TRANSIMS Training Course at TRACC

Transportation Research and Analysis Computing Center

Part 9

TRANSIMS Control Files

Dr.-Ing. Hubert Ley

Transportation Research and Analysis Computing Center

Last Updated: December 14, 2009

Unit **9**

Contents

- TRANSIMS Control File History
- Invocation of TRANSIMS Tools
- TRANSIMS Control File Semantics
- Typical Problems
- Log Files
- Network File Specifications and the Project Directory
- Arrays of Control Keys and Value Lists
- A Sample Control File for the Router

TRANSIMS Control File History

- The specific implementation of the control keys in TRANSIMS dates back to early developments
- TRANSIMS used to allow for include file logic, which could lead to a level of confusion about settings such as the project directory or network directory
- It was also common practice to place the control keys for many tools into a single file, having unrelated executables disregard the keys that they would not use internally
- Current practice is to provide a single control file for each invocation of a tool. A new option "-K" permits the user to receive warnings for any key that is specified in the control file but is not used by the tool.

Invocation of TRANSIMS Tools

Router –K –B Router.ctl

- This will execute the router based on the keys specified in the Router.ctl control file

Router

- If no control file is specified, the tool will ask for the name of a control file interactively.

Router Router.ctl 3

- If a number is specified at the end, the control file is interpreted differently. Partitionable files are opened using special extensions such as ".tAA", ".tAB", based on the file specifications given in the control files

Router -H

- The –H option lists all allowable control keys for the executable

TRANSIMS Control File Semantics

- Control files follow these specific semantics:
  - Both DOS/Windows and Linux text files are allowed on both platforms
  - Empty lines (white space) are allowable anywhere in the control file
  - The “#” sign at the beginning of the line marks the line as a comment
  - All input is provided as pairs of a control key and a corresponding value or list of values (key/value pair)
  - Comments are not allowed at the end of a key/value pair because the text will become part of the value (it may work at times though)
  - A key/value pair is separated by white space, meaning either tab characters or blanks. The value starts with the first non-white character and ends at the end of the line.
  - Control keys are all upper case and are case sensitive. Values are mixed case or numeric and may also be case sensitive.
  - Boolean keys take the values true/false/yes/no/1/0 in both upper and lower case

Typical Problems

- Misspelled key names
  - Make sure all keys are spelled in upper case
  - Add the –K option to the executable to turn key checking on (this will create warnings for keys that are misspelled or that are not used by the current tool)
  - Use the –H option to learn about applicable keys

- Misinterpreted values
  - Each key’s value is interpreted differently. It is important to understand the range of allowed values that a key may accept. Integers are typically allowable for keys expecting floating point values.
  - The units used are usually SI units, but care must be taken to provide some of the keys with the correct value. There are in particular many different ways how times are specified for different keys (seconds, minutes, hours, 24hour formats, and more)

Typical Problems

- Verification of Values
  - The TRANSIMS executables report the values provided in control files in their log file (".prn" by default). The names are not intuitively connected with the control keys and may use different units than the ones used in the control file.
- Default values
  - It is important to understand the defaults chosen for certain values if they are not specified in the control file. The use of certain keys turns defaults for other keys on, which may then be overwritten by specification in the control file.

Log Files

- The invocation of any TRANSIMS tool will write a log file
- The log file is written by default into the same directory where the corresponding control file is located
- The file name is derived from the control file name with the new extension ".prn"
- The log file name can be specified through the first control key in the control file to choose a different name:
  - REPORT_FILE ..\log\Router.log
- When running in partition mode, the log file will be named according to the partition number, e.g. "Router_3.prn"
- This convention failed until recently if the "REPORT_FILE" key was specified. In this case, all partitions used the same specified log file name. This has been fixed in the latest versions of the sofwtare.
- These log files can be opened in "append" mode by specifying the "REPORT_FLAG" key.

Network File Specifications and the Project Directory

- Each network table must be specified as file
- A common directory for network files can be defined using the "NET_DIRECTORY" key
- A common project directory can be specified in some tools using the "PROJECT_DIRECTORY" key. This is prepended to all non-network file specifications.
- All directory and file name logic and concatenation is done using simple string operations and is not "path-logic" aware. If not used with care, complex directory and file names may lead to invalid paths.

Arrays of Control Keys and Value Lists

- Some control keys are expected to be specified as array elements
- The syntax is somewhat unusual and is expressed using underscore characters for historical reasons:
  - Example: OUTPUT_SUMMARY_FILE_1 LinkDelay
- The example above indicates the first output summary file name
- Up to 8 dimensions are theoretically allowed, but a maximum of three is used.
- Some keys expect a comma or otherwise separated list of values, e.g.
  - Example: ROUTE_SELECTED_MODES 1,2,3,4,5,6,7,8,9

A Sample Control File for the Router

```
TITLE                               Mode Router Test
#---- Network Files -----#
NET_DIRECTORY                       /u1/projects/mpo5/network
NET_SIZE_TABLE                      Size
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
NET_TRANSIT_STOP_TABLE              Transit_Stop
NET_TRANSIT_FARE_TABLE              Transit_Fare
NET_TRANSIT_ROUTE_TABLE             Transit_Route
NET_TRANSIT_SCHEDULE_TABLE                  Transit_Schedule
#---- Input Files ----#
LINK_DELAY_FILE             /u1/projects/mpo5/msim/LinkDelay.90.tim
HOUSEHOLD_LIST              /u1/projects/mpo5/population/hhold_list.100
VEHICLE_FILE                /u1/projects/mpo5/vehicle/Total_Vehicle
ACTIVITY_FILE               /u1/projects/mpo5/activity/Total_Activity
```

A Sample Control File for the Router

```
#---- Output Files ----#
NEW_PLAN_FILE               /u1/projects/mpo5/plans/Total_Plans.100
NEW_PROBLEM_FILE            /u1/projects/mpo5/output/Problems.100
#---- Options -----#
NODE_LIST_PATHS                     YES
ROUTE_SELECTED_MODES                1,2,3,4,5,6,7,8,9
ROUTE_WITH_SPECIFIED_MODE           3
LIMIT_PARKING_ACCESS                YES
IGNORE_TIME_CONSTRAINTS NO
PERCENT_RANDOM_IMPEDANCE            20
RANDOM_NUMBER_SEED                  0
#---- Parameters -----#
WALK_SPEED                          1.0
BICYCLE_SPEED                       4.0
WALK_TIME_VALUE                     20.0
BICYCLE_TIME_VALUE                  15.0
FIRST_WAIT_VALUE                    20.0
TRANSFER_WAIT_VALUE                 20.0
VEHICLE_TIME_VALUE                  10.0
DISTANCE_VALUE                      0.0
```

A Sample Control File for the Router

```
COST_VALUE                144.0 #---- $5/hour of walk time ----
TRANSFER_PENALTY                0.0
RAIL_BIAS_FACTOR                1.0
#---- Constraints -----#
MAX_WALK_DISTANCE               2000
MAX_BICYCLE_DISTANCE            10000
MAX_WAIT_TIME                   60
MAX_NUMBER_OF_TRANSFERS                 5
MAX_PARK_RIDE_PERCENTAGE                50
MAX_CIRCUITY_RATIO                      2.0
MIN_CIRCUITY_DISTANCE                   2000
MAX_CIRCUITY_DISTANCE                   20000
MAX_ROUTING_ERRORS                      100000
MAX_LINK_DELAY_ERRORS                   100000
#----- V/C Controls ----#
LINK_DELAY_UPDATE_RATE                  0
EQUATION_PARAMETERS_1                   BPR, 0.15, 4.0, 0.75
```

Credits and Acknowledgements

- GIS visualization materials were mostly developed at Argonne based on the TRANSIMS tools developed by AECOM for USDOT
- Chicago road and transit network data used in some of the examples was provided by the Chicago Metropolitan Agency for Planning
- USDOT provided the funding for the development of these training materials
- USDOT provided the funding for the TRACC computing center and the resources necessary to perform these training session