TRANSIMS Version 5  
File Reference

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

[List of Figures 4](#_Toc362520204)

[List of Tables 5](#_Toc362520205)

[ACCESS\_FILE 7](#_Toc362520206)

[ACTIVITY\_FILE 8](#_Toc362520207)

[ARC\_...FILE 8](#_Toc362520208)

[ARC\_ACCESS\_FILE 9](#_Toc362520209)

[ARC\_ACCESSIBILITY\_FILE 9](#_Toc362520210)

[ARC\_ BANDWIDTH\_FILE 10](#_Toc362520211)

[ARC\_CENTERLINE \_FILE 12](#_Toc362520212)

[ARC\_CONNECTION\_FILE 12](#_Toc362520213)

[ARC\_ DETECTOR\_FILE 12](#_Toc362520214)

[ARC\_DISTANCE\_CONTOUR \_FILE 13](#_Toc362520215)

[ARC\_ LANE\_USE \_FILE 13](#_Toc362520216)

[ARC\_ LINK\_FILE 13](#_Toc362520217)

[ARC\_LOCATION \_FILE 13](#_Toc362520218)

[ARC\_ NODE\_FILE 13](#_Toc362520219)

[ARC\_ PARKING\_DEMAND\_FILE 14](#_Toc362520220)

[ARC\_ PARKING\_FILE 14](#_Toc362520221)

[ARC\_ PHASING\_PLAN \_FILE 14](#_Toc362520222)

[ARC\_ PLAN\_FILE 15](#_Toc362520223)

[ARC\_ POCKET\_FILE 16](#_Toc362520224)

[ARC\_PROBLEM \_FILE 16](#_Toc362520225)

[ARC\_RIDERSHIP \_FILE 16](#_Toc362520226)

[ARC\_ ROUTE\_NODES \_FILE 17](#_Toc362520227)

[ARC\_ SIGN\_FILE 17](#_Toc362520228)

[ARC\_ SIGNAL \_FILE 17](#_Toc362520229)

[ARC\_ SNAPSHOT \_FILE 17](#_Toc362520230)

[ARC\_ STOP\_DEMAND \_FILE 17](#_Toc362520231)

[ARC\_ STOP\_GROUP \_FILE 17](#_Toc362520232)

[ARC\_ SUBZONE\_DATA \_FILE 18](#_Toc362520233)

[ARC\_ TIME\_CONTOUR \_FILE 18](#_Toc362520234)

[ARC\_ TIMING\_PLAN \_FILE 18](#_Toc362520235)

[ARC\_ TRANSIT\_DRIVER \_FILE 18](#_Toc362520236)

[ARC\_ TRANSIT\_ROUTE \_FILE 18](#_Toc362520237)

[ARC\_ TRANSIT\_STOP \_FILE 19](#_Toc362520238)

[ARC\_ TURN\_PENALTY \_FILE 19](#_Toc362520239)

[ARC\_ ZONE \_FILE 19](#_Toc362520240)

[COMPARE\_PERFORMANCE\_FILE 19](#_Toc362520241)

[COMPARE\_PLAN\_FILE 19](#_Toc362520242)

[Configuration File 19](#_Toc362520243)

[CONNECTION\_FILE 20](#_Toc362520244)

[Control Files/Printout Files (.CTL, .PRN) 24](#_Toc362520245)

[CONVERSION\_SCRIPT 25](#_Toc362520246)

[COST\_DISTRIBUTION\_FILE 27](#_Toc362520247)

[NEW\_COST\_DISTRIBUTION\_FILE 27](#_Toc362520248)

[DATA\_FILE OR DATA\_FILE\_# 27](#_Toc362520249)

[Definition Files (\*.DEF) 27](#_Toc362520250)

[DELETE\_LINK\_FILE 29](#_Toc362520251)

[DELETE\_NODE\_CONTROL\_FILE 30](#_Toc362520252)

[DELETE\_NODE\_FILE 32](#_Toc362520253)

[DESTINATION\_LOCATION\_FILE 32](#_Toc362520254)

[DESTINATION\_ZONE\_FILE 33](#_Toc362520255)

[DETECTOR\_FILE 33](#_Toc362520256)

[DIRECTIONAL\_DATA\_FILE 35](#_Toc362520257)

[EVENT\_FILE 36](#_Toc362520258)

[GROUP\_TRAVEL\_FILE 37](#_Toc362520259)

[HOUSEHOLD\_FILE 38](#_Toc362520260)

[INPUT\_LINK\_FILE 39](#_Toc362520261)

[INPUT\_NODE\_FILE 40](#_Toc362520262)

[INPUT\_SIGN\_FILE 40](#_Toc362520263)

[INPUT\_SIGNAL\_FILE 41](#_Toc362520264)

[INPUT\_SPDCAP\_FILE 42](#_Toc362520265)

[INPUT\_ZONE\_FILE 42](#_Toc362520266)

[KEEP\_LINK\_FILE 43](#_Toc362520267)

[KEEP\_NODE\_FILE 43](#_Toc362520268)

[LANE\_USE\_FILE 44](#_Toc362520269)

[LINK\_ACTIVITY\_FILE 46](#_Toc362520270)

[LINK\_DATA\_FILE 46](#_Toc362520271)

[LINK\_DELAY\_FILE 47](#_Toc362520272)

[LINK\_DETAIL\_FILE 48](#_Toc362520273)

[LINK\_EQUIVALENCE\_FILE 48](#_Toc362520274)

[LINK\_NODE\_EQUIVALENCE 49](#_Toc362520275)

[LINK\_FILE 49](#_Toc362520276)

[LINK\_NODE\_LIST\_FILE 52](#_Toc362520277)

[LINK\_SUMMARY\_FILE 52](#_Toc362520278)

[LINK\_VOLUME\_FILE 52](#_Toc362520279)

[LOCATION\_FILE 53](#_Toc362520280)

[MERGE\_LINK\_DELAY\_FILE 55](#_Toc362520281)

[MERGE\_PLAN\_FILE 55](#_Toc362520282)

[MERGE\_TRIP\_FILE 55](#_Toc362520283)

[Nested Data Files 55](#_Toc362520284)

[NEW\_ARC\_...FILE 55](#_Toc362520285)

[NEW\_ZONE\_LOCATION\_MAP\_FILE 56](#_Toc362520286)

[NODE\_FILE 57](#_Toc362520287)

[OCCUPANCY\_FILE 58](#_Toc362520288)

[ORIGIN\_LOCATION\_FILE 59](#_Toc362520289)

[ORIGIN\_ZONE\_FILE 60](#_Toc362520290)

[PARKING\_FILE 60](#_Toc362520291)

[PARKING\_PENALTY\_FILE 62](#_Toc362520292)

[PERFORMANCE\_DATA\_FILE 62](#_Toc362520293)

[PERFORMANCE\_FILE 62](#_Toc362520294)

[PERSON\_FILE 64](#_Toc362520295)

[PHASING\_PLAN\_FILE 65](#_Toc362520296)

[PLAN\_FILE 67](#_Toc362520297)

[POCKET\_FILE 70](#_Toc362520298)

[PROBLEM\_FILE 72](#_Toc362520299)

[REPORT\_FILE 76](#_Toc362520300)

[RIDERSHIP\_FILE 76](#_Toc362520301)

[ROUTE\_NODES\_FILE 78](#_Toc362520302)

[SELECTION\_FILE 78](#_Toc362520303)

[SHAPE\_FILE 79](#_Toc362520304)

[SIGN\_FILE 81](#_Toc362520305)

[SIGNAL\_FILE 82](#_Toc362520306)

[SKIM\_FILE 85](#_Toc362520307)

[SNAPSHOT\_FILE 87](#_Toc362520308)

[STOP\_EQUIVALENCE\_FILE 89](#_Toc362520309)

[SUBZONE\_DATA\_FILE 89](#_Toc362520310)

[SUBZONE\_ZONE\_FACTOR\_FILE 89](#_Toc362520311)

[TIME\_DISTRIBUTION\_FILE\_\* 90](#_Toc362520312)

[TIMING\_PLAN\_FILE 90](#_Toc362520313)

[TOLL\_FILE 94](#_Toc362520314)

[TRANSIT\_DRIVER\_FILE 94](#_Toc362520315)

[TRANSIT\_FARE\_FILE 95](#_Toc362520316)

[TRANSIT\_PENALTY\_FILE 97](#_Toc362520317)

[TRANSIT\_ROUTE\_FILE 97](#_Toc362520318)

[TRANSIT\_SCHEDULE\_FILE 99](#_Toc362520319)

[TRANSIT\_STOP\_FILE 101](#_Toc362520320)

[TRAVELER\_FILE 102](#_Toc362520321)

[TRIP\_COST\_GAP\_FILE 103](#_Toc362520322)

[TRIP\_FILE 103](#_Toc362520323)

[TRIP\_TABLE\_FILE\_1 104](#_Toc362520324)

[TRIP\_TIME\_FILE 105](#_Toc362520325)

[TRIP\_TIME\_GAP\_FILE 105](#_Toc362520326)

[TURN\_PENALTY\_FILE 105](#_Toc362520327)

[TURN\_VOLUME\_FILE 106](#_Toc362520328)

[UPDATE\_LINK\_FILE 107](#_Toc362520329)

[UPDATE\_NODE\_FILE 107](#_Toc362520330)

[VEHICLE\_FILE 107](#_Toc362520331)

[VEHICLE\_TYPE\_FILE 108](#_Toc362520332)

[VERSION4\_PLAN\_FILE 109](#_Toc362520333)

[ZONE\_BOUNDARY\_FILE 109](#_Toc362520334)

[ZONE\_EQUIVALENCE\_FILE 109](#_Toc362520335)

[ZONE\_FILE 110](#_Toc362520336)

[ZONE\_LOCATION\_MAP\_FILE 111](#_Toc362520337)

[ZONE\_TRAVEL\_FILE 111](#_Toc362520338)

# List of Figures

Figure 1 Process and Access links: Version 4 and Version 5 7

Figure 2 ARC\_ACCESSIBILITY\_FILE Example 10

Figure 3 ARC\_BANDWIDTH File Example 1 11

Figure 4 ARC\_BANDWIDTH\_FILE Example 2 12

Figure 5 ARC\_PARKING\_DEMAND\_FILE Example 14

Figure 6 ARC\_PLAN\_FILE Example 16

Figure 7 CONNECTION\_FILE Illustration 21

Figure 8 Lane Number Difference Between V4 and V5 22

Figure 9 Version 4 Lane Connectivity Edits 23

Figure 10 Version 5 Connection Edits 24

Figure 11 LINK\_EQUIVALENCE\_FILE Example Network 49

Figure 12 Example of Links 51

Figure 13 Example of Locations 54

Figure 14 Hierarchical Relationships Among the Signalized Intersection Files 66

Figure 15 Pocket Lanes 72

Figure 16 Links, Nodes and Shape Points 81

Figure 17 Entity-Relationship Diagram - Signal File Dependencies 85

# List of Tables

Table 1 CONNECTION\_FILE Field Definitions 21

Table 2 CONNECTION\_FILE Example 22

Table 3 Example of a Nested File Structure 29

Table 4: DETECTOR\_FILE Field Definitions 34

Table 5 DETECTOR\_FILE Example 35

Table 6 EVENT\_FILE Field Definitions 36

Table 7 EVENT\_FILE 37

Table 8 HOUSEHOLD\_FILE Field Definitions 39

Table 9 HOUSEHOLD\_FILE Example 39

Table 10 LANE\_USE File Field Definitions 44

Table 11 LANE\_USE File Example 46

Table 12 LINK\_DELAY\_FILE Field Definitions 47

Table 13 LINK\_DELAY\_FILE Example 48

Table 14 LINK\_FILE Field Definitions 50

Table 15 LINK\_FILE Fields Example 51

Table 16 LOCATION\_FILE Fields Example 54

Table 17 NODE\_FILE Example 58

Table 18 OCCUPANCY\_FILE Field Definitions 58

Table 19 OCCUPANCY\_FILE Example 59

Table 20 PARKING\_FILE - Simple Example 61

Table 21 PARKING\_FILE Example with Nested Records 61

Table 22 PERFORMANCE\_FILE Field Definitions 63

Table 23 PERFORMANCE\_FILE Example 64

Table 24 PHASING\_PLAN\_FILE Example 65

Table 25 PHASING\_PLAN\_FILE Field Definitions 67

Table 26 PLAN\_FILE Field Definitions 69

Table 27 PLAN\_FILE Example: Primary Trip Record 70

Table 28 PLAN\_FILE Example: Path Records 70

Table 29 POCKET\_FILE Field Definitions 71

Table 30 POCKET\_FILE Example 72

Table 31 Problem Codes 72

Table 32 PROBLEM\_FILE Example 76

Table 33 RIDERSHIP\_FILE Field Definitions 77

Table 34 RIDERSHIP\_FILE Example 77

Table 35 SELECTION\_FILE Example 78

Table 36 SHAPE\_FILE Example 80

Table 37 Information for Links 62 and 63 80

Table 38 Information for Nodes 123, 132, 133 80

Table 39 SIGNAL\_FILE Field Definitions 84

Table 40 SIGNAL\_FILE Example 85

Table 41 SKIM\_FILE Example 87

Table 42 SNAPSHOT\_FILE Field Definitions 88

Table 43 SNAPSHOT\_FILE Example 88

Table 44 TIME\_DISTRIBUTION\_FILE Example 90

Table 45 TIMING\_PLAN\_FILE First Example 91

Table 46 TIMING\_PLAN\_FILE Field Definitions 92

Table 47 TIMING\_PLAN\_FILE Second Example 93

Table 48 TRANSIT\_DRIVER\_FILE Field Definitions 95

Table 49 TRANSIT\_DRIVER\_FILE Example 95

Table 50 TRANSIT\_FARE\_FILE Field Definitions 96

Table 51 TRANSIT\_FARE\_FILE Example (Taken from Version 4) 96

Table 52 Unbundled TRANSIT\_FARE\_FILE Example (after NewFormat) 97

Table 53 TRANSIT\_ROUTE\_FILE Field Definitions 98

Table 54 TRANSIT\_ROUTE\_FILE Example 98

Table 55 TRANSIT\_SCHEDULE\_FILE Field Definitions 100

Table 56 TRANSIT\_SCHEDULE\_FILE Example 100

Table 57 TRANSIT\_STOP\_FILE Field Definitions 101

Table 58 TRANSIT\_STOP\_FILE Example 102

Table 59 NEW\_TRAVELER\_FILE Field Definitions 103

Table 60 NEW\_TRAVELER\_FILE Example 103

Table 61 TRIP\_FILE Example 104

Table 62 TRIP\_TABLE\_FILE Example 105

Table 63 TURN\_PENALTY\_FILE Field Definitions 106

Table 64 TURN\_PENALTY\_FILE Example 106

Table 65 VEHICLE\_FILE Example 108

Table 66 VEHICLE\_TYPE\_FILE Example 108

Table 67 ZONE\_FILE Example 111

# 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 (TRANSIMS 5), Process Links (Version 4) are replaced by Access Links. 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 Figure 1). In TRANSIMS 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.

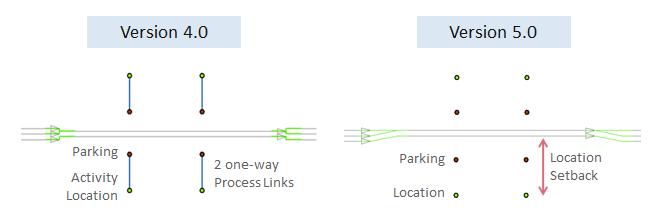


Figure Process and Access links: Version 4 and Version 5

# ACTIVITY\_FILE

Used in NewFormat

The name of the Version 4 ACTIVITY\_FILE that is optionally used by NewFormat.

# ARC\_...FILE

Names:

NEW\_ARC\_ACCESS\_FILE

NEW\_ARC\_ACCESSIBILITY\_FILE

NEW\_ARC\_ BANDWIDTH\_FILE

NEW\_ARC\_ CENTERLINE \_FILE

NEW\_ARC\_ CONNECTION \_FILE

NEW\_ARC\_ DETECTOR \_FILE

NEW\_ARC\_ DISTANCE\_CONTOUR \_FILE

NEW\_ARC\_ LANE\_USE \_FILE

NEW\_ARC\_ LINK \_FILE

NEW\_ARC\_ LOCATION \_FILE

NEW\_ARC\_NODE \_FILE

NEW\_ARC\_ PARKING**\_**DEMAND \_FILE

NEW\_ARC\_ PARKING \_FILE

NEW\_ARC\_ PHASING\_PLAN \_FILE

NEW\_ARC\_POCKET \_FILE

NEW\_ARC\_ RIDERSHIP \_FILE

NEW\_ARC\_ ROUTE\_NODES \_FILE

NEW\_ARC\_ SIGN\_FILE

NEW\_ARC\_ SIGNAL \_FILE

NEW\_ARC\_ SNAPSHOT \_FILE

NEW\_ARC\_ STOP\_DEMAND \_FILE

NEW\_ARC\_ STOP\_GROUP \_FILE

NEW\_ARC\_ SUBZONE\_DATA \_FILE

NEW\_ARC\_ TIME\_CONTOUR \_FILE

NEW\_ARC\_ TIMING\_PLAN \_FILE

NEW\_ARC\_ TRANSIT\_DRIVER \_FILE

NEW\_ARC\_ TRANSIT\_ROUTE \_FILE

NEW\_ARC\_ TRANSIT\_STOP \_FILE

NEW\_ARC\_ TURN\_PENALTY \_FILE

NEW\_ARC\_ ZONE \_FILE

Used in ArcNet

These are shape files produced by ArcNet, ArcPlan and ArcSnapshot. The file name in the control file 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 a Geographic Information System such as ArcView or QGIS to read and display the path.

# ARC\_ACCESS\_FILE

Names: ARC\_ACCESS\_FILE, NEW\_ARC\_ACCESS\_FILE

This NEW\_ARC\_ACCESS\_FILE is output by ArcNet, and is a shape file of access (formerly, process) links. See ACCESS\_FILE, on page 7, for further information on the underlying file.

# ARC\_ACCESSIBILITY\_FILE

Names: ARC\_ACCESSIBILITY\_FILE, NEW\_ARC\_ACCESSIBILITY\_FILE

Used in ArcPlan

The NEW\_ARC\_ACCESSIBILITY\_FILE is output by ArcPlan. It is a shapefile of points showing the travel time and trip distance from a given origin to other activity locations on the network.

Fields in this file include the following:

ORIGIN, INTEGER, 1, 10

START, STRING, 11, 20

LOCATION, INTEGER, 31, 10

TTIME, INTEGER, 41, 10

DISTANCE, INTEGER, 51, 10

Figure 2 shows an example of the shapefile. Here distances are calculated from location 27, which is at the upper right corner of the figure. They are plotted by color: red is closer to the origin, green further away.

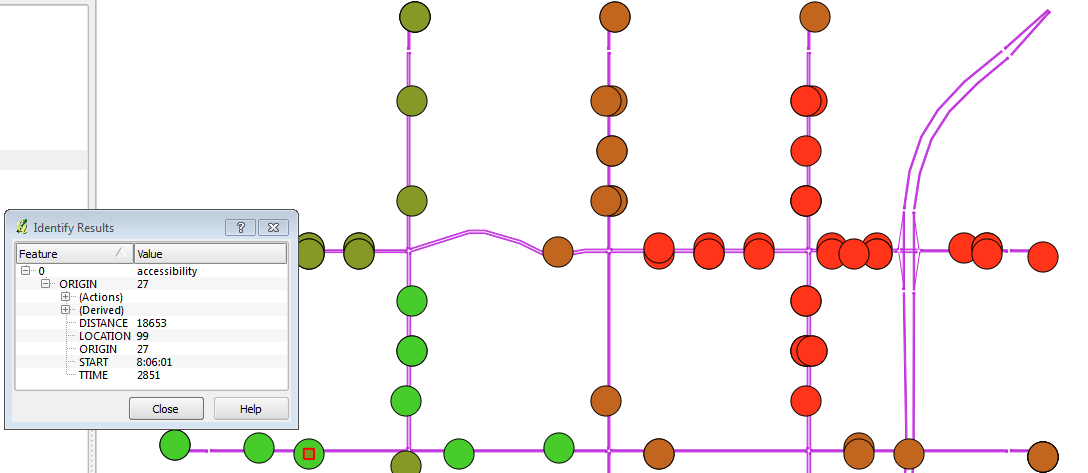


Figure ARC\_ACCESSIBILITY\_FILE Example

# ARC\_ BANDWIDTH\_FILE

Names: ARC\_ BANDWIDTH\_FILE, NEW\_ARC\_ BANDWIDTH\_FILE

Used in ArcPlan

The NEW\_ARC\_ BANDWIDTH\_FILE is output by ArcPlan. It is a shapefile of polygons showing the number of trips on particular links of the network.

Fields in this file include the following:

LINK, INTEGER, 1, 10

DIR, INTEGER, 11, 1

VOLUME, INTEGER, 12, 10

In the first example (Figure 3), bandwidths are shown for trips from a single origin (at the upper right corner of the screen).

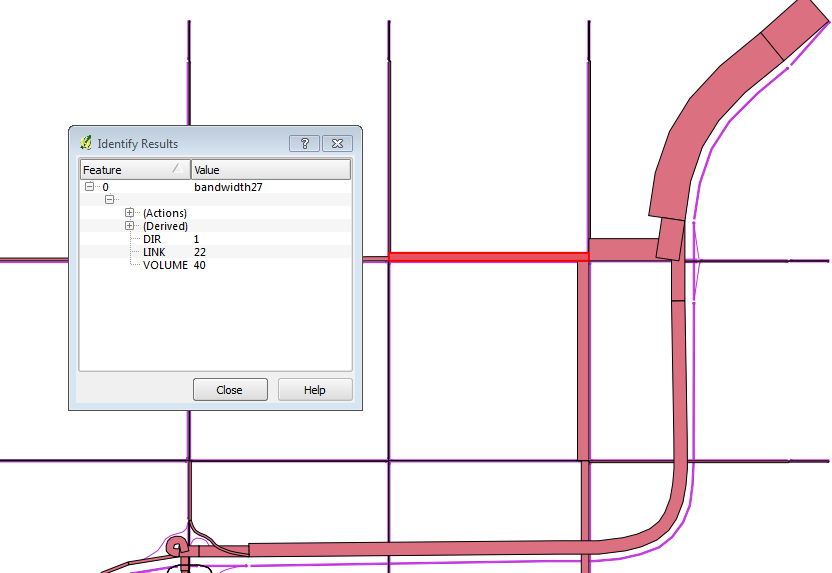


Figure ARC\_BANDWIDTH File Example 1

The second example () shows all trips. In this example, the BANDWIDTH\_SCALING\_FACTOR was set to 10 units / meter (rather than the default of 1) so that the bands would be of reasonable size. When using ArcPlan to produce bandwidth files, it is often necessary to experiment with the scaling factor. If the bands cover your map, the BANDWIDTH\_SCALING\_FACTOR should be increased.

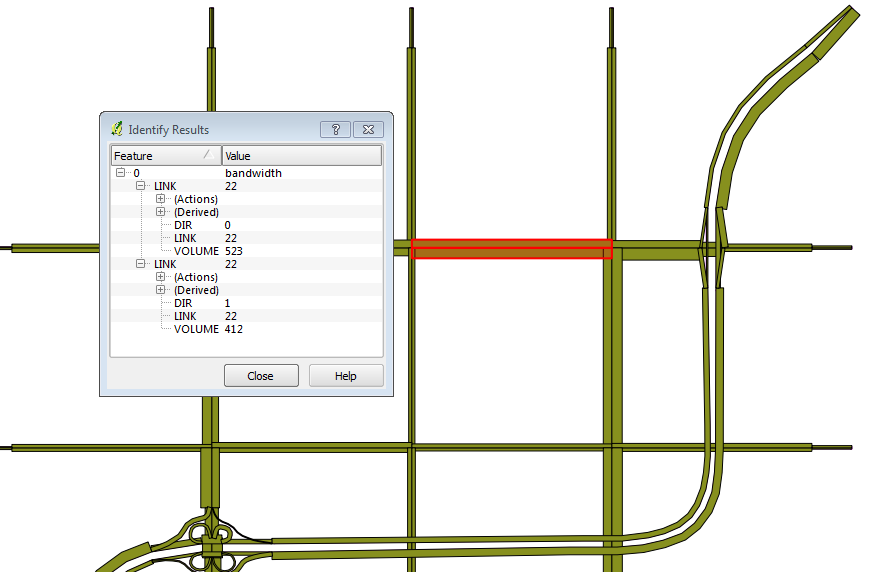


Figure ARC\_BANDWIDTH\_FILE Example 2

# 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

The NEW\_ARC\_CONNECTION\_FILE is output by ArcNet, and is a shape file of connections between inbound and outbound lanes at an intersection. For example, the connection file might show that pocket lane 1 on link 24 connects to lanes 1 and 2 on link 25. See CONNECTION\_FILE, on page 20, for further information on the underlying file.

# ARC\_ DETECTOR\_FILE

Names: ARC\_ DETECTOR \_FILE, NEW\_ARC\_ DETECTOR \_FILE

Used in ArcNet

The NEW\_ARC\_DETECTOR\_FILE is output by ArcNet, and is a shape file of traffic signal detectors. See DETECTOR\_FILE, on page 27, for further information on the underlying file.

# ARC\_DISTANCE\_CONTOUR \_FILE

Names: ARC\_ DISTANCE\_CONTOUR \_FILE, NEW\_ARC\_ DISTANCE\_CONTOUR \_FILE

Used in ArcPlan

A shapefile showing trip length contours from a given origin to all destinations. It does not appear to have been implemented yet.

# ARC\_ LANE\_USE \_FILE

Names: ARC\_ LANE\_USE \_FILE, NEW\_ARC\_ LANE\_USE \_FILE

Used in ArcNet

The NEW\_ARC\_LANE\_USE\_FILE is output by ArcNet, and is a shape file of lane use restrictions. See LANE\_USE\_FILE, on page 38, for further information on the underlying file.

# ARC\_ LINK\_FILE

Names: ARC\_ LINK \_FILE, NEW\_ARC\_ LINK \_FILE

Used in ArcNet

The NEW\_ARC\_LINK\_FILE is output by ArcNet, and is a shape file of links. It combines information from the link file, along with shape point information from the TRANSIMS node and shape files. See LINK\_FILE on page 49 and SHAPE\_FILE on page 79 for further information on the underlying files.

# ARC\_LOCATION \_FILE

Names: ARC\_ LOCATION \_FILE, NEW\_ARC\_ LOCATION \_FILE

Used in ArcNet

The NEW\_ARC\_LOCATION\_FILE is output by ArcNet, and is a shape file of locations (formerly, activity locations). See LOCATION\_FILE on page 53 for further information on the underlying file.

# ARC\_ NODE\_FILE

Names: ARC\_NODE \_FILE, NEW\_ARC\_NODE \_FILE

Used in ArcNet

The NEW\_ARC\_NODE\_FILE is output by ArcNet, and is a shape file of nodes.

# ARC\_ PARKING\_DEMAND\_FILE

Names: ARC\_ PARKING**\_**DEMAND \_FILE, NEW\_ARC\_ PARKING**\_**DEMAND \_FILE

Used in ArcPlan

The NEW\_ARC\_ PARKING**\_**DEMAND \_FILE is output by ArcPlan, and shows arrival and departure demand at the parking lots associated with activity locations. Fields in the file include:

PARKING, INTEGER, 1, 10

DEPART, INTEGER, 11, 10

ARRIVE, INTEGER, 21, 10

TOTAL, INTEGER, 31, 10

Figure 5 shows an example of a arc\_parking\_demand file. The pie charts were created by the GIS, using the arrive, depart and total demand data from TRANSIMS. To create them, the yellow part (on the bottom) was set to indicate arrivals, the blue part departures and the overall size of the pie is proportional to total demand.

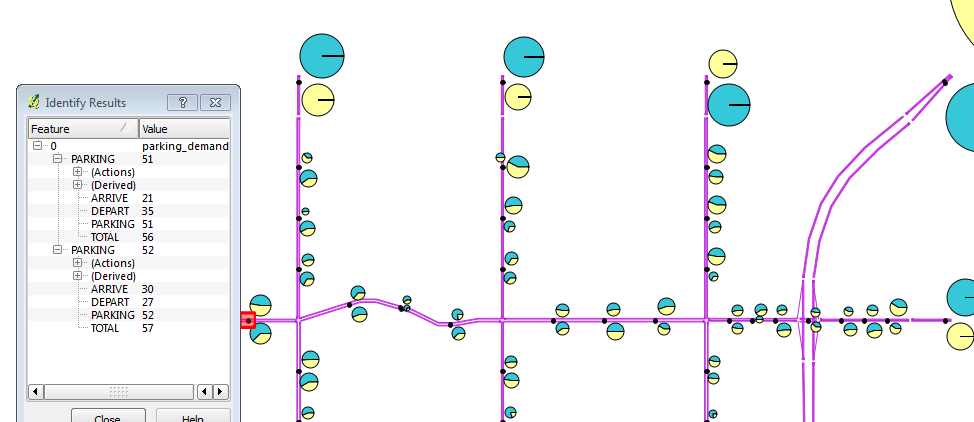


Figure ARC\_PARKING\_DEMAND\_FILE Example

# ARC\_ PARKING\_FILE

Names: ARC\_ PARKING \_FILE, NEW\_ARC\_ PARKING \_FILE

Used in ArcNet

The NEW\_ARC\_PARKING\_FILE is output by ArcNet, and is a shape file of parking locations. See PARKING\_FILE, on page 60 for further information on the underlying file.

# ARC\_ PHASING\_PLAN \_FILE

Names: ARC\_ PHASING\_PLAN \_FILE, NEW\_ARC\_ PHASING\_PLAN \_FILE

Used in ArcNet

The NEW\_ARC\_PHASING\_PLAN\_FILE is output by ArcNet, and is a shape file of traffic signal phasing plans. See PHASING\_PLAN\_FILE, on page 65, for further information on the underlying file.

# ARC\_ PLAN\_FILE

Names: ARC\_ PLAN\_FILE, NEW\_ARC\_PLAN \_FILE

Used in ArcPlan

The NEW\_ARC\_PLAN\_FILE is output by ArcPlan, and is a shape file of travel plans. See PLAN\_FILE, on page 67, for further information on the underlying file. Fields in the ARC\_PLAN\_FILE include the following:

HHOLD, INTEGER, 1, 10

PERSON, INTEGER, 11, 5

TOUR, INTEGER, 16, 3

TRIP, INTEGER, 19, 3

START, TIME, 22, 16, HOUR\_CLOCK

END, TIME, 38, 16, HOUR\_CLOCK

DURATION, TIME, 54, 16, HOUR\_CLOCK

ORIGIN, INTEGER, 70, 10

DESTINATION, INTEGER, 80, 10

PURPOSE, INTEGER, 90, 2

MODE, STRING, 92, 12, MODE\_TYPE

CONSTRAINT, STRING, 104, 14, CONSTRAINT\_TYPE

PRIORITY, STRING, 118, 10, PRIORITY\_TYPE

VEHICLE, INTEGER, 128, 4

PASSENGERS, INTEGER, 132, 2

TYPE, INTEGER, 134, 4

DEPART, TIME, 138, 16, HOUR\_CLOCK

ARRIVE, TIME, 154, 16, HOUR\_CLOCK

ACTIVITY, TIME, 170, 16, HOUR\_CLOCK

WALK, TIME, 186, 12, SECONDS

DRIVE, TIME, 198, 12, SECONDS

TRANSIT, TIME, 210, 12, SECONDS

WAIT, TIME, 222, 12, SECONDS

OTHER, TIME, 234, 12, SECONDS

LENGTH, INTEGER, 246, 10, FEET

COST, FIXED, 256, 6.1, CENTS

IMPEDANCE, UNSIGNED, 262, 10, IMPEDANCE

NUM\_LEGS, INTEGER, 272, 5, NEST\_COUNT

LEG\_MODE, STRING, 277, 12, MODE\_TYPE

LEG\_ID, INTEGER, 289, 10

LEG\_TIME, TIME, 299, 10, SECONDS

LEG\_LENGTH, INTEGER, 309, 10, FEET

LEG\_COST, DOUBLE, 319, 6.1, CENTS

LEG\_IMPED, INTEGER, 325, 10, IMPEDANCE

is an example of an ARC\_PLAN\_FILE, for the following plans:

SELECT\_ORIGINS 27 // Upper right corner of the figure

SELECT\_DESTINATIONS 53..58, 80..84 // As shown in Figure 5.

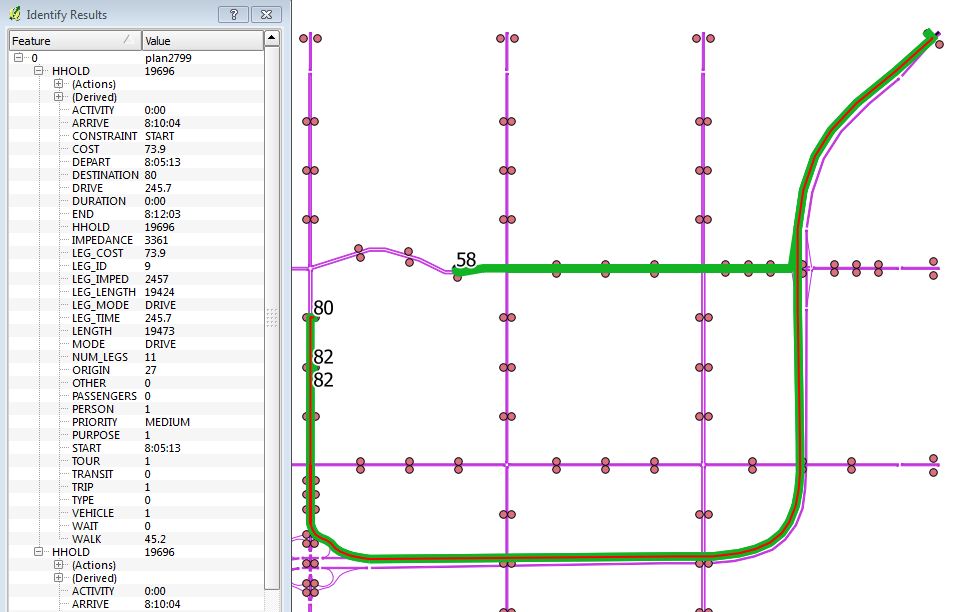


Figure ARC\_PLAN\_FILE Example

# ARC\_ POCKET\_FILE

Names: ARC\_ POCKET\_FILE, NEW\_ARC\_POCKET \_FILE

Used in ArcNet

The NEW\_ARC\_POCKET\_FILE is output by ArcNet, and is a shape file of pocket lanes. See POCKET\_FILE, on page 70, for further information on the underlying file.

# 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.

# 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

The NEW\_ARC\_SIGN\_FILE is output by ArcNet, and is a shape file of traffic control signs (typically, stop and yield signs)..

# ARC\_ SIGNAL \_FILE

Names: ARC\_ SIGNAL \_FILE, NEW\_ARC\_ SIGNAL \_FILE

Used in ArcNet

The NEW\_ARC\_SIGNAL\_FILE is output by ArcNet, and is a shape file of traffic control signals. See SIGNAL\_FILE, on page 82, for further information on the underlying file.

# ARC\_ SNAPSHOT \_FILE

Names: ARC\_ SNAPSHOT \_FILE, NEW\_ARC\_ SNAPSHOT \_FILE

Used in ArcSnapshot

The NEW\_ARC\_ SNAPSHOT \_FILE is output by ArcSnapshot, and is a shape file of vehicle positions (snapshots) on the network. See SNAPSHOT\_FILE, on page 86, for further information on the underlying file.

# 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

An optional shapefile showing trip time contours from a given origin to all destinations. It does not appear to have been implemented yet.

# ARC\_ TIMING\_PLAN \_FILE

Names: ARC\_ TIMING\_PLAN \_FILE, NEW\_ARC\_ TIMING\_PLAN \_FILE

Used in ArcNet

The NEW\_ARC\_TIMING\_PLAN\_FILE is output by ArcNet, and is a shape file of traffic signal timing plans.

# ARC\_ TRANSIT\_DRIVER \_FILE

Names: ARC\_ TRANSIT\_DRIVER \_FILE, NEW\_ARC\_ TRANSIT\_DRIVER \_FILE

Used in ArcNet

The NEW\_ARC\_ TRANSIT\_DRIVER \_FILE is output by ArcNet. See TRANSIT\_DRIVER\_FILE, on page 93, for further information on the underlying file.

# ARC\_ TRANSIT\_ROUTE \_FILE

Names: ARC\_ TRANSIT\_ROUTE \_FILE, NEW\_ARC\_ TRANSIT\_ROUTE \_FILE

Used in ArcNet

The NEW\_ARC\_ TRANSIT\_ROUTE \_FILE is a set of transit routes output by ArcNet. See TRANSIT\_ROUTE\_FILE, on page 96, for further information on the underlying file.

# ARC\_ TRANSIT\_STOP \_FILE

Names: ARC\_ TRANSIT\_STOP \_FILE, NEW\_ARC\_ TRANSIT\_STOP \_FILE

Used in ArcNet

The NEW\_ARC\_ TRANSIT\_STOP \_FILE is a set of transit stops output be ArcNet. See TRANSIT\_STOP\_FILE, on page 100, for further information on the underlying file.

# ARC\_ TURN\_PENALTY \_FILE

Names: ARC\_ TURN\_PENALTY \_FILE, NEW\_ARC\_ TURN\_PENALTY \_FILE

Used in ArcNet

The NEW\_ARC\_TURN\_PENALTY\_FILE is output by ArcNet, and is a shape file of turn penalties and restrictions. See TURN\_PENALTY\_FILE, on page 104, for further information on the underlying file

# ARC\_ ZONE \_FILE

Names: ARC\_ ZONE \_FILE, NEW\_ARC\_ ZONE \_FILE

Used in ArcNet

The NEW\_ARC\_ZONE\_FILE is output by ArcNet, and is a shape file of transportation analysis zones. See ZONE\_FILE, on page 109, for further information on the underlying file.

# COMPARE\_PERFORMANCE\_FILE

This file is an input to LinkSum. It is the second performance file, used in a comparison of two performance files. See the LinkSum Program reference for further information. See PERFORMANCE\_FILE, on page 62, for a definition of this file’s format.

# COMPARE\_PLAN\_FILE

This file is an input to PlanCompare. It is the second plan file, used in a comparison of two sets of travel plans. See the PlanCompare Program reference for further information. See PLAN\_FILE, on page 67, for a definition of this file’s format.

# Configuration File

Used by all programs (global settings; can be overridden 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 listed below:

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

Table 1 defines the fields.

Table 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

Figure 7 and Table 2 provide examples of intersection connections.

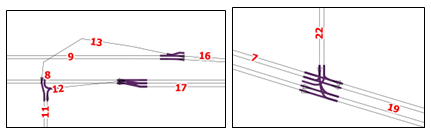


Figure CONNECTION\_FILE Illustration

Table 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 8).

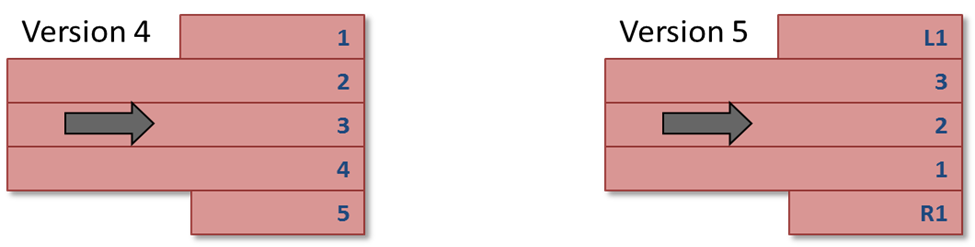


Figure Lane Number Difference Between V4 and V5

This simplifies the lane connectivity edits. In Version 4, the edits would often cascade from one intersection to another (Figure 9).

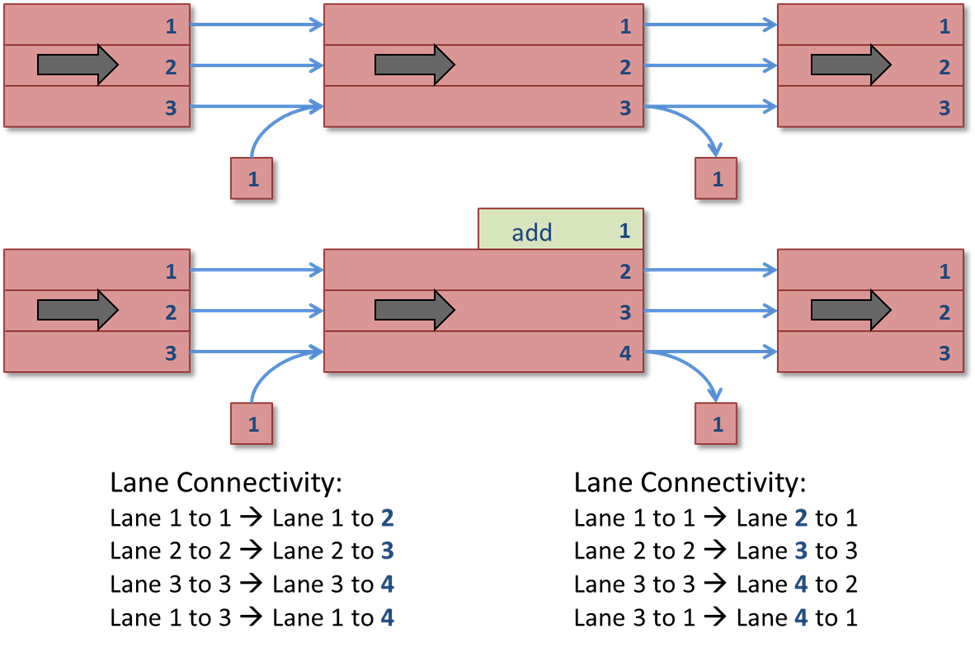


Figure Version 4 Lane Connectivity Edits

In Version 5, the edits are simpler (Figure 10):

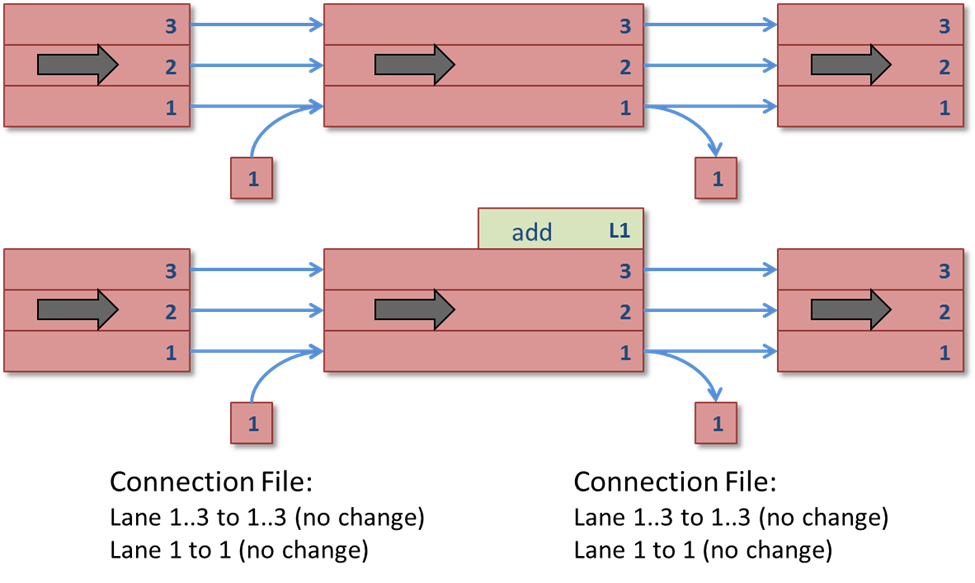


Figure Version 5 Connection Edits

# 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 is shown 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\_#

Used in Location\_Data

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 CONVERSION\_SCRIPT, on page ) 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. shows an example of a nested file structure. In Table 3, 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 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 in Table 3 is listed below:

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 unit 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 shown below:

**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 this output file is shown below:

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. This 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.

The definition file listing the data fields for the 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

Table 4 lists the field definitions for the DETECTOR\_FILE:

Table : 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.) |

Table 5 is an example of a DETECTOR\_FILE populated with data:

Table 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

# EVENT\_FILE

The NEW\_EVENT\_FILE is an output from the Simulator. It lists the scheduled and actual time and link direction and offset for each traveler and trip event (i.e., start time and end time).

An example of a typical definition file is listed below:

TRANSIMS50, TAB\_DELIMITED, 1

HHOLD, INTEGER, 1, 10

PERSON, INTEGER, 2, 5

TOUR, INTEGER, 3, 3

TRIP, INTEGER, 4, 3

MODE, STRING, 5, 12, MODE\_TYPE

EVENT, STRING, 6, 20, EVENT\_TYPE

SCHEDULE, TIME, 7, 16, HOUR\_CLOCK

ACTUAL, TIME, 8, 16, HOUR\_CLOCK

LINK, INTEGER, 9, 10

DIR, INTEGER, 10, 1

LANE, STRING, 11, 3, LANE\_ID\_TYPE

OFFSET, DOUBLE, 12, 8.1, METERS

ROUTE, INTEGER, 13, 10

Table 6 lists the field definitions for the EVENT\_FILE.

Table EVENT\_FILE Field Definitions

| **Field(s)** | **Description** | **Data Type** |
| --- | --- | --- |
| HHOLD | Household identifier | Integer |
| PERSON | Person identifier | Integer |
| TOUR | Tour identifier | Integer |
| TRIP | Trip identifier | Integer |
| MODE | Traveler’s mode code (MODE\_TYPE) | String |
| EVENT | Event (EVENT\_TYPE) | String |
| SCHEDULE | The time of day when the event was scheduled to take place. | Hour\_Clock |
| ACTUAL | The time of day when the event actually took place. | Hour\_Clock |
| LINK | Link identifier | Integer |
| DIR | Direction on the link | Integer |
| LANE | Lane identifier | String |
| OFFSET | Offset of the front of the vehicle from the beginning of the link (defaults to METERS) | Decimal |
| ROUTE | For transit, the route number | Integer |

Event types include the following:

* TRIP\_START\_EVENT, also called TRIP\_START\_TIME, START\_TIME
* TRIP\_END\_EVENT, also called TRIP\_END\_TIME, END\_TIME
* DURATION\_EVENT, also called TRIP\_DURATION, RUN\_TIME
* VEH\_START\_EVENT, also called VEH\_START\_TIME, DEPARTURE\_TIME, VEHICLE\_START
* VEH\_END\_EVENT, also called VEH\_END\_TIME, ARRIVAL\_TIME, VEHICLE\_END
* VEH\_LOST\_EVENT, also called VEH\_LOST\_TIME, LOST\_TIME, VEHICLE\_LOST, LOST\_VEHICLE
* TRANSIT\_WAIT\_EVENT, also called TRANSIT\_WAIT, START\_WAITING, WAITING
* TRANSIT\_ON\_EVENT, also called TRANSIT\_BOARDING, TRANSIT\_ON, BOARDING
* TRANSIT\_OFF\_EVENT, also called TRANSIT\_ALIGHTING, TRANSIT\_OFF, ALIGHTING

Table 7 shows an excerpt from an EVENT\_FILE.

Table EVENT\_FILE

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **HHOLD** | **PERSON** | **TOUR** | **TRIP** | **MODE** | **EVENT** | **SCHEDULE** | **ACTUAL** | **LINK** | **DIR** | **LANE** | **OFFSET** | **ROUTE** |
| 353434 | 1 | 1 | 1 | DRIVE | TRIP\_END\_TIME | 5:15:27 PM | 5:14:43 PM | 0 | 0 | 0 | 0 | 0 |
| 353434 | 1 | 1 | 1 | DRIVE | VEH\_END\_TIME | 5:15:27 PM | 5:14:41 PM | 6233 | 0 | 0 | 1638.7 | 0 |
| 353434 | 1 | 1 | 1 | DRIVE | VEH\_START\_TIME | 5:12:13 PM | 5:12:14 PM | 6229 | 0 | 3 | 18.7 | 0 |
| 353434 | 1 | 1 | 1 | DRIVE | TRIP\_START\_TIME | 5:12:13 PM | 5:12:13 PM | 0 | 0 | 0 | 0 | 0 |

# 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. The Version 5 definition file is listed below:

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 8 lists the field definitions for the HOUSEHOLD\_FILE.

Table 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 |

Table 9 is an example of a HOUSEHOLD\_FILE with one person:

Table 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](#_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

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.

An example of an IntControl control file is shown below, with the relevant key highlighted in bold:

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.

An example of IntControl control file is shown 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.

An example of a KEEP \_LINK\_FILE is listed 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 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.

An example of a 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

In TRANSIMS 5, the LANE\_USE\_FILE takes the place of both the LANE\_USE\_FILE and the TOLL\_FILE from Version 4. It includes lane ranges, lengths and offsets. Tolls are enabled by lane (e.g., HOT lanes). A minimum and maximum delay is added that enables the modeling of toll plaza delay, ramp metering and other flow interruptions.

Table 10 lists the field definitions for the LANE\_USE\_FILE.

Table LANE\_USE File Field Definitions

| **Field(s)** | **Description** | **Default Units** |
| --- | --- | --- |
| LINK | Link number (an integer) |  |
| DIR | Direction (0 = A->B, 1 = B->A) |  |
| LANES | Range of lanes. This is either a single lane number, or a range of lanes. | LANE\_RANGE\_TYPE |
| TYPE | Type of lane\_use restriction. Values include APPLY, A or blank; LIMIT, L or O; PROHIBIT, P or N; REQUIRE or R. | RESTRICTION\_TYPE |
| USE | Type of USE. Values include ANY, WALK, BIKE, BICYCLE, CAR, AUTO , TRUCK, BUS, SOV, HOV2 , HOV3, HOV4 , LIGHTTRUCK, HEAVYTRUCK , RESTRICTED, NONE , TAXI , RAIL, TROLLEY , STREETCAR, LIGHTRAIL, RAPIDRAIL , REGIONRAIL | USE\_TYPE |
| MIN\_TYPE | UNSIGNED | VEHICLE\_TYPE |
| MAX\_TYPE | UNSIGNED | VEHICLE\_TYPE |
| MIN\_TRAV | UNSIGNED | UNSIGNED |
| MAX\_TRAV |  |  |
| START | TIME | HOUR\_CLOCK |
| END | TIME | HOUR\_CLOCK |
| LENGTH | DOUBLE | FEET |
| OFFSET | DOUBLE | FEET |
| TOLL | UNSIGNED | CENTS |
| TOLL\_RATE | DOUBLE | CENTS/MILE |
| MIN\_DELAY | DOUBLE | SECONDS |
| MAX\_DELAY | DOUBLE | SECONDS |
| NOTES | STRING |  |

LANES is a range of lanes. Zero implies all lanes.

The TYPE field corresponds to the RESTRICT field in version 4. LIMIT (“O” in version 4) means that only a particular vehicle type may use the lane. PROHIBIT (“N” in version 4) means that the lane is not allowed to be used by the vehicle type. REQUIRE (“R” in version 4) means that the lane is required to be used by this vehicle type.

MIN\_TYPE and MAX\_TYPE refer to vehicle type numbers, which appear to be indexed as follows: ANY, CAR, TRUCK, CAR, BUS, BUS, RAIL, RAIL, RAIL, RAIL, RAIL, HOV2, HOV3, HOV4, LIGHTTRUCK, HEAVYTRUCK, RESTRICTED, NONE

The LENGTH is the length of the use restriction, measured from the offset. Zero implies the whole link.

The OFFSET is the starting position of the use restriction, measured from the beginning of the link.

The amount of delay is selected as a random number between MIN\_DELAY and MAX\_DELAY.

A typical field definition (.def file or .DEF) file is listed below:

TRANSIMS50, TAB\_DELIMITED, 1

LINK, INTEGER, 1, 10

DIR, INTEGER, 2, 1

LANES, STRING, 3, 8, LANE\_RANGE\_TYPE

TYPE, STRING, 4, 10, RESTRICTION\_TYPE

USE, STRING, 5, 128, USE\_TYPE

MIN\_TYPE, UNSIGNED, 6, 3, VEHICLE\_TYPE

MAX\_TYPE, UNSIGNED, 7, 3, VEHICLE\_TYPE

MIN\_TRAV, UNSIGNED, 8, 3

MAX\_TRAV, UNSIGNED, 9, 3

START, TIME, 10, 16, HOUR\_CLOCK

END, TIME, 11, 16, HOUR\_CLOCK

LENGTH, DOUBLE, 12, 8.1, FEET

OFFSET, DOUBLE, 13, 8.1, FEET

TOLL, UNSIGNED, 14, 5, CENTS

TOLL\_RATE, DOUBLE, 15, 8.1, CENTS/MILE

MIN\_DELAY, DOUBLE, 16, 8.1, SECONDS

MAX\_DELAY, DOUBLE, 17, 8.1, SECONDS

NOTES, STRING, 18, 128

Table 11 is an example of a LANE\_USE\_FILE populated with data.

Table LANE\_USE File Example

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| LINK | DIR | LANES | TYPE | USE | MIN\_TYPE | MAX\_TYPE | MIN\_TRAV | MAX\_TRAV | START | END | LENGTH | OFFSET | TOLL | TOLL\_RATE | MIN\_DELAY | MAX\_DELAY | NOTES |
| 17 | 0 | L1 | LIMIT | TRUCK|SOV|HOV2 | 0 | 0 | 0 | 0 | 15:30 | 18:00 | 0 | 0 | 0 | 0 | 0 | 0 | I395 Exp Lane Restriction |
| 17 | 0 | 2 | LIMIT | TRUCK|SOV|HOV2 | 0 | 0 | 0 | 0 | 15:30 | 18:00 | 0 | 0 | 0 | 0 | 0 | 0 | I395 Exp Lane Restriction |
| 17 | 0 | 1 | LIMIT | TRUCK|SOV|HOV2 | 0 | 0 | 0 | 0 | 15:30 | 18:00 | 0 | 0 | 0 | 0 | 0 | 0 | I395 Exp Lane Restriction |
| 17 | 0 | R1..L1 | LIMIT | TRUCK|SOV|HOV2 | 0 | 0 | 0 | 0 | 15:30 | 18:00 | 0 | 0 | 0 | 0 | 0 | 0 | I395 Exp Lane Restriction |
| 53 | 0 | 1 | PROHIBIT | AUTO|TRUCK|BUS | 0 | 0 | 0 | 0 | 0:00 | 7:00 | 0 | 0 | 0 | 0 | 0 | 0 | Park on 1st St: Right Lane |
| 42 | 0 | L1 | PROHIBIT | AUTO|TRUCK|BUS | 0 | 0 | 0 | 0 | 0:00 | 7:00 | 0 | 0 | 0 | 0 | 0 | 0 | Park on 3rd St: Right Lane |
| 42 | 0 | 2 | PROHIBIT | AUTO|TRUCK|BUS | 0 | 0 | 0 | 0 | 9:30 | 24:00 | 0 | 0 | 0 | 0 | 0 | 0 | Park on 3rd St: Right Lane |
| 42 | 0 | 2 | PROHIBIT | AUTO|TRUCK|BUS | 0 | 0 | 0 | 0 | 0:00 | 7:00 | 0 | 0 | 0 | 0 | 0 | 0 | Park on 3rd St: Right Lane |
| 42 | 0 | 1 | PROHIBIT | AUTO|TRUCK|BUS | 0 | 0 | 0 | 0 | 9:30 | 24:00 | 0 | 0 | 0 | 0 | 0 | 0 | Park on 3rd St: Right Lane |
| 39 | 0 | R1..L1 | APPLY | SOV|HOV3 | 0 | 0 | 0 | 0 | 6:00 | 9:00 | 0 | 0 | 26 | 0 | 0 | 0 |  |
| 39 | 0 | R1..L1 | APPLY | SOV|HOV3 | 0 | 0 | 0 | 0 | 9:00 | 16:00 | 0 | 0 | 15 | 0 | 0 | 0 |  |
| 39 | 0 | R1..L1 | APPLY | SOV|HOV3 | 0 | 0 | 0 | 0 | 16:00 | 19:00 | 0 | 0 | 26 | 0 | 0 | 0 |  |
| 39 | 0 | R1..L1 | APPLY | SOV|HOV3 | 0 | 0 | 0 | 0 | 19:00 | 24:00 | 0 | 0 | 10 | 0 | 0 | 0 |  |

# 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 fromLinkDelay**,** NewFormat**,** PathSkim**,** PlanSum**, and** Router

If the input LINK\_DELAY\_FILE key is provided to the router, the program uses the information in the file to initialize the link flows and travel times for each time period. The header record in 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 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. 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).

The definition file is listed below:

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 12 lists the field definitions for the LINK\_DELAY\_FILE.

Table 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 13 is an example of a LINK\_DELAY\_FILE populated with data:

Table 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). Figure 11 illustrates a sample file, along with a sketch of the network. There are two groups: Group 1 represents eastbound flow, while Group 2 represents westbound flow.

Three link, 2 node network.  Link numbers, going left to right are 2, 3 and 4. 

Figure 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

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 listed 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

Table 14 lists the field definitions for the LINK\_FILE:

Table 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 12 illustrates a small example of using links. 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).

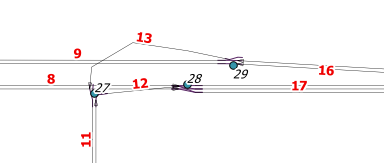


Figure Example of Links

Table 15 lists the LINK\_FILE fields used in this example.

Table LINK\_FILE Fields 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 15 (continued) LINK\_FILE Fields 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

The LOCATION\_ 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 file is listed below:

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

C:\Users\scott.smith\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\X5CU5R0E\MC900329243[1].wmf

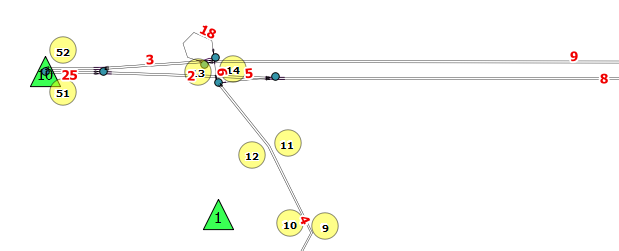
North points up

Figure Example of Locations

Table 16 lists the LOCATION\_FILE fields used in this example.

Table LOCATION\_FILE Fields 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

The MERGE\_LINK\_DELAY\_FILE is an input to LinkDelay. It is the name of the second link delay file when two link delay files are being combined. This file name replaces PREVIOUS\_LINK\_DELAY\_FILE in Version 4 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 shown below:

-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

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

# NEW\_ARC\_...FILE

Names:

NEW\_ARC\_ACCESS\_FILE

NEW\_ARC\_ACCESSIBILITY\_FILE

NEW\_ARC\_ BANDWIDTH\_FILE

NEW\_ARC\_ CENTERLINE \_FILE

NEW\_ARC\_ CONNECTION \_FILE

NEW\_ARC\_ DETECTOR \_FILE

NEW\_ARC\_ DISTANCE\_CONTOUR \_FILE

NEW\_ARC\_ LANE\_USE \_FILE

NEW\_ARC\_ LINK \_FILE

NEW\_ARC\_ LOCATION \_FILE

NEW\_ARC\_NODE \_FILE

NEW\_ARC\_ PARKING**\_**DEMAND \_FILE

NEW\_ARC\_ PARKING \_FILE

NEW\_ARC\_ PHASING\_PLAN \_FILE

NEW\_ARC\_POCKET \_FILE

NEW\_ARC\_ RIDERSHIP \_FILE

NEW\_ARC\_ ROUTE\_NODES \_FILE

NEW\_ARC\_ SIGN\_FILE

NEW\_ARC\_ SIGNAL \_FILE

NEW\_ARC\_ SNAPSHOT \_FILE

NEW\_ARC\_ STOP\_DEMAND \_FILE

NEW\_ARC\_ STOP\_GROUP \_FILE

NEW\_ARC\_ SUBZONE\_DATA \_FILE

NEW\_ARC\_ TIME\_CONTOUR \_FILE

NEW\_ARC\_ TIMING\_PLAN \_FILE

NEW\_ARC\_ TRANSIT\_DRIVER \_FILE

NEW\_ARC\_ TRANSIT\_ROUTE \_FILE

NEW\_ARC\_ TRANSIT\_STOP \_FILE

NEW\_ARC\_ TURN\_PENALTY \_FILE

NEW\_ARC\_ ZONE \_FILE

These are shape files produced by ArcNet, ArcPlan and ArcSnapshot. The file name in the control file 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 a Geographic Information System such as ArcView, ArcMap or QGIS to read and display the object.

# 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 of this output file is listed 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

LinkDelay

LinkSum

LocationData

Microsimulator

NetPrep

NewFormat

PathSkim

PlanSelect

PlanSum

ProblemSelect

Router

TransimsNet

Network nodes are listed below:

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.

Table 17 is an example of a NODE\_FILE populated with data:

Table 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

The OCCUPANCY\_FILE is an input to ArcSnapshot, and the NEW\_OCCUPANCY\_FILE is an output from the Simulator. It lists the link direction, offset, lane, and number of seconds that the cell is occupied by time increment (e.g., 15 minutes).

A typical field definition (.def or .DEF) file is listed below:

TRANSIMS50, TAB\_DELIMITED, 1

LINK, INTEGER, 1, 10

DIR, INTEGER, 2, 1

START, TIME, 3, 16, HOUR\_CLOCK

END, TIME, 4, 16, HOUR\_CLOCK

LANE, STRING, 5, 4, LANE\_ID\_TYPE

OFFSET, DOUBLE, 6, 8.1, METERS

OCCUPANCY, INTEGER, 7, 10

Table 18 lists the field definitions for the OCCUPANCY\_FILE.

Table OCCUPANCY\_FILE Field Definitions

| **Field(s)** | **Description** | **Data Type** |
| --- | --- | --- |
| LINK | Link identifier | Integer |
| DIR | Direction on the link | Integer |
| START | Start time | Hour\_Clock |
| END | End time | Hour\_Clock |
| LANE | Lane number on which the cell is located | String |
| OFFSET | Offset in meters of the cell along the link | Decimal |
| OCCUPANCY | Number of seconds vehicles occupy the cell | Integer |

shows a partial example of an OCCUPANCY\_FILE. It was produced by the Simulator using the following parameters:

NEW\_OCCUPANCY\_FILE\_1 results/Occupancy.txt

NEW\_OCCUPANCY\_INCREMENT\_1 900 seconds

NEW\_OCCUPANCY\_TIME\_RANGE\_1 17:00..17:30

Table OCCUPANCY\_FILE Example

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| LINK | DIR | START | END | LANE | OFFSET | OCCUPANCY |
| 4206 | 0 | 17:00 | 17:15 | L1 | 810.0 | 8 |
| 4206 | 0 | 17:00 | 17:15 | L1 | 817.5 | 12 |
| 4206 | 0 | 17:00 | 17:15 | L1 | 825.0 | 12 |
| 4206 | 0 | 17:00 | 17:15 | 2 | 7.5 | 7 |
| 4206 | 0 | 17:00 | 17:15 | 2 | 15.0 | 7 |
| 4206 | 0 | 17:00 | 17:15 | 2 | 22.5 | 7 |
| 4213 | 0 | 17:01 | 17:01:20 | 1 | 315 | 11 |
| 4213 | 0 | 17:01 | 17:01:20 | 1 | 321.9 | 20 |

In Version 5, the effect of setting NEW\_OCCUPANCY\_MAX\_FLAG to TRUE is unclear.

# 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 this output file is listed below:

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

Microsimulator

NewFormat

PathSkim

Router

TransimsNet

The PARKING\_FILE contains parking lot information. It is a nested file that optionally includes cost, delay and the hours that a parking lot is open.

An example of a typical definition file is listed below:

TRANSIMS50, TAB\_DELIMITED, 2, NESTED

PARKING, INTEGER, 1, 10

LINK, INTEGER, 2, 10

DIR, INTEGER, 3, 1

OFFSET, DOUBLE, 4, 8.1, METERS

TYPE, STRING, 5, 10, PARKING\_TYPE

NUM\_NEST, INTEGER, 6, 2, NEST\_COUNT

NOTES, STRING, 7, 128

USE, STRING, 1, 128, USE\_TYPE, NESTED

START, TIME, 2, 16, HOUR\_CLOCK, NESTED

END, TIME, 3, 16, HOUR\_CLOCK, NESTED

SPACE, UNSIGNED, 4, 5, NO, NESTED

TIME\_IN, TIME, 5, 12, SECONDS, NESTED

TIME\_OUT, TIME, 6, 12, SECONDS, NESTED

HOURLY, UNSIGNED, 7, 5, CENTS, NESTED

DAILY, UNSIGNED, 8, 5, CENTS, NESTED

The location of the parking facility is given by the link, dir (0, 1 direction on the link) and the offset (distance along the link)

Possible values for the TYPE of parking facility include STREET, LOT, BOUNDARY, and PARKRIDE. Aliases for STREET include PRSTR, HISTR. Aliases for LOT include DRVWY. An alias for BOUNDARY is BNDRY and for PARKRIDE is PARKR.

The NUM\_NEST is the number of records in the nest. For a PARKING\_FILE generated by TransimsNet, this number is 0; there are no nested records. The Notes are open for user annotations, by default they contain a longer version of the type of parking facility.

Table 16 is a simple example of a PARKING\_FILE that could be generated by TransimsNet.

Table PARKING\_FILE - Simple Example

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| PARKING | LINK | DIR | OFFSET | TYPE | NUM\_NEST | NOTES |
| USE | START | END | SPACE | TIME\_IN | TIME\_OUT | HOURLY | DAILY |
| 1 | 1 | 0 | 558.2 | BOUNDARY | 0 | External Station |
| 2 | 1 | 1 | 98.0 | BOUNDARY | 0 | External Station |
| … | … | … | … | … | … | … |
| 44 | 19 | 1 | 1640.4 | LOT | 0 | Parking Lot |
| 45 | 19 | 0 | 2460.6 | LOT | 0 | Parking Lot |
| 46 | 19 | 1 | 820.2 | LOT | 0 | Parking Lot |
| 47 | 20 | 0 | 820.2 | LOT | 0 | Parking Lot |
| 48 | 20 | 1 | 2460.6 | LOT | 0 | Parking Lot |
| 49 | 20 | 0 | 1640.4 | LOT | 0 | Parking Lot |

Table 21 is an example of a PARKING\_FILE with nested records. The nest includes number of spaces, time (in seconds) to enter or exit the facility, and the hourly and daily costs in cents.

Table PARKING\_FILE Example with Nested Records

| PARKING | LINK | DIR | OFFSET | TYPE | NUM\_NEST | NOTES |
| --- | --- | --- | --- | --- | --- | --- |
| USE | START | END | SPACE | TIME\_IN | TIME\_OUT | HOURLY | DAILY |
| 1 | 1 | 0 | 558.2 | BOUNDARY | 0 | External Station |
| 2 | 1 | 1 | 98.0 | BOUNDARY | 0 | External Station |
| … | … | … | … | … | … | … |
| 44 | 19 | 1 | 1640.4 | LOT | 0 | Parking Lot |
| 45 | 19 | 0 | 2460.6 | LOT | 1 | Parking Lot |
| ANY | 0:00 | 27:00 | 100 | 20 | 40 | 50 | 200 |
| 46 | 19 | 1 | 820.2 | LOT | 3 | Parking Lot |
| AUTO | 0:00 | 7:00 | 100 | 10 | 20 | 50 | 100 |
| AUTO | 7:00 | 17:00 | 100 | 20 | 40 | 100 | 400 |
| AUTO | 17:00 | 27:00 | 100 | 10 | 30 | 50 | 200 |
| 47 | 20 | 0 | 820.2 | LOT | 1 | Parking Lot |
| AUTO | 0:00 | 27:00 | 50 | 20 | 20 | 50 | 300 |
| 48 | 20 | 1 | 2460.6 | LOT | 0 | Parking Lot |
| 49 | 20 | 0 | 1640.4 | LOT | 0 | Parking Lot |

# PARKING\_PENALTY\_FILE

**PARKING\_PENALTY\_FILE**

Microsimulator

PathSkim

Router

The optional PARKING\_PENALTY\_FILE penalizes parking at certain facilities. It is a file with two columns: the parking lot ID, and the penalty.

# PERFORMANCE\_DATA\_FILE

**NEW\_PERFORMANCE\_DATA\_FILE**

LinkSum

# PERFORMANCE\_FILE

ThePERFORMANCE\_FILE is an input to ArcPlan, LinkSum, and NewFormat

TheNEW\_PERFORMANCE\_FILEis an output from NewFormat and the Simulator. The simulator may output multiple performance files, indexed by number (e.g., NEW\_PERFORMANCE\_FILE\_1).

The PERFORMANCE\_FILE is similar to the LINK\_DELAY\_FILE in that it reports link-based performance. Note that the first 6 fields in the PERFORMANCE\_FILE are identical to those in the LINK\_DELAY\_FILE. This means that if a program such as LinkSum is looking for a performance file, it will accept a link delay file in its place.

A significant difference between the version 4 and version 5 LINK\_DELAY\_FILE / PEFORMANCE\_FILE is that the Version 4 file had volume, an integer number of vehicles entering or exiting the link during a time period. The Version 5 file has flow, which is not necessarily integer (for example, a vehicle traversing half of the link would add 0.5 to the flow). The flow is defined as the distance traveled / link length, where travel units could be vehicles, persons, or passenger car equivalents.

The difference is that the link\_based performance that comes from the Router is placed in a LINK\_DELAY\_FILE, while the link\_based performance that comes from the Simulator is placed in a PERFORMANCE\_FILE. The PERFORMANCE\_FILE includes delay, density and queuing information. It is typically a nested file that also includes turning movement information.

A typical field definition (.def or .DEF) file is listed below:

TRANSIMS50, TAB\_DELIMITED, 2, NESTED

LINK, INTEGER, 1, 10

DIR, INTEGER, 2, 1

START, TIME, 3, 16, HOUR\_CLOCK

END, TIME, 4, 16, HOUR\_CLOCK

FLOW, DOUBLE, 5, 9.2, VEHICLES

TIME, TIME, 6, 8.1, SECONDS

NCONNECT, INTEGER, 7, 2, NEST\_COUNT

AVG\_SPEED, DOUBLE, 8, 8.1, KPH

AVG\_DELAY, TIME, 9, 8.1, SECONDS

AVG\_DENSITY, DOUBLE, 10, 8.1, FLOW/LANE-KM

MAX\_DENSITY, DOUBLE, 11, 8.1, FLOW/LANE-KM

TIME\_RATIO, DOUBLE, 12, 8.3, RATIO

AVG\_QUEUE, DOUBLE, 13, 8.1, STOPPED\_FLOW

MAX\_QUEUE, INTEGER, 14, 10, STOPPED\_FLOW

NUM\_FAIL, INTEGER, 15, 10, VEHICLES

OUT\_LINK, INTEGER, 1, 10, NO, NESTED

OUT\_FLOW, DOUBLE, 2, 9.2, VEHICLES, NESTED

OUT\_TIME, TIME, 3, 8.1, SECONDS, NESTED

lists the field definitions for the PERFORMANCE\_FILE.

Table PERFORMANCE\_FILE Field Definitions

| **Field(s)** | **Description** | **Data Type** |
| --- | --- | --- |
| LINK | Link Number | Integer |
| DIR | Direction (0 = A->B, 1 = B->A) | Integer |
| 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. Default is VEHICLES | Decimal |
| TIME | Travel time: link length x VHT / VMT. Default units are seconds | Decimal |
| NCONNECT | The number of nested records | Integer |
| AVG\_SPEED | Average speed of the vehicles using the link during the time increment. Default units are meters/second | Decimal |
| AVG\_DELAY | Average travel time minus free flow travel time. Default units are seconds. Note that in uncongested conditions, simulated speeds are often faster than free-flow, leading to negative values for AVG\_DELAY | Decimal |
| AVG\_DENSITY | The average number of vehicles occupying the link during each second of the time increment divided by the number of lane meters (i.e., vehicles / (length \* lanes)) | Decimal |
| MAX\_DENSITY | The maximum number of vehicles that occupied the link during the time increment divided by the number of lane meters (i.e., max vehicles / (length \* lanes)) | Decimal |
| TIME\_RATIO | Average travel time during the time increment divided by the free flow time | Decimal |
| AVG\_QUEUE | Average number of stopped vehicles (Sum of the number of seconds each vehicle is stopped on the link during the time increment divided by the length of the time increment) | Decimal |
| MAX\_QUEUE | The maximum number of stopped vehicles on  the link during the time increment | Integer |
| NUM\_FAIL | Number of vehicles experiencing a cycle failure, i.e., number of vehicles that occupied the link when the signal phase turned green and were still on the link when the signal phase turned red | Integer |
| OUT\_LINK | Link ID of the link leaving the end of LINK | Integer |
| OUT\_FLOW | Number of vehicles turning onto OUT\_LINK | Decimal |
| OUT\_TIME | The average travel time on the link for vehicles  making the turning movement (i.e., link length /  (turn meters / turn seconds)) Default units are SECONDS | Decimal |

Table 19 is an example of a PERFORMANCE\_FILE populated with data:

Table PERFORMANCE\_FILE Example

| LINK | DIR | START | END | FLOW | TIME | NCONNECT | AVG\_SPEED | AVG\_DELAY | AVG\_DENSITY | MAX\_DENSITY | TIME\_RATIO | AVG\_QUEUE | MAX\_QUEUE | NUM\_FAIL |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| OUT\_LINK | OUT\_FLOW | OUT\_TIME |
| 4206 | 0 | 15:00 | 15:15 | 23.4 | 39.4 | 2 | 75.3 | -11.9 | 0.6 | 2.4 | 0.769 | 0 | 2 | 0 |
| 11465 | 6.86 | 37.4 |
| 4209 | 16.14 | 40.5 |
| 4206 | 1 | 15:00 | 15:15 | 37.55 | 38.6 | 1 | 76.9 | -12.7 | 1 | 3.6 | 0.753 | 0 | 0 | 0 |
| 4207 | 34.94 | 38.3 |
| 4207 | 0 | 15:00 | 15:15 | 35.06 | 50 | 0 | 77.6 | -17.1 | 0.9 | 2.8 | 0.745 | 0 | 0 | 0 |
| 4207 | 1 | 15:00 | 15:15 | 23.29 | 49.8 | 1 | 77.9 | -17.3 | 0.6 | 1.9 | 0.742 | 0 | 0 | 0 |
| 4206 | 23.02 | 49.8 |
| 4209 | 0 | 15:00 | 15:15 | 29.37 | 7.6 | 2 | 70.6 | -1.8 | 0.9 | 13.5 | 0.806 | 0 | 1 | 0 |
| 4213 | 20.39 | 7.3 |
| 4212 | 5.05 | 8.5 |
| 4209 | 1 | 15:00 | 15:15 | 34.32 | 9.8 | 2 | 54.8 | 0.4 | 1.3 | 10.1 | 1.039 | 0 | 1 | 0 |
| 4206 | 18.17 | 10.5 |
| 11465 | 14.79 | 9.2 |

# PERSON\_FILE

**PERSON\_FILE**

NewFormat

This is the name of the Version 4 person file that is optionally used by 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.

Table 24 is an example of a PHASING\_PLAN\_FILE populated with data:

Table 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?” Figure 14 illustrates this hierarchical relationship:

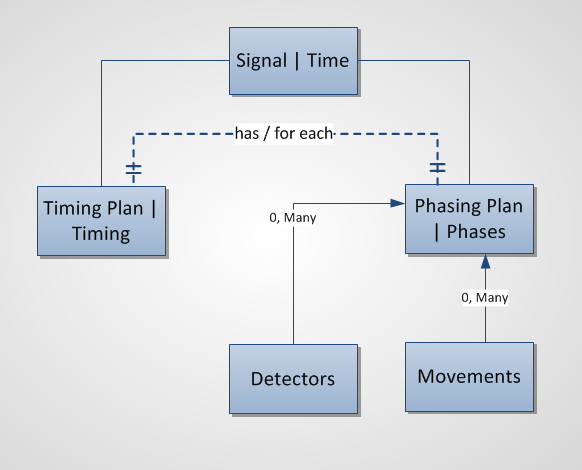


Figure Hierarchical Relationships Among the Signalized Intersection Files

The definition file is listed below:

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

Table 25 lists the field definitions for the PHASING\_PLAN\_FILE.

Table 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\_FILEis an output file, used inNewFormat**,** PathSkim**,** PlanCompare**,** PlanPrep**,** PlanSelect, and Router.

PLAN\_FILE is an input file, used in ArcPlan, Simulator**,** NewFormat**,** PlanCompare**,** PlanPrep**,** PlanSelect**,** PlanSumand 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

The definition file is listed below:

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

Table 26 lists the field definitions for the PLAN\_FILE:

Table 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 |
| 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 27) 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 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 28) 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 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 the list of pocket lanes in the network.

A typical field definition (.def or .DEF) file is listed below:

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. Table 29 lists the field definitions for the POCKET\_FILE.

Table 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 15 illustrates a small example of using pocket lanes.

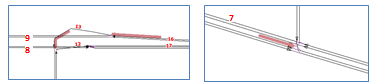


Figure Pocket Lanes

Table 30 lists the POCKET\_FILE fields used in this example

Table 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 25. 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 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. |
| 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. |
| 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 |  |
| 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. This 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 32). This 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 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

The NEW\_RIDERSHIP\_FILE is an output from NewFormat and the Simulator. The RIDERSHIP\_FILE is an input to NewFormat and Validate. The RIDERSHIP\_FILE summarizes the boardings and alightings at each stop on each route based on the scheduled and actual departure time for each run.

A typical field definition (.def or .DEF) file is listed below:

TRANSIMS50, TAB\_DELIMITED, 1

MODE, STRING, 1, 16, TRANSIT\_TYPE

ROUTE, INTEGER, 2, 10

RUN, INTEGER, 3, 5

STOP, INTEGER, 4, 10

SCHEDULE, TIME, 5, 16, HOUR\_CLOCK

TIME, TIME, 6, 16, HOUR\_CLOCK

BOARD, UNSIGNED, 7, 5

ALIGHT, UNSIGNED, 8, 5

LOAD, UNSIGNED, 9, 5

FACTOR, DOUBLE, 10, 5.2, RATIO

Table 33 lists the field definitions for the RIDERSHIP\_FILE.

Table RIDERSHIP\_FILE Field Definitions

| **Field(s)** | **Description** | **Data Type** |
| --- | --- | --- |
| MODE | An optional mode string indicating the type of transit, e.g., BUS | String |
| ROUTE | Route number (a required field) | Integer |
| RUN | Run (or trip) number (a required field) | Integer |
| STOP | Stop number (a required field) | Integer |
| SCHEDULE | Scheduled departure time | Hour\_Clock |
| TIME | Actual departure time | Hour\_Clock |
| BOARD | Number of persons boarding at this stop (a required field) | Integer |
| ALIGHT | Number of persons alighting at this stop (a required field) | Integer |
| LOAD | Number of persons on-board the vehicle as it leaves this stop (an optional field) | Integer |
| FACTOR | Ratio of LOAD to vehicle capacity (Load Factor) (an optional field) | Decimal |

is an example of a RIDERSHIP\_FILE populated with data: [There appears to be a bug in the factor calculation coming out of NewFormat. A bus should have a capacity of 52, but this calculation is behaving as though it is 5.2]

Table RIDERSHIP\_FILE Example

| MODE | ROUTE | RUN | STOP | SCHEDULE | TIME | BOARD | ALIGHT | LOAD | FACTOR |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| BUS | 100 | 3 | 2 | 0:20:12 | 0:20:15 | 2 | 0 | 2 | 0.40 |
| BUS | 100 | 3 | 3 | 0:20:23 | 0:20:23 | 0 | 0 | 2 | 0.40 |
| BUS | 100 | 3 | 4 | 0:20:34 | 0:20:38 | 2 | 0 | 4 | 0.80 |
| BUS | 100 | 3 | 5 | 0:20:45 | 0:20:53 | 2 | 0 | 6 | 1.20 |
| BUS | 100 | 3 | 6 | 0:20:56 | 0:21:09 | 2 | 0 | 8 | 1.50 |
| BUS | 100 | 3 | 7 | 0:21:11 | 0:22 | 1 | 0 | 9 | 1.70 |
| BUS | 100 | 3 | 8 | 0:21:22 | 0:22:08 | 0 | 0 | 9 | 1.70 |
| BUS | 100 | 3 | 9 | 0:21:33 | 0:22:23 | 2 | 0 | 11 | 2.10 |
| BUS | 100 | 3 | 10 | 0:21:44 | 0:22:38 | 2 | 0 | 13 | 2.50 |
| BUS | 100 | 3 | 11 | 0:21:55 | 0:22:51 | 1 | 0 | 14 | 2.70 |
| BUS | 100 | 3 | 12 | 0:22:06 | 0:23:09 | 3 | 0 | 17 | 3.30 |
| BUS | 100 | 3 | 13 | 0:22:20 | 0:23:40 | 3 | 1 | 19 | 3.70 |
| BUS | 100 | 3 | 14 | 0:22:31 | 0:23:55 | 2 | 0 | 21 | 4.00 |
| BUS | 100 | 3 | 15 | 0:22:42 | 0:24:08 | 1 | 0 | 22 | 4.20 |
| BUS | 100 | 3 | 16 | 0:22:53 | 0:24:30 | 2 | 3 | 21 | 4.00 |
| BUS | 100 | 3 | 17 | 0:23:04 | 0:24:38 | 0 | 0 | 21 | 4.00 |
| BUS | 100 | 3 | 18 | 0:23:15 | 0:24:55 | 0 | 2 | 19 | 3.70 |
| BUS | 100 | 3 | 19 | 0:23:29 | 0:25:25 | 1 | 7 | 13 | 2.50 |
| BUS | 100 | 3 | 20 | 0:23:51 | 0:25:44 | 0 | 1 | 12 | 2.30 |
| BUS | 100 | 3 | 21 | 0:24:13 | 0:26:06 | 2 | 0 | 14 | 2.70 |
| BUS | 100 | 3 | 22 | 0:24:38 | 0:26:40 | 1 | 6 | 9 | 1.70 |

# 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\_NODES \_ILE 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 35). It indicates the household, person, tour, trip and, for parallel processing applications, the partition of the router that will be used.

Table 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 or .DEF) file is listed below:

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

Table 36 is an example of a SHAPE\_FILE populated with data. 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 SHAPE\_FILE Example

| 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 |

Table 37 and Table 38 provide information on the links and nodes corresponding to the above SHAPE\_FILE.

Table 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 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. Figure 16 illustrates the end result of part of a cloverleaf freeway interchange:

links, nodes and shape points in a cloverleaf interchange

Figure 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, it 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 the PHASING\_PLAN \_FILE and TIMING\_PLAN\_FILE.

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\_FILE (or SIGN\_FILE) 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 \_FILE, TIMING\_PLAN\_FILE, and DETECTOR\_FILE).

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 TRANSIMS. 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. A typical field definition (.def or .DEF) file for the Version 5 SIGNAL\_FILE is listed 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

Table 39 lists the field definitions for the SIGNAL\_FILE:

Table 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.

The entity-relationship diagram in Figure 17 illustrates the rather complex nested file structure that interconnects the Signal, Phasing Plan, and Timing Plan file records. TRANSIMS 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:

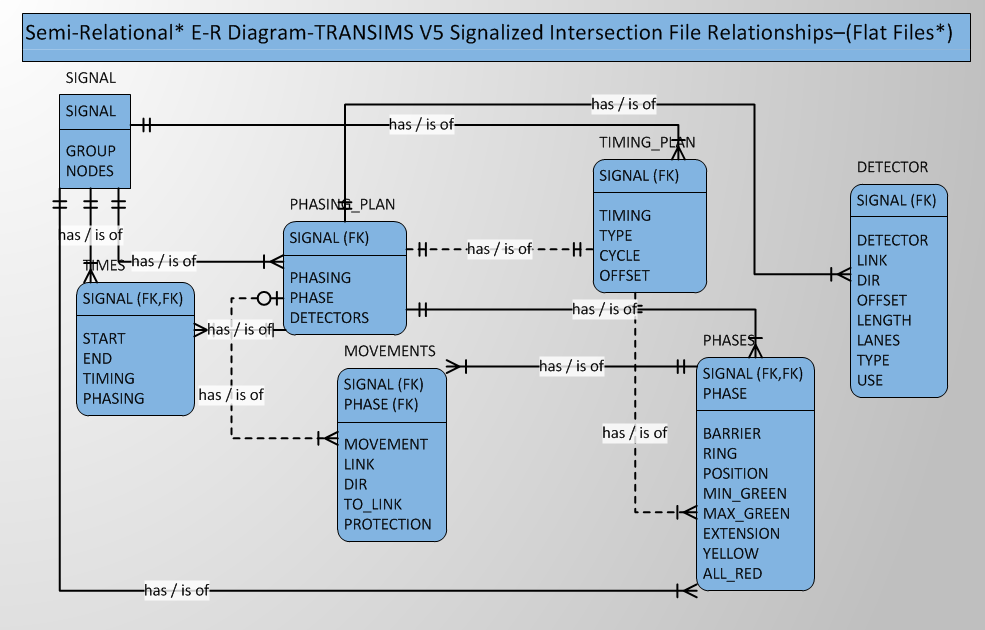


Figure Entity-Relationship Diagram - Signal File Dependencies

Table 40 is an example of a SIGNAL\_FILE populated with data:

Table 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)

Definitions for the SKIM\_FILE columns are listed below:

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]](#footnote-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 field definition (.def or .DEF) file is listed below:

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

IMPEDANCE, INTEGER, 10, 10, IMPEDANCE

Table 41 is an example of a SKIM\_FILE (tab-delimited) populated with data. 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 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

NewFormat

The SNAPSHOT\_FILE, output by the simulator, lists the link direction, offset, lane, and speed of each vehicle at specified time points (e.g., every 5 minutes).

A typical field definition (.def or .DEF) file is listed below:

TRANSIMS50, TAB\_DELIMITED, 1

HHOLD, INTEGER, 1, 10

VEHICLE, INTEGER, 2, 4

TIME, TIME, 3, 16, HOUR\_CLOCK

LINK, INTEGER, 4, 10

DIR, INTEGER, 5, 1

LANE, STRING, 6, 4, LANE\_ID\_TYPE

OFFSET, DOUBLE, 7, 8.1, METERS

SPEED, DOUBLE, 8, 8.1, KPH

PASSENGERS, INTEGER, 9, 4

TYPE, INTEGER, 10, 4, VEHICLE\_TYPE

lists the field definitions for the SNAPSHOT\_FILE.

Table SNAPSHOT\_FILE Field Definitions

| **Field(s)** | **Description** | **Data Type** |
| --- | --- | --- |
| HHOLD | Household identifier | Integer |
| VEHICLE | Vehicle identifier | Integer |
| TIME | Time of day | Hour\_Clock |
| LINK | Link identifier | Integer |
| DIR | Direction on the link | Integer |
| LANE | Lane identifier (LANE\_ID\_TYPE) | String |
| OFFSET | Offset along the link (default units are meters) | Decimal |
| SPEED | Speed of the vehicle (default units are km / hr) | Decimal |
| PASSENGERS | Passengers in the vehicle | Integer |
| TYPE | Vehicle type code | Integer |

shows an excerpt from a SNAPSHOT\_FILE of vehicles over 15 minutes, in a 3-cell left turn lane.

Table SNAPSHOT\_FILE Example

| HHOLD | VEHICLE | TIME | LINK | DIR | LANE | OFFSET | SPEED | PASSENGERS | TYPE |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1889808 | 9 | 17:00:08 | 4206 | 0 | L1 | 810 | 81 | 0 | 11 |
| 1889808 | 9 | 17:00:09 | 4206 | 0 | L1 | 817.5 | 0 | 0 | 11 |
| 1889808 | 9 | 17:00:10 | 4206 | 0 | L1 | 825 | 27 | 0 | 11 |
| 1892449 | 10 | 17:00:48 | 4206 | 0 | L1 | 810 | 81 | 0 | 11 |
| 564315 | 6 | 17:02:03 | 4206 | 0 | L1 | 810 | 81 | 0 | 1 |
| 699721 | 2 | 17:02:36 | 4206 | 0 | L1 | 810 | 81 | 0 | 1 |
| 563810 | 1 | 17:02:47 | 4206 | 0 | L1 | 810 | 81 | 0 | 1 |
| 568164 | 5 | 17:02:51 | 4206 | 0 | L1 | 810 | 81 | 0 | 1 |
| 831495 | 6 | 17:03:01 | 4206 | 0 | L1 | 810 | 81 | 0 | 2 |
| 1889620 | 1 | 17:03:58 | 4206 | 0 | L1 | 810 | 81 | 0 | 11 |
| 1889620 | 1 | 17:03:59 | 4206 | 0 | L1 | 817.5 | 0 | 0 | 11 |
| 1889620 | 1 | 17:04 | 4206 | 0 | L1 | 825 | 27 | 0 | 11 |
| 570781 | 2 | 17:04:13 | 4206 | 0 | L1 | 810 | 81 | 0 | 1 |
| 1889438 | 9 | 17:05:29 | 4206 | 0 | L1 | 810 | 81 | 0 | 11 |
| 1889438 | 9 | 17:05:30 | 4206 | 0 | L1 | 817.5 | 0 | 0 | 11 |
| 1889438 | 9 | 17:05:31 | 4206 | 0 | L1 | 825 | 27 | 0 | 11 |
| 467149 | 10 | 17:05:55 | 4206 | 0 | L1 | 810 | 81 | 0 | 1 |
| 467149 | 10 | 17:05:56 | 4206 | 0 | L1 | 817.5 | 0 | 0 | 1 |
| 467149 | 10 | 17:05:57 | 4206 | 0 | L1 | 825 | 27 | 0 | 1 |
| 564932 | 3 | 17:07:40 | 4206 | 0 | L1 | 810 | 81 | 0 | 1 |
| 1853617 | 8 | 17:07:47 | 4206 | 0 | L1 | 810 | 81 | 0 | 11 |
| 570784 | 5 | 17:08:25 | 4206 | 0 | L1 | 810 | 81 | 0 | 1 |
| 1892301 | 2 | 17:09 | 4206 | 0 | L1 | 810 | 81 | 0 | 11 |
| 1892301 | 2 | 17:09:01 | 4206 | 0 | L1 | 817.5 | 0 | 0 | 11 |
| 1892301 | 2 | 17:09:02 | 4206 | 0 | L1 | 825 | 27 | 0 | 11 |
| 569431 | 2 | 17:09:13 | 4206 | 0 | L1 | 810 | 81 | 0 | 1 |
| 569442 | 3 | 17:11:12 | 4206 | 0 | L1 | 810 | 81 | 0 | 1 |
| 565004 | 5 | 17:11:37 | 4206 | 0 | L1 | 810 | 81 | 0 | 1 |
| 1889437 | 8 | 17:11:39 | 4206 | 0 | L1 | 810 | 81 | 0 | 11 |
| 1889622 | 3 | 17:12:11 | 4206 | 0 | L1 | 810 | 81 | 0 | 11 |
| 1889622 | 3 | 17:12:12 | 4206 | 0 | L1 | 817.5 | 0 | 0 | 11 |
| 1889622 | 3 | 17:12:13 | 4206 | 0 | L1 | 825 | 27 | 0 | 11 |
| 564805 | 6 | 17:12:51 | 4206 | 0 | L1 | 810 | 81 | 0 | 1 |
| 1891937 | 8 | 17:12:53 | 4206 | 0 | L1 | 810 | 81 | 0 | 11 |
| 563843 | 4 | 17:12:58 | 4206 | 0 | L1 | 810 | 81 | 0 | 1 |
| 563849 | 10 | 17:13:48 | 4206 | 0 | L1 | 810 | 81 | 0 | 1 |
| 563840 | 1 | 17:14:10 | 4206 | 0 | L1 | 810 | 81 | 0 | 1 |

# 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.

# 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.

Table 44 is an example of a TIME\_DISTRIBUTION\_FILE populated with data:

Table 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 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 45 is the first example of a TIMING\_PLAN\_FILE populated with data:

Table TIMING\_PLAN\_FILE First 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 |

The definition file listing the data fields for a TIMING\_PLAN\_FILE is shown below:

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 46 lists the field definitions for the TIMING\_PLAN\_FILE.

Table 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 |

Table 47 is the second example of a 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 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 |
| 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

TOLL\_FILE is the filename for the version 4 toll file, used by NewFormat for toll to version 5 lane use conversion

# TRANSIT\_DRIVER\_FILE

**NEW\_TRANSIT\_DRIVER\_FILE**

NewFormat

**TRANSIT\_DRIVER\_FILE**

ArcNet

ArcPlan

Microsimulator

NewFormat

The TRANSIT\_DRIVER\_FILE is similar to its version 4 counterpart. It is a nested table of transit driver paths. A typical field definition (.def or .DEF) file is listed below:

TRANSIMS50, TAB\_DELIMITED, 2, NESTED

ROUTE, INTEGER, 1, 10

NLINKS, INTEGER, 2, 4, NEST\_COUNT

TYPE, INTEGER, 3, 4, VEHICLE\_TYPE

NOTES, STRING, 4, 128

LINK, INTEGER, 1, 10, NO, NESTED

DIR, INTEGER, 2, 1, NO, NESTED

Table 48 lists the field definitions for the TRANSIT\_DRIVER\_FILE.

Table TRANSIT\_DRIVER\_FILE Field Definitions

| **Field(s)** | **Description** | **Use** | **Values** |
| --- | --- | --- | --- |
| ROUTE | Route ID Number | Key | Integer |
| NLINKS | Number of nested records | Req | Integer |
| TYPE | Vehicle type for the route | Opt | Integer (Vehicle type code) |
| NOTES | User annotations | Opt |  |
| Nested Fields | | | |
| LINK | Link ID number on the driver’s path | Req | Integer |
| DIR | Direction on the link | Opt | Integer |

Table 49 is an example of the TRANSIT\_DRIVER\_FILE.

Table TRANSIT\_DRIVER\_FILE Example

|  |  |  |  |
| --- | --- | --- | --- |
| ROUTE | NLINKS | TYPE | NOTES |
| LINK | DIR |
| 100 | 6 | 7 | Bus Route |
| 20 | 0 |
| 21 | 0 |
| 22 | 0 |
| 28 | 0 |
| 39 | 0 |
| 46 | 0 |
| 101 | 6 | 7 | Bus Route |
| 46 | 1 |
| 39 | 1 |
| 28 | 1 |
| 22 | 1 |
| 21 | 1 |
| 20 | 1 |

# TRANSIT\_FARE\_FILE

**NEW\_TRANSIT\_FARE\_FILE**

NewFormat

**TRANSIT\_FARE\_FILE**

Microsimulator

NewFormat

PathSkim

Router

The TRANSIT\_FARE\_FILE lists zone to zone transit fares.

A typical field definition file (.def or .DEF) is listed below:

TRANSIMS50, TAB\_DELIMITED, 2

0:00 10:00

FROM\_ZONE, STRING, 1, 128, FARE\_ZONE\_RANGE

TO\_ZONE, STRING, 2, 128, FARE\_ZONE\_RANGE

FROM\_MODE, STRING, 3, 128, TRANSIT\_TYPE\_RANGE

TO\_MODE, STRING, 4, 128, TRANSIT\_TYPE\_RANGE

PERIOD, STRING, 5, 128, TIME\_PERIOD\_RANGE

CLASS, STRING, 6, 128, FARE\_CLASS\_RANGE

FARE, INTEGER, 7, 5, CENTS

NOTES, STRING, 8, 128

Table 50 lists the field definitions for the TRANSIT\_FARE\_FILE, which are the same as TRANSIMS 4.

Table TRANSIT\_FARE\_FILE Field Definitions

| **Field(s)** | **Description** | **Use** | **Values** |
| --- | --- | --- | --- |
| FROM\_ZONE | Zone range for the applicable boarding stops | Req | String (FARE\_ZONE\_RANGE) |
| TO\_ZONE | Zone range for the applicable alighting stops | Req | String  (FARE\_ZONE\_RANGE) |
| FROM\_MODE | Transit mode ranges from which the trip came | Opt. | String (TRANSIT\_CODE\_RANGE) |
| TO\_MODE | Transit mode ranges to which the cost applies | Req | String (TRANSIT\_CODE\_RANGE) |
| PERIOD | Start and end time range at the boarding stop (defaults to all) | Opt | Range of time codes |
| CLASS | Traveler or payment classification ranges | Opt | Range of class codes (CASH, CARD, SPECIAL) |
| FARE | Boarding cost in cents | Req | Integer |
| NOTES | User annotations | Opt | String |

Table 51 is an example of a TRANSIT\_FARE\_FILE populated with data.

Table TRANSIT\_FARE\_FILE Example (Taken from Version 4)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| FROM\_ZONE | TO\_ZONE | FROM\_MODE | TO\_MODE | PERIOD | CLASS | FARE | NOTES |
| 9, 10, 11 | 9, 10, 11 | NONE, RAPIDRAIL, REGIONRAIL | BUS, EXPRESS, REGIONRAIL | 0:00..24:00 | CASH, CARD | 135 | Base BUS Fare |
| 9..37 | 9..37 | RAPIDRAIL | EXPRESS | 0:00..24:00 | CASH | 220 | Metro-Rail to Express Bus Transfer |
| 9..37 | 9..37 | RAPIDRAIL | EXPRESS | 0:00..24:00 | CARD | 210 | Metro-Rail to Express Bus Transfer |

Table 52 is an example of a TRANSIT\_FARE\_FILE after NewFormat was run on the previous file (MODEL\_END\_TIME = 10:00)

Table Unbundled TRANSIT\_FARE\_FILE Example (after NewFormat)

|  |
| --- |
| 0:00 10:00 |
| FROM\_ZONE | TO\_ZONE | FROM\_MODE | TO\_MODE | PERIOD | CLASS | FARE | NOTES |
| 9 | 9 | 0 | 1 | 0 | 0 | 135 |  |
| 9 | 9 | 0 | 1 | 0 | 1 | 135 |  |
| 9 | 9 | 0 | 2 | 0 | 0 | 135 |  |
| 9 | 9 | 0 | 2 | 0 | 1 | 135 |  |
| 9 | 9 | 0 | 7 | 0 | 0 | 135 |  |
| 9 | 9 | 0 | 7 | 0 | 1 | 135 |  |
| 9 | 9 | 6 | 1 | 0 | 0 | 135 |  |
| 9 | 9 | 6 | 1 | 0 | 1 | 135 |  |
| 9 | 9 | 6 | 2 | 0 | 0 | 220 |  |
| 9 | 9 | 6 | 2 | 0 | 1 | 210 |  |
| 9 | 9 | 6 | 7 | 0 | 0 | 135 |  |
| 9 | 9 | 6 | 7 | 0 | 1 | 135 |  |
| 9 | 9 | 7 | 1 | 0 | 0 | 135 |  |
| 9 | 9 | 7 | 1 | 0 | 1 | 135 |  |
| 9 | 9 | 7 | 2 | 0 | 0 | 135 |  |
| 9 | 9 | 7 | 2 | 0 | 1 | 135 |  |

# 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

The TRANSIT\_ROUTE\_FILE is similar to its version 4 counterpart. It is a nested table of transit routes.

A typical field definition file (.def or .DEF) is listed below:

TRANSIMS50, TAB\_DELIMITED, 2, NESTED

ROUTE, INTEGER, 1, 10

STOPS, INTEGER, 2, 10, NEST\_COUNT

MODE, STRING, 3, 16, TRANSIT\_TYPE

TYPE, INTEGER, 4, 4, VEHICLE\_TYPE

NAME, STRING, 5, 40

NOTES, STRING, 6, 128

STOP, INTEGER, 1, 10, NO, NESTED

ZONE, INTEGER, 2, 5, FARE\_ZONE, NESTED

TIMEPT, INTEGER, 3, 3, NO, NESTED

Table 53 lists the field definitions for the TRANSIT\_ROUTE\_FILE.

Table TRANSIT\_ROUTE\_FILE Field Definitions

| **Field(s)** | **Description** | **Use** | **Values** |
| --- | --- | --- | --- |
| ROUTE | Route ID Number | Key | Integer |
| STOPS | Number of nested stop records | Req | Integer |
| MODE | Transit Mode for the routes | Req | String (transit mode code, e.g., BUS) |
| TYPE | Vehicle type for the route | Opt | Integer (Vehicle type code) |
| NAME | Name of the transit route | Opt | String |
| NOTES | User annotations | Opt |  |
| Nested Fields | | | |
| STOP | Stop ID number | Req | Integer |
| ZONE | ID of the fare zone | Opt | Integer |
| TIMEPT | Non-zero indicates a timepoint | Opt | Integer |

Transit mode codes include:

NO\_TRANSIT (aliases include NONE, NO\_MODE, N/A or blank)

LOCAL\_BUS (alias includes BUS)

EXPRESS\_BUS (alias includes EXPRESS)

TROLLEY

STREETCAR

LRT (alias includes LIGHTRAIL)

RAPIDRAIL

REGIONRAIL

ANY\_TRANSIT (aliases include ANY\_MODE or ANY)

Table 54 is an example of a TRANSIT\_ROUTE\_FILE populated with data.

Table TRANSIT\_ROUTE\_FILE Example

| ROUTE | STOPS | MODE | TYPE | NAME | NOTES |
| --- | --- | --- | --- | --- | --- |
| STOP | ZONE | TIMEPT |
| 100 | 32 | BUS | 7 | Route 100 | Bus Route |
| 1 | 1 | 0 |
| 2 | 1 | 0 |
| 3 | 1 | 0 |
| 4 | 1 | 0 |
| … | … | … |
| 22 | 2 | 0 |
| … | … | … |
| 32 | 3 | 0 |
| 101 | 32 | BUS | 7 | Route 100 | Bus Route |
| 33 | 2 | 0 |
| … | … | … |
| 37 | 1 | 0 |
| 38 | 1 | 0 |

# TRANSIT\_SCHEDULE\_FILE

**NEW\_TRANSIT\_SCHEDULE\_FILE**

NewFormat

**TRANSIT\_SCHEDULE\_FILE**

ArcNet

LocationData

Microsimulator

NewFormat

PathSkim

Router

The TRANSIT\_SCHEDULE\_FILE is now formatted as a time table. It is a nested table of transit schedules, with up to 8 runs per group.

A typical field definition (.def or .DEF) file is listed below:

TRANSIMS50, TAB\_DELIMITED, 2, NESTED

ROUTE, INTEGER, 1, 10

STOPS, INTEGER, 2, 4, NEST\_COUNT

RUN1, INTEGER, 3, 4

RUN2, INTEGER, 4, 4

RUN3, INTEGER, 5, 4

RUN4, INTEGER, 6, 4

RUN5, INTEGER, 7, 4

RUN6, INTEGER, 8, 4

RUN7, INTEGER, 9, 4

RUN8, INTEGER, 10, 4

NOTES, STRING, 11, 128

STOP, INTEGER, 1, 10, NO, NESTED

TIME1, TIME, 2, 16, HOUR\_CLOCK, NESTED

TIME2, TIME, 3, 16, HOUR\_CLOCK, NESTED

TIME3, TIME, 4, 16, HOUR\_CLOCK, NESTED

TIME4, TIME, 5, 16, HOUR\_CLOCK, NESTED

TIME5, TIME, 6, 16, HOUR\_CLOCK, NESTED

TIME6, TIME, 7, 16, HOUR\_CLOCK, NESTED

TIME7, TIME, 8, 16, HOUR\_CLOCK, NESTED

TIME8, TIME, 9, 16, HOUR\_CLOCK, NESTED

Table 55 lists the field definitions for the TRANSIT\_SCHEDULE\_FILE.

Table TRANSIT\_SCHEDULE\_FILE Field Definitions

| **Field(s)** | **Description** | **Use** | **Values** |
| --- | --- | --- | --- |
| ROUTE | Route ID Number | Req | Integer |
| RUN1 | Number of the first run in this set | Opt | Integer |
| RUN2 | Number of the second run in this set | Opt | Integer |
| RUN3 | Number of the third run in this set | Opt | Integer |
| RUN4 | Number of the fourth run in this set | Opt | Integer |
| RUN5 | Number of the fifth run in this set | Opt | Integer |
| RUN6 | Number of the sixth run in this set | Opt | Integer |
| RUN7 | Number of the seventh run in this set | Opt | Integer |
| RUN8 | Number of the eighth run in this set | Opt | Integer |
| NOTES | User annotations | Opt | String |
| Nested Fields | | | |
| STOP | Stop ID number | Req | Integer |
| TIME1 | Scheduled time at the stop for Run 1 | Opt | Time of Day |
| TIME2 | Scheduled time at the stop for Run 2 | Opt | Time of Day |
| TIME3 | Scheduled time at the stop for Run 3 | Opt | Time of Day |
| TIME4 | Scheduled time at the stop for Run 4 | Opt | Time of Day |
| TIME5 | Scheduled time at the stop for Run 5 | Opt | Time of Day |
| TIME6 | Scheduled time at the stop for Run 6 | Opt | Time of Day |
| TIME7 | Scheduled time at the stop for Run 7 | Opt | Time of Day |
| TIME8 | Scheduled time at the stop for Run 8 | Opt | Time of Day |

Table 56 is an example of a TRANSIT\_SCHEDULE\_FILE populated with data. The route in this example has a headway of 10 minutes.

Table TRANSIT\_SCHEDULE\_FILE Example

| ROUTE | STOPS | RUN1 | RUN2 | RUN3 | RUN4 | RUN5 | RUN6 | RUN7 | RUN8 | NOTES |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| STOP | TIME1 | TIME2 | TIME3 | TIME4 | TIME5 | TIME6 | TIME7 | TIME8 |
| 100 | 32 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |  |
| 1 | 0:00:01 | 0:10:01 | 0:20:01 | 0:30:01 | 0:40:01 | 0:50:01 | 1:00:01 | 1:10:01 |
| 2 | 0:00:12 | 0:10:12 | 0:20:12 | 0:30:12 | 0:40:12 | 0:50:12 | 1:00:12 | 1:10:12 |
| 3 | 0:00:23 | 0:10:23 | 0:20:23 | 0:30:23 | 0:40:23 | 0:50:23 | 1:00:23 | 1:10:23 |
| 4 | 0:00:34 | 0:10:34 | 0:20:34 | 0:30:34 | 0:40:34 | 0:50:34 | 1:00:34 | 1:10:34 |
| … | … | … | … | … | … | … | … | … |
| 22 | 0:04:38 | 0:14:38 | 0:24:38 | 0:34:38 | 0:44:38 | 0:54:38 | 1:04:38 | 1:14:38 |
| … | … | … | … | … | … | … | … | … |
| 32 | 0:06:52 | 0:16:52 | 0:26:52 | 0:36:52 | 0:46:52 | 0:56:52 | 1:06:52 | 1:16:52 |
| 100 | 32 | 9 | 10 | 11 | 12 |  |  |  |  |  |
| 1 | 1:20:01 | 1:30:01 | 1:40:01 | 1:50:01 |  |  |  |  |
| 2 | 1:20:12 | 1:30:12 | 1:40:12 | 1:50:12 |  |  |  |  |
| 3 | 1:20:23 | 1:30:23 | 1:40:23 | 1:50:23 |  |  |  |  |
| 4 | 1:20:34 | 1:30:34 | 1:40:34 | 1:50:34 |  |  |  |  |
| … | … | … | … | … |  |  |  |  |

# TRANSIT\_STOP\_FILE

**NEW\_TRANSIT\_STOP\_FILE**

NewFormat

**TRANSIT\_STOP\_FILE**

ArcNet

ArcPlan

LocationData

Microsimulator

NewFormat

PathSkim

Router

The TRANSIT\_STOP\_FILE lists transit stops and is similar to its version 4 counterpart.

A typical field definition (.def or .DEF) file is listed below:

TRANSIMS50, TAB\_DELIMITED, 1

STOP, INTEGER, 1, 10

NAME, STRING, 2, 50

LINK, INTEGER, 3, 10

DIR, INTEGER, 4, 1

OFFSET, DOUBLE, 5, 8.1, FEET

USE, STRING, 6, 128, USE\_TYPE

TYPE, STRING, 7, 16, STOP\_TYPE

SPACE, UNSIGNED, 8, 3

NOTES, STRING, 9, 128

Table 57 lists the field definitions for the TRANSIT\_STOP\_FILE.

Table TRANSIT\_STOP\_FILE Field Definitions

| **Field(s)** | **Description** | **Use** | **Values** |
| --- | --- | --- | --- |
| STOP | Transit Stop ID number | Key | Integer |
| NAME | Name of the stop | Opt | String |
| LINK | Link ID where the stop is located | Req | Integer |
| DIR | Direction on the link | Opt | Integer |
| OFFSET | Location of the transit stop along the link | Req | Decimal distance (default is METERS) |
| USE | Vehicle types that may stop at the transit stop (USE\_CODE) | Opt | String: Vehicle types that may stop (default of ANY) |
| TYPE | Type of transit stop (STOP\_CODE) | Opt | String: STOP or STATION |
| SPACE | Number of vehicles that the stop can accommodate (0 for unlimited) | Opt | Integer |
| NOTES | User annotations | Opt | String |

Table 58 is an example of a TRANSIT\_STOP\_FILE populated with data.

Table TRANSIT\_STOP\_FILE Example

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| STOP | NAME | LINK | DIR | OFFSET | USE | TYPE | SPACE | NOTES |
| 1 | Stop Name | 20 | 0 | 32.8 | BUS | STOP | 2 | Transit Stop |
| 2 |  | 20 | 0 | 560.3 | BUS | STOP | 2 | Transit Stop |
| 3 |  | 20 | 0 | 1087.9 | BUS | STOP | 2 | Transit Stop |
| 4 |  | 20 | 0 | 1615.4 | BUS | STOP | 2 | Transit Stop |
| 5 |  | 20 | 0 | 2143.0 | BUS | STOP | 2 | Transit Stop |
| 6 |  | 20 | 0 | 2670.6 | BUS | STOP | 2 | Transit Stop |
| 7 |  | 21 | 0 | 213.3 | BUS | STOP | 2 | Transit Stop |
| 8 |  | 21 | 0 | 732.6 | BUS | STOP | 2 | Transit Stop |
| 9 |  | 21 | 0 | 1252.0 | BUS | STOP | 2 | Transit Stop |
| 10 |  | 21 | 0 | 1771.4 | BUS | STOP | 2 | Transit Stop |
| 11 |  | 21 | 0 | 2290.7 | BUS | STOP | 2 | Transit Stop |
| 12 |  | 21 | 0 | 2810.1 | BUS | STOP | 2 | Transit Stop |

# TRAVELER\_FILE

The NEW\_TRAVELER\_FILE is an output from the Simulator. It lists the link direction, offset, lane, and speed for each selected traveler by time step (e.g., second). A typical field definition (.def) file is listed below:

TRANSIMS50, TAB\_DELIMITED, 1

HHOLD, INTEGER, 1, 10

PERSON, INTEGER, 2, 5

TOUR, INTEGER, 3, 3

TRIP, INTEGER, 4, 3

MODE, STRING, 5, 12, MODE\_TYPE

TIME, TIME, 6, 16, HOUR\_CLOCK

DISTANCE, DOUBLE, 7, 8.1, METERS

SPEED, DOUBLE, 8, 5.1, KPH

LINK, INTEGER, 9, 10

DIR, INTEGER, 10, 1

LANE, STRING, 11, 3, LANE\_ID\_TYPE

OFFSET, DOUBLE, 12, 8.1, METERS

ROUTE, INTEGER, 13, 10

lists the field definitions for the NEW\_TRAVELER\_FILE.

Table NEW\_TRAVELER\_FILE Field Definitions

| **Field(s)** | **Description** | **Data Type** |
| --- | --- | --- |
| HHOLD | Household identifier | Integer |
| PERSON | Person in the household | Integer |
| TOUR | Tour number | Integer |
| TRIP | Trip Number | Integer |
| MODE | Mode of travel (MODE\_TYPE) | Integer |
| TIME | Time of day | Hour\_Clock |
| DISTANCE | Distance traveled by the person | Decimal |
| SPEED | Speed of the person (default units are km / hr) | Decimal |
| LINK | Link identifier | Integer |
| DIR | Direction on the link | Integer |
| LANE | Lane identifier (LANE\_ID\_TYPE) | String |
| OFFSET | Offset along the link (default units are meters) | Decimal |
| ROUTE | Route taken | Integer |

Table 60 is an example of a NEW\_TRAVELER\_FILE populated with data.

Table NEW\_TRAVELER\_FILE Example

| HHOLD | PERSON | TOUR | TRIP | MODE | TIME | DISTANCE | SPEED | LINK | DIR | LANE | OFFSET | ROUTE |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1889435 | 1 | 1 | 1 | DRIVE | 17:00 | 30.0 | 108.0 | 6233 | 0 | 1 | 1635.0 | 0 |
| 1930992 | 1 | 1 | 1 | DRIVE | 17:00 | 22.5 | 81.0 | 4207 | 0 | 1 | 945.0 | 0 |

# 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. Table 61 is an example of a TRIP\_FILE populated with data:

Table 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\_1 key is appended to the PROJECT\_DIRECTORY key to specify the file name for the input TRIP\_TABLE\_FILE\_1 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 the TRANSIMS 3 format. The default TRANSIMS 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.

Table 62 is an example of a TRIP\_TABLE\_FILE\_1 populated with data.

Table 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

The optional TURN\_PENALTY\_FILE lists turn penalties and prohibitions. A typical field definition (.def or .DEF) file is listed below:

TRANSIMS50, TAB\_DELIMITED, 1

LINK, INTEGER, 1, 10

DIR, INTEGER, 2, 1

TO\_LINK, INTEGER, 3, 10

START, TIME, 4, 16, HOUR\_CLOCK

END, TIME, 5, 16, HOUR\_CLOCK

USE, STRING, 6, 128, USE\_TYPE

MIN\_TYPE, UNSIGNED, 7, 3, VEHICLE\_TYPE

MAX\_TYPE, UNSIGNED, 8, 3, VEHICLE\_TYPE

PENALTY, UNSIGNED, 9, 5, IMPEDANCE

NOTES, STRING, 10, 128

Table 63 lists the field definitions for the TURN\_PENALTY\_FILE.

Table TURN\_PENALTY\_FILE Field Definitions

| **Field(s)** | **Description** | **Use** | **Values** |
| --- | --- | --- | --- |
| LINK | Inbound link | Req[[2]](#footnote-2) | Integer |
| DIR | Direction on the link | Req | Integer |
| TO\_LINK | Outbound link | Req | Integer |
| START | Start time (defaults to 0) | Opt | Time of Day |
| END | End time (defaults to 24:00) | Opt | Time of Day |
| USE | USE\_CODE[[3]](#footnote-3) | Opt |  |
| MIN\_TYPE | VEH\_TYPE | Opt |  |
| MAX\_TYPE | VEH\_TYPE | Opt |  |
| PENALTY | IMPEDANCE in seconds. 0 denotes a turn prohibition | Opt | Integer |
| NOTES | User annotations | Opt | String |

Table 64 is an example of a TURN\_PENALTY\_FILE populated with data.

Table TURN\_PENALTY\_FILE Example

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| LINK | DIR | TO\_LINK | START | END | USE | MIN\_TYPE | MAX\_TYPE | PENALTY | NOTES |
| 26 | 0 | 29 | 0:00 | 24:00 | ANY | 0 | 0 | 120 | Left Turn From M Street to Key Bridge |
| 26 | 1 | 20 | 0:00 | 24:00 | ANY | 0 | 0 | 0 | No turn from N Capitol St NE to Florida Ave for ALLDAY |
| 27 | 0 | 31 | 7:00 | 19:00 | ANY | 0 | 0 | 120 | Right Turn Penalty from M Street to Key Bridge |
| 44 | 1 | 40 | 0:00 | 24:00 | ANY | 0 | 0 | 0 | No turn from Ramp to Ramp for ALLDAY |

# 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 TRANSIMS 4 “Location” field becomes a “Parking” field. Finally, the Version 4 vehicle type and subtype are combined into a Version 5 vehicle type.

Table 66 is an example of a VEHICLE\_FILE populated with data.

Table 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 TRANSIMS 4 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 TRANSIMS 4 conversion is V5\_Vehicle\_Type = 10 \* V4\_Vehicle\_Type + V4\_Subtype.

Table 66 is an example of a VEHICLE\_TYPE\_FILE populated with data.

Table VEHICLE\_TYPE\_FILE Example

| TYPE | LENGTH | MAX\_SPEED | MAX\_ACCEL | MAX\_DECEL | OP\_COST | USE | CAPACITY | LOADING | UNLOADING | METHOD | MIN\_DWELL | MAX\_DWELL | 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 |
| 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

The input plan file, used by NewFormat, for version 4 to version 5 plan file conversions.

# 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. An example of this output file is listed 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 or .DEF) file is listed below:

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.

Table 67 is an example of a ZONE\_FILE populated with data.

Table 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

1. an integer that corresponds to SKIM\_TIME\_PERIODS or SKIM\_TIME\_INCREMENT [↑](#footnote-ref-1)
2. Links are required unless nodes (FROMNODE, TONODE, NODE) are used. The node option is provided for backwards compatibility. [↑](#footnote-ref-2)
3. Use codes include ANY, WALK, BICYCLE, AUTO, TRUCK, BUS, RAIL, SOV, HOV2, HOV3, HOV4, LIGHTTRUCK, HEAVYTRUCK, RESTRICTED, CAR, BIKE, TAXI, TROLLEY, STREETCAR, LIGHTRAIL, RAPIDRAIL, REGIONRAIL [↑](#footnote-ref-3)