TRANSIMS Version 5
File Reference

Table of Contents - Yellow shading indicates files that have not yet been defined

List of Figures

List of Tables

ACCESS_FILE

Names: ACCESS_FILE and NEW_ACCESS_FILE

Used In:

ArcNet
ConvertTrips Default Control Key
LocationData
Microsimulator
NewFormat
PathSkim
Router

TransimsNet

In TRANSIMS Version 5, Process Links (Version 4) are replaced by Access Links (Version 5). Some important functional differences exist between the two link types as well. In Version 4, process links are required to connect activity locations to parking lots, and to connect activity locations to transit stops. For vehicles to be loaded onto the network they must move from parking lots to activity locations to the actual link/road via process links which are located on the network as paired, one-way links on either side of the actual network link (see illustration below). In Version 5, process links are no longer needed. Instead, link-offsets are used to build direct, two-way connections for loading and unloading of vehicles from the network. Walk links now have travel time, distance, and cost associated with them in Version 5. In addition, transit stops no longer need activity locations, and activity locations with a zone number equal to zero are deleted. In contrast with process links, access links are only used for special connections (as two-way or one-way links), between nodes, locations, parking lots and transit stops Figure 1.

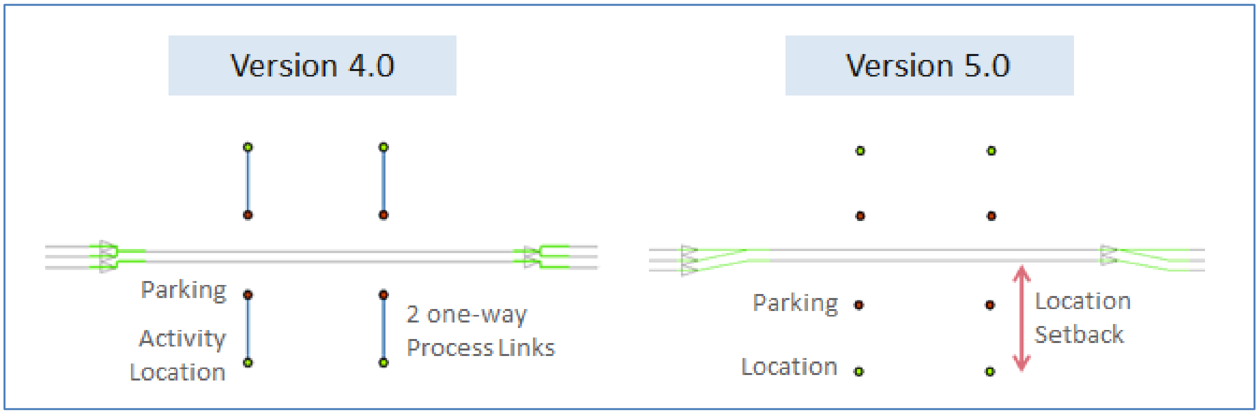


Figure 1 Version 4 Process Links do not always need version 5 access links

ACTIVITY_FILE

Name: ACTIVITY_FILE

Used In New Format

ARC_ACCESS_FILE

Names: ARC_ACCESS_FILE, NEW_ARC_ACCESS_FILE

Used in ArcNet

ARC_ACCESSIBILITY_FILE

Names: ARC_ACCESSIBILITY_FILE, NEW_ARC_ACCESSIBILITY_FILE

Used in ArcPlan

ARC_ BANDWIDTH_FILE

Names: ARC_ BANDWIDTH_FILE, NEW_ARC_ BANDWIDTH_FILE
Used in ArcPlan

ARC_CENTERLINE _FILE

Names: ARC_ CENTERLINE _FILE, NEW_ARC_ CENTERLINE _FILE
Used in ArcNet

ARC_CONNECTION_FILE

Names: ARC_ CONNECTION _FILE, NEW_ARC_ CONNECTION _FILE
Used in ArcNet

ARC_ DETECTOR_FILE

Names: ARC_ DETECTOR _FILE, NEW_ARC_ DETECTOR _FILE
Used in ArcNet

ARC_DISTANCE_CONTOUR _FILE

Names: ARC_ DISTANCE_CONTOUR _FILE, NEW_ARC_ DISTANCE_CONTOUR _FILE
Used in ArcPlan

ARC_ LANE_USE _FILE

Names: ARC_ LANE_USE _FILE, NEW_ARC_ LANE_USE _FILE
Used in ArcNet

ARC_ LINK_FILE

Names: ARC_ LINK _FILE, NEW_ARC_ LINK _FILE
Used in ArcNet

ARC_LOCATION _FILE

Names: ARC_ LOCATION _FILE, NEW_ARC_ LOCATION _FILE
Used in ArcNet

ARC_ NODE_FILE

Names: ARC_NODE _FILE, NEW_ARC_NODE _FILE
Used in ArcNet

ARC_ PARKING_DEMAND_FILE

Names: ARC_ PARKING_DEMAND _FILE, NEW_ARC_ PARKING_DEMAND _FILE
Used in ArcPlan

ARC_ PARKING_FILE

Names: ARC_ PARKING _FILE, NEW_ARC_ PARKING _FILE
Used in ArcNet

ARC_ PHASING_PLAN _FILE

Names: ARC_ PHASING_PLAN _FILE, NEW_ARC_ PHASING_PLAN _FILE
Used in ArcNet

ARC_ PLAN_FILE

Names: ARC_ PLAN_FILE, NEW_ARC_PLAN _FILE
Used in ArcPlan

The ArcView plan file key specifies the file name for the output Arcview shape file. The file name should end with ".shp". The program automatically creates three files in the output directory. These are the ArcView shape file with the ".shp" extension, the ArcView index file with a ".shx" extension, and the ArcView data file with a ".dbf" extension. All three files are required for ArcView or ArcMap to read and display the path.

ARC_ POCKET_FILE

Names: ARC_ POCKET_FILE, NEW_ARC_POCKET _FILE
Used in ArcNet

ARC_PROBLEM _FILE

Names: ARC_ PROBLEM_FILE, NEW_ARC_PROBLEM _FILE
Used in ArcPlan

ARC_RIDERSHIP _FILE

Names: ARC_ RIDERSHIP _FILE, NEW_ARC_ RIDERSHIP _FILE
Used in ArcPlan

The ArcView transit ridership file key is optional. If provided, the key value is appended to the PROJECT_DIRECTORY key to specify the file name for the output Arcview shape file. The file name should end with ".shp". The program automatically creates three files in the output directory. These are the ArcView shape file with the ".shp" extension, the ArcView index file with a ".shx" extension, and the ArcView data file with a ".dbf" extension. All three files are required for ArcView or ArcMap to read and display the path.

ARC_ ROUTE_NODES _FILE

Names: ARC_ ROUTE_NODES _FILE, NEW_ARC_ ROUTE_NODES _FILE
Used in ArcNet

ARC_ SIGN_FILE

Names: ARC_SIGN _FILE, NEW_ARC_ SIGN_FILE
Used in ArcNet

ARC_ SIGNAL _FILE

Names: ARC_ SIGNAL _FILE, NEW_ARC_ SIGNAL _FILE
Used in ArcNet

ARC_ SNAPSHOT _FILE

Names: ARC_ SNAPSHOT _FILE, NEW_ARC_ SNAPSHOT _FILE
Used in ArcSnapshot

ARC_ STOP_DEMAND _FILE

Names: ARC_ STOP_DEMAND _FILE, NEW_ARC_ STOP_DEMAND _FILE
Used in ArcPlan

The ArcView transit stop demand file key is optional. If provided, the key value is appended to the PROJECT_DIRECTORY key to specify the file name for the output Arcview shape file. The file name should end with ".shp". The program automatically creates three files in the output directory. These are the ArcView shape file with the ".shp" extension, the ArcView index file with a ".shx" extension, and the ArcView data file with a ".dbf" extension. All three files are required for ArcView or ArcMap to read and display the path.

ARC_ STOP_GROUP _FILE

Names: ARC_ STOP_GROUP _FILE, NEW_ARC_ STOP_GROUP _FILE
Used in ArcPlan

The arc transit stop group file key is optional. If provided, the key value is appended to the PROJECT_DIRECTORY key to specify the file name for the output Arcview shape file. The file name should end with ".shp". The program automatically creates three files in the output directory. These are the ArcView shape file with the ".shp" extension, the ArcView index file with a ".shx" extension, and the ArcView data file with a ".dbf" extension. All three files are required for ArcView or ArcMap to read and display the path.

ARC_ SUBZONE_DATA _FILE

Names: ARC_ SUBZONE_DATA _FILE, NEW_ARC_ SUBZONE_DATA _FILE
Used in ArcNet

ARC_ TIME_CONTOUR _FILE

Names: ARC_ TIME_CONTOUR _FILE, NEW_ARC_ TIME_CONTOUR _FILE
Used in ArcPlan

The arc time contour file key is optional. If provided, the key value is appended to the PROJECT_DIRECTORY key to specify the file name for the output Arcview shape file. The file name should end with ".shp". The program automatically creates three files in the output directory. These are the ArcView shape file with the ".shp" extension, the ArcView index file with a ".shx" extension, and the ArcView data file with a ".dbf" extension. All three files are required for ArcView or ArcMap to read and display the path.

ARC_ TIMING_PLAN _FILE

Names: ARC_ TIMING_PLAN _FILE, NEW_ARC_ TIMING_PLAN _FILE
Used in ArcNet

ARC_ TRANSIT_DRIVER _FILE

Names: ARC_ TRANSIT_DRIVER _FILE, NEW_ARC_ TRANSIT_DRIVER _FILE
Used in ArcNet

ARC_ TRANSIT_ROUTE _FILE

Names: ARC_ TRANSIT_ROUTE _FILE, NEW_ARC_ TRANSIT_ROUTE _FILE
Used in ArcNet

ARC_ TRANSIT_STOP _FILE

Names: ARC_ TRANSIT_STOP _FILE, NEW_ARC_ TRANSIT_STOP _FILE
Used in ArcNet

ARC_ TURN_PENALTY _FILE

Names: ARC_ TURN_PENALTY _FILE, NEW_ARC_ TURN_PENALTY _FILE
Used in ArcNet

ARC_ ZONE _FILE

Names: ARC_ ZONE _FILE, NEW_ARC_ ZONE _FILE
Used in ArcNet

COMPARE_PERFORMANCE_FILE

Used in LinkSum

COMPARE_PLAN_FILE

Used in PlanCompare

Configuration File

Used by all programs (global settings; can be over-ridden by local settings)

In most TRANSIMS applications there are a significant number of keys that are common to all programs. Many of the Execution Service keys fall into this category. They tend to be global keys that define the default behavior of the model. If the modeler wishes to set these keys once and use them in all model applications, a TRANSIMS configuration file can be created. A configuration file is exactly like any other control file and can include any number of control keys and key values. Each TRANSIMS program looks for a configuration file using the operating system environment variable TRANSIMS_CONFIG_FILE. The variable points to a file name that stores the configuration keys. The program reads the configuration keys into memory before it reads the control file keys. If a control key is defined in both files, the value from the control file will override the value in the configuration file.

The path to a configuration file can be set dynamically for a particular application using the SET command within a batch file or at the command prompt. For example:

```
SET_TRANSIMS_CONFIG_FILE=c:\myproject\config.txt
```

CONNECTION_FILE

Names: CONNECTION_FILE, NEW_CONNECTION_FILE

Used In:

ArcNet
ArcPlan
ArcSnapshot
IntControl **(Required)**
LinkDelay
LinkSum
Microsimulator
NewFormat
PathSkim
PlanSelect
PlanSum
Router

TransimsNet

Formerly known as LANE_CONNECTIVITY, this is a list of intersection connections in the network. A typical field definition (.def or .DEF) file is as follows:

```
TRANSIMS50, TAB_DELIMITED, 1
LINK, INTEGER, 1, 10
DIR, INTEGER, 2, 1
TO_LINK, INTEGER, 3, 10
LANES, STRING, 4, 8, LANE_RANGE_TYPE
TO_LANES, STRING, 5, 8, LANE_RANGE_TYPE
TYPE, STRING, 6, 8, CONNECTION_TYPE
PENALTY, UNSIGNED, 7, 5, IMPEDANCE
SPEED, DOUBLE, 8, 5.1, KPH
CAPACITY, UNSIGNED, 9, 8, VPH
NOTES, STRING, 10, 128
```

The fields are defined as follows:

Table 1 CONNECTION_FILE Field Definitions

Field(s)	Description	Use	Default Units
LINK	The link number (an integer)	Key	
DIR	Direction on the link AB=0, BA=1	Req.	
TO_LINK	Outbound Link	Req.	
LANES	Range of inbound lanes, numbered from right to left	Req.	Note 1
TO_LANES	Range of outbound lanes, numbered from right to left	Req.	Note 1
TYPE	Connection Type	Req.	Note 2
PENALTY	Penalty for the movement	Opt.	Seconds
SPEED	Maximum turning speed	Opt.	m/s
CAPACITY	Hourly vehicle capacity for the turn	Opt.	veh/hr
NOTES	Character string for user notes	Opt.	

Note 1: Could either be a single lane number, or a range, e.g., 1..2

Note 2: Connection types include NO_TYPE, THRU, R_SPLIT, L_SPLIT, R_MERGE, L_MERGE, RIGHT, LEFT, and UTURN

The following figure and table shows some examples:

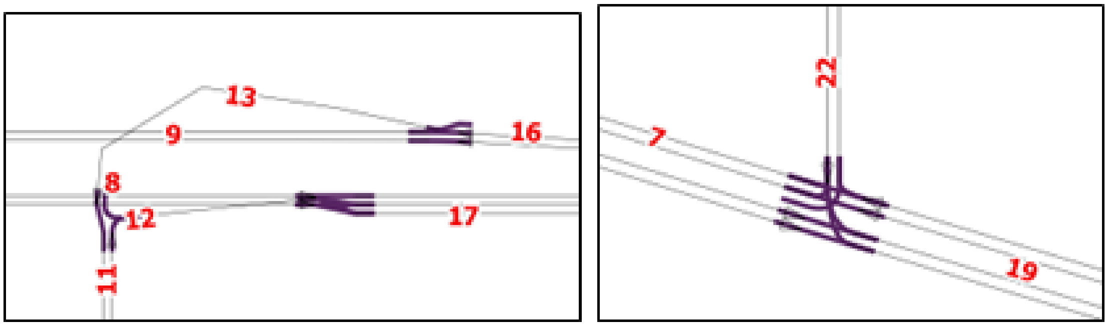


Figure 2 CONNECTION_FILE Illustration

Table 2 CONNECTION_FILE Example

LINK	DIR	TO_LINK	LANES	TO_LANES	TYPE	PENALTY	SPEED	CAPACITY	NOTES
16	0	13	R1	1	R_SPLIT	0	0	0	Off Ramp
16	0	9	1..2	1..2	THRU	0	0	0	Thru lanes at off ramp
8	0	17	1..2	1..3	THRU	0	0	0	Thru lanes at on ramp
12	0	17	1	1..2	R_MERGE	0	0	0	On ramp
7	0	19	1..2	1..2	THRU	0	0	0	Eastbound thru lanes
7	0	22	L1	1	LEFT	0	0	0	Eastbound left turn
19	1	22	1	1	RIGHT	0	0	0	Westbound right turn
19	1	7	1..2	1..2	THRU	0	0	0	Westbound thru lanes
22	1	7	1	1..2	RIGHT	0	0	0	Southbound right turn
22	1	19	1	1..2	LEFT	0	0	0	Southbound left turn

Differences from Version 4

Lane numbering has changed significantly from version 4. In version 5, lanes are numbered from right to left, and pocket lanes are treated separately (Figure 3).

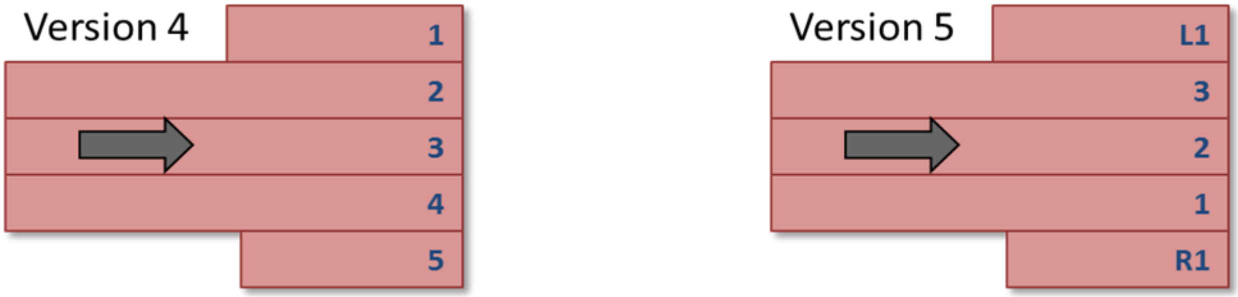
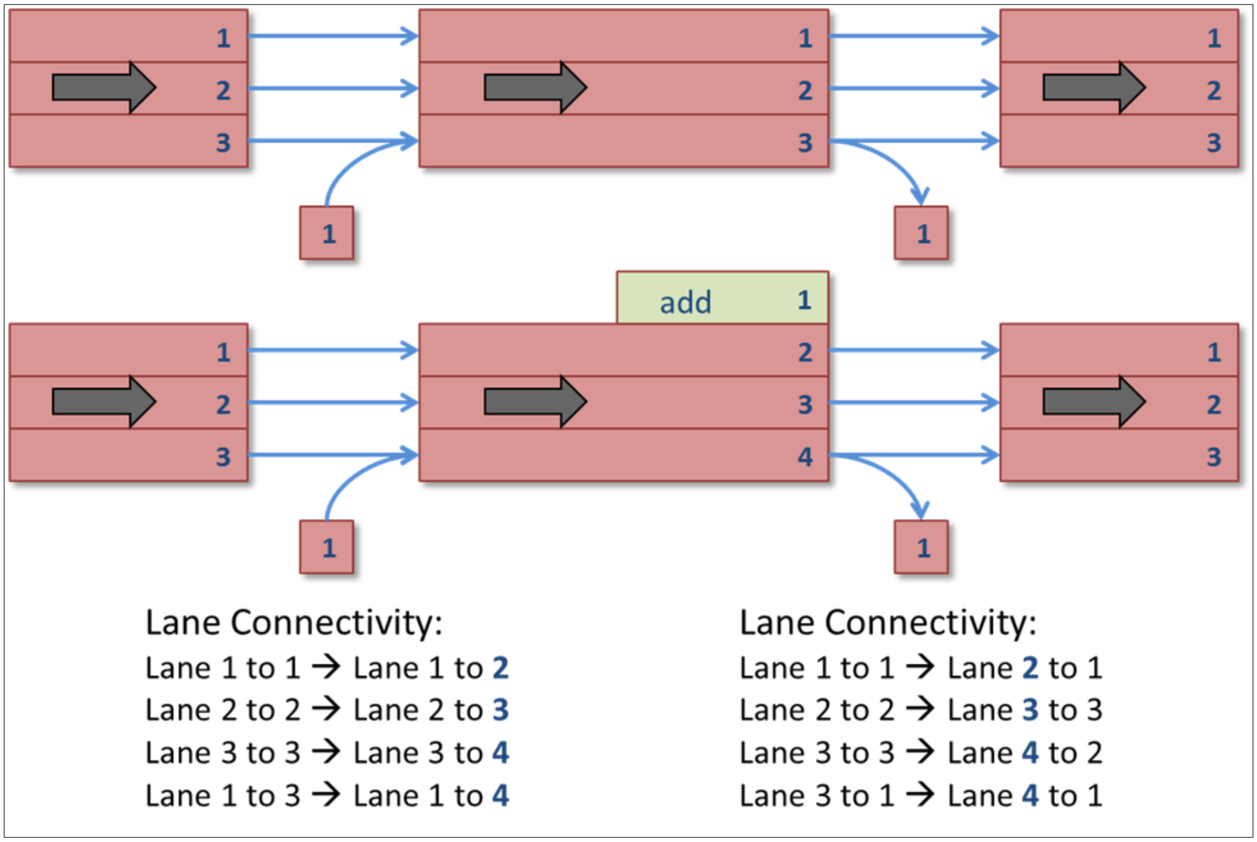


Figure 3 Lane Number Difference between V4 and V5

This simplifies the lane connectivity edits. In version 4, the edits would often cascade from one intersection to another.



Lane Connectivity:

Lane 1 to 1 → Lane 1 to **2**
Lane 2 to 2 → Lane 2 to **3**
Lane 3 to 3 → Lane 3 to **4**
Lane 1 to 3 → Lane 1 to **4**

Lane Connectivity:

Lane 1 to 1 → Lane **2** to 1
Lane 2 to 2 → Lane **3** to 3
Lane 3 to 3 → Lane **4** to 2
Lane 3 to 1 → Lane **4** to 1

Figure 4 Version 4 Lane Connectivity Edits

In version 5, the edits are simpler:

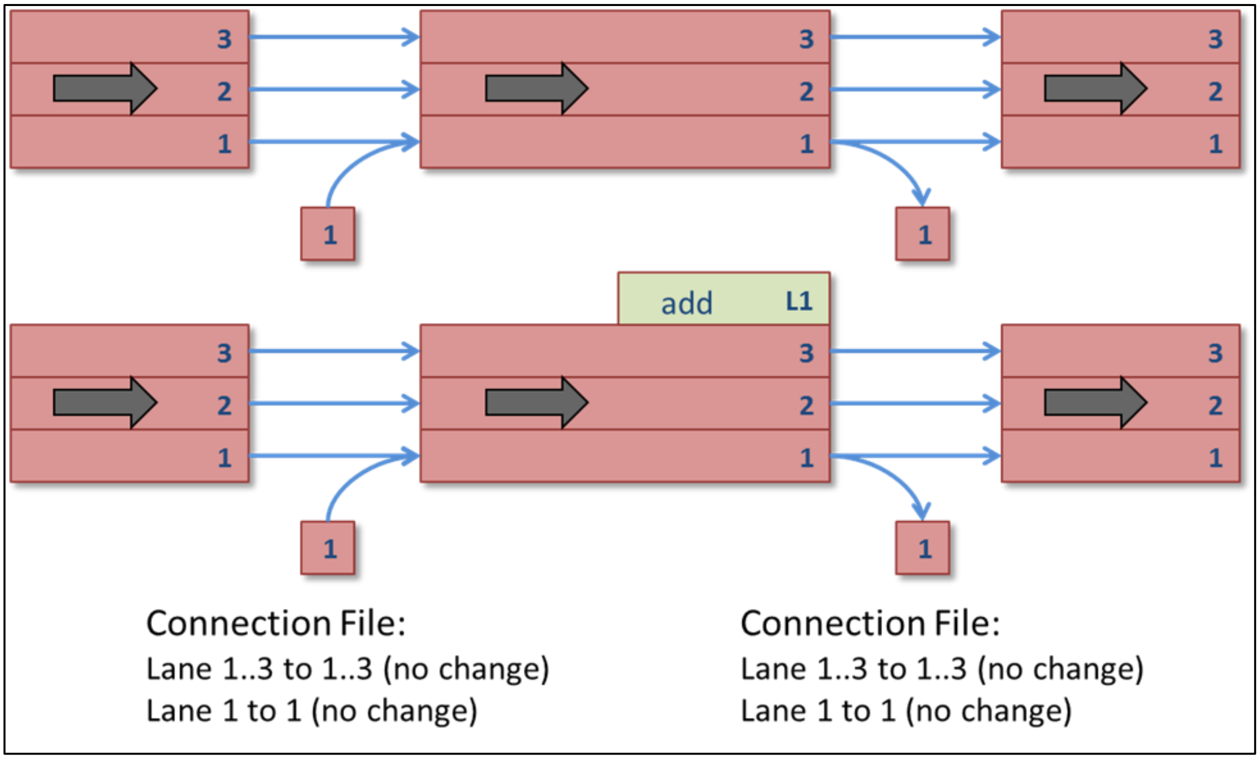


Figure 5 Version 5 Connection Edits

Control Files / Printout Files (.CTL, .PRN)

Used with all TRANSIMS programs

The control_file field on the command line is the directory path and file name of a text file that contains the control strings expected by the program. If a file name is not provided, the program will prompt the user to enter a file name. The program automatically creates a printout file based on the control file name. If the file name includes an extension (e.g., ".ctl") the extension is removed, and the ".prn" suffix is added. The printout file will be created in the current working directory and will overwrite an existing file with the same name.

If the program command syntax includes the partition option, the program can be instructed to process a subset of file partitions by specifying a partition number or partition range after the control file name. For example, the Router can execute a subset of partitions using a command line like:

```
Router.exe  Router.ctl  10
Router.exe  Router.ctl  0..4
```

The first command generates plans for the households assigned to partition 10. The second command generates plans for households assigned to partitions 0 through 4. In these cases, the printout file generated by the program includes the partition number or range in the file name:

```
Router_10.prn
Router_0-4.prn
```

If the program command syntax includes the parameter option, the printout file will include the parameter information. For example, the command

```
RunSetup.exe  TripModel.ctl  2010
```

…will create the printout file:

```
TripModel_2010.prn
```

CONVERSION_SCRIPT

Used In:

UserPrograms
LocationData
NetPrep

This is an optional key though it may be required for specific applications. The conversion script key value is a directory with the conversion script filename appended to it. NetPrep is the primary program in TRANSIMS 5 which uses a TRANSIMS UserProgram-type conversion script for input. However, LocationData also makes extensive use of conversion scripts. The programming / scripting language syntax and features are described in the UserPrograms documentation. By default, the data field names found in the GIS link file are copied to their corresponding field names in the TRANSIMS link file. If the GIS link file was created using ArcNet, this means the data from the GIS file will automatically be copied to the TRANSIMS fields (provided the input and output files are in the same general file structure (i.e., Version4 vs. Version5)). If the GIS link file includes different field names or different units of measure, a conversion script is typically used to manipulate the data or map the input field names to the output field names. The input link shape file fields are referenced as "Link.*field*" and the TRANSIMS link fields are referenced as NewLink.*field*".

An example of a conversion script appears below:

```
NewLink.USE = "ANY"
NewLink.LINK = Link.ID

NewLink.LENGTH = 1609 * Link.LENGTH

NewLink.LANES_AB = Link.AB_LANE
NewLink.LANES_BA = Link.BA_LANE
NewLink.SPEED_AB = Link.AB_PKSPD
NewLink.SPEED_BA = Link.BA_PKSPD

NewLink.FSPD_AB = Link.SPDLIM
NewLink.FSPD_BA = Link.SPDLIM

NewLink.CAP_AB = Link.AB_CAP
NewLink.CAP_BA = Link.BA_CAP

IF (Link.FT == 1) THEN
    NewLink.TYPE = "FREEWAY"
ELSE IF (Link.FT == 2) THEN
    NewLink.TYPE = "Expressway"
ELSE IF (Link.FT == 3) THEN
    NewLink.TYPE = "Principal"
ELSE IF (Link.FT == 4) THEN
    NewLink.TYPE = "Major"
ELSE IF (Link.FT == 5) THEN
    NewLink.TYPE = "Minor"
ELSE IF (Link.FT == 6) THEN
    NewLink.TYPE = "Collector"
ELSE IF (Link.FT == 7) THEN
    NewLink.TYPE = "Local"
ELSE IF (Link.FT == 8) THEN
    NewLink.TYPE = "Local"
ELSE IF (Link.FT == 9) THEN
    NewLink.TYPE = "Frontage"
ELSE IF (Link.FT == 20) THEN
    NewLink.TYPE = "External"
ENDIF

RETURN (1)
END
```

LocationData

The conversion script key is a file name that includes a TRANSIMS User Program script. Any field in the input activity location file can be referenced using the file label IN (e.g., IN.*field*). Any field in the output activity location file (including all newly created fields) can be referenced using the field label OUT (e.g., OUT.*field*). All fields in each Data File are referenced using DATA and the key group number. For example, a field in DATA_FILE_2 is accessed as DATA2.*field*. An additional field called "AL_COUNT" is added to each data file and is set to the number of activity locations with the same join field. Note that “Location” can be used in the place of “IN” and “NewLocation” can be used instead of “OUT” if desired.

An example of a script that sets up external stations fields (ORIG_COEF and DEST_COEF) is shown below:

```
#---- check for external stations ----
OUT.ORIG_COEF = 1
OUT.DEST_COEF = 1
IF (IN.NOTES == "External Destination") THEN
OUT.ORIG_COEF = 0
OUT.DEST_COEF = 1
ENDIF
IF (IN.NOTES == "External Origin") THEN
OUT.ORIG_COEF = 1
OUT.DEST_COEF = 0
ENDIF
RETURN (1)
```

COST_DISTRIBUTION_FILE

NEW_COST_DISTRIBUTION_FILE
Used in PlanCompare

DATA_FILE OR DATA_FILE_#

LocationData Control Keys

This key points to a filename. Each data file group consists of up to four keys. The two join fields must exist in their respective files. The appropriate data record from each data file is passed to the conversion script (see the CONVERSION_SCRIPT entry in this document) for each location. The program counts the number of locations with the same join field value and saves this value to the AL_COUNT field added to each data file. This field can be used to proportionally distribute data items to locations based on the number of locations associated with the data record. For example, population and employment data from traffic analysis zones can be distributed equally to each location within the zone by dividing the data by the value in the AL_COUNT field. Note that “locations” in TRANSIMS 5.0 are the equivalent of activity locations in TRANSIMS 4.0.

Definition Files (*.DEF)

Used by all TRANSIMS programs

TRANSIMS uses definition files to interpret and define data fields within most input and output files generated by the modeling process. A definition file is automatically created when the associated data file is created the majority of the time; however, a few exceptions exist that require the user to manually create a *.DEF file. It has the same path and file name as the data file with a ".def" extension added at the end. For example, the program control keys below…

```
        NEW_LINK_FILE              network\link.txt
        NEW_LINK_FORMAT  TAB_DELIMITED
```

…create a new link file in the network directory called "link.txt". The format key indicates that the link file will be created in tab delimited format. A definition file called "link.txt.def" will also be created in the network directory. The definition file is a standard text file containing the following information:

```
TRANSIMS50, TAB_DELIMITED, 1
LINK, INTEGER, 1, 10
NAME, STRING, 2, 40
NODE_A, INTEGER, 3, 10
NODE_B, INTEGER, 4, 10
LENGTH, DOUBLE, 5, 8.1, FEET
TYPE, STRING, 10, 12, FACILITY_TYPE
AREA_TYPE, UNSIGNED, 12, 3
LANES_AB, UNSIGNED, 14, 2
SPEED_AB, DOUBLE, 15, 5.1, MPH
FSPD_AB, DOUBLE, 16, 5.1, MPH
CAP_AB, UNSIGNED, 17, 8, VPH
USE, STRING, 22, 128, USE_TYPE
```

The first record in the *.def file specifies the software version that created the file (TRANSIMS 5.0), the data file format (tab delimited), and the number of header records in the data file (1). The header record is followed by one record for each data field. These records include the field name, the data type, the field offset within the data record, the maximum field length and number of decimal places, and, if appropriate, the units or enumeration type of the field. The units field facilitates conversions between English and metric systems. It also automates the process of converting text strings to internal type codes (i.e., enumerations) and back again. Binary files, for example, store the type codes as numbers rather than strings to reduce file size and improve performance.

When an existing file is read by a program, the program looks for the definition file to automatically determine how to read the file and process the data fields. If a definition file is not found, the program will look for a *.FORMAT control key where the user identifies the file format. In many cases, the program can used the file format information to read header records from the data file and construct a definition file. If the file is delimited, the program will read the first 100 records of the file to estimate the data types and field widths. This information is written to a new definition file constructed for the data file. If the estimation process is inaccurate, the user can edit the definition file to correct any inaccuracies.

Binary and fixed column file format definition files cannot be constructed automatically. These file formats do not store field header information in the data file. All information about how to read and interpret the file must be provided in the definition file. The user must manually create a definition file for these file types if they are to be read into a TRANSIMS program. This is also true for delimited files that do not include field names as the first record in the file.

TRANSIMS also supports nested files that include two record types. The first record is the master record that includes a field that identifies the number of nested records that follow. A link delay file is a typical example of a nested data file. The master records define the link, time period, flow and travel time on the link while the nested records define the turning movement links, flows, and travel times. Table 3 shows an example of a nested file structure. In this table, the master records are shaded in dark blue, while the nested records are shaded in light blue. (In reality, the file is typically a tab-delimited text file, with no formatting.)

Table 3 Example of a nested file structure

LINK	DIR	TYPE	START	END	FLOW	TIME	NCONNECT
OUT_LINK	OUT_FLOW	OUT_TIME					
37	0	0	2:00	2:15	2.0	19.4	2
44	1.0	19.4					
41	1.0	19.4					
37	1	0	2:00	2:15	0.5	19.4	0
39	0	0	2:00	2:15	8.0	63.8	3
42	4.0	63.8					
46	11.0	63.8					
43	1.0	63.8					
40	1	0	2:00	2:15	2.0	63.8	1
10	2.0	63.8					

The definition file for the link delay file shown above looks like this:

```
TRANSIMS50, TAB_DELIMITED, 2, NESTED
LINK, INTEGER, 1, 10
DIR, INTEGER, 2, 1
TYPE, INTEGER, 3, 1
START, TIME, 4, 16, HOUR_CLOCK
END, TIME, 5, 16, HOUR_CLOCK
```

```
FLOW, DOUBLE, 6, 8.1, VEHICLES
TIME, TIME, 7, 8.1, SECONDS
NCONNECT, INTEGER, 8, 2, NEST_COUNT
OUT_LINK, INTEGER, 1, 10, NO, NESTED
OUT_FLOW, DOUBLE, 2, 8.1, VEHICLES, NESTED
OUT_TIME, TIME, 3, 8.1, SECONDS, NESTED
```

The first record indicates that the data file has two header records and includes the NESTED key word. The field specifications for the master record are exactly like any other definition file. The nested fields add the NESTED key word after the units field. Note that the record offsets restarts from 1 as well. The field with the NEST_COUNT identifier is used to determine how many nested records follow each master record.

DELETE_LINK_FILE

DELETE_LINK_FILE

NetPrep

The delete link file is optional and if specified defines a series of link numbers where the pocket lanes, activity locations, parking lots, processing links and link are deleted. The lane connectivity at both ends of the link is also updated. Each record in the file is interpreted as a comma separated list of link ranges. A link range is specified using two period (e.g., 100..200). The file could also be a simple list of link numbers. The values in the link range and the link file are combined if both keys are provided.

DELETE_NODE_CONTROL_FILE

DELETE_NODE_CONTROL_FILE

IntControl Default Control Key

The delete node control file key is optional and if provided, specifies the location of a file containing a list of nodes for which deletion of the associated signal control files is desired. A delete node control definition (*.DEF) file must also be present in order for IntControl to process this key without returning an error. Refer to the Definition Files (*.DEF) entry in this document for additional information. In contrast to most files used in or produced by TRANSIMS, an associated definition file for this key and referenced input/data file will **not** be automatically created if one is not present. Consequently, the user must manually create this definition file. An existing *.DEF.txt file (e.g., Signal.txt) can be copied, renamed, and the file contents replaced with the proper field names and metadata for this file. A sample delete node control file and its associated definition file (manually created) are shown below:

Delete_Node_Control.txt (Input File):

```
NODE
```

22

Delete_Node_Control.txt.def (Definition File):

```
TRANSIMS50, TAB_DELIMITED, 1
NODE, INTEGER, 1, 10
```

Note that this key requires IntControl to be run at least twice. The first run of IntControl cannot use this key unless the typical IntControl-produced output network files already exist. Including it prematurely or forgetting to comment the key out of the control file will typically return a run-time error. The synthetic intersection control files produced by running IntControl (signal file, phasing plan file, timing plan file, detector file, etc.) must be present (synthetic and/or edited) in order for this operation to successfully execute. The IntControl control (CTL) files for a typical two-iteration execution of IntControl using this key are provided below for illustration.

First Run:

```
TITLE                               IntControl Example – Part 1 of 2

INPUT_SIGN_FILE                     ../network/sign_warrant.txt
INPUT_SIGNAL_FILE                   ../network/signal_warrant.txt

NODE_FILE                           ../network/node.txt
LINK_FILE                           ../network/link.txt
SHAPE_FILE                          ../network/shape.txt
POCKET_FILE                         ../network/pocket.txt
CONNECTION_FILE                     ../network/connection.txt
##DELETE_NODE_CONTROL_FILE          ../network/Delete_Node_Control.txt

NEW_SIGN_FILE                       ../network/sign.txt
NEW_SIGNAL_FILE                     ../network/signal.txt
NEW_TIMING_PLAN_FILE                ../network/timing_plan.txt
NEW_PHASING_PLAN_FILE               ../network/phasing_plan.txt
NEW_DETECTOR_FILE                   ../network/detector.txt

SIGNAL_TYPE_CODE_1                  ACTUATED
NUMBER_OF_RINGS_1                   1
SIGNAL_TIME_BREAKS_1                7:00, 10:00
SIGNAL_CYCLE_LENGTH_1               90 seconds
MINIMUM_PHASE_TIME_1                5 seconds
YELLOW_PHASE_TIME_1                 3 seconds
RED_CLEAR_PHASE_TIME_1              1 seconds
SIGNAL_SPLIT_METHOD_1               CAPACITY
MINIMUM_LANE_CAPACITY_1             500
```

```
MAXIMUM_LANE_CAPACITY_1          1500
POCKET_LANE_FACTOR_1             0.5
SHARED_LANE_FACTOR_1             0.5
TURN_MOVEMENT_FACTOR_1           0.9
PERMITTED_LEFT_FACTOR_1          0.5
GENERAL_GREEN_FACTOR_1           1.0
EXTENDED_GREEN_FACTOR_1          0.5
MAXIMUM_GREEN_FACTOR_1           2.0
SIGNAL_DETECTOR_LENGTH_1         30 feet
```

Second Run:

```
TITLE                       IntControl Example – Part 2 of 2

SIGN_FILE                   ../network/sign.txt
SIGNAL_FILE                 ../network/signal.txt
TIMING_PLAN_FILE            ../network/timing_plan.txt
PHASING_PLAN_FILE           ../network/phasing_plan.txt
DETECTOR_FILE               ../network/detector.txt

NODE_FILE                   ../network/node.txt
LINK_FILE                   ../network/link.txt
SHAPE_FILE                  ../network/shape.txt
POCKET_FILE                 ../network/pocket.txt
CONNECTION_FILE             ../network/connection.txt
DELETE_NODE_CONTROL_FILE    ../network/Delete_Node_Control.txt

NEW_SIGN_FILE               ../network/sign2.txt
NEW_SIGNAL_FILE             ../network/signal2.txt
NEW_TIMING_PLAN_FILE        ../network/timing_plan2.txt
NEW_PHASING_PLAN_FILE       ../network/phasing_plan2.txt
NEW_DETECTOR_FILE           ../network/detector2.txt

SIGNAL_TYPE_CODE_1          ACTUATED
NUMBER_OF_RINGS_1           1
SIGNAL_TIME_BREAKS_1        7:00, 10:00
SIGNAL_CYCLE_LENGTH_1       90 seconds
MINIMUM_PHASE_TIME_1        5 seconds
YELLOW_PHASE_TIME_1              3 seconds
RED_CLEAR_PHASE_TIME_1           1 seconds
SIGNAL_SPLIT_METHOD_1            CAPACITY
MINIMUM_LANE_CAPACITY_1          500
MAXIMUM_LANE_CAPACITY_1          1500
POCKET_LANE_FACTOR_1             0.5
```

SHARED_LANE_FACTOR_1	0.5
TURN_MOVEMENT_FACTOR_1	0.9
PERMITTED_LEFT_FACTOR_1	0.5
GENERAL_GREEN_FACTOR_1	1.0
EXTENDED_GREEN_FACTOR_1	0.5
MAXIMUM_GREEN_FACTOR_1	2.0
SIGNAL_DETECTOR_LENGTH_1	30 feet

DELETE_NODE_FILE

DELETE_NODE_FILE

NetPrep

The delete node file is optional and if specified defines a series of node numbers where the lane connectivity, traffic control warrants, and node are deleted. Each record in the file is interpreted as a comma separated list of node ranges. A node range is specified using two period (e.g., 100..200). The file could also be a simple list of node numbers. The values in the node range and the node file are combined if both keys are provided.

DESTINATION_LOCATION_FILE

DESTINATION_LOCATION_FILE

PathSkim

NEW_DESTINATION_LOCATION_FILE

PathSkim

When PathSkim calculates zone-to-zone skims, it is actually calculating multiple location-to-location travel times, distances, and costs. The DESTINATION_LOCATION_FILE is an input file,indicating which locations should be used in the destination zone. The NEW_DESTINATION_LOCATION_FILE is the output file, indicating which locations were used by PathSkim's calculations.

An example of such a file is as follows:

ZONE	LOCATIONS
1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18
3	36,37,38,39,40,41,42,43,44,45,46,47,48,49,50
10	51, 52
11	53, 54
12	55, 56
13	57, 58
14	59, 60

DESTINATION_ZONE_FILE

DESTINATION_ZONE_FILE

PathSkim

DETECTOR_FILE

DETECTOR_FILE

ArcNet
IntControl Default Control Key
Microsimulator
NewFormat

A Detector file is required when **actuated** traffic signals are specified as part of the highway network. The Detector file contains a unique record for each detector associated with a signalized node. A detector file is unnecessary when signals (fixed-time or actuated) are not used in the network construction process (e.g., sign controls at intersections or no controls). For completely fixed-time signals (e.g., not just fixed time at night), a detector file is not required.

In addition, the Signal Coordinator file and any associated field instances have been removed from TRANSIMS starting with Version 5. The Signal file now serves as a consolidated replacement for the Version 4 Signal Coordinator file.

Nested data fields are not used in the Detector file, but they are used in the three other required and interdependent files that define traffic signals (Signal, Phasing_Plan, Timing_Plan). Also note that a detector file record may be specified at a given location (e.g., a signalized intersection node), but is not necessarily active all the time. Specifically, time periods can be used to define both timed signal operation time periods and actuated signal operation time periods.

A definition file specifying the data fields for a Detector file is shown below:

```
TRANSIMS50, TAB_DELIMITED, 1
DETECTOR, INTEGER, 1, 10
LINK, INTEGER, 2, 10
DIR, INTEGER, 3, 1
OFFSET, DOUBLE, 4, 8.1, METERS
LENGTH, DOUBLE, 5, 8.1, METERS
LANES, STRING, 6, 8, LANE_RANGE_TYPE
TYPE, STRING, 7, 10, DETECTOR_TYPE
USE, STRING, 8, 128, USE_TYPE
NOTES, STRING, 9, 128
```

The fields listed above are defined below:

Table 4: DETECTOR_FILE Field Definitions

Field(s)	Descriptions	Default Units
DETECTOR	Detector ID number	Integer (Starts at 1 and increments by 1 for each additional record number)
LINK	Link ID on which the detector is located	Integer
DIR	The Direction Code, now used instead of Node, to indicate the direction of the link	Integer (0 or 1 only)
OFFSET	Offset from the beginning of the link (meters/feet) – helps define the starting location of the detector	Decimal (Meters/Feet)
LENGTH	The length of the detector in meters	Decimal
LANES	Defines lane ranges with pocket lane codes	String (R1..2, 1..L1, 1, etc.)
TYPE	The type of detector as concerns its activation or lack thereof at an actuated, signalized intersection	String (keyword values) PRESENCE = sense vehicles on detector PASSAGE = sense vehicles crossing detector
USE	Specifies the vehicle use type	String (e.g., Any, Buses, Trains)
NOTES	Optional field; can specify type of movement through an intersection	String (Left_Turn, Thru_Right, Approach, etc.)

Here is an example of a Detector file populated with data as described above:

Table 5 DETECTOR_FILE Example

DETECTOR	LINK	DIR	OFFSET	LENGTH	LANES	TYPE	USE	NOTES
1	23	0	129.6	9.1	1	PRESENCE	ANY	Left Turn
2	17	1	71.1	9.1	1	PRESENCE	ANY	Left Turn
3	7	0	280.4	9.1	1	PRESENCE	ANY	Left Turn
4	1	1	94.6	9.1	1	PRESENCE	ANY	Left Turn
5	1	1	94.6	9.1	R1..1	PRESENCE	ANY	Thru Right
6	21	0	34.0	9.1	1..L1	PRESENCE	ANY	Approach
7	15	1	192.6	9.1	L1	PRESENCE	ANY	Left Turn
8	15	1	192.6	9.1	1..2	PRESENCE	ANY	Thru Right
9	2	0	161.6	9.1	1..2	PRESENCE	ANY	Thru Right
10	26	0	63.0	9.1	L1	PRESENCE	ANY	Left Turn
11	26	0	63.0	9.1	1..2	PRESENCE	ANY	Thru Right
12	22	1	64.2	9.1	R1..1	PRESENCE	ANY	Approach
13	10	0	180.9	9.1	1..2	PRESENCE	ANY	Thru Right

NEW_DETECTOR_FILE

IntControl Default Control Key
NewFormat

The NEW_DETECTOR_FILE control key specifies the location of the newly created Detector file that is output from IntControl. IntControl can be run and re-run iteratively for various purposes, and in fact will likely need to be re-run in most TRANSIMS implementations. This is necessary to ensure that the resultant highway network is free of any significant coding errors or otherwise unanticipated system behavior.

DIRECTIONAL_DATA_FILE

DIRECTIONAL_DATA_FILE

LinkData

NEW_DIRECTIONAL_DATA_FILE

LinkData

GROUP_TRAVEL_FILE

NEW_GROUP_TRAVEL_FILE

LinkSum

HOUSEHOLD_FILE

Names: HOUSEHOLD_FILE, NEW_HOUSEHOLD_FILE

Both HOUSEHOLD_FILE and NEW_HOUSEHOLD_FILE are used in
ConvertTrips
NewFormat

Additionally, HOUSEHOLD_FILE is used in
Simulator
PathSkim
RandomSelect
Router

The household file key is appended to the PROJECT_DIRECTORY key to specify the file name for the input household file copied to the output household file by the program. One household is generated for each trip in the input trip tables.

The Version 5 household file replaces the household and population files in version 4. A Version 5 definition file for the household file is as follows:

```
TRANSIMS50, TAB_DELIMITED, 2, NESTED
HHOLD, INTEGER, 1, 10
LOCATION, INTEGER, 2, 10
PERSONS, INTEGER, 3, 2, NEST_COUNT
WORKERS, INTEGER, 4, 2
VEHICLES, INTEGER, 5, 2
PERSON, INTEGER, 1, 5, NO, NESTED
AGE, INTEGER, 2, 3, YEARS, NESTED
RELATE, STRING, 3, 12, RELATE_TYPE, NESTED
GENDER, STRING, 4, 8, GENDER_TYPE, NESTED
WORK, STRING, 5, 6, TRUE/FALSE, NESTED
DRIVE, STRING, 6, 6, TRUE/FALSE, NESTED
```

Table 6 HOUSEHOLD_FILE Field Definitions

Field(s)	Description	Default Units
HHOLD	Household number	
LOCATION	Location (activity location) for the household	
PERSONS	Number of persons in the household	
WORKERS	Number of workers	
VEHICLES	Number of vehicles	
PERSON	(nested field) Person number	
AGE	(nested field) Age of that person	Years
RELATE	(nested field) Relationship of that person. Options include blank, NO_RELATE, HEAD_HHOLD, SPOUSE, CHILD, FAMILY	
GENDER	(nested field) Gender of that person. Options include NO_SEX, MALE, FEMALE	MALE
WORK	(nested field) Is that person a worker (TRUE, FALSE)	TRUE
DRIVE	(nested field) Does that person drive (TRUE, FALSE)	TRUE

An example, with one person, is as follows:

Table 7 HOUSEHOLD_FILE Example

HHOLD	LOCATION	PERSONS	WORKERS	VEHICLES	
PERSON	AGE	RELATE	GENDER	WORK	DRIVE
1	41	1	1	1	
1	25		MALE	TRUE	TRUE

INPUT_LINK_FILE

INPUT_LINK_FILE

NetPrep

The link file key is optional (depending on the purpose of using NetPrep). It specifies the name of a shapefile containing the links in the network. If an input node shapefile is provided, the node coordinates will be extracted from the shapefile point location. If an input node shapefile is not provided, the node coordinates will be extracted from the first and last points in the input link shapefile. The value for this key specifies the relative path of the directory and the filename of the input link shape file. Note that this file MUST have a “.shp” extension. Use of a “.txt” extension file for the value of this key will result in processing error(s). When this key is included, a conversion script is likely to be needed as well.

INPUT_NODE_FILE

INPUT_NODE_FILE

NetPrep

The input node file key is optional. It specifies the name of a shapefile containing the nodes in the network. If a node shapefile is provided, the node coordinates will be extracted from the point locations in this shapefile. If a node shapefile is not provided, the node coordinates will be extracted from the first and last points in the input link shapefile.

INPUT_SIGN_FILE

INPUT_SIGN_FILE

IntControl Default Control Key

The input sign file key is optional unless the SIGN_FILE key is not provided in the control file for IntControl. The value for this key specifies the relative path of the directory and the filename of the input sign file. This key is used to manipulate or add to existing sign files using IntControl. The SIGN_FILE key file reference is either omitted or used to read in the existing TRANSIMS network sign file. However, either the INPUT_SIGNAL_FILE or the SIGNAL_FILE control key and referenced file must be specified and included in order for IntControl to execute successfully. The updated sign file is specified using the NEW_SIGN_FILE control key. Note that changes to the sign file, directly or indirectly (via warrants), generally result in changes to related records in output, interdependent intersection files. Consider the example control file below:

```
TITLE                              IntControl Example

INPUT_SIGN_FILE                    ../network/sign_warrant.txt
INPUT_SIGNAL_FILE                  ../network/signal_warrant.txt

NODE_FILE                          ../network/node.txt
LINK_FILE                          ../network/link.txt
SHAPE_FILE                         ../network/shape.txt
POCKET_FILE                        ../network/pocket.txt
CONNECTION_FILE                    ../network/connection.txt
##DELETE_NODE_CONTROL_FILE         ../network/Delete_Node_Control.txt

NEW_SIGN_FILE                      ../network/sign.txt
NEW_SIGNAL_FILE                    ../network/signal.txt
NEW_TIMING_PLAN_FILE               ../network/timing_plan.txt
NEW_PHASING_PLAN_FILE              ../network/phasing_plan.txt
NEW_DETECTOR_FILE                  ../network/detector.txt

SIGNAL_TYPE_CODE_1                 ACTUATED
NUMBER_OF_RINGS_1                  1
SIGNAL_TIME_BREAKS_1               7:00, 10:00
SIGNAL_CYCLE_LENGTH_1              90 seconds
```

```
MINIMUM_PHASE_TIME_1          5 seconds
YELLOW_PHASE_TIME_1           3 seconds
RED_CLEAR_PHASE_TIME_1        1 seconds
SIGNAL_SPLIT_METHOD_1         CAPACITY
MINIMUM_LANE_CAPACITY_1       500
MAXIMUM_LANE_CAPACITY_1       1500
POCKET_LANE_FACTOR_1          0.5
SHARED_LANE_FACTOR_1          0.5
TURN_MOVEMENT_FACTOR_1        0.9
PERMITTED_LEFT_FACTOR_1       0.5
GENERAL_GREEN_FACTOR_1        1.0
EXTENDED_GREEN_FACTOR_1       0.5
MAXIMUM_GREEN_FACTOR_1        2.0
SIGNAL_DETECTOR_LENGTH_1      30 feet
```

INPUT_SIGNAL_FILE

INPUT_SIGNAL_FILE

IntControl Default Control Key

The input sign file key is optional unless the SIGNAL_FILE key is not provided in the control file for IntControl. The value for this key specifies the relative path of the directory and the filename of the input signal file. This key is used to manipulate or add to existing signal files using IntControl. The SIGNAL_FILE key file reference is either omitted or used to read in the existing TRANSIMS network signal file. However, either the INPUT_SIGNAL_FILE or the SIGNAL_FILE control key and referenced file must be specified and included in order for IntControl to execute successfully. The updated signal file is specified using the NEW_SIGNAL_FILE control key. Note that changes to the signal file, directly or indirectly (via warrants), generally result in changes to related records in the output intersection files. Consider the example control file below:

```
TITLE                         IntControl Example

INPUT_SIGN_FILE               ../network/sign_warrant.txt
INPUT_SIGNAL_FILE             ../network/signal_warrant.txt

NODE_FILE                     ../network/node.txt
LINK_FILE                     ../network/link.txt
SHAPE_FILE                    ../network/shape.txt
POCKET_FILE                   ../network/pocket.txt
CONNECTION_FILE               ../network/connection.txt
##DELETE_NODE_CONTROL_FILE    ../network/Delete_Node_Control.txt

NEW_SIGN_FILE                 ../network/sign.txt
```

```
NEW_SIGNAL_FILE                ../network/signal.txt
NEW_TIMING_PLAN_FILE           ../network/timing_plan.txt
NEW_PHASING_PLAN_FILE          ../network/phasing_plan.txt
NEW_DETECTOR_FILE              ../network/detector.txt

SIGNAL_TYPE_CODE_1             ACTUATED
NUMBER_OF_RINGS_1              1
SIGNAL_TIME_BREAKS_1           7:00, 10:00
SIGNAL_CYCLE_LENGTH_1          90 seconds
MINIMUM_PHASE_TIME_1           5 seconds
YELLOW_PHASE_TIME_1            3 seconds
RED_CLEAR_PHASE_TIME_1         1 seconds
SIGNAL_SPLIT_METHOD_1          CAPACITY
MINIMUM_LANE_CAPACITY_1        500
MAXIMUM_LANE_CAPACITY_1        1500
POCKET_LANE_FACTOR_1           0.5
SHARED_LANE_FACTOR_1           0.5
TURN_MOVEMENT_FACTOR_1         0.9
PERMITTED_LEFT_FACTOR_1        0.5
GENERAL_GREEN_FACTOR_1         1.0
EXTENDED_GREEN_FACTOR_1        0.5
MAXIMUM_GREEN_FACTOR_1         2.0
SIGNAL_DETECTOR_LENGTH_1       30 feet
```

INPUT_SPDCAP_FILE

INPUT_SPDCAP_FILE

NetPrep

INPUT_ZONE_FILE

INPUT_ZONE_FILE

NetPrep

The network zone table key is optional. If provided, it specifies the filename and relative path of the input zone file. For example, the value “network/Input_Zone.txt” could be used, given that the Project Directory key has been set to a value of “../” (quotation marks should not be included). If a Project Directory key is not specified, the full path of the input zone file should be used instead. The zone file contains: the zone number (ZONE), X and Y coordinates in UTM meters (X_COORD, Y_COORD), and an area type (AREATYPE) code between 1 and 8. If a zone file is not provided, the zone centroids are extracted from the node file. In this case, all area types will be equal to 2. The input zone file, if produced by NetPrep, can be used subsequently as one of the network files.

KEEP_LINK_FILE

KEEP_LINK_FILE

NetPrep

The keep link file key is optional and specifies the full path and file name of the file that lists the link IDs that need to be retained in the highway network. The NetPrep 5 program deletes and/or replaces certain links that are not required for the highway simulation. If transit routes will be included in the network, the links associated with transit stations and stops should not be removed from the network. The TransimsNet program retains all the links specified in the keep link file even when they are not required for the highway simulation. This functionality is sometimes necessary for highway networks to prevent programmatic deletion of important links by TRANSIMS. A sample keep link file is shown below.

```
LINK
72
612
4050
4088
5988
6201
12006
12009
20133
```

KEEP_NODE_FILE

KEEP_NODE_FILE

NetPrep

The keep node file key is optional and specifies the full path and file name of the file that lists the node IDs that need to be retained in the highway network. In TRANSIMS Version 5, this functionality is located in NetPrep V5; previously, it was incorporated into TransimsNet V4. The NetPrep 5 program removes node that are not required for the highway simulation. If transit routes will be included in the network, the nodes associated with transit stations and stops should not be removed from the network. The TransimsNet program retains all the nodes specified in the keep node file even when they are not required for the highway simulation. This functionality is sometimes necessary for highway networks to prevent programmatic deletion of important nodes by TRANSIMS. Typically, few nodes are adversely affected by the TRANSIMS network pruning algorithm, but when this issue is present, subsequent network synthesis and trip assignment, routing, and simulation can result. A sample keep node file is shown below:

```
NODE
```

288
583
3930
3931
3932
3933
3934
3935
3936

LANE_USE_FILE

LANE_USE_FILE
- ArcNet
- IntControl Default Control Key
- LinkSum
- Microsimulator
- NewFormat
- PathSkim
- PlanSelect
- PlanSum
- Router

NEW_LANE_USE_FILE
- NewFormat

LINK_ACTIVITY_FILE

NEW_LINK_ACTIVITY_FILE
- LinkSum

LINK_DATA_FILE

NEW_LINK_DATA_FILE
- LinkData

LINK_DELAY_FILE

LINK_DELAY_FILE is an input to ArcPlan, IntControl, LinkDelay, NewFormat, PathSkim, PlanSelect, PlanSum, and Router

NEW_LINK_DELAY_FILE is an output from LinkDelay**,** NewFormat**,** PathSkim**,** PlanSum**, and** Router

If the input LINK_DELAY_FILE key is provided to the router, the program uses the information in the link delay file to initialize the link flows and travel times for each time period. The header record in the link delay file is used to determine the size of each time period. The time periods are typically 15 minutes long. If a link delay file is not provided (or the key is “NULL”), free flow speeds are used for all times of day. Free flow speeds are also used for all links and time periods not included in the link delay file.

The NEW_LINK_DELAY_FILE is the output file of link delays. By default, it produces flows and travel times at 15-minute increments (Table 9). The Version 4 LINK_DELAY file had volume, an integer number of vehicles entering or exiting the link during a time period. The Version 5 LINK_DELAY file has flow, which is not necessarily integer (for example, a vehicle traversing half of the link would add 0.5 to the flow).

A definition file is as follows:

```
TRANSIMS50, TAB_DELIMITED, 1
LINK, INTEGER, 1, 10
DIR, INTEGER, 2, 1
START, TIME, 3, 16, HOUR_CLOCK
END, TIME, 4, 16, HOUR_CLOCK
FLOW, DOUBLE, 5, 8.1, VEHICLES
TIME, TIME, 6, 8.1, SECONDS
```

Table 8 LINK_DELAY_FILE Field Definitions

Field(s)	Description	Default Units
LINK	Link Number	
DIR	Direction (0 = A->B, 1 = B->A)	
START	Starting time of day	HOUR_CLOCK
END	Ending time of day	HOUR_CLOCK
FLOW	Distance traveled by vehicles on the link during the START-END interval divided by the link length. Travel units could be VEHICLES, PERSONS, or Passenger Car Equivalents	VEHICLES
TIME	Travel time: link length x VHT / VMT	SECONDS

Table 9 LINK_DELAY_FILE Example

LINK	DIR	START	END	FLOW	TIME
11	0	0:00	0:15	2.7	83.4
11	1	0:00	0:15	0.7	83.4
…	…	…	…	…	…
11	0	7:00	7:15	147	114.9
11	1	7:00	7:15	83.5	85.9
11	0	7:15	7:30	159.2	128.4
11	1	7:15	7:30	95	87.8
11	0	7:30	7:45	190.5	184.5
11	1	7:30	7:45	97.2	88.3
11	0	7:45	8:00	191.5	186.9
11	1	7:45	8:00	98.2	88.6
11	0	8:00	8:15	208.7	235.7
11	1	8:00	8:15	121.8	97
11	0	8:15	8:30	227.5	308.1
11	1	8:15	8:30	115	93.8
11	0	8:30	8:45	229.7	317.8
11	1	8:30	8:45	122.7	97.4
…	…	…	…	…	…
11	1	23:45	24:00:00	18.3	83.4

LINK_DETAIL_FILE

NEW_LINK_DETAIL_FILE

NetPrep

LINK_EQUIVALENCE_FILE

LINK_EQUIVALENCE_FILE

LinkSum

PlanSum

The link equivalence file key is used to set up groups of links (e.g., progression groups). A sample file, along with a sketch of the network, is shown below. There are two groups: Group 1 represents eastbound flow, while Group 2 represents westbound flow.

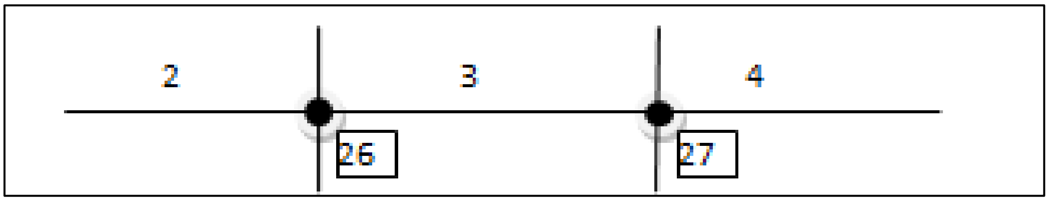


Figure 6: LINK_EQUIVALENCE_FILE example network

1 0 Eastbound
1 1 2, 3, 4
2 0 Westbound
2 1 -4, -3, -2

LINK_NODE_EQUIVALENCE

LINK_NODE_EQUIVALENCE_FILE

TransimsNet

The LINK_NODE_EQUIVALENCE file control key has not yet been implemented in TransimsNet 5.0.

LINK_FILE

Names: LINK_FILE, NEW_LINK_FILE

Used In:

ArcNet
ArcPlan
ArcSnapshot
ConvertTrips
IntControl **(Required)**
LinkDelay
LinkSum
LocationData
Microsimulator
NetPrep
NewFormat
PathSkim
PlanSelect
PlanSum
ProblemSelect
Router
TransimsNet

The link file and new link file both refer to a list of links in the network. A typical field definition (.def) file is shown below. Note that the NEW_LINK_FILE key is a required control key in NetPrep 5 and specifies the relative location and the name of the output link file. This key may not be a required value in the other TRANSIMS program modules in which it is used. Also note that the definition (.def) file associated with the NEW_LINK_FILE produced by NetPrep does not contain a “NAME” field. Consequently, including a reference to a “NAME” field in a NetPrep conversion script (if one is used) will result in a run-time error that prevents the program from executing to completion. Refer to the Quick Reference and/or the Program Reference associated with each TRANSIMS module which uses link and new link files for additional details. The NetPrep conversion script is described in this document as well.

```
TRANSIMS50, TAB_DELIMITED, 1
LINK, INTEGER, 1, 10
NAME, STRING, 2, 40
NODE_A, INTEGER, 3, 10
NODE_B, INTEGER, 4, 10
LENGTH, DOUBLE, 5, 8.1, METERS
SETBACK_A, DOUBLE, 6, 5.1, METERS
SETBACK_B, DOUBLE, 7, 5.1, METERS
BEARING_A, INTEGER, 8, 4, DEGREES
BEARING_B, INTEGER, 9, 4, DEGREES
TYPE, STRING, 10, 12, FACILITY_TYPE
DIVIDED, UNSIGNED, 11, 1
AREA_TYPE, UNSIGNED, 12, 3
GRADE, DOUBLE, 13, 5.1, PERCENT
LANES_AB, UNSIGNED, 14, 2
SPEED_AB, DOUBLE, 15, 5.1, KPH
FSPD_AB, DOUBLE, 16, 5.1, KPH
CAP_AB, UNSIGNED, 17, 8, VPH
LANES_BA, UNSIGNED, 18, 2
SPEED_BA, DOUBLE, 19, 5.1, KPH
FSPD_BA, DOUBLE, 20, 5.1, KPH
CAP_BA, UNSIGNED, 21, 8, VPH
USE, STRING, 22, 128, USE_TYPE
NOTES, STRING, 23, 128
```

The fields are defined as follows:

Table 10 LINK_FILE Field Definitions

Field(s)	Description	Use	Default Units
LINK	The link number (an integer)	Key	
NAME	Typically, the name of the street	Opt.	
NODE_A	The node at one end of the link (an integer)	Req.	
NODE_B	The node at the other end of the link (an integer)	Req.	
LENGTH	Length of the link	Req.	M
SETBACK_A, SETBACK_B	When the link is drawn, the setback from each end to its corresponding node	Opt.	M
BEARING_A	Compass direction entering the link at the A end	Opt.	Degrees
BEARING_B	Compass direction exiting the link at the B end	Opt.	Degrees
TYPE	Facility type (functional classification) of the link	Req.	Note 1
DIVIDED	Is it a divided highway?	Opt.	
AREA_TYPE			
GRADE	Percent grade from A to B	Opt.	Pct.
LANES_AB, LANES_BA	Number of thru lanes in the indicated direction. For a one-way link going from A to B, LANES_BA = 0	Req.	
SPEED_AB, SPEED_BA	Speed limit in the indicated direction	Opt.	m/s
FSPD_AB, FSPD_BA	Free flow speed in the indicated direction	Opt.	m/s
CAP_AB, CAP_BA	Hourly vehicle capacity in the indicated direction (used for Volume / Capacity functions)	Opt.	veh/hr
USE	Vehicle types, modes, or use types permitted on the link	Req.	Note 2
NOTES	Character string for user notes	Opt.	

Note 1: Facility types include FREEWAY, EXPRESSWAY, PRINCIPAL, MAJOR, MINOR, COLLECTOR, LOCAL_THRU, LOCAL, FRONTAGE, RAMP, BRIDGE, TUNNEL, OTHER, WALKWAY, BIKEWAY, BUSWAY, LIGHTRAIL, HEAVYRAIL, FERRY, and EXTERNAL

Note 2: Uses include ANY, WALK, BIKE, CAR, TRUCK, BUS, RAIL, SOV, HOV2, HOV3, HOV4, LIGHTTRUCK, HEAVYTRUCK, TAXI, and RESTRICTED

Figure 7 and Table 2 illustrate a small example. Here, links 8 and 9 are freeway links, link 11 is a minor arterial, and links 12 and 13 are ramps. Typically, limited access roads are represented with separate links for each direction of travel, while other roads have a single link for both directions of travel (even if the road is divided).

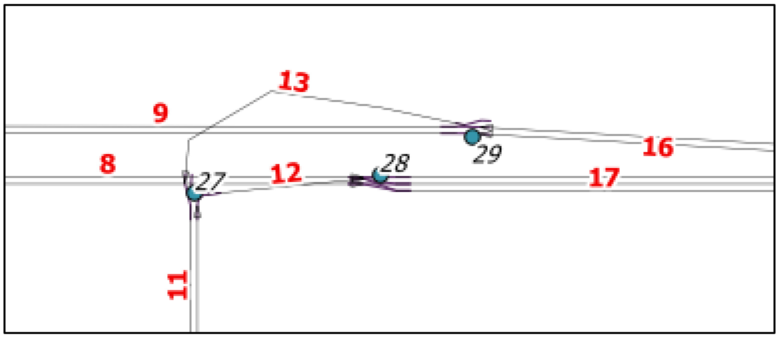


Figure 7 Example Links

Table 11 LINK_FILE Example

LINK	NAME	NODE_A	NODE_B	LENGTH	SETBACK_A	SETBACK_B	BEARING_A	BEARING_B	TYPE	DIVIDED	AREA_TYPE	GRADE
8	0	24	28	1650	9.1	9.1	90	90	FREEWAY	0	2	0
9	0	29	21	1650	7.5	7.5	270	270	FREEWAY	0	2	
11	0	16	27	1000	9.1	7.5	0	0	MINOR	0	2	
12	0	27	28	110	7.5	9.1	84	84	RAMP	0	2	
13	0	29	27	300	7.5	7.5	282	207	RAMP	0	2	

Table 11 (continued) LINK_FILE Example

LINK	LANES_AB	SPEED_AB	FSPD_AB	CAP_AB	LANES_BA	SPEED_BA	FSPD_BA	CAP_BA	USE	NOTES
8	2	97	96	4000	0	0	0	0	AUTO|TRUCK|BUS	
9	2	97	96	4000	0	0	0	0	AUTO|TRUCK|BUS	
11	1	43	43	800	1	43	43	800	ANY	
12	1	72	71	1000	0	0	0	0	AUTO|TRUCK|BUS	
13	1	72	72	1000	0	0	0	0	AUTO|TRUCK|BUS	

LINK_NODE_LIST_FILE

LINK_NODE_LIST_FILE

LinkData

NEW_LINK_NODE_LIST_FILE

NetPrep

LINK_SUMMARY_FILE

NEW_LINK_SUMMARY_FILE

- ArcSnapshot Default Control Key

LINK_VOLUME_FILE

NEW_LINK_VOLUME_FILE

- PlanSum

LOCATION_FILE

Used In

- ArcNet
- ArcPlan
- ConvertTrips Default Control Key
- LinkSum
- LocationData
- Microsimulator
- NewFormat
- PathSkim
- PlanSelect
- ProblemSelect Default Control
- Router
- TransimsNet Default Control Key

NEW_LOCATION_FILE is used in

- LocationData
- NewFormat
- TransimsNet

This file lists the locations (formerly, activity_locations) in the network. The locations are created by TransimsNet, and represent places where traffic can enter or leave the network. In TRANSIMS they take the place of the zone centroids in an older four-step model.

The definition (*.def) file is as follows:

```
TRANSIMS50, TAB_DELIMITED, 1
LOCATION, INTEGER, 1, 10
LINK, INTEGER, 2, 10
DIR, INTEGER, 3, 1
OFFSET, DOUBLE, 4, 8.1, METERS
SETBACK, DOUBLE, 5, 8.1, METERS
ZONE, INTEGER, 6, 10
ORIG_WGT, INTEGER, 7, 2
DEST_WGT, INTEGER, 8, 2
NOTES, STRING, 9, 128
```

The location file includes the following predefined fields:

LOCATION – the index number of the location
LINK – the link to which the location connects
DIR – direction of the link
OFFSET – offset along the link
SETBACK – setback
ZONE – associated zone
NOTES – optional notes

It may also include user defined fields. A commonly used pair of fields is ORIG_WGT, DEST_WGT, which is used in ConvertTrips to determine whether trips should be assigned to a particular location.

Figure 8 and Table 12 show some locations. The circled numbers are activity locations. The green triangles are zone centroids, and the links are numbered in red.

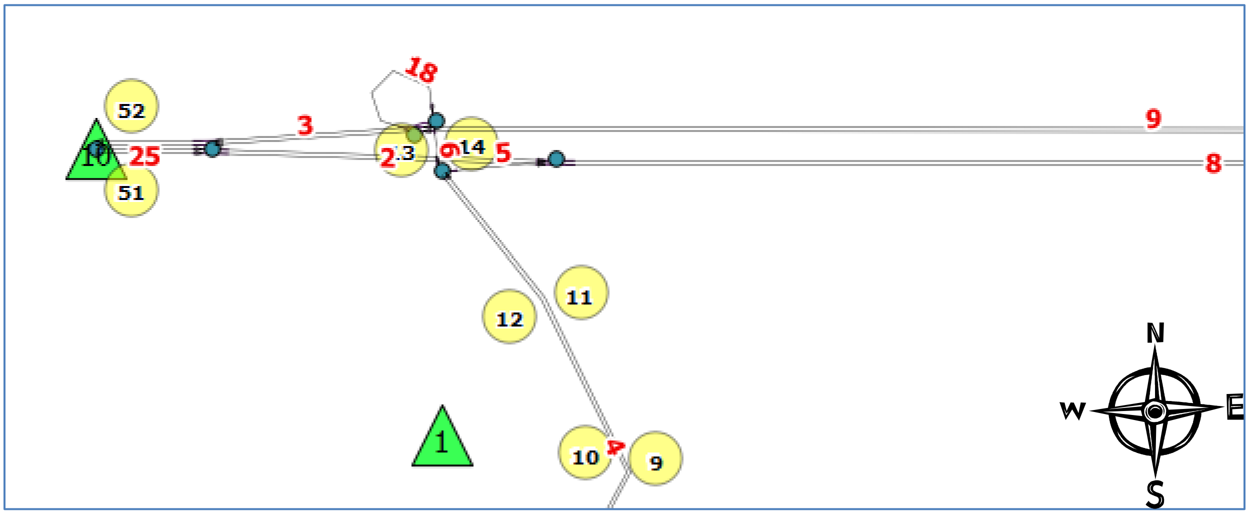


Figure 8 Locations

Table 12 LOCATION_FILE Example

LOCATION	LINK	DIR	OFFSET	SETBACK	ZONE	ORIG_WGT	DEST_WGT	NOTES
9	4	0	500.0	15.2	1	1	1	Activity Location
10	4	1	500.0	15.2	1	1	1	Activity Location
11	4	0	750.0	15.2	1	1	1	Activity Location
12	4	1	250.0	15.2	1	1	1	Activity Location
13	6	0	30.0	15.2	1	1	1	Activity Location
14	6	1	30.0	15.2	1	1	1	Activity Location
51	25	0	30.0	30.0	10	1	0	External Origin
52	25	1	70.0	30.0	10	0	1	External Destination

Link 4 is oriented south to north. It has activity locations 9 and 11 on the A->B side (direction 0) and locations 12 and 10 on the B->A side. The very short link 6 has two locations associated with it. Links 2, 3, 5, 8, 9 and 18 have no locations, as they are freeways or ramps. Locations 51 and 52 are external locations, associated with external zone 10.

MERGE_LINK_DELAY_FILE

MERGE_LINK_DELAY_FILE

LinkDelay

MERGE_PLAN_FILE

Used in PlanPrep

When plans are combined, the output plan file consists of records from the INPUT and MERGE plan files, as follows:

-If a particular plan exists in the input file, it is used, superseding the plan in the merge file.

-If a particular plan exists in the merge file, but not the input file, it is used.

MERGE_TRIP_FILE

MERGE_TRIP_FILE

TripPrep

Nested Data Files

Used In:
IntControl and other TRANSIMS modules

Nested data files are used extensively in TRANSIMS Version 5; however, their use is optional in some cases. Refer to the entry for Definition Files (*.def) in this document for additional information concerning their use and significance.

NEW_ZONE_LOCATION_MAP_FILE

Used in:

LocationData

This is the name of a file that contains a list of zones along with nearby activity locations that are not currently assigned to the zones. The zones that are listed include those zones that are currently assigned to fewer activity locations than the number specified in MINIMUM_ZONE_LOCATIONS. For each such zone, one or more locations, near to but not currently assigned to the zone, are listed. An example appears below.

ZONE	LOCATIONS
2	48, 9
5	46
10	13, 56
11	57, 14
12	51, 1
13	29, 37
14	58, 41

NODE_FILE

Names: NODE_FILE, NEW_NODE_FILE

Used In:

ArcNet
ArcPlan
ArcSnapshot
ConvertTrips
IntControl **(Required)**
LinkDelay
LinkSum
LocationData

Microsimulator
NetPrep
NewFormat
PathSkim
PlanSelect
PlanSum
ProblemSelect
Router
TransimsNet

This is a list of nodes in the network. A typical field definition (.def) file is as follows:

```
TRANSIMS50, TAB_DELIMITED, 1
NODE, INTEGER, 1, 10
X_COORD, DOUBLE, 2, 14.1, METERS
Y_COORD, DOUBLE, 3, 14.1, METERS
Z_COORD, DOUBLE, 4, 14.1, METERS
SUBAREA, INTEGER, 5, 4
NOTES, STRING, 6, 128
```

Essential information includes the node number (an integer) and the X and Y coordinates. These are typically UTM coordinates.

A new field, not in version 4, is the subarea.

Node numbers do not have to be consecutive. However, for external links (zone connectors), the TransimsNet program assumes that the lower node number attached to a zone connector represents the external station zone number. The simplest way to meet this requirement is to assign numbers higher than the highest external zone number all nodes that are NOT associated with zone centroids (internal or external). For example, if the internal zones are in the range 1 – 500, and external zone numbers are in 600 - 620, the non-centroid nodes might be given node numbers of 700 or higher.

An example of a node file appears below:

Table 13 NODE_FILE Example

NODE	X_COORD	Y_COORD	Z_COORD	SUBAREA	NOTES
600	180054.9	4768512.4	0.0	0	External Node
601	179481.0	4767920.0	0.0	0	External Node
602	179397.8	4767815.8	0.0	0	External Node
3802	179740.0	4767650.0	0.0	0	Subarea Node
3803	180724.8	4766966.0	0.0	0	Subarea Node
3808	178366.1	4768820.6	0.0	0	Subarea Node
4660	179865.9	4767545.0	0.0	0	Subarea Node
4665	179620.0	4767750.0	0.0	0	Subarea Node
8819	179705.0	4767730.0	0.0	0	Subarea Node
9511	179685.0	4767705.0	0.0	0	Subarea Node

OCCUPANCY_FILE

OCCUPANCY_FILE

ArcSnapshot Default Control Key

ORIGIN_LOCATION_FILE

NEW_ORIGIN_LOCATION_FILE

PathSkim

ORIGIN_LOCATION_FILE

PathSkim

When PathSkim calculates zone-to-zone skims, it is actually calculating multiple location-to-location travel times, distances and costs. The ORIGIN_LOCATION_FILE is an input, indicating which locations should be used to represent the origin zone. The NEW_ORIGIN_LOCATION_FILE is an output, indicating which locations were used by PathSkim's calculations.

An example of such a file is as follows:

```
ZONE   LOCATIONS
1       1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18
3       36,37,38,39,40,41,42,43,44,45,46,47,48,49,50
10      51, 52
11      53, 54
12      55, 56
13      57, 58
14      59, 60
```

ORIGIN_ZONE_FILE

ORIGIN_ZONE_FILE

- PathSkim

PARKING_FILE

NEW_PARKING_FILE

- NewFormat
- TransimsNet

PARKING_FILE

- ArcNet
- ArcPlan
- ConvertTrips Default Control Key
- Microsimulator
- NewFormat
- PathSkim
- Router
- TransimsNet Default Control Key

PARKING_PENALTY_FILE

PARKING_PENALTY_FILE

- Microsimulator
- PathSkim
- Router

PERFORMANCE_DATA_FILE

NEW_PERFORMANCE_DATA_FILE

- LinkSum

PERFORMANCE_FILE

NEW_PERFORMANCE_FILE

- NewFormat

PERFORMANCE_FILE

- ArcPlan
- LinkSum
- NewFormat

PERSON_FILE

PERSON_FILE

- NewFormat

PHASING_PLAN_FILE

NEW_PHASING_PLAN_FILE

IntControl Default Control Key
NewFormat

PHASING_PLAN_FILE

ArcNet
IntControl Default Control Key
Microsimulator
NewFormat

The phasing plan file is produced by IntControl and contains the link connections, link direction, detectors, movement, movements, and protections associated with a traffic signal phase. It uses nested movements records in order to improve record management and minimize coding mistakes. Each signal phase has multiple movements. Similarly, each timing plan has multiple phases. In Version 5, user help and improved linkages to traffic signal software (e.g., Synchro®) have been incorporated. In addition, the Node now corresponds to the controller number and direction code. Multi-node signals and cross-referencing are also new additions in Version 5 of TRANSIMS. A phasing plan example file is illustrated below:

Table 14: PHASING_PLAN_FILE Example

Signal	Phasing	Phase	Movements	Detectors
Movement	Link	Dir	To_Link	Protection
1	1	1	4	1|3
EB_Left	4892	1	3164	Protected
WB_Left	4202	0	439	Protected
NB_Right	439	0	4202	Stop_Permit
SB_Right	3164	1	4892	Stop_Permit

Both the timing plan and the phasing plan are tightly integrated with the signal plan. What the phasing plan describes is analogous to answering the question of "where are we going?" while the timing plan focuses more on answering "how long until we get there?" This can perhaps better be seen from the diagram below:

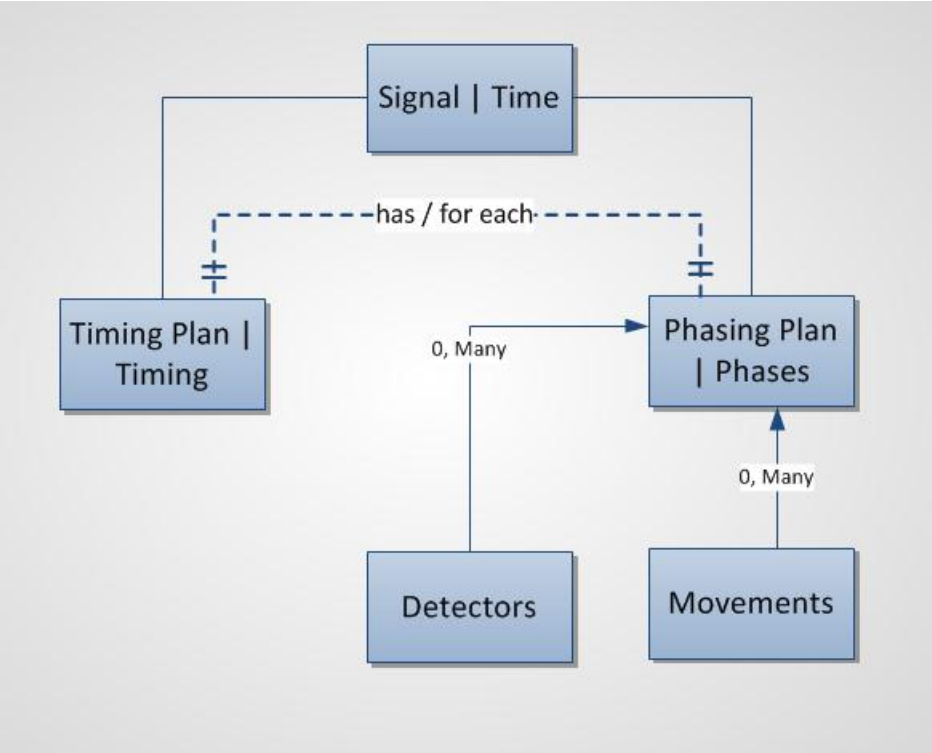


Figure 9: Hierarchical Relationships among the Signalized Intersection files

A definition file (*.def) for the phasing plan file is as follows:

```
TRANSIMS50, TAB_DELIMITED, 2, NESTED
SIGNAL, INTEGER, 1, 10
PHASING, INTEGER, 2, 3
PHASE, INTEGER, 3, 3
MOVEMENTS, INTEGER, 4, 2, NEST_COUNT
DETECTORS, STRING, 5, 128
MOVEMENT, STRING, 1, 10, MOVEMENT_TYPE, NESTED
LINK, INTEGER, 2, 10, NO, NESTED
DIR, INTEGER, 3, 1, NO, NESTED
TO_LINK, INTEGER, 4, 10, NO, NESTED
PROTECTION, STRING, 5, 12, PROTECTION_TYPE, NESTED
```

Fields in the phasing plan file are as follows:

Table 15 PHASING_PLAN_FILE Field Definitions

Field(s)	Description	Default Units
SIGNAL	The signal field indicates the signal number and replaces the signal controller's functions	INTEGER
PHASING	Where multiple phasing plans exist over the course of a day, this is the phasing plan number	INTEGER
PHASE	The phase number; each phase has multiple movements associated with it	INTEGER
MOVEMENTS	Number of associated movement with this phase	INTEGER
DETECTORS	The ID number of detectors related to this movement and separated by slashes (e.g., 1|2); this is required only for actuated controls	STRING
MOVEMENT	Part of the inner nest of fields, this field refers to the movement made at an intersection	STRING
LINK	Incoming link	INTEGER
DIR	Direction on the incoming link	INTEGER
TO_LINK	Link ID number of the link that comes out of the intersection	INTEGER
PROTECTION	Movement protection indicator	STRING

PLAN_FILE

NEW_PLAN_FILE is an output file, used in NewFormat**,** PathSkim**,** PlanCompare**,** PlanPrep**,** PlanSelect, and Router

PLAN_FILE is an input file, used **in** ArcPlan**, S**imulator**,** NewFormat**,** PlanCompare**,** PlanPrep**,** PlanSelect**,** PlanSum and Router

This is the name of the file of travel plans. Travel plans may be partitioned, in which case the file will have a numeric suffix, e.g., TripPlan.0, TripPlan.1.

The version 5 plan file is significantly different from plan files in previous versions of TRANSIMS. Version 4 plan files must be converted, using NewFormat, for use in version 5.

All trip data and path legs are stored in a single nested record. This eliminates problems created by incomplete trips, and simplifies comparisons, update processing and sorting. The file stores detailed information about each component of the path. This provides greater accuracy and fidelity, eliminates data estimates and approximations, and facilitates more detailed analysis of congested locations. The result is a significantly larger plan file with more information. Binary format should be used in most production runs of TRANSIMS.

The Version 5 plan file is a nested file that includes a primary trip record and several nested path records for each leg on the path. The primary trip record includes

- A full copy of the input trip file record
- Path departure and arrival times
- Trip travel time by mode (walk, drive, transit, wait, other)
- Total trip length, cost, and impedance

The nested path records include

- Mode, ID type, facility ID, travel time, distance, cost and impedance for each leg / link on the path

A definition file for the plan file is as follows:

```
TRANSIMS50, TAB_DELIMITED, 2, NESTED
HHOLD, INTEGER, 1, 10
PERSON, INTEGER, 2, 5
TOUR, INTEGER, 3, 3
TRIP, INTEGER, 4, 3
START, TIME, 5, 16, HOUR_CLOCK
END, TIME, 6, 16, HOUR_CLOCK
DURATION, TIME, 7, 16, HOUR_CLOCK
ORIGIN, INTEGER, 8, 10
DESTINATION, INTEGER, 9, 10
PURPOSE, INTEGER, 10, 2
MODE, STRING, 11, 12, MODE_TYPE
CONSTRAINT, STRING, 12, 14, CONSTRAINT_TYPE
PRIORITY, STRING, 13, 10, PRIORITY_TYPE
VEHICLE, INTEGER, 14, 4
PASSENGERS, INTEGER, 15, 2
TYPE, INTEGER, 16, 4
DEPART, TIME, 17, 16, HOUR_CLOCK
ARRIVE, TIME, 18, 16, HOUR_CLOCK
ACTIVITY, TIME, 19, 16, HOUR_CLOCK
WALK, TIME, 20, 12, SECONDS
DRIVE, TIME, 21, 12, SECONDS
TRANSIT, TIME, 22, 12, SECONDS
WAIT, TIME, 23, 12, SECONDS
OTHER, TIME, 24, 12, SECONDS
LENGTH, INTEGER, 25, 10, METERS
COST, FIXED, 26, 6.1, CENTS
IMPEDANCE, UNSIGNED, 27, 10, IMPEDANCE
NUM_LEGS, INTEGER, 28, 5, NEST_COUNT
LEG_MODE, STRING, 1, 12, MODE_TYPE, NESTED
LEG_TYPE, STRING, 2, 8, ID_TYPE, NESTED
LEG_ID, INTEGER, 3, 10, NO, NESTED
```

LEG_TIME, TIME, 4, 12, SECONDS, NESTED
LEG_LENGTH, UNSIGNED, 5, 5, METERS, NESTED
LEG_COST, FIXED, 6, 6.1, CENTS, NESTED
LEG_IMPED, INTEGER, 7, 10, IMPEDANCE, NESTED

Fields in the plan file are as follows:

Table 16 PLAN_FILE Field Definitions

Field(s)	Description	Default Units
HHOLD	Household Number (from the trip file)	
PERSON	Person Number (from the trip file)	
TOUR	Tour Number (from the trip file)	
TRIP	Trip Number (from the trip file)	
START	Trip start time (from the trip file)	HOUR_CLOCK
END	Trip end time (from the trip file)	HOUR_CLOCK
DURATION	Activity duration at the end of this trip (used in tours, otherwise 0) (from the trip file)	HOUR_CLOCK
ORIGIN	Origin location (from the trip file)	
DESTINATION	Destination location (from the trip file)	
PURPOSE	Trip purpose (from the trip file)	
MODE	Mode (from the trip file). Options include WALK, BIKE, DRIVE, RIDE, TRANSIT, PNR_OUT, PNR_IN, KNR_OUT, KNR_IN, TAXI, OTHER, HOV2, HOV3, HOV4	MODE_TYPE
CONSTRAINT	Timing constraint (from the trip file) NONE, START, ARRIVE, FIXED, DURATION, PASSENGER	CONSTRAINT_ TYPE
PRIORITY	Priority for the activity (from the trip file) LOW, MEDIUM, HIGH, CRITICAL	PRIORITY_TYPE
VEHICLE	Vehicle number (from the trip file) (generally 1)	
PASSENGERS	Passengers in the vehicle (from the trip file)	
TYPE	Vehicle type (from the trip file)	
DEPART	Departure time (from the router)	HOUR_CLOCK
ARRIVE	Arrival time	HOUR_CLOCK
ACTIVITY	Duration of the activity	HOUR_CLOCK
WALK	Time spent walking	SECONDS
DRIVE	Time spent driving	SECONDS
TRANSIT	Time spent in public transit	SECONDS
WAIT	Time spent waiting	SECONDS
OTHER	Time spent in other activities	SECONDS
LENGTH	Distance traveled	METERS
COST	Out of pocket cost	CENTS
IMPEDANCE	Total impedance	IMPEDANCE
NUM_LEGS	Number of legs in the trip	NEST_COUNT
LEG_MODE	(nested field) Mode for the leg	MODE_TYPE
LEG_TYPE	(nested field) Type of leg. LOCATION, PARKING, LINK	ID_TYPE
LEG_ID	(nested field) Identifier for the leg, a location or link id (Similar to version 4, if a link is traversed in the B->A direction, it is given a minus sign)	NO

Field(s)	Description	Default Units
LEG_TIME	(nested field) Time to traverse the leg	SECONDS
LEG_LENGTH	(nested field) Distance	METERS
LEG_COST	(nested field) Out of pocket cost	CENTS
LEG_IMPED	(nested field) Total Impedance	IMPEDANCE

The Primary Trip record (Table 17) contains two types of data. The first 16 columns are a copy of the input trip record; the last 12 columns are generated by the router, including trip departure and arrival time, length, cost and impedance.

Table 17 PLAN_FILE Example: Primary Trip Record

HHOLD	PERSON	TOUR	TRIP	START	END	DURATION	ORIGIN	DESTINATION	PURPOSE	MODE	CONSTRAINT	PRIORITY	VEHICLE	PASSENGERS	TYPE	DEPART	ARRIVE	ACTIVITY	WALK	DRIVE	TRANSIT	WAIT	OTHER	LENGTH	COST	IMPEDANCE	NUM_LEGS
1	1	1	1	7:18:02	7:33:15	4:58:07	46	54	1	DRIVE	NONE	MEDIUM	1	0	1	7:18:02	7:21:27	4:58:07	45.2	160	0	0	0	2320	0.0	2504	10

The path records (Table 18) contain one record for each leg of the path. Each record contains the following information:

- Mode for the leg
- Type of leg, typically a link in the network, a parking location, or an activity location
- Leg identifier. Similar to version 4, if a link is traversed in the B->A direction, it is given a minus sign.
- Time to traverse the leg, in seconds
- Length of the leg, in meters
- Cost of the leg in cents (for example, a parking fee or toll would be placed here)
- Total leg impedance

Table 18 PLAN_FILE Example: Path Records

LEG_MODE	LEG_TYPE	LEG_ID	LEG_TIME	LEG_LENGTH	LEG_COST	LEG_IMPED
WALK	LOCATION	46	15.2	15	0.0	304
OTHER	PARKING	46	0	0	0.0	0
DRIVE	LINK	-22	15.5	185	0.0	155
DRIVE	LINK	19	37.6	525	0.0	376
DRIVE	LINK	11	83.4	1000	0.0	834
DRIVE	LINK	12	5.7	110	0.0	57
DRIVE	LINK	17	15.1	400	0.0	151
DRIVE	LINK	-26	2.7	70	0.0	27
OTHER	PARKING	54	0	0	0.0	0
WALK	LOCATION	54	30	30	0.0	600

POCKET_FILE

Names: POCKET_FILE, NEW_ POCKET_FILE

Used In:

ArcNet
ArcPlan
ArcSnapshot
IntControl **(Required)**
Microsimulator
NewFormat
PathSkim
Router
TransimsNet

This is a list of pocket lanes in the network. A typical field definition (.def) file is as follows:

```
TRANSIMS50, TAB_DELIMITED, 1
LINK, INTEGER, 1, 10
DIR, INTEGER, 2, 1
TYPE, STRING, 3, 12, POCKET_TYPE
LANES, UNSIGNED, 4, 2
LENGTH, DOUBLE, 5, 8.1, METERS
OFFSET, DOUBLE, 6, 8.1, METERS
NOTES, STRING, 7, 128
```

A pocket lane is an auxiliary lane that approaches or leaves an intersection. It is associated with a link, direction, node and movement. If a pocket is only available part-time (for example, if parking is permitted at off-peak times), that can be handled via a lane use restriction. The fields are defined as follows:

Table 19 POCKET_FILE Field Definitions

Field(s)	Description	Use	Default Units
LINK	The link number (an integer)	Req.	
DIR	Direction of the link AB=0, BA=1	Req.	
TYPE	Specifies the pocket type (e.g., Left_Turn, Left_Merge, Right_Turn, Right_Merge). Turn pockets are those approaching an intersection; merge pockets are those leaving an intersection.	Req.	Note 1
LANES	The number of pockets associated with the above LINK & DIR	Req.	
LENGTH	Length of the pocket	Req.	m
OFFSET	Specifies the length of the pocket offset	Req.	m
NOTES	Character string for user notes	Opt.	

Note 1: Pocket types include LEFT_TURN, RIGHT_TURN, LEFT_MERGE, RIGHT_MERGE, POCKET_LANE, and AUX_LANE

Figure 10 and Table 20 illustrate a small example.

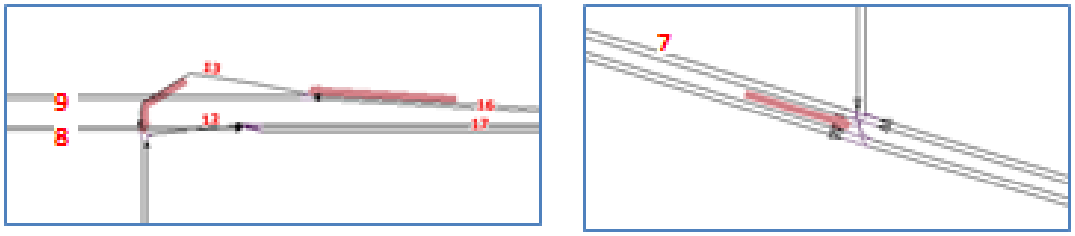


Figure 10 Pocket Lanes

Table 20 POCKET_FILE Example

LINK	DIR	TYPE	LANES	LENGTH	OFFSET	NOTES
7	0	LEFT_TURN	1	50.0	0.0	Left Turn Lane
13	0	LEFT_TURN	1	100.0	0.0	Left Turn Lane
16	0	RIGHT_TURN	1	200.0	0.0	Pocket Lane for Right off-ramp

PROBLEM_FILE

The **NEW_PROBLEM_FILE** is output by the Router or Simulator to indicate trips that could not be routed or simulated. Other programs that use **NEW_PROBLEM_FILE** as output include NewFormat and PathSkim.

Programs that use **PROBLEM_FILE** as input include ArcPlan, NewFormat, and ProblemSelect

Problem codes output by the router and simulator are listed in Table 21. The first column is the problem number, the second is the code that is used in the TRANSIMS source code; the third is the problem name as shown in the problem file; the fourth is the most typical meaning.

Table 21 Problem Codes

No.	Code	Name	Most typical meaning
0	TOTAL_PROBLEM	Total	
1	PATH_PROBLEM	Path Building	No feasible path between the origin and destination. It could be caused by lane connectivity or one-way street conditions or by a network coding error.
2	TIME_PROBLEM	Time Schedule	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_PROBLEM	Zero Length	The zero-node error occurs when the origin and the destination activity locations lie on the same link, at zero distance
4	TYPE_PROBLEM	Vehicle Type	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.
5	DIST_PROBLEM	Path Circuity	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	MODE_PROBLEM	Travel Mode	The Router records a travel mode error when the mode on the activity file cannot be built. This generally means that the transit, walk, or bike networks have not been enabled.

No.	Code	Name	Most typical meaning
7	ACCESS_PROBLEM	Vehicle Access	An access error is generated when the vehicle listed in the activity 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_PROBLEM	Walk Distance	This message is generated when the cumulative walk distance required by the path exceeds the MAX_WALK_DISTANCE parameter.
9	WAIT_PROBLEM	Wait Time	This message indicates that potential transit routes exist, but the wait time required to board the routes exceed the MAX_WAIT_TIME parameter. In the Simulator, a wait time problem is generated when a vehicle remains in the same cell unable to advance for an amount of time greater than the MAX_WAIT_TIME key. The most frequent cause of this problem is excessive congestion. It can also be caused by incorrect signal coding that does not provide a phase for all the eligible movements.
10	LINK_PROBLEM	Walk Access	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 indicated a walk or bike access restriction at one of the trip ends.
11	LOAD_PROBLEM	Load Time	
12	PARK_PROBLEM	Park-&-Ride Lot	In order to building a park-&-ride trip (mode 5), the must be a parking lot 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	BIKE_PROBLEM	Bike Distance	This message is generated when the bicycling distance exceeds the MAX_BICYCLE_DISTANCE parameter.

No.	Code	Name	Most typical meaning
14	DEPARTURE_PROBLEM	Departure Time	When a vehicle cannot start its trip at the time specified in the trip file plus the amount of slack time defined by the MAX_DEPARTURE_TIME_VARIANCE, a departure time problem is generated. The most frequent cause of this problem is excessive congestion close to the starting parking lot, which prevents the vehicle from being loaded onto the first link.
15	ARRIVAL_PROBLEM	Arrival Time	If a vehicle is still traveling at the time it is scheduled to arrive at its destination plus the slack time defined by the MAX_ARRIVAL_TIME_VARIANCE key, it will be removed from the network, and an arrival time problem error will be generated. The most frequent cause of this type of problem is congestion. You can allocate more time for the vehicle to finish its trip by adjusting the value for the END_TIME_CONSTRAINT key in the Router.
16	LINK_ACCESS_PROBLEM	Link Access	
17	CONNECT_PROBLEM	Link Connection	This message is generated when no lane connectivity exists between two successive links of the vehicle's path. This most often occurs when the network has been changed, but the travel plans have not been rebuilt. You should inspect the location generating the problem and restore the lane connectivity or re-route the traveler.
18	PARKING_PROBLEM	Parking Access	This problem message is generated when a vehicle is not able to move from the parking lot to the first link in its journey. You should check that the link does not restrict vehicles of the particular vehicle type from using the link.
19	MERGE_PROBLEM	Lane Merging	
20	LANE_PROBLEM	Lane Changing	
21	TURN_PROBLEM	Turning Speed	
22	POCKET_PROBLEM	Pocket Merge	
23	SPACING_PROBLEM	Vehicle Spacing	

No.	Code	Name	Most typical meaning
24	CONTROL_PROBLEM	Traffic Control	
25	USE_PROBLEM	Access Restriction	
26	STOP_PROBLEM	Transit Stop	
27	LOCATION_PROBLEM	Activity Location	
28	PASSENGER_PROBLEM	Vehicle Passenger	
29	DURATION_PROBLEM	Activity Duration	
30	KISS_PROBLEM	Kiss-&-Ride Lot	
31	VEHICLE_PROBLEM	Vehicle ID	
32	SORT_PROBLEM	Data Sort	
33	WALK_LOC_PROBLEM	Walk Location	
34	BIKE_LOC_PROBLEM	Bike Location	
35	TRANSIT_LOC_PROBLEM	Transit Location	
36	MATCH_PROBLEM	Person Match	
37	CONSTRAINT_PROBLEM	Schedule Constraint	
38	BOARDING_PROBLEM	Transit Capacity	
39	DWELL_PROBLEM	Transit Dwell	
40	TRANSFER_PROBLEM	Number of Transfers	
41	LOCAL_PROBLEM	Local Facility	

A router problem file includes one line for each problem trip. That line includes the following columns:

- Problem number
- Columns to identify the trip, including HHOLD, PERSON, TOUR, TRIP, START, END, DURATION, ORIGIN, DESTINATION, PURPOSE, MODE, CONSTRAINT, PRIORITY, VEHICLE, PASSENGERS, TYPE
- A notes column that contains the problem name (e.g., Path Building)

A simulator problem file is similar, with one line per problem trip (Table 22). That line includes the following columns:

- Problem number
- Columns to identify the trip, including HHOLD, PERSON, TOUR, TRIP, START, END, DURATION, ORIGIN, DESTINATION, PURPOSE, MODE, CONSTRAINT, PRIORITY, VEHICLE, PASSENGERS, TYPE

- Time of day
- Link
- Direction for the link
- Lane
- Offset (within the link)
- Route (for transit)
- A notes column that contains the problem name (e.g., Traffic Control)

Table 22 PROBLEM_FILE Example

PROBLEM	TRIP INFORMATION	TIME	LINK	DIR	LANE	OFFSET	ROUTE	NOTES
24	HHOLD, etc.	0:04:12	7	1	2	1113.7	0	Traffic Control
23		4:38:20	1	0	L1	183.7	0	Vehicle Spacing
24		4:39:26	1	0	L1	191.2	0	Traffic Control
23		4:42:25	11	1	1	978.7	0	Vehicle Spacing

REPORT_FILE

Execution Service Keys

Used in nearly every TRANSIMS program

The report file name is optional. If a file name is not provided, the program automatically creates a report file name based on the input control file name. The report file will overwrite an existing file with the same name if the Report Flag key is False or not specified.

RIDERSHIP_FILE

NEW_RIDERSHIP_FILE

NewFormat

RIDERSHIP_FILE

NewFormat

ROUTE_NODES_FILE

NEW_ROUTE_NODES_FILE

NewFormat

ROUTE_NODES_FILE

ArcNet

The network route nodes key is optional. It specifies the name of the input route nodes file used by the TransitNet program to synthetically generate the TRANSIMS transit network. This file is created by the user or generated by the Emme2Route or TPPlusRoute programs. The full path and file name for the route node file is constructed by appending the value of this key to the value of the DIRECTORY/NEW_DIRECTORY key(s). This key is only read if the route header is also provided and read.

SELECTION_FILE

NEW_SELECTION_FILE is an output from NewFormat, PlanCompare, PlanSelect, ProblemSelect, and RandomSelect

SELECTION_FILE is an input to ArcPlan, NewFormat, PathSkim, PlanCompare, PlanPrep, PlanSum, ProblemSelect, Router, and TripPrep

The selection file key is appended to the value of the PROJECT_DIRECTORY key to identify the full path to a list of households and trips that will be processed by the router. A sample selection file is shown below (Table 23). It indicates the household, person, tour, trip and, for parallel processing applications, the partition of the router that will be used.

Table 23 SELECTION_FILE Example

HHOLD	PERSON	TOUR	TRIP	PARTITION
1	1	1	1	0
1	1	1	2	0
1	1	1	3	0
1	1	1	4	0
1	1	1	5	0
1	1	2	1	0
1	1	2	2	0
1	1	2	3	0
100	1	1	1	0
101	1	1	1	0
102	1	1	1	1
103	1	1	1	1
104	1	1	1	1
105	1	1	1	0
106	1	1	1	1
107	1	1	1	1
108	1	1	1	0
109	1	1	1	0

SHAPE_FILE

Names: SHAPE_FILE, NEW_SHAPE_FILE

Used In:
- ArcNet
- ArcPlan
- ArcSnapshot
- ConvertTrips
- IntControl
- LocationData
- NetPrep
- NewFormat
- TransimsNet

This is the name of the TRANSIMS shape file within the network directory, which provides plain text lists of shape points for links in the network. The full path and name for the shape file is constructed by appending the value of this key to the value of the PROJECT_DIRECTORY key.

A typical field definition (.def) file is as follows:

TRANSIMS50, TAB_DELIMITED, 2, NESTED

```
LINK, INTEGER, 1, 10
POINTS, INTEGER, 2, 4, NEST_COUNT
NOTES, STRING, 3, 128
X_COORD, DOUBLE, 1, 14.1, FEET, NESTED
Y_COORD, DOUBLE, 2, 14.1, FEET, NESTED
```

An example of a shape file appears below. After the two-line header, the third line contains the link number and the number of shape points (n) for that link. The next n lines contain the X and Y coordinates of the shape points. The process is then repeated for the next link:

Table 24 SHAPE_FILE Field Definitions

LINK	POINTS
X_COORD	Y_COORD
62	10
6532.8	7935.0
6497.4	7870.7
6439.3	7832.0
6361.9	7822.2
6287.7	7838.2
6226.4	7883.5
6197.5	7938.3
6200.4	7996.4
6235.9	8070.5
6310.4	8109.2
63	11
6816.6	8115.8
6880.9	8093.2
6922.9	8044.6
6948.8	7977.0
6942.2	7912.7
6903.5	7854.6
6842.2	7815.9
6768.0	7802.8
6684.4	7822.2
6632.5	7870.7
6590.0	7934.0

The next two tables provide information on the links and nodes corresponding to the above shape file.

Table 25 Information for links 62 and 63

Link	Node A	Node B	Length	Bearing A	Bearing B	Type
62	123	132	656.2	196	73	RAMP
63	133	123	656.2	103	346	RAMP

Table 26 Information for nodes 123, 132, 133

Node	X_Coord	Y_Coord
123	6561.7	8038
132	6397.6	8136.5
133	6725.7	8136.5

This file indicates that link 62 and link 63 should be drawn as follows: Start at Node A for the link, go through the points in the shape file, end at Node B.

The link file indicates that link 62 runs from node 123 to 132, and that link 63 runs from node 133 to 123. The node file gives the location of these nodes. The end result is as follows (part of a cloverleaf freeway interchange):

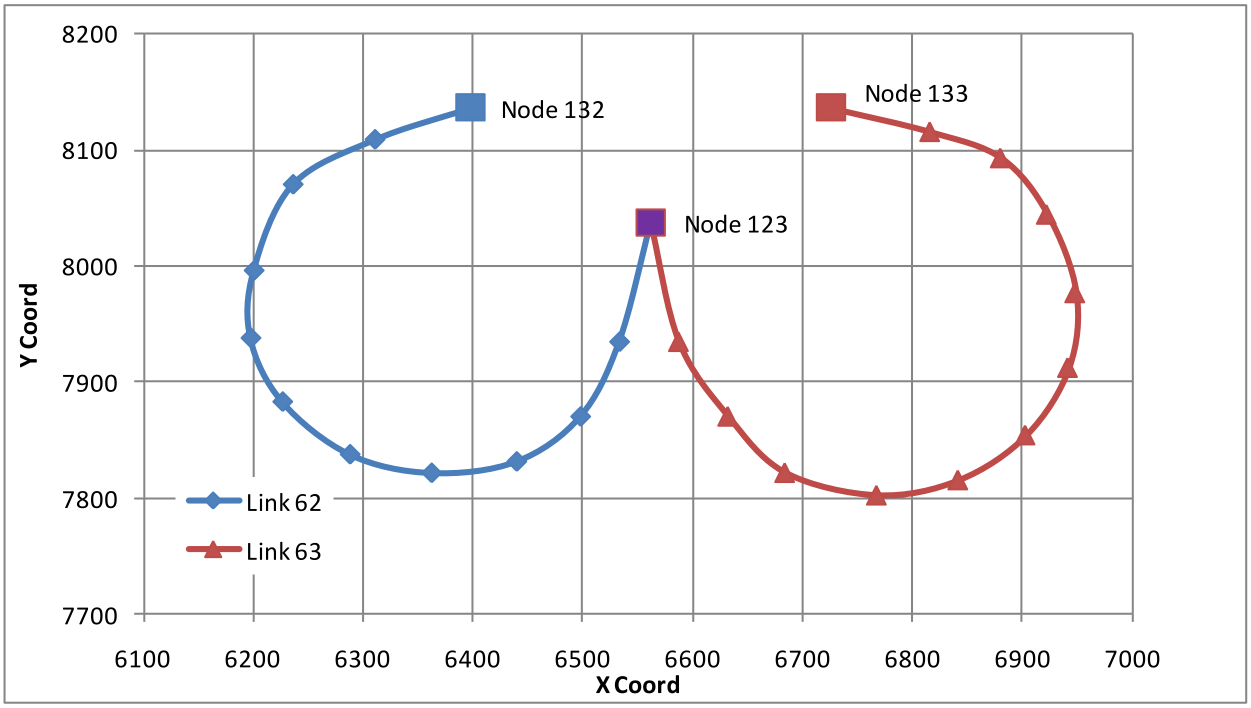


Figure 11 Links, nodes and shape points

Normally, TRANSIMS shape files are not created by hand, but are generated from ArcView shapefiles by programs such as NetPrep. However, it might be necessary to clean-up a TRANSIMS shapefile by hand. In this case shape points are added, deleted or corrected. The total number of shapepoints for the link must then also be checked and updated.

Although shapefiles are not absolutely necessary to run TRANSIMS, they are helpful for two reasons:

They enable a more realistic depiction of the network in a GIS.

They ensure that TransimsNet has the correct connection angles between links when generating connections within a network.

SIGN_FILE

NEW_SIGN_FILE

IntControl Default Control Key
NewFormat
TransimsNet Default Control Key

SIGN_FILE

ArcNet
IntControl Default Control Key
Microsimulator
NewFormat
TransimsNet Default Control Key

The Sign file specifies the location and type of all unsignalized intersection controls (e.g., Stop, Yield, None).

SIGNAL_FILE

NEW_SIGNAL_FILE

IntControl Default Control Key
NewFormat
TransimsNet Default Control Key

SIGNAL_FILE

ArcNet
IntControl Default Control Key
Microsimulator
NewFormat
TransimsNet Default Control Key

The Signal file is among the most important network/supply-side files for the majority of TRANSIMS implementations (e.g., a transit-only model and simulation, or a highway network with only signs and no signals are the only exceptions). This file plays an integral part in defining signalized nodes at synthetically generated and/or manually indicated highway network signalized intersections. More importantly, the Signal file is used directly in combination with three other interdependent/relational network files also created by IntControl (specifically, the Phasing Plan, Timing Plan, and Detector files) to describe signalized intersections at the fine level of detail required for microsimulation.

Starting with Version 5 of TRANSIMS, the Signalized node field (Signal) represents the controller number as well, and consequently replaces the functions of the Version 4 Signal Coordinator file. Consequently, the Signal Controller file is not used in TRANSIMS 5, and each node in the Signal file controls a list of nodes. The records are nested time period records with end times. The purpose in using nested fields is to improve record management and minimize network coding mistakes. Also, the Timing and Phasing ID numbers are now re-useable, as are these two respective files (Phasing Plan and Timing Plan).

Any time a new Signal file is generated, it is imperative to review the IntControl output files for network coding errors (preferably via both graphical/GIS software and text file review). This is strongly recommended because IntControl typically requires several re-iterations to achieve accurate and desired sign and signal placement on the network. A Keep Node File or Delete Node File may be necessary to prevent pruning/collapsing of important nodes by the TRANSIMS synthetic network generation algorithm. Edits to the Sign and Signal Warrants generated by TransimsNet may also be necessary, followed by re-running IntControl using the updated warrant files. Directly editing the Signal (or Sign) files to correct network coding errors is not recommended due to the associated file interdependencies. Manual review and sanity checks of intermediate outputs are both important for many TRANSIMS 5 files, particularly the Signal file and its dependencies (Phasing Plan, Timing Plan, and Detector files).

Common coding issues may relate to incorrect network coding (file content and/or parameter value aberrations or inconsistencies). Other potential issues include unexpected placement of signals by TR0ANSIMS. In addition, residual artifacts may be present on the network following an update or delete signalized intersection-type operation (or delete node or link operation, etc.). For instance, a Detector file record may be found at a previously signalized node, but not an associated Signal, Phasing Plan, or Timing Plan entry associated with the node prior to the update or delete operation. If it is later decided to place a Sign file entry (via the Sign Warrant file) at that intersection/node, a sign-signal conflict error may result.

Since most MPOs do not have the highly detailed intersection-level data required by TRANSIMS in their regional planning models, the network coder will have to perform some sanity checks on the results, at intermediate and final breakpoints in the overall network generation process.

Multi-node signal coordination is another feature new to TRANSIMS 5. Additionally, changes in barrier, ring, and position codes are employed to enhance compatibility with traffic signal software. Also, Detectors now include use types (e.g., buses or trains).

The Signal file consists of several fields, and uses nested file structure. The definition file for the Version 5 Signal file is provided below:

```
TRANSIMS50, TAB_DELIMITED, 2, NESTED
```

```
SIGNAL, INTEGER, 1, 10
GROUP, INTEGER, 2, 4
TIMES, INTEGER, 3, 2, NEST_COUNT
NODES, STRING, 4, 128
START, TIME, 1, 16, HOUR_CLOCK, NESTED
END, TIME, 2, 16, HOUR_CLOCK, NESTED
TIMING, INTEGER, 3, 3, NO, NESTED
PHASING, INTEGER, 4, 3, NO, NESTED
NOTES, STRING, 5, 128, NO, NESTED
```

Fields in the phasing plan file are as follows:

Table 27: SIGNAL_FILE Field Definitions

Field(s)	Description	Default Units
SIGNAL	The signal field indicates the signal number and replaces the signal controller's functions	INTEGER
GROUP	The signal group number	INTEGER
TIMES	The total number of timing plans associated with a signal group	INTEGER
NODES	A list of nodes separated using '|'	STRING
START	Start time for a signal period; multiple time periods, each with a unique set of associated, grouped parameters may be defined in the control file	TIME
END	End time for a signal period; multiple time periods, each with a unique set of associated, grouped parameters may be defined in the control file	TIME
TIMING	The timing plan associated with a signal group	INTEGER
PHASING	Where multiple phasing plans exist over the course of a day, this is the phasing plan number	INTEGER
NOTES	A character string for user annotations	STRING

Of particular note is that the Timing field above corresponds and links to the Timing Plan. Also, the Phasing field noted above similarly interconnects with the Phasing Plan.

To illustrate the rather complex nested file structure that interconnects the Signal, Phasing Plan, and Timing Plan file records, consider the entity-relationship diagram shown in Figure 12. TRANSIMS Version 5 uses flat files; however, these Signal-centric files are related to each other to varying extents by means of a unique field or combination of fields that can be considered a primary key. Note that this is not an official version, but rather, abstracted:

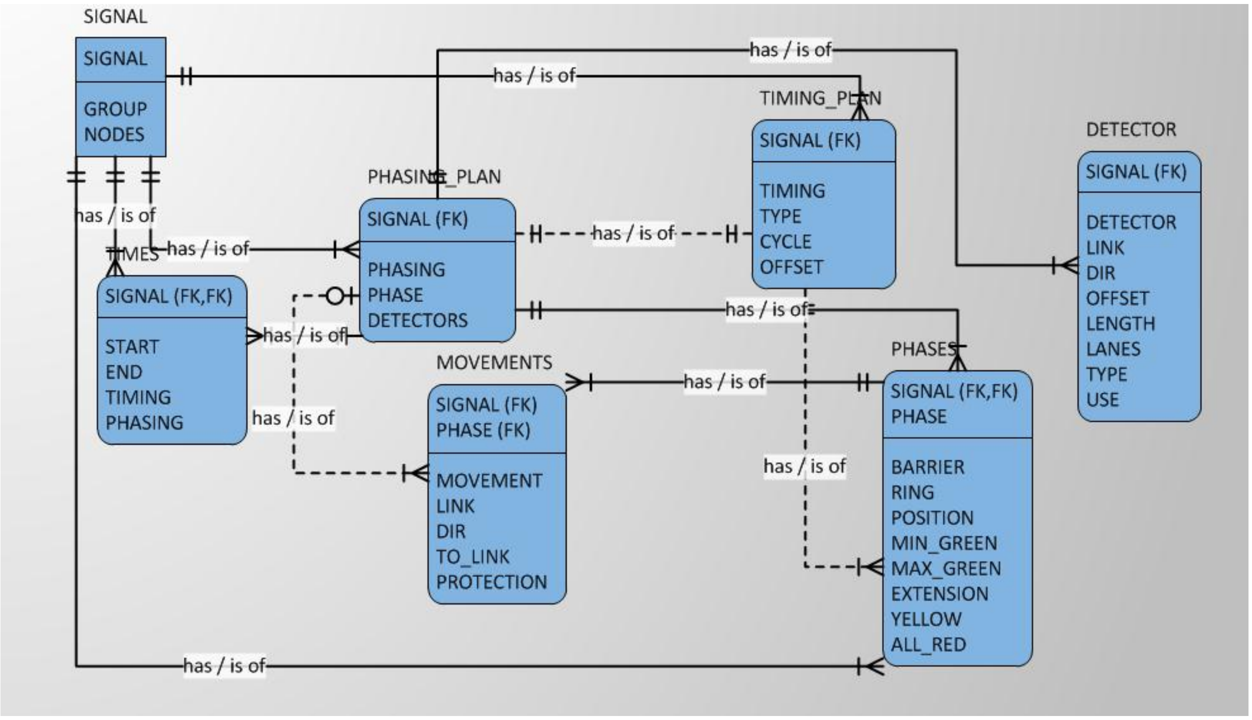


Figure 12 Entity-Relationship Diagram - Signal file dependencies

Last, an example Signal file with data pre-populated is provided below:

Table 28 SIGNAL_FILE Example

Signal	Group	Times	Nodes
Start	End	Timing	Phasing
1	2	5	101|102|103
0:00	6:00	1	1
6:00	9:30	2	1
9:30	16:00	1	1
16:00	19:00	3	1
19:00	27:00	1	1

SKIM_FILE

NEW_SKIM_FILE is written by

NewFormat
PathSkim

SKIM_FILE is read by
ConvertTrips
NewFormat

A skim file contains the skimmed travel times and impedances for various origin-destination pairs and time periods. It is used by ConvertTrips to provide initial travel times for a set of newly generated trips.

Features include the following:

- Skim files include OD size and time period meta-data
- Partitioned time periods or merged time periods
- Location or zone-based origins and destinations
- Total travel time or time components (walk, drive, transit, wait, other)
- Trip length, cost, and impedance
- User-specified output units (e.g., minutes, miles)

Columns of the skim file are defined as follows:

ORIGIN - The origin location index (zone, location or district)

DESTINATION - The destination location index (zone, location or district)

PERIOD - Time period for this skim[1]

COUNT - Number of location/time period pairs considered for this origin-destination pair

WALK - Walking time, typically in seconds

DRIVE - Driving time, typically in seconds

OTHER - Other time, typically in seconds

LENGTH - Length of the trip, typically in meters

COST - Out of pocket cost of the trip in cents

IMPEDANCE – Total impedance for the trip

A typical .def file for the skim file is as follows:

```
ORIGIN, UNSIGNED, 1, 5, ZONES
DESTINATION, UNSIGNED, 2, 5, ZONES
PERIOD, UNSIGNED, 3, 3
COUNT, INTEGER, 4, 5
WALK, TIME, 5, 12, SECONDS
DRIVE, TIME, 6, 12, SECONDS
OTHER, TIME, 7, 12, SECONDS
LENGTH, INTEGER, 8, 10, METERS
COST, INTEGER, 9, 5, CENTS
```

[1] an integer that corresponds to SKIM_TIME_PERIODS or SKIM_TIME_INCREMENT

IMPEDANCE, INTEGER, 10, 10, IMPEDANCE

An example skim file (tab-delimited) appears below. The first line gives the number of origins, destinations and the time period span, e.g., NUM_ORG=2; NUM_DES=1; PERIODS=6:00..6:15, 8:00..8:15, 8:15..8:30, 8:30..9:00

Table 29 SKIM_FILE Example

ORIGIN	DESTINATION	PERIOD	COUNT	WALK	DRIVE	OTHER	LENGTH	COST	IMPEDANCE
3	54	0	1	45	167	0	3360	0	2578
4	54	0	1	45	167	0	3360	0	2578
3	54	1	1	45	176	0	3360	0	2666
4	54	1	1	45	176	0	3360	0	2666
3	54	2	1	45	181	0	3360	0	2710
4	54	2	1	45	181	0	3360	0	2710
3	54	3	2	45	180	0	3360	0	2701
4	54	3	2	45	180	0	3360	0	2701

SNAPSHOT_FILE

NEW_SNAPSHOT_FILE

ArcSnapshot
NewFormat
Simulator

SNAPSHOT_FILE

ArcSnapshot Default Control Key
NewFormat

STOP_EQUIVALENCE_FILE

STOP_EQUIVALENCE_FILE

ArcPlan

The stop equivalence file is required if the ArcView Stop Group File is requested. This file provides the list of transit stops included in each stop group. The total boardings and alightings for all selected transit routes that use one or more stops in the stop group are output to the shapefile. The location of the shape point is the simple average X and Y coordinates associated with each stop in the group.

PlanSum

SUBZONE_DATA_FILE

SUBZONE_DATA_FILE

ArcNet

SUBZONE_ZONE_FACTOR_FILE

SUBZONE_ZONE_FACTOR_FILE

LocationData

TIME_DISTRIBUTION_FILE_*

Required in ConvertTrips. **NEW_TIME_DISTRIBUTION_FILE** is used in PlanCompare.

The time distribution file key is appended to the PROJECT_DIRECTORY key to specify the file name for the input trip time file for the trip group. If the trip time format is not specified and a Definition file is not found, the program assumes the file is in Version 3 format. The default Version 3 format is a tab-delimited text file with three floating point data fields and no header record. The first field is the start time in hours, the second field is the end time in hours, and the third field is the relative share of trips assigned to the period between the start time and end time. The shares are automatically normalized to 1.0, and are therefore NOT required to add up to 1.0. An example file appears below:

Table 30 TIME_DISTRIBUTION_FILE Example

Start	End	Share
0	5	0.005
5	6	0.02
6	7	0.04
7	8	0.075
8	9	0.1
9	10	0.06
10	14	0.16
14	15	0.06
15	16	0.07
16	17	0.08
17	18	0.09
18	19	0.08
19	20	0.06
20	21	0.04
21	22	0.03
22	23	0.02
23	24	0.01

TIMING_PLAN_FILE

NEW_TIMING_PLAN_FILE

IntControl Default Control Key
NewFormat

TIMING_PLAN_FILE

ArcNet
IntControl Default Control Key
Microsimulator
NewFormat

The Timing Plan file has been significantly expanded in TRANSIMS Version 5 compared to previous releases. The new Timing Plan file makes extensive use of data field nesting. In particular, nested phase records are used to improve record management and minimize coding mistakes compared with prior versions. Also, the new timing plan file utilizes barrier, ring, and position codes in order to provide clear sequencing and improved linkages to traffic signal software (e.g., Synchro®). Lastly, the timing plan uses the signal control number and the timing ID together for record indexing. Signal type, offset, and cycle length are also included in the Version 5 timing plan file.

Table 31: TIMING_PLAN_FILE Example

Signal	Timing	Type	Cycle	Offset	Phases	Notes		
Phase	Barrier	Ring	Position	Min_Green	Max_Green	Extension	Yellow	All_Red
1	1	Actuated	100	0	4	0:00..6:00		
1	1	1	1	5	5	0	0	0
2	1	1	2	20	39	12	3	1
3	1	1	3	5	9	3	0	0
4	1	1	4	20	39	12	3	1

A simple example Timing Plan file is provided above.

The Timing Plan file's list of fields and related data taken from the timing plan definition (*.def) file follows:

```
TRANSIMS50, TAB_DELIMITED, 2, NESTED
SIGNAL, INTEGER, 1, 10
TIMING, INTEGER, 2, 10
TYPE, STRING, 3, 10, SIGNAL_TYPE
CYCLE, INTEGER, 4, 5, SECONDS
OFFSET, INTEGER, 5, 5, SECONDS
PHASES, INTEGER, 6, 3, NEST_COUNT
```

```
NOTES, STRING, 7, 128
PHASE, INTEGER, 1, 3, NO, NESTED
BARRIER, INTEGER, 2, 3, NO, NESTED
RING, INTEGER, 3, 3, NO, NESTED
POSITION, INTEGER, 4, 3, NO, NESTED
MIN_GREEN, INTEGER, 5, 5, SECONDS, NESTED
MAX_GREEN, INTEGER, 6, 5, SECONDS, NESTED
EXTENSION, INTEGER, 7, 5, SECONDS, NESTED
YELLOW, INTEGER, 8, 3, SECONDS, NESTED
ALL_RED, INTEGER, 9, 3, SECONDS, NESTED
```

Table 32: TIMING_PLAN_FILE Field Definitions

Field(s)	Description	Default Units
SIGNAL	The signal field indicates the signal number and replaces the signal controller's functions	INTEGER
TIMING	The ID number of the timing plan	INTEGER
TYPE	The signal type (e.g., Actuated, Timed)	STRING
CYCLE	The time required to complete one sequence of intervals (i.e., a cycle), measured in seconds	INTEGER
OFFSET	Relative offset in seconds for timed signals	INTEGER
PHASES	Where multiple phasing plans exist over the course of a day, this is the phasing plan number	INTEGER
NOTES	A character string for user annotations	STRING
PHASE	The phase number	INTEGER
BARRIER	The barrier number associated with a given phase	INTEGER
RING	Number of actuated signal rings	INTEGER
POSITION	The movement number associated with a given phase	INTEGER
MIN_GREEN	The minimum green time in seconds for an actuated signal	INTEGER
MAX_GREEN	The maximum green time in seconds for an actuated signal; the default is minimum green plus one extension	INTEGER
EXTENSION	The number of seconds the green time is extended each time vehicles are detected	INTEGER
YELLOW	The yellow interval in seconds	INTEGER
ALL_RED	The all red interval in seconds	INTEGER

Below is an example Timing Plan file populated with real data. Of particular note are the nested groupings. The master records are shaded in blue; the inner nest has a lighter shade:

Table 33 TIMING_PLAN_FILE Second Example

SIGNAL	TIMING	TYPE	CYCLE	OFFSET	PHASES	NOTES		
PHASE	BARRIER	RING	POSITION	MIN_GREEN	MAX_GREEN	EXTENSION	YELLOW	ALL_RED
1	1	ACTUATED	90	0	4	0:00..7:00		
1	1	1	1	5	5	0	0	0
2	1	1	2	13	25	7	3	1
3	1	1	3	14	27	7	0	0
4	1	1	4	13	25	7	3	1
1	2	ACTUATED	90	0	4	7:00..10:00		
1	1	1	1	5	5	0	0	0
2	1	1	2	13	25	7	3	1
3	1	1	3	14	27	7	0	0
4	1	1	4	13	25	7	3	1
1	3	ACTUATED	90	0	4	10:00..27:00		
1	1	1	1	5	5	0	0	0
2	1	1	2	13	25	7	3	1
3	1	1	3	14	27	7	0	0
4	1	1	4	13	25	7	3	1
2	1	ACTUATED	90	0	3	0:00..7:00		
1	1	1	1	5	5	0	3	1
2	1	1	2	17	33	9	0	0
3	1	1	3	22	44	11	3	1
2	2	ACTUATED	90	0	3	7:00..10:00		
1	1	1	1	5	5	0	3	1
2	1	1	2	17	33	9	0	0
3	1	1	3	22	44	11	3	1
2	3	ACTUATED	90	0	3	10:00..27:00		
1	1	1	1	5	5	0	3	1
2	1	1	2	17	33	9	0	0
3	1	1	3	22	44	11	3	1
3	1	ACTUATED	90	0	3	0:00..7:00		
1	1	1	1	5	5	0	0	0
2	1	1	2	23	46	12	3	1
3	1	1	3	16	31	8	3	1

SIGNAL	TIMING	TYPE	CYCLE	OFFSET	PHASES	NOTES		
PHASE	BARRIER	RING	POSITION	MIN_GREEN	MAX_GREEN	EXTENSION	YELLOW	ALL_RED
3	2	ACTUATED	90	0	3	7:00..10:00		
1	1	1	1	5	5	0	0	0
2	1	1	2	23	46	12	3	1
3	1	1	3	16	31	8	3	1
3	3	ACTUATED	90	0	3	10:00..27:00		
1	1	1	1	5	5	0	0	0
2	1	1	2	23	46	12	3	1
3	1	1	3	16	31	8	3	1

TOLL_FILE

TOLL_FILE

NewFormat

TRANSIT_DRIVER_FILE

NEW_TRANSIT_DRIVER_FILE

NewFormat

TRANSIT_DRIVER_FILE

ArcNet
ArcPlan
Microsimulator
NewFormat

TRANSIT_FARE_FILE

NEW_TRANSIT_FARE_FILE

NewFormat

TRANSIT_FARE_FILE

Microsimulator
NewFormat
PathSkim
Router

TRANSIT_PENALTY_FILE

TRANSIT_PENALTY_FILE

Microsimulator
PathSkim
Router

TRANSIT_ROUTE_FILE

NEW_TRANSIT_ROUTE_FILE

NewFormat

TRANSIT_ROUTE_FILE

ArcNet
ArcPlan
LocationData
Microsimulator
NewFormat
PathSkim
Router

TRANSIT_SCHEDULE_FILE

NEW_TRANSIT_SCHEDULE_FILE

NewFormat

TRANSIT_SCHEDULE_FILE

ArcNet
LocationData
Microsimulator
NewFormat
PathSkim
Router

TRANSIT_STOP_FILE

NEW_TRANSIT_STOP_FILE

NewFormat

TRANSIT_STOP_FILE

ArcNet
ArcPlan
LocationData
Microsimulator
NewFormat
PathSkim
Router

TRIP_COST_GAP_FILE

NEW_TRIP_COST_GAP_FILE

PlanCompare

TRIP_FILE

NEW_TRIP_FILE is used by the following programs:

ConvertTrips

NewFormat
TripPrep

TRIP_FILE is used by the following programs:
ConvertTrips
NewFormat
RandomSelect
Router
TripPrep

The trip file key is appended to the PROJECT_DIRECTORY key to specify the file name for the input trip file copied to the output trip file by the program. The new trip file key is appended to the PROJECT_DIRECTORY key to specify the file name for the output trip file created by the program. The program generates one trip record for each trip in the input trip tables.

In TRANSIMS 5, the trip file includes both trip and activity-related data. An example of trip file output appears below:

Table 34 TRIP_FILE Example

HHOLD	PERSON	TOUR	TRIP	START	END	DURATION	ORIGIN	DESTINATION	PURPOSE	MODE	CONSTRAINT	PRIORITY	VEHICLE	PASSENGERS	TYPE
1	1	1	1	7:33:28	7:48:30	9:43:48	41	54	1	DRIVE		MEDIUM	1	0	
1	1	1	2	17:32:17	17:45:16	0:05	54	26	1	DRIVE		MEDIUM	1	0	
1	1	1	3	17:50:16	17:53:07	1:08:59	26	41	1	DRIVE		MEDIUM	1	0	
1	1	2	1	19:02:06	19:07:33	0:05	41	19	1	DRIVE		MEDIUM	1	0	
1	1	2	2	19:12:33	19:23:46	1:49:17	19	56	1	DRIVE		MEDIUM	1	0	
1	1	2	3	21:13:03	21:29:43	0:00	56	41	1	DRIVE		MEDIUM	1	0	

Activity-related data include the household (HHOLD), person number (PERSON), tour number (TOUR), activity duration (DURATION), activity purpose (PURPOSE), activity constraint (CONSTRAINT), activity priority (PRIORITY), and traveler type (TYPE). Trip-related data include the trip number (TRIP), start time (START), end time (END), origin location (ORIGIN), destination location (DESTINATION), travel mode (MODE), vehicle number (VEHICLE) and number of passengers.

TRIP_TABLE_FILE_1

Used by ConvertTrips

The trip table file key is appended to the PROJECT_DIRECTORY key to specify the file name for the input trip table file for the trip group. If the trip table format is not specified and a Definition file is not found, the program assumes the file is in Version 3 format. The default Version 3 format is a tab-delimited text file with three integer data fields and no header record. The first field is the origin zone number, the second field is the destination zone number, and the third field is the number of trips. An example appears below:

Table 35 TRIP_TABLE_FILE Example

ORG	DES	TRIPS
1	2	500
1	3	500
2	1	500
2	3	500
3	1	500
3	2	500
1	11	10000

TRIP_TIME_FILE

NEW_TRIP_TIME_FILE

PlanSum

TRIP_TIME_GAP_FILE

NEW_TRIP_TIME_GAP_FILE

PlanCompare

TURN_PENALTY_FILE

NEW_TURN_PENALTY_FILE

NewFormat

TransimsNet Default Control Key

TURN_PENALTY_FILE

ArcNet

Microsimulator

NewFormat

PathSkim

Router

TransimsNet Default Control Key

TURN_VOLUME_FILE

NEW_TURN_VOLUME_FILE

LinkSum

TURN_VOLUME_FILE

IntControl Default Control Key

UPDATE_LINK_FILE

UPDATE_LINK_FILE

TransimsNet Default Control Key

The update link file is optional and if specified defines a series of link numbers where the pocket lanes, activity locations, parking lots, and processing links are recalculated. The lane connectivity at both ends of the link is also updated. Each record in the file is interpreted as a comma separated list of link ranges. A link range is specified using two period (e.g., 100..200). The file could also be a simple list of link numbers. The values in the link range and the link file are combined if both keys are provided. In update mode, the program reads existing network files and deletes the existing records for the link and adds new records at the end of the file.

UPDATE_NODE_FILE

UPDATE_NODE_FILE

TransimsNet Default Control Key

The update node file is optional and if specified defines a series of node numbers where the lane connectivity and traffic control warrants are recalculated. Each record in the file is interpreted as a comma separated list of node ranges. A node range is specified using two period (e.g., 100..200). The file could also be a simple list of node numbers. The values in the node range and the node file are combined if both keys are provided. In update mode, the program reads existing network files and deletes the existing records for the node and adds new records at the end of the file.

VEHICLE_FILE

NEW_VEHICLE_FILE is used by the following programs:

ConvertTrips
NewFormat

VEHICLE_FILE is used by the following programs:

ConvertTrips
Simulator
NewFormat
PathSkim
Router

The vehicle_file lists the vehicles in the network. Each vehicle is uniquely identified by a household number and a household vehicle number. The Version 4 “Location” field becomes a “Parking” field. Finally, the Version 4 vehicle type and subtype are combined into a Version 5 vehicle type.

Table 36 VEHICLE_FILE Example

HHOLD	VEHICLE	PARKING	TYPE
1	1	41	1

VEHICLE_TYPE_FILE

NEW_VEHICLE_TYPE_FILE is used by the following programs:
NewFormat

VEHICLE_TYPE_FILE is used by the following programs:
ArcNet
ArcSnapshot
ConvertTrips
Microsimulator
NewFormat
PathSkim
Router

The V4 vehicle subtype no longer exists. Rather, there is a simple type index with valid values ranging from 1 to 99. Operating cost and vertical grade impacts have been added.

The default V4 conversion is V5_Vehicle_Type = 10 * V4_Vehicle_Type + V4_Subtype

Table 37 VEHICLE_TYPE_FILE Example

TYPE	LENGTH	MAX_SPEED	MAX_ACCEL	MAX_DECEL	OP_COST	USE	CAPACITY	LOADING	UNLOADING	METHOD	MIN_DWELL	MAX_DWEL	GRADE_1	GRADE_2	GRADE_3	GRADE_4	GRADE_5	GRADE_6	GRADE_7	GRADE_8	GRADE_9	GRADE_10
1	5.5	162	6	12	20	SOV	5	3	2	SERIAL	0	0	1	1	1	1	1	1	1	1	1	1
2	11.5	162	3	9	25	TRUCK	2	3	2	SERIAL	0	0	1	1	1	1	1	1	1	1	1	1
3	5.5	162	6	12	20	SOV	5	3	2	SERIAL	0	0	1	1	1	1	1	1	1	1	1	1
4	17.5	162	2	6	0	BUS	250	3	2	SERIAL	0	0	1	1	1	1	1	1	1	1	1	1

TYPE	LENGTH	MAX_SPEED	MAX_ACCEL	MAX_DECEL	OP_COST	USE	CAPACITY	LOADING	UNLOADING	METHOD	MIN_DWELL	MAX_DWEL	GRADE_1	GRADE_2	GRADE_3	GRADE_4	GRADE_5	GRADE_6	GRADE_7	GRADE_8	GRADE_9	GRADE_10
5	17.5	162	2	6	0	BUS	250	3	2	SERIAL	0	0	1	1	1	1	1	1	1	1	1	1
11	5.5	162	6	12	20	SOV	5	3	2	SERIAL	0	0	1	1	1	1	1	1	1	1	1	1
12	5.5	162	6	12	20	SOV	5	3	2	SERIAL	0	0	1	1	1	1	1	1	1	1	1	1
13	5.5	162	6	12	20	SOV	5	3	2	SERIAL	0	0	1	1	1	1	1	1	1	1	1	1
14	5.5	162	6	12	20	SOV	5	3	2	SERIAL	0	0	1	1	1	1	1	1	1	1	1	1
15	5.5	162	6	12	20	SOV	5	3	2	SERIAL	0	0	1	1	1	1	1	1	1	1	1	1
16	5.5	162	6	12	20	SOV	5	3	2	SERIAL	0	0	1	1	1	1	1	1	1	1	1	1
21	11.5	162	3	9	25	TRUCK	2	3	2	SERIAL	0	0	1	1	1	1	1	1	1	1	1	1
22	11.5	162	3	9	25	TRUCK	2	3	2	SERIAL	0	0	1	1	1	1	1	1	1	1	1	1
23	17.5	162	2	6	50	TRUCK	2	3	2	SERIAL	0	0	1	1	1	1	1	1	1	1	1	1
24	17.5	162	2	6	50	TRUCK	2	3	2	SERIAL	0	0	1	1	1	1	1	1	1	1	1	1

VERSION4_PLAN_FILE

VERSION4_PLAN_FILE

NewFormat

ZONE_BOUNDARY_FILE

ZONE_BOUNDARY_FILE

LocationData
TransimsNet Default Control Key

ZONE_FIELD_NAME, "ZONE_FIELD_NAME", LEVEL0, OPT_KEY, TEXT_KEY, "", "ZONE, TAZ, Z, ID", NO_HELP

ZONE_EQUIVALENCE_FILE

Used in:

ConvertTrips
LinkSum
PathSkim
PlanSum

The zone equivalence file is required for the trip adjustment factors. The key specifies the name of the file that defines a group of zones. Zone Groups typically represent large geographic areas or governmental entities (i.e., cities and counties). Each zone may only be associated with one Zone Group. The software generates warning messages if a zone is used more than once or appears to be missing from the sequence of zone numbers.

The zone equivalence file is a tab, space, or comma-delimited ASCII file with special format rules. A sample equivalence file is shown below.

```
1 0 Portland CBD - 1
1 1 1..16
2 0 West Suburbs - 2
2 1 79..307, 1248..1253
3 0 Southwest Suburbs - 3
3 1 308..403, 931..933
4 0 Southeast Suburbs - 4
4 1 404..557, 934..943, 1254..1258
5 0 East Portland - 55 1 561..563, 714..721, 731..738, 763..929, 949..961
6 0 East Suburbs - 6
6 1 558..560, 564..713, 722..730, 739..762, 1259..1260
7 0 West Portland - 7
7 1 17..78, 930, 944..948, 962, 1247
8 0 Clark County - 8
8 1 970..1246
```

If the file contains a header record, it is ignored by the software. The first integer on each subsequent record is the district or zone group number. This number is followed by an index number that is used to associate multiple records with a given district. If the index number is zero, the software interprets everything that follows the index number as the district label. The first 25 characters of the label are printed in reports.

If the index number is not zero, the values that follow are interpreted as a range of zone numbers.

Individual zone numbers and ranges of zone numbers can be specified on a given record. A range of zone numbers is specified using the first and last number in the sequence connected by two or more periods. For example, “79..307” represents all of the zone numbers between 79 and 307.

ZONE_FILE

Names: ZONE_FILE, NEW_ZONE_FILE

Used In:
ArcNet
ConvertTrips
LocationData
NetPrep
NewFormat
PathSkim
TransimsNet

The TRANSIMS zone file provides a list of zones in the network. The full path and file name for the zone table is constructed by appending the value of this key to the value of the PROJECT_DIRECTORY key.

A typical field definition (.def) file is as follows:

```
TRANSIMS50, TAB_DELIMITED, 1
ZONE, INTEGER, 1, 10
X_COORD, DOUBLE, 2, 14.1, FEET
Y_COORD, DOUBLE, 3, 14.1, FEET
Z_COORD, DOUBLE, 4, 14.1, FEET
AREA_TYPE, INTEGER, 5, 3
NOTES, STRING, 6, 128
```

Essential information includes the zone number (an integer) and the X and Y coordinates. These are typically UTM coordinates.

Zone numbers do not have to be consecutive. However, external zones are typically assigned higher numbers than internal zones.

An example of a zone file appears below:

Table 38 ZONE_FILE Example

ZONE	X_COORD	Y_COORD	Z_COORD	AREA_TYPE	NOTES
1	4921.3	14763.8	0.0	2	Internal Zone
2	8202.1	14763.8	0.0	2	Internal Zone
20	6561.7	17060.3	0.0	0	External Zone
21	9842.5	17060.3	0.0	0	External Zone

ZONE_LOCATION_MAP_FILE

NEW_ZONE_LOCATION_MAP_FILE

LocationData

ZONE_LOCATION_MAP_FILE

ConvertTrips

PathSkim

ZONE_TRAVEL_FILE

NEW_ZONE_TRAVEL_FILE

LinkSum