the function to terminate with an error, it is good practice to run the tools with the –K option. This option checks whether all control keys listed in the control file are actually supported. In this case, the Router produced a warning message indicating that "NET_NODE_TABL" was not used. This warning along with the error that the "NET_NODE_TABLE" was missing would help the user identify that he/she had simply omitted the letter 'E' from the control key variable. Other options for running the TRANSIMS functions are listed below.

1.3 Command Line Options

Command line options have been implemented to turn off potential interactive feedback and provide some information about the particular TRANSIMS function.

-Q[uiet]	= execute without screen messages
-H[elp]	= show program syntax and control keys
-K[eyCheck]	= list unrecognized control file keys
-P[ause]	= pause before exiting
-N[oPause]	= never pause before exiting
-B[atch]	= execute in batch processing mode

The –H command line option is very useful. Running a TRANSIMS tool with the –H command will list all of the control keys which can be used with the given executable. It will also provide the version number of the tool, in this example Router – Version 4.0.63. This is a good way of identifying which release version of a particular tool is being used as updates and new minor releases are frequent.

The syntax below will produce a Router_control_keys.txt file.

bin\Router.exe –H –N > Router_control_keys.txt

A portion of the Router_control_keys.txt created by running a batch file with the syntax above is provided below. Note, this, like any TRANSIMS execution can be initiated at the command prompt or in a batch file or via a script developed in Python or another scripting language.

```
********************************************
|                                          |
|         Router - Version 4.0.63          |
|   Copyright (c) 2009 by AECOM Consult    |
|         Fri May 14 08:27:52 2010         |
|                                          |
********************************************


Syntax is Router [-flag] [control_file] [partition]


Optional Flags:
       -Q[uiet]     = execute without screen messages
       -H[elp]      = show program syntax and control keys
       -K[eyCheck] = list unrecognized control file keys
       -P[ause]     = pause before exiting
       -N[oPause]   = never pause before exiting
       -B[atch]     = execute in batch processing mode

Control File Keys:
       TITLE
       REPORT_FILE
       REPORT_FLAG
       PROJECT_DIRECTORY
       DEFAULT_FILE_FORMAT
       MAX_WARNING_MESSAGES
       MAX_WARNING_EXIT_FLAG
       TRAVELER_SCALING_FACTOR
       NET_DIRECTORY
       NET_NODE_TABLE
       NET_LINK_TABLE
       NET_LANE_USE_TABLE
       NET_TOLL_TABLE
       NET_LANE_CONNECTIVITY_TABLE
       NET_TURN_PROHIBITION_TABLE
       NET_PARKING_TABLE
       NET_ACTIVITY_LOCATION_TABLE
       NET_PROCESS_LINK_TABLE
       <more>

Report Options:
       HOUSEHOLD_TYPE_SCRIPT
       HOUSEHOLD_TYPE_STACK
       FARE_DATA_REPORT
```

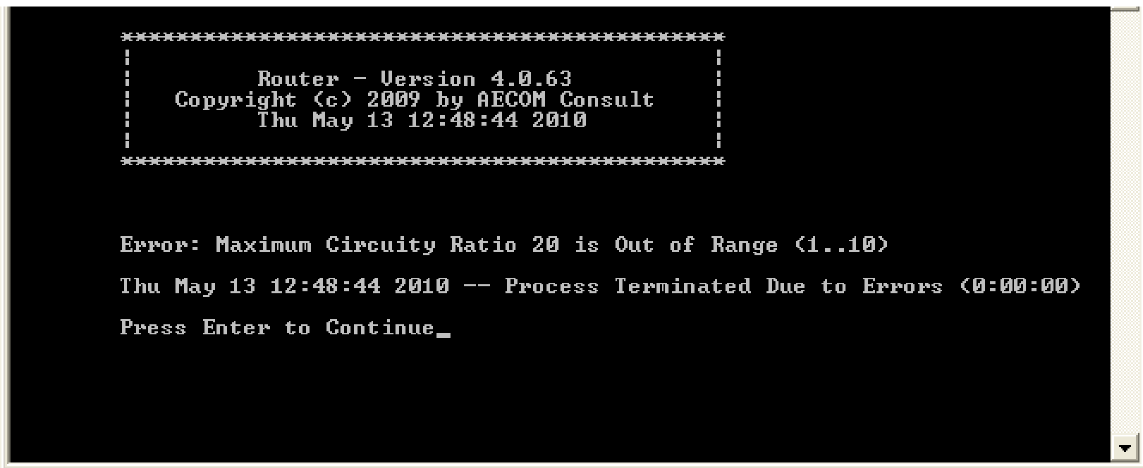
The Quick Reference documentation also contains a listing of the control key variables and identifies which are required and which are optional along with the range of permitted control key parameter values. The –H command will always provide the most up-to-date listing of control key variables as it is wedded to the software directly.

1.4 Out of Range Variables

The final major type of execution error that can be encountered is when a control key variable exceeds the range of permitted values. For instance, in the Router, the control key variable MAX_CIRCUITY_RATIO can be any value between 0 and 10. However a value of 20 could be specified in the Router.ctl file with the line:

```
MAX_CIRCUITY_RATIO          20
```

The Router will terminate with an error because the permitted variable range defined within the Router has been exceeded. In this case, the Maximum Circuity Ratio of 20 is identified as being Out of Range and the error message provides information that the accepted range is (1..10).



1.5 Default Control Key Parameter Values

The TRANSIMS software tools, like other transportation planning software packages, use default control key parameter values in some cases if they are not specified in the user-configured control file. It is important to understand these default values. In addition, the use of certain control keys activates the use of other keys and their default settings unless they are specifically overwritten by specification in the control file.

For example, the default value for the control key MAX_CIRCUITY_RATIO is 2.0. This key defines the maximum permissible ratio between the sum of the distance a path node is from the trip origin and destination and the straight-line distance between the trip origin and destination. If the value is zero, no circuity checks are made. A value of 2.0 implies that the length of the travel path is limited to approximately twice the straight-line distance between the origin and the destination.

Even if the control key MAX_CIRCUITY_RATIO is not specified in the Router control file, a value of 2.0 will be used during the calculations. Given certain network geography or other regional conditions, a user may want to permit more circuitous routing, so the user can add the MAX_CIRCUITY_RATIO key to the Router control file and specify a control key value of 3.0 for example.

The Router can be executed without a trip, activity or vehicle file by removing the ACTIVITY_FILE, TRIP_FILE, and VEHICLE_FILE control keys from the control file. However, the Router will not execute unless origins and destinations are specified to save paths from and to. This is accomplished using the control keys:

ROUTE_FROM_SPECIFIED_LOCATIONS
ROUTE_TO_SPECIFIED_LOCATIONS

The sample Router control file removes the trip file and vehicle file by using the '#' symbol to comment out the control key and identifies a range of activity location origin and destinations to save paths for, in this case locations 1 to 1000.

```
TITLE                  Route Travelers for defined Os & Ds

#---- Input Files ----

NET_NODE_TABLE                   network\Node
NET_LINK_TABLE                   network\Link
NET_LANE_CONNECTIVITY_TABLE      network\Lane_Connectivity
NET_PARKING_TABLE                network\Parking
NET_ACTIVITY_LOCATION_TABLE      network\Activity_Location
NET_PROCESS_LINK_TABLE           network\Process_Link

# TRIP_FILE                       trips\Trip
# VEHICLE_FILE                    trips\Vehicle

ROUTE_FROM_SPECIFIED_LOCATIONS       1..1000
ROUTE_TO_SPECIFIED_LOCATIONS         1..1000

#---- Output Files ----

NEW_PROBLEM_FILE                 plans\Problems
NEW_PLAN_FILE                    plans\Plans
```

The sample control file above will execute without errors, however no paths will be generated and saved by the Router. This is due to the fact that use of the Route-From and Route-To control keys activates other control keys which were not specified in the control file. The ROUTE_AT_SPECIFIED_TIMES and ROUTE_BY_TIME_INCREMENT must also be specified since these four control keys are co-dependent. The sample control file below will successfully save travel plans from activity locations 1-1000 to activity locations 1-1000 between the times of 8am and 9am at 15 minute increments.

```
TITLE                  Route Travelers for defined Os & Ds

#---- Input Files ----

NET_NODE_TABLE                     network\Node
NET_LINK_TABLE                     network\Link
NET_LANE_CONNECTIVITY_TABLE        network\Lane_Connectivity
NET_PARKING_TABLE                  network\Parking
NET_ACTIVITY_LOCATION_TABLE        network\Activity_Location
NET_PROCESS_LINK_TABLE             network\Process_Link

# TRIP_FILE                         trips\Trip
# VEHICLE_FILE                      trips\Vehicle

ROUTE_FROM_SPECIFIED_LOCATIONS        1..1000
ROUTE_TO_SPECIFIED_LOCATIONS          1..1000
ROUTE_AT_SPECIFIED_TIMES              8:00..9:00
ROUTE_BY_TIME_INCREMENT               15

#---- Output Files ----

NEW_PROBLEM_FILE                   plans\Problems
NEW_PLAN_FILE                      plans\Plans
```

Information about control key default values and their use can be found in the Quick Reference documents and by utilizing the –H command option when executing the TRANSIMS functions.

2.0 TRANSIMS File Formats

TRANSIMS supports a wide range of input and output file formats which provides great flexibility for the user when configuring the model simulation system. Input and output files are typically saved as text files that can be viewed with standard text editors (e.g. EditPad, NotePad++). The table below shows the different file formats which can be used in TRANSIMS.

File Format	TRANSIMS File Format Name
Version3	VERSION3
Binary	BINARY
Fixed Column	FIXED_COLUMN
Comma Delimited	COMMA_DELIMITED, CSV_DELIMITED
Space Delimited	SPACE_DELIMITED
Tab Delimited	TAB_DELIMITED
Database	DBASE
SQLITE3	SQLITE3

There are 185 control keys which deal with the file and data specification formats of the various TRANSIMS input, output, and parameter settings. The control keys generally end with the suffix _FORMAT. A list of some of the prominent file format keys is provided below.

DEFAULT_FILE_FORMAT	NET_ACTIVITY_LOCATION_FORMAT
DEFAULT_OUTPUT_FORMAT	NET_DEFAULT_FORMAT
INPUT_PLAN_FORMAT	NET_LANE_CONNECTIVITY_FORMAT
LINK_DELAY_FORMAT	NET_LINK_FORMAT
MERGE_PLAN_FORMAT	NET_NODE_FORMAT
NEW_PLAN_FORMAT	NET_PARKING_FORMAT
OUTPUT_DATA_FORMAT	NET_PHASING_PLAN_FORMAT
OUTPUT_EVENT_FORMAT	NET_POCKET_LANE_FORMAT
OUTPUT_OCCUPANCY_FORMAT	NET_PROCESS_LINK_FORMAT
OUTPUT_PLAN_FORMAT	NET_SIGNALIZED_NODE_FORMAT
OUTPUT_PROBLEM_FORMAT	NET_SIGNAL_COORDINATOR_FORMAT
OUTPUT_SNAPSHOT_FORMAT	NET_TIMING_PLAN_FORMAT
PLAN_FORMAT	
TRIP_TABLE_FORMAT	

It is important to keep in mind that these _FORMAT control keys also have default values, in most cases the default file format is "Version3". TRANSIMS uses definition files (*.def) to maintain information about input and output files including the format, file attribute data fields, and variable types. If the *.def files are not present when a TRANSIMS function is executed they will be automatically generated by the utility. Let's examine a case where the Router is run with the sample control file we've been using for illustrative purposes thus far. In addition, we will assume that the **\network** directory does not contain the *.def definition files for the Node, Link and other TRANSIMS network files.

```
TITLE                              Route Travelers

#---- Input Files ----

NET_NODE_TABLE                     network\Node
NET_LINK_TABLE                     network\Link
NET_LANE_CONNECTIVITY_TABLE        network\Lane_Connectivity
NET_PARKING_TABLE                  network\Parking
NET_ACTIVITY_LOCATION_TABLE        network\Activity_Location
NET_PROCESS_LINK_TABLE             network\Process_Link

TRIP_FILE                         trips\Trip
VEHICLE_FILE                      trips\Vehicle

#---- Output Files ----

NEW_PROBLEM_FILE                   plans\Problems
NEW_PLAN_FILE                      plans\Plans
```

Following execution of the Router, the **\network** directory will now contain 6 additional files that were created by the Router when the Node, Link, Lane_Connectivity, Parking, Activity_Location, and Process_Link files were read. The directory will now contain definition files for each network file with the *.def suffix appended to the network file name. The content of the newly created Node.def is shown below.

```
AECOM HEADER, VERSION3, 1
NODE, INTEGER, 1, 10
X_COORD, DOUBLE, 2, 10, 1
Y_COORD, DOUBLE, 3, 10, 1
NOTES, STRING, 4, 16
```

The definition file includes the file format (Version3) and indicates there are four data attribute fields called NODE, X_COORD, YCOORD, and NOTES. The NODE data are integer variables, the X and Y coordinates are doubles and the NOTES variable is a string. A portion of the Node file is presented below.

```
NODE  X_COORD     Y_COORD     NOTES
301   649803.3    4944339.6   External Station
351   637400.9    4944034.4   External Station
369   642709.7    4926477.8   Network Node
370   643100.7    4926518.0   Network Node
371   641660.6    4926383.7   Network Node
```

The Node file is actually tab-delimited. However, the default value for the NET_DEFAULT_FORMAT control key is Version3. Therefore TRANSIMS believes the Node file is in the Version3 format. If we delete the *.def files that were created and rerun the Router with a slightly different control file we can make sure the definition file is correct. The control key NET_DEFAULT_FORMAT has been specified as tab-delimited.

```
TITLE                             Route Travelers

NET_DEFAULT_FORMAT                TAB_DELIMITED

#---- Input Files ----

NET_NODE_TABLE                    network\Node
NET_LINK_TABLE                    network\Link
NET_LANE_CONNECTIVITY_TABLE       network\Lane_Connectivity
NET_PARKING_TABLE                 network\Parking
NET_ACTIVITY_LOCATION_TABLE       network\Activity_Location
NET_PROCESS_LINK_TABLE            network\Process_Link

TRIP_FILE                        trips\Trip
VEHICLE_FILE                     trips\Vehicle

#---- Output Files ----

NEW_PROBLEM_FILE                  plans\Problems
NEW_PLAN_FILE                     plans\Plans
```

Examination of the new Node.def reveals that the correct file format associated with the Node file is now specified in the definition file.

```
AECOM HEADER, TAB_DELIMITED, 1
NODE, INTEGER, 1, 10
X_COORD, DOUBLE, 2, 10, 1
Y_COORD, DOUBLE, 3, 10, 1
NOTES, STRING, 4, 16
```

The Router.prn output file can also be reviewed to confirm file formats for each input file. A portion of the Router.prn file is shown below.

```
********************************************
|                                          |
|         Router - Version 4.0.63          |
|   Copyright (c) 2009 by AECOM Consult    |
|         Fri May 14 12:31:05 2010         |
|                                          |
********************************************

Control File = controls\Router.ctl
Report_File  = controls\Router.prn (Create)

Route Selected Travelers

Node File = network\Node
Node File Format = TAB_DELIMITED
Link File = network\Link
Link File Format = TAB_DELIMITED
Lane Connectivity File = network\Lane_Connectivity
Lane Connectivity File Format = TAB_DELIMITED
Parking File = network\Parking
Parking File Format = TAB_DELIMITED
Activity Location File = network\Activity_Location
Activity Location File Format = TAB_DELIMITED
Process Link File = network\Process_Link
Process Link File Format = TAB_DELIMITED

Vehicle File = trips\Vehicle
Vehicle File Format = TAB_DELIMITED
Vehicle File will be Sorted by Vehicle ID
```

If the *.def definition files for input files are already present in the directory, they will be read by the utility. If they are not present they will be created. Therefore, deleting all the *.def files was necessary in the step above or the *.def files would not have been changed by re-running the Router with the NET_DEFAULT_FORMAT key added to the control file. This fact coupled with control key default formats in some cases can lead to incompatibilities when executing a TRANSIMS function that will cause the utility to fail. New definition *.def files will always be created for outputs generated by running a utility.

Users should pay special attention to the file formats they are using in their model applications because different file formats can contain different data field attributes. For example, the PlanSum function is used to derive link delays from a set of regional travel plans created by running the Router. A sample PlanSum control file is provided below. The default file format for the output Link_Delay file is Version3

```
TITLE                              PlanSum on Route Plans

# ---- Input Files ----

NET_DIRECTORY
NET_LINK_TABLE                     network\Link
NET_NODE_TABLE                     network\Node
NET_ACTIVITY_LOCATION_TABLE        network\Activity_Location
NET_PARKING_TABLE                  network\Parking

PLAN_FILE                          plans\Plans

# ---- Output Files ----

NEW_LINK_DELAY_FILE                network\Link_Delay
```

The definition file and the output Link_Delay file created by PlanSum for a portion of the file records are presented below. The output file has 11 unique data fields.

```
AECOM HEADER, VERSION3, 2
METADATA      Fri May 14 12:45:51 2010   TIME_STEP    900
COUNT, INTEGER, 1, 10
LANE, INTEGER, 2, 2
LINK, INTEGER, 3, 10
NODE, INTEGER, 4, 10
SUM, DOUBLE, 5, 14, 2
SUMSQUARES, DOUBLE, 6, 20, 2
TIME, INTEGER, 7, 10
TURN, INTEGER, 8, 2
VCOUNT, INTEGER, 9, 10
VSUM, DOUBLE, 10, 14, 2
VSUMSQUARES, DOUBLE, 11, 20, 2
```

METADATA TIME_STEP 900

COUNT	LANE	LINK	NODE	SUM	SUMSQUARES	TIME	TURN	VCOUNT	VSUM	VSUMSQUARES
1	1	9	681	469.6	220524.16	1800	0	1	22	484.05
1	1	9	681	469.6	220524.16	14400	0	1	22	484.05
3	1	9	681	1408.8	661572.48	15300	0	3	66	1452.14
1	1	9	681	469.6	220524.16	16200	0	1	22	484.05

If we re-run PlanSum but specify a NEW_LINK_DELAY_FORMAT of TAB_DELIMITED a different Link_Delay output file and definition file are generated. The definition file and the output Link_Delay file created by PlanSum using the tab-delimited file format for a portion of the file records are presented below. The output file this time has 19 unique data fields and 4 nested data fields.

```
AECOM HEADER, TAB_DELIMITED, 2, NESTED
LINK, INTEGER, 1, 10
DIR, INTEGER, 2, 1
START_TIME, STRING, 3, 16
END_TIME, STRING, 4, 16
AVG_VOLUME, INTEGER, 5, 10
IN_VOLUME, INTEGER, 6, 10
OUT_VOLUME, INTEGER, 7, 10
AVG_SPEED, DOUBLE, 8, 10, 2
AVG_TIME, DOUBLE, 9, 10, 2
AVG_DELAY, DOUBLE, 10, 10, 2
AVG_DENSITY, DOUBLE, 11, 10, 2
MAX_DENSITY, DOUBLE, 12, 10, 2
TIME_RATIO, DOUBLE, 13, 10, 2
AVG_QUEUE, DOUBLE, 14, 10, 2
MAX_QUEUE, INTEGER, 15, 10
NUM_FAIL, INTEGER, 16, 10
VMT, DOUBLE, 17, 12, 1
VHT, DOUBLE, 18, 12, 1
NCONNECT, INTEGER, 19, 2
OUT_LINK, INTEGER, 1, 10, 0, NESTED
OUT_DIR, INTEGER, 2, 1, 0, NESTED
OUT_TURN, INTEGER, 3, 10, 0, NESTED
OUT_TIME, DOUBLE, 4, 10, 2, NESTED
```

LINK	DIR	START_TIME	END_TIME	AVG_VOLUME	IN_VOLUME	OUT_VOLUME	AVG_SPEED	AVG_TIME	AVG_DELAY
OUT_LINK	OUT_DIR	OUT_TURN	OUT_TIME						
9	0	0:15	0:30	1	1	1	22	469.6	0
9	0	3:45	4:00	1	1	1	22	469.6	0
9	0	4:00	4:15	3	3	3	22	469.6	0
9	0	4:15	4:30	1	1	1	22	469.6	0
9	0	4:30	4:45	3	3	3	22	469.6	0

AVG_DENSITY	MAX_DENSITY	TIME_RATIO	AVG_QUEUE	MAX_QUEUE	NUM_FAIL	VMT	VHT	NCONNECT
0.01	0.01	1	0	0	0	10331.7	469.6	0
0.01	0.01	1	0	0	0	10331.7	469.6	0
0.01	0.01	1	0	0	0	30995.1	1408.8	0
0.01	0.01	1	0	0	0	10331.7	469.6	0
0.01	0.01	1	0	0	0	30995.1	1408.8	0

The tab-delimited link delay file contains many more data fields than the Version3 link delay file. However, the data in both files is the same. For example SUM equals 469.6 for the first record in Version3 link delay file. The AVG_TIME also equals 469.6 for the first record in the tab-delimited link delay file. Both data attributes represent the sum of the vehicle travel times (in seconds) for all vehicles leaving the link.

Which file formats are used depends on the needs and preferences of the user. However it is important that users recognize the differences between the supported file formats and the potential for errors if consistent file formats are not maintained throughout the model system. The *.def definition files help to ensure that TRANSIMS is aware of the file formats but these too can be susceptible to problems if the user is unaware of the default format settings for certain inputs or outputs.

3.0 Problem Files

3.1 Router Problems

The Router program generates two important outputs, 1) plans and 2) problems. The plans contain the exact list of nodes or links that the traveler goes through to complete a trip including the start time and end time. The problem file lists the travelers for whom complete plans could not be generated. If a trip cannot be routed the problem type is logged to the problem file so it can be investigated and resolved by the user and/or addressed in future model iterations.

The NEW_PROBLEM_FILE control key is optional. When included in the Router control file, it specifies the file name for the output problem file created by the program. If the key is not provided, the path building problems will be summarized in the output *.prn file but will not be saved to a file. If the command line includes a partition parameter, the program will add ".t*" to this key. If the partition number is "0", the "tAA" extension is added. If the partition number is "1", the "tAB" extension is added and so on.

A sample problem file generated by the Router is shown below.

HHOLD	PERSON	TRIP	MODE	PROBLEM	START	ORIGIN	ARRIVE	DESTINATION
178	1	1	2	3	3272	5	3485	7
343	1	1	2	3	1089	24	1302	22
356	1	1	2	3	3178	11	3391	10
359	1	1	2	3	3264	11	3477	10
503	1	1	2	3	997	10	1210	11

A sample portion of the Router output *.prn file which summarizes the problems is shown below.

```
Number of Trip Records = 430893
Number of Trips Processed = 430893
Number of Households Processed = 430893
Number of Vehicle Trips Saved = 422958
Number of Travel Time Updates = 4229

Number of Output Plans = 1276809
Number of Output Records = 8850299
Number of Output Travelers = 430893
Number of Output Trips = 422958

Percent of Total Trips with Problems = 1.8%

Total Number of Problems = 7935
Number of Zero Node Problems (#3) = 5628 (70.9%)
Number of Path Circuity Problems (#5) = 2307 (29.1%)
```

In this example, the total input trip demand is 430,893 trips (Number of Trip Records). The Router was able to successfully route 422,958 of those trips (Number of Output Trips). However, 7,935 of the total 430,893 input trips had problems and were not routed successfully, representing 1.8% of the total input trip demand. The summary report in the output *.prn file also indicates that 5,628 of those problems were type #3 and 2,307 were type #5. Resolving these problems is important because the resulting plan file will only contain travel plans for 422,958 trips. If for example, 50% of the input trips had problems then only 50% of the travel demand would be represented in the output plan file created by the Router.

The PROBLEM field identifies the problem code for a given record. The message corresponding to each problem code is listed below. An interpretation of the messages generated by the Router program is also provided in Section 3.3 of this document.

1. Path Building
2. Time Schedule
3. Zero Node
4. Vehicle Type
5. Path Circuity
6. Travel Mode
7. Vehicle Access
8. Walk Distance
9. Wait Time
10. Walk Access
11. Path Size
12. Park-&-Ride Lot
13. Bike Distance
14. Departure Time
15. Arrival Time
16. Link Access
17. Lane Connectivity
18. Parking Access
19. Lane Merging
20. Lane Changing
21. Turning Speed
22. Pocket Merge
23. Vehicle Spacing
24. Traffic Control
25. Access Restriction
26. Transit Stop
27. Activity Location
28. Vehicle Passenger
29. Activity Duration
30. Kiss-&-Ride Lot
31. Vehicle ID
32. Data Sort
33. Walk Location
34. Bike Location
35. Transit Location
36. Person Match
37. Transit Capacity
38. Transit Dwell

3.2 Microsimulator Problems

The Microsimulator also generates a problem file as one of its principal outputs. The problem file lists the travelers for whom their complete plan could not be simulated. If a trip cannot be simulated the problem type is logged to the problem file so it can be investigated and resolved by the user and/or addressed in future model iterations.

The NEW_PROBLEM_FILE control key is optional. When included in the microsimulator control file it specifies the file name for the output problem file created by the program. The problem file is a tab-delimited ASCII file that lists the households and persons for whom complete plans could not be simulated. A sample problem file generated by the Microsimulator is shown below.

TIME	LINK	HHOLD	PERSON	TRIP	MODE	PROBLEM	START	ORIGIN	ARRIVE	DESTINATION
3272	1	178	1	1	2	3	3272	5	3485	7
1089	1	343	1	1	2	3	1089	24	1302	22
3178	1	356	1	1	2	3	3178	11	3391	10
3264	1	359	1	1	2	3	3264	11	3477	10
997	1	503	1	1	2	3	997	10	1210	11

A sample portion of the Microsimulator output *.prn file which summarizes the problems is shown below.

```
Number of Plan Files = 1
Number of Input Plans = 1458152
Number of Input Records = 10522924
Number of Input Travelers = 416464
Number of Input Trips = 419201

Number of Vehicle Trips Processed = 384667
Number of Vehicle Trips Started   = 383496 (99.7%)
Number of Vehicle Trips Completed = 336636 (87.5%)

Number of Travelers with Problems = 53368 (12.8%)

Total Number of Problems = 55201
Number of Wait Time (#9) Problems = 23302 (42.2%)
Number of Departure Time (#14) Problems = 8120 (14.7%)
Number of Arrival Time (#15) Problems = 23659 (42.9%)
Number of Lane Connectivity (#17) Problems = 102 (0.2%)
Number of Vehicle Spacing (#23) Problems = 18 (0.0%)
```

In this example, the total input trip demand is 419,201 trips (Number of Input Trips). The Microsimulator was able to successfully simulate 336,636 of those vehicle trips (Number of Vehicle Trips Completed). However, 55,201 of the total input trips had problems and were not simulated successfully. The summary report in the output *.prn file also summarizes the problems by type. In this case there were a large number of Wait Time and Arrival Time problems. Again, resolving these problems is important because the resulting network performance measures will only represent travel plans for 336,636 trips. If for example, 50% of the input trips had problems then the network performance measures produced by the Microsimulator would only reflect a partially loaded transportation system.

The PROBLEM field identifies the problem code for a given record. The problem codes are the same as those presented earlier in this How-To. An interpretation of all the messages generated by this program is provided in Section 3.3 of this document.

3.3 Problem Types

If a problem is encountered during the path building process, a message is written to the Problem file and added to the summary statistics. The most likely explanation for a given problem message and strategy for resolving the problem is listed below:

1. *Path Building Problem* – This message typically means that there is no feasible path between the origin and destination. It could be caused by lane connectivity or one-way street conditions or by another network coding error.

2. *Time Schedule Problem* – This message indicates that the trip travel time exceeded the upper bound of the activity start time. It could be caused by excessive congestion or no path options.

3. *Zero Node Path Problem* – The zero-node error occurs when the origin and the destination activity locations lie on the same link and node list plans are requested. Zero Node Path problems can be eliminated by saving link list plans using the NODE_LIST_PATHS (true/false) control key parameter.

4. *Vehicle Type Problem* – The origin parking lot is located on a link that does not permit the corresponding vehicle type. This most often occurs when autos are loaded to transit only links or trucks to auto only links. Check the 'USE' variable in the network link file to identify permitted vehicle types for each link.

5. *Path Circuity Problem* – A circuity error indicates that the path building process was limited by one or more of the circuity parameters. It either means that a path does not exist or the path is highly circuitous. The user can set the maximum circuity ratio parameter to zero to eliminate these messages. If a path does not exist, a path building or time schedule message will be generated.

6. *Travel Mode Problem* – The Router records a travel mode error when the mode on the activity or trip file cannot be built. This generally means that the transit, walk, or bike networks have not been enabled.

7. *Vehicle Access Problem* – An access error is generated when the vehicle listed in the activity or trip file is not found in the vehicle file or when the vehicle is located at a parking lot that is not attached to the activity location with a process link.

8. *Walk Distance Problem* – This message is generated when the cumulative walk distance required by the path exceeds the MAX_WALK_DISTANCE control key parameter.

9. *Wait Time Problem* – This message indicates that potential transit routes exist, but the wait time required to board the routes exceed the MAX_WAIT_TIME control key parameter.

10. *Walk Access Problem* – This message is generated when the link associated with the origin or destination activity location does permit travel by the chosen mode. It most often indicates a walk or bike access restriction at one of the trip ends.

11. *Path Size Problem* – This message is generated when the number of links or nodes in the path exceeds the maximum path size. The maximum path size parameter is defined in the Network Size file and defaults to the number of network nodes. Assuming the maximum path size is a relatively large number, this error typically indicates a problem in the path-building algorithm that resulted

in an infinitely looping path. This can only occur if the percent random impedance is high. Re-running the Router will most likely eliminate the problem.

12. *Park & Ride Lot Problem* – In order to build a park-&-ride trip (mode 5), a parking lot must be designated with the PARKRIDE style in the general proximity of the trip origin. The MAX_PARK_RIDE_PERCENTAGE parameter determines how far away from the origin the software can search for possible park-&-ride lots. If no lots are found within the search area, the park-&-ride lot error message is recorded.

13. *Bicycle Distance Error* – This message is generated when the bicycling distance exceeds the MAX_BICYCLE_DISTANCE control key parameter.

14. *Departure Time Problem* – This message indicates that a trip could not be started within the MAX_DEPARTURE_TIME_VARIANCE parameter (in minutes) from the stipulated start time in the plan file. It is typically caused by broken parking lot access, vehicle access restriction or heavy congestion on the link adjacent to the parking lot.

15. *Arrival Time Problem* – This message means that a trip could not be successfully completed within the MAX_ARRIVAL_TIME_VARIANCE parameter (in minutes) from the stipulated arrival time in the plan file. It is typically caused by congestion or access restrictions on the destination link or parking lot.

16. *Link Access Problem* – This message indicates that one or more links in the path are restricted. This could happen due to the difference in the time-of-day when the Router and Microsimulator expect the given link(s) to be used given the presence of time-of-day 'Lane_Use' controls on the link.

17. *Lane Connectivity Problem* – This is an important message indicating broken lane-connections between links. Typically this is caused when older plans are used on newer networks which are not completely compatible. The vehicles that encounter such a situation will remain stuck on the last link and choke the traffic until they exceed the MAXIMUM_WAITING_TIME parameter, thus badly impacting the simulation. Lane connectivity problems should be fixed.

18. *Parking Access Problem* – This message is generated when the vehicle cannot access the destination parking lot at the end of its path. This can happen if the destination parking lot is not the last link of the path or it is not within two lanes of the vehicles position when the ENFORE_PARKING_LANES parameter is active.

19. *Lane Merging Problem* – This message is issued in the Microsimulator when the instantaneous gap-distance is smaller than the instantaneous stopping distance for a vehicle maneuvering a lane-merge. There can be multiple messages for a given

vehicle over the course of its simulation. This could typically happen in heavily congested situations and simulations with multi-cell vehicles.

20. *Lane Changing Problem* – This message is similar to the *Lane Merging Problem* and is issued when the gap-distance is smaller than the instantaneous stopping distance for a vehicle attempting a lane-change maneuver. There can be multiple messages for a given vehicle over the course of the simulation. This could typically happen in heavily congested situations and simulations with multi-cell vehicles.

21. *Turning Speed Problem* – This message is issued when a vehicle cannot maintain a safe gap distance after making a turning movement. The vehicle then undergoes abrupt dynamics to prevent a collision. There can be multiple messages for a given vehicle over the course of the simulation. This could typically happen in heavily congested situations and simulations with multi-cell vehicles.

22. *Pocket Merge Problem* – This message is issued when a vehicle cannot maintain a safe gap while approaching the end of a merge-pocket. The vehicle then undergoes abrupt dynamics to prevent a collision. There can be multiple messages for a given vehicle over the course of the simulation. This could typically happen in heavily congested situations and simulations with multi-cell vehicles.

23. *Vehicle Spacing Problem* – This message is issued when a vehicle cannot maintain a safe gap between itself and the vehicle in front of it. The vehicle then undergoes abrupt dynamics to prevent a collision. There can be multiple messages for a given vehicle over the course of the simulation. This could typically happen in heavily congested situations and simulations with multi-cell vehicles.

24. *Traffic Control Problem* – This message indicates that a vehicle could not come to a complete stop at signal or stop controlled intersections. The vehicle is then brought to a complete stop and this message is issued. There can be multiple messages for a given vehicle over the course of the simulation.

25. *Access Restriction Problem* – This message is related to the *Link Access Problem*. If a vehicle is able to make a lane-change and avoid a potential *Link Access Problem* then only this message is issued. This typically happens at turning movements when the lanes on the destination link are inaccessible due to 'Lane_Use'controls. The presence of time-of-day parking lanes also typically can cause these messages. These messages can be given a relatively low priority compared to other problem message types.

26. *Transit Stop Problem* – If a transit vehicle cannot come to a complete stop as required at a transit-stop, this message is issued.

27. *Activity Location Problem* – This message is issued when either the origin or the destination activity location of a trip cannot be found in the activity location file.

28. *Vehicle Passenger Problem* – If the vehicle is not available for a passenger trip, this message is issued. This could typically happen if the driver of the vehicle encounters problems during the simulation. Such passenger trips can be “magic moved” to their destination if their travel mode is set to mode 8.

29. *Activity Duration Problem* – This message is issued in the ActGen and LocationChoice programs when duration of the activity is negative or in other words, if a subsequent activity is scheduled to begin before the end of the preceding activity.

30. *Kiss-&-Ride Lot Problem* – This message is issued when a suitable Kiss-&-Ride lot cannot be located for a building a Kiss-&-Ride path under the specified parameters in the Router control.

31. *Vehicle ID Problem* – This message indicates that the traveler’s expected vehicle cannot be found in the vehicle file or when the vehicle ID is negative. These problems are not generated when the IGNORE_VEHICLE_ID parameter is activated in the Router.

32. *Data Sort Problem* – This message indicates that the input activities are not in time-sorted order.

33. *Walk Location Problem* – This message is issued as a sub-classification of the *Activity Location Problem*. It indicates that the mode of travel is walk and the origin or destination locations cannot be found in the activity location file.

34. *Bike Location Problem*– This message is issued as a sub-classification of the *Activity Location Problem*. It indicates that the mode of travel is bike and the origin or destination locations cannot be found in the activity location file.

35. *Transit Location Problem* – This message is issued as a sub-classification of the *Activity Location Problem*. It indicates that the mode of travel is transit and the origin or destination locations cannot be found in the activity location file.

36. *Person Match Problem* – This message is issued when a person cannot be matched to the survey population after all the trials.

37. *Transit Capacity Problem* – This message is issued when a transit person could not board its bus or train because the bus or train was at its maximum capacity. These messages give a sense of capacity needed on transit systems under the given demand.

38. *Transit Dwell Problem* – This message is issued when a transit person could not board their bus or train because the bus or train closes its doors to further boarding after the MAX_DWELL parameter in the vehicle type file has been

exceeded. These messages give a sense of the residual demand at transit stop or stations.

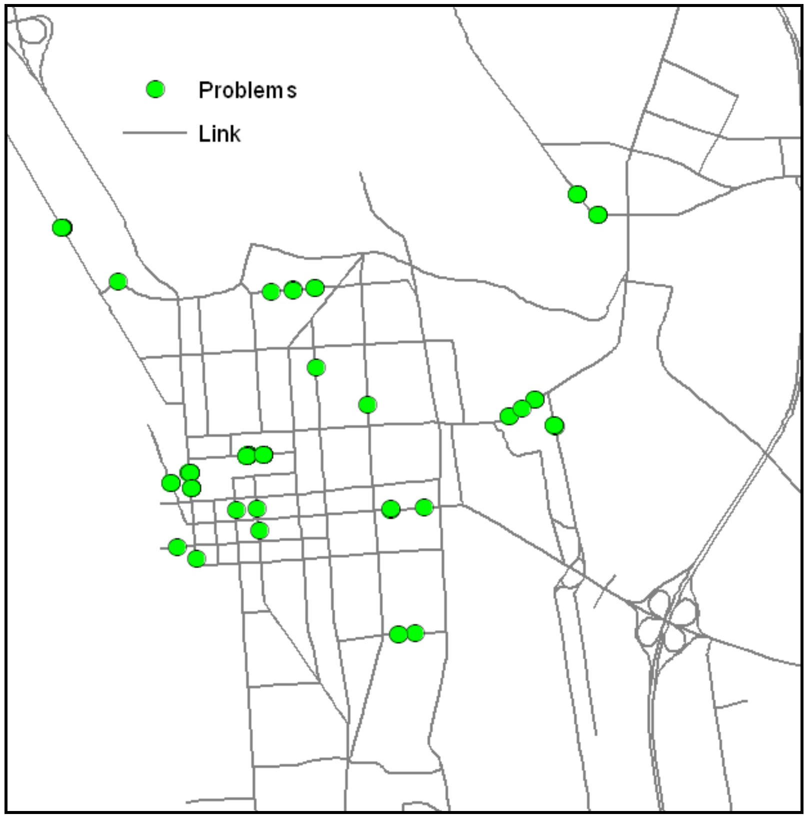
3.4 Using ArcProblem

The ArcProblem function can be used to visualize the location of both Router and Microsimulator problems. The ArcProblem utility can create Arcview shapefiles from problem files generated by the Router based on the trip origin and destination activity locations. The function can also create Arcview shapefiles from problem files generated by the Microsimulator based on the link offset and lane where the problem occurred during the simulation with the ability to draw them as vehicle polygons or points. Problems can also be selected by type, time period, and/or subarea polygon. The ArcProblem utility can be very useful in resolving problems especially if they are network related.

A sample ArcProblem control file is shown below.

```
TITLE                            Make Problem shapefile

#---- Input Files ----

NET_NODE_TABLE                   network\Node
NET_LINK_TABLE                   network\Link
NET_Shape_TABLE                  network\Shape

PROBLEM_FILE                     plans\Router_Problems

#---- Output Files ----

ARCVIEW_PROBLEM_FILE             plans\Router_Problems.shp
```

The figure below shows shapefile representations of the TRANSIMS link file generated using the ArcNet function and a shapefile representation of the problem file generated using the ArcProblem function.

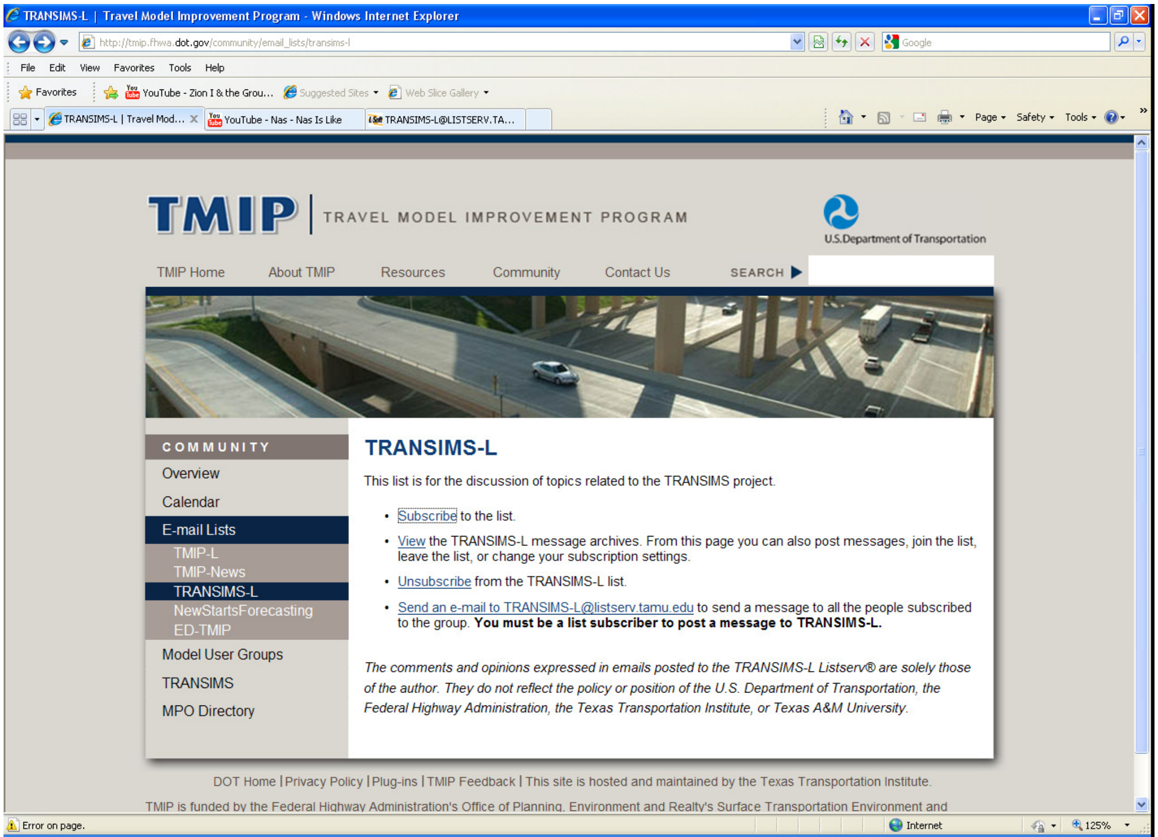


4.0 TRANSIMS Email Listserv

An email listserv is maintained by Texas A&M University (TAMU) to provide a forum for TRANSIMS users. The developers of the TRANSIMS software along with a dedicated group of users who have wide ranging experience developing model implementations monitor the TRANSIMS-L listserv to assist and support the growing user's community. There are currently about 200 subscribers to the listserv. This is a great place to seek guidance and assistance from a number of individuals who are more than eager to help. A response to a posting to the listserv describing a particular problem, error and/or software bug is often provided by the community within hours.

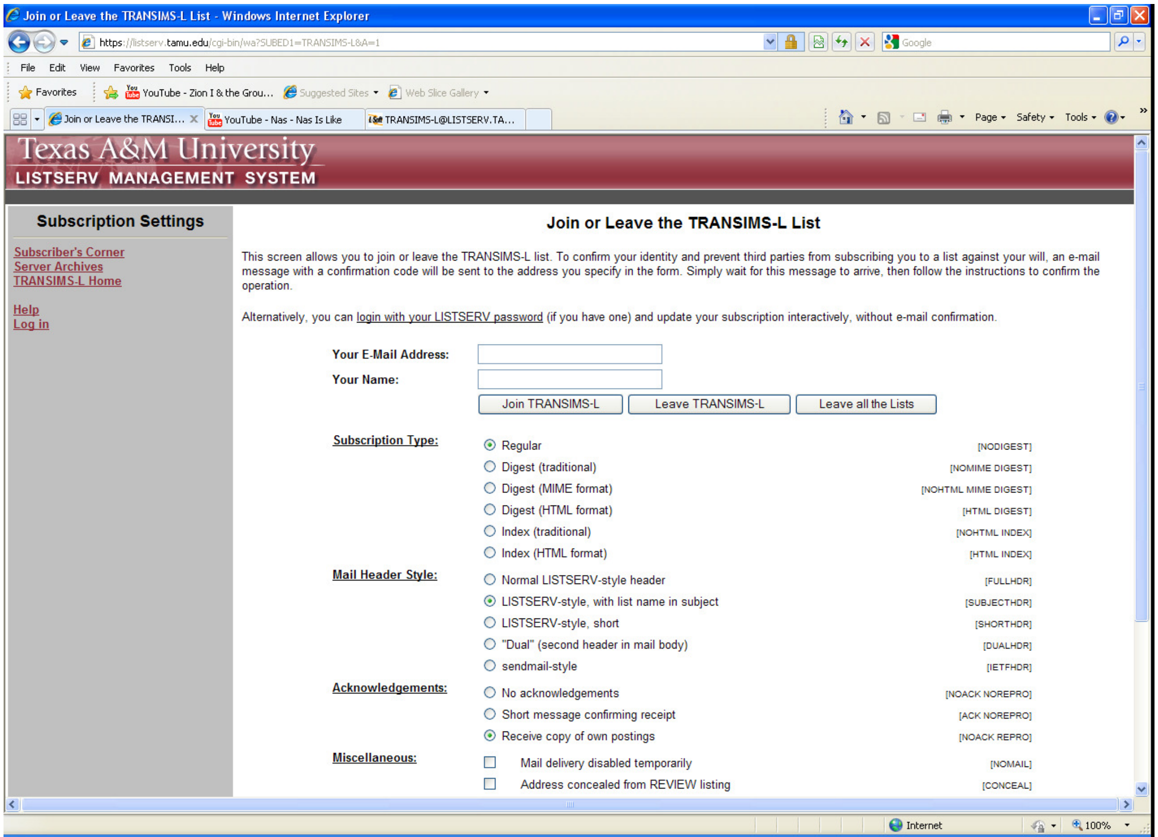
To subscribe to the TRANSIMS-L listserv go to the following link and 'Subscribe':

http://tmip.fhwa.dot.gov/community/email_lists/transims-l



You can also access the TAMU listserv management system directly at:

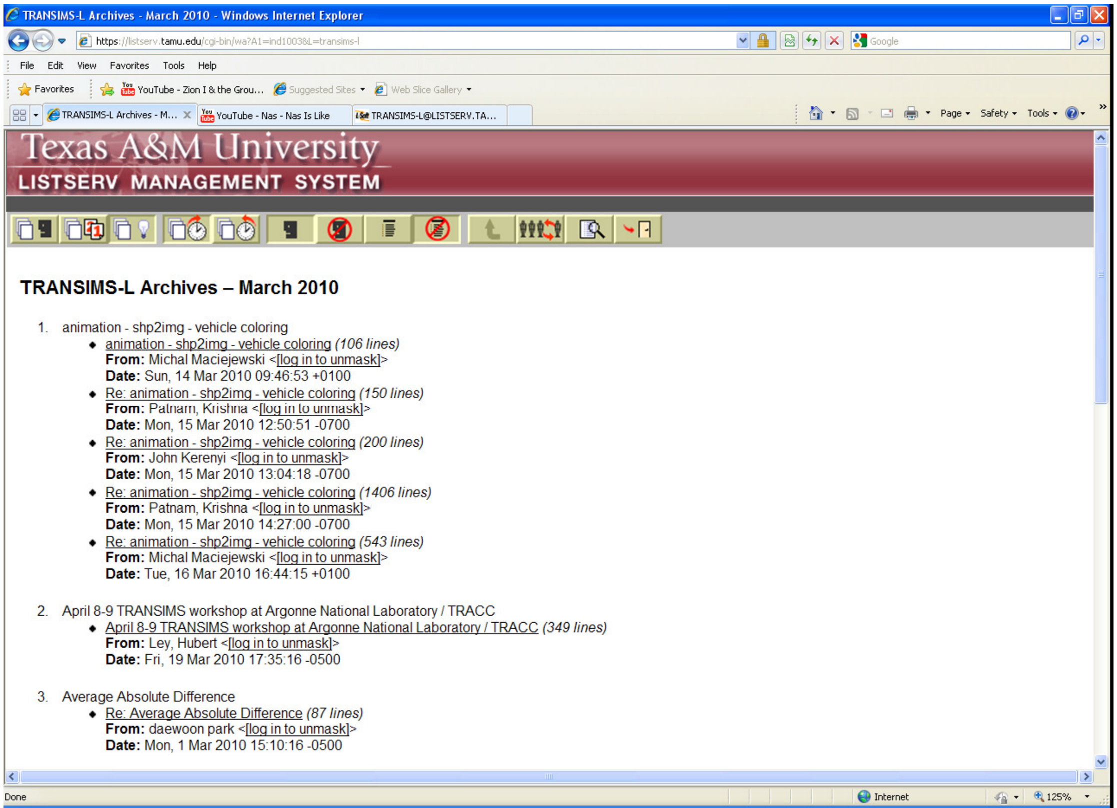
https://listserv.tamu.edu/cgi-bin/wa?SUBED1=TRANSIMS-L&A=1



The TAMU listserv system also maintains a searchable web archive which allows users to search all emails posted to the listserv since its inception. Therefore a user can see all the questions submitted to the listserv and the responses provided by other users. The web archive contains postings that date back to May 2005.

The listserv archive can be accessed at:

http://listserv.tamu.edu/archives/transims-l.html

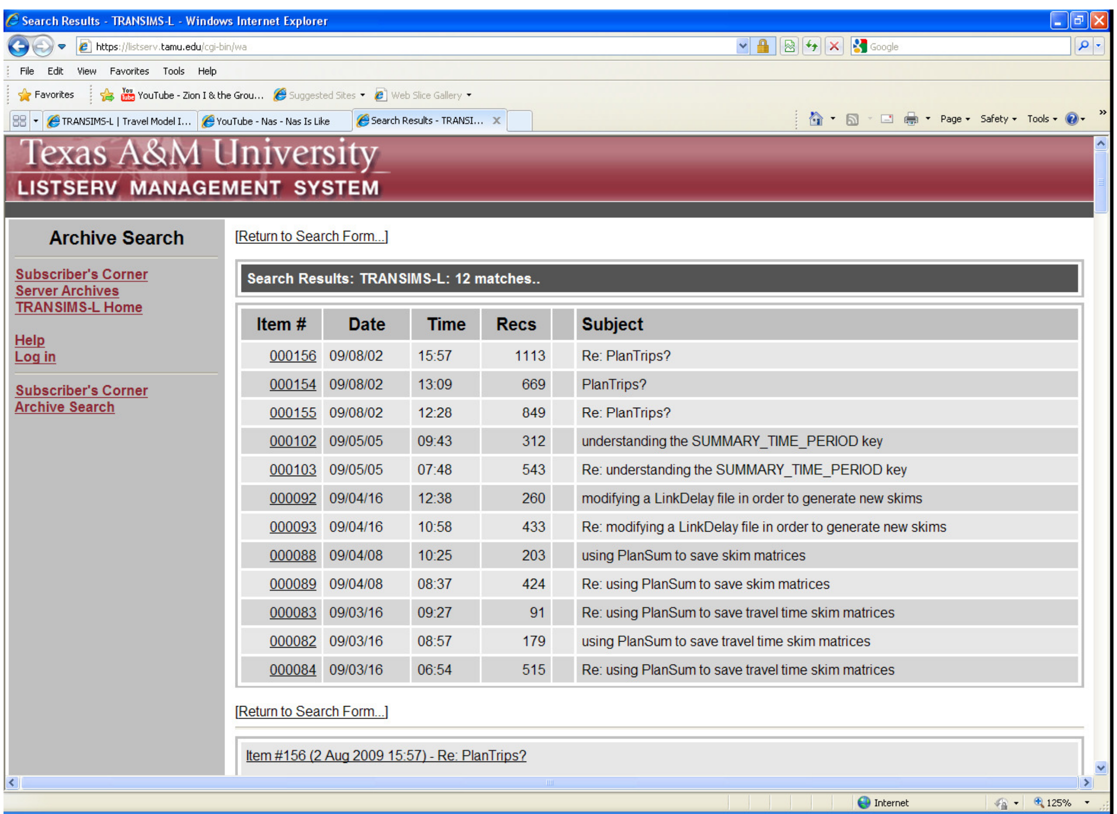


TRANSIMS-L Archives - March 2010 - Windows Internet Explorer

Texas A&M University
LISTSERV MANAGEMENT SYSTEM

TRANSIMS-L Archives – March 2010

1. animation - shp2img - vehicle coloring
   - animation - shp2img - vehicle coloring *(106 lines)*
     From: Michal Maciejewski <[log in to unmask]>
     Date: Sun, 14 Mar 2010 09:46:53 +0100
   - Re: animation - shp2img - vehicle coloring *(150 lines)*
     From: Patnam, Krishna <[log in to unmask]>
     Date: Mon, 15 Mar 2010 12:50:51 -0700
   - Re: animation - shp2img - vehicle coloring *(200 lines)*
     From: John Kerenyi <[log in to unmask]>
     Date: Mon, 15 Mar 2010 13:04:18 -0700
   - Re: animation - shp2img - vehicle coloring *(1406 lines)*
     From: Patnam, Krishna <[log in to unmask]>
     Date: Mon, 15 Mar 2010 14:27:00 -0700
   - Re: animation - shp2img - vehicle coloring *(543 lines)*
     From: Michal Maciejewski <[log in to unmask]>
     Date: Tue, 16 Mar 2010 16:44:15 +0100
2. April 8-9 TRANSIMS workshop at Argonne National Laboratory / TRACC
   - April 8-9 TRANSIMS workshop at Argonne National Laboratory / TRACC *(349 lines)*
     From: Ley, Hubert <[log in to unmask]>
     Date: Fri, 19 Mar 2010 17:35:16 -0500
3. Average Absolute Difference
   - Re: Average Absolute Difference *(87 lines)*
     From: daewoon park <[log in to unmask]>
     Date: Mon, 1 Mar 2010 15:10:16 -0500



Search Results - TRANSIMS-L - Windows Internet Explorer

Texas A&M University
LISTSERV MANAGEMENT SYSTEM

Archive Search

Subscriber's Corner
Server Archives
TRANSIMS-L Home

Help
Log in

Subscriber's Corner
Archive Search

[Return to Search Form...]

Search Results: TRANSIMS-L: 12 matches..

Item #	Date	Time	Recs	Subject
000156	09/08/02	15:57	1113	Re: PlanTrips?
000154	09/08/02	13:09	669	PlanTrips?
000155	09/08/02	12:28	849	Re: PlanTrips?
000102	09/05/05	09:43	312	understanding the SUMMARY_TIME_PERIOD key
000103	09/05/05	07:48	543	Re: understanding the SUMMARY_TIME_PERIOD key
000092	09/04/16	12:38	260	modifying a LinkDelay file in order to generate new skims
000093	09/04/16	10:58	433	Re: modifying a LinkDelay file in order to generate new skims
000088	09/04/08	10:25	203	using PlanSum to save skim matrices
000089	09/04/08	08:37	424	Re: using PlanSum to save skim matrices
000083	09/03/16	09:27	91	Re: using PlanSum to save travel time skim matrices
000082	09/03/16	08:57	179	using PlanSum to save travel time skim matrices
000084	09/03/16	06:54	515	Re: using PlanSum to save travel time skim matrices

[Return to Search Form...]

Item #156 (2 Aug 2009 15:57) - Re: PlanTrips?