

NetPrep

Version 5.0.11

Revision History

May 2012 - Created by Volpe Center

The **NetPrep** (*Network Preparation Utility*) program is used to:

1. *Combine GISNet, TPPlusNet, and EMMENet features.*
 - a. *Convert and edit source data to create Input Link and Node files.*
2. *Perform TransimsNet functions related to link and node selection.*
 - a. *Review and edit link and node information/files.*
 - b. *Reformat, merge, select, and collapse nodes and links.*
3. *Perform spatial network manipulations using new controls.*
4. *Develop a User Program script to manipulate link data fields.*
5. *Merge networks.*

Syntax is NetPrep [-flag] [control_file]

The control_file is the file name of an ASCII file that contains the control strings expected by the program. The control_file is optional. If a file name is not provided, the program will prompt the user to enter a file name. The flag parameters are also optional. Any combination of the following flag parameters can be included on the command line:

Optional Flags:

- Q[uiet] = execute without screen messages
- H[elp] = show program syntax and control keys
- C[ontrol] = create/update a default control file
- K[eyCheck] = list unrecognized control file keys
- P[ause] = pause before exiting
- N[oPause] = never pause before exiting
- D[etail] = execute with detailed status messages
- X[ML] = write an XML file with control keys

The program automatically creates a printout file based on the control file name. If the filename includes an extension (e.g., ".ctl"), the extension is replaced with ".prn". The printout file will be created in the current working directory and will overwrite an existing file with the same name.

Version 5 Features (as of Version 5.0.11)

- The speed-capacity lookup table logic was corrected when area type and facility type fields are provided.
- Route Nodes file processing was added. The new keys include TRANSIT_TIME_PERIODS, PERIOD_TRAVEL_TIMES, FIRST_ROUTE_NUMBER, INPUT_ROUTE_FORMAT, ROUTE_MODE_MAP, INPUT_ROUTE_FILE, ROUTE_PERIOD_MAP, and FLIP_ROUTE_FLAG.
- The Version 4 Time Format key also translates Router Header headways, offsets, and travel times.

- Network node and link selecting and collapsing moved to NetPrep (formerly part of TransimsNet)

Control Key List

The list of control file keys appears in the table below:

- Req / Opt indicates whether the key is **required** or **optional**
- The types include **Text**, Input **Filename**, **New** file, **Boolean**, **Path** (to a file), **Time**, **Integer**, **Decimal**, and **List** of items
- The Default is the default value, used if the key does not appear in the control file.
- I/O/P indicates Input, Output or Parameter.

For a more detailed description of the Parameter control keys, refer to the Parameter Reference. For a more detailed description of the Input or Output control keys, refer to the File Reference. These two documents also provide the possible values or range of values allowed for each control key listed below. For instance, files can usually be output to numerous formats beyond TAB_DELIMITED for additional post-processing / file manipulation actions.

Configuration Keys

Control File Keys:	Req/Opt	Type	Default	I/O/P
TITLE	Opt	Text		O
REPORT_FILE	Opt	File		O
REPORT_FLAG	Opt	Bool	FALSE	P
PROJECT_DIRECTORY	Opt	Path		P
DEFAULT_FILE_FORMAT ¹	Opt	Text	TAB_DELIMITED	P
TIME_OF_DAY_FORMAT ¹	Opt	Text	DAY_TIME	P
MODEL_START_TIME	Opt	Time	0:00	P
MODEL_END_TIME	Opt	Time	24:00	P
UNITS_OF_MEASURE	Opt	Text	METRIC	P
RANDOM_NUMBER_SEED	Opt	Int.	0	P
MAX_WARNING_MESSAGES	Opt	Int.	100000	P
MAX_WARNING_EXIT_FLAG	Opt	Bool	TRUE	P
MAX_PROBLEM_COUNT	Opt	Int.	0	P
NUMBER_OF_THREADS	Opt	Int.	1	P
NETPREP_REPORT_*	Opt	Text		P

System File Keys

Control File Keys:	Req/Opt	Type	Default	I/O/P
NODE_FILE	Opt	File		I
NODE_FORMAT	Opt	Text	TAB_DELIMITED	P
ZONE_FILE	Opt	File		I
ZONE_FORMAT	Opt	Text	TAB_DELIMITED	P
SHAPE_FILE	Opt	File		I
SHAPE_FORMAT	Opt	Text	TAB_DELIMITED	P
LINK_FILE	Opt	File		I

LINK_FORMAT	Opt	Text	TAB_DELIMITED	P
NEW_NODE_FILE	Opt	File		O
NEW_NODE_FORMAT	Opt	Text	TAB_DELIMITED	P
NEW_ZONE_FILE	Opt	File		O
NEW_ZONE_FORMAT	Opt	Text	TAB_DELIMITED	P
NEW_SHAPE_FILE	Opt	File		O
NEW_SHAPE_FORMAT	Opt	Text	TAB_DELIMITED	P
NEW_LINK_FILE	Req	File		O
NEW_LINK_FORMAT	Opt	Text	TAB_DELIMITED	P

File Service Keys

Control File Keys:	Req/Opt	Type	Default	I/O/P
NOTES_AND_NAME_FIELDS	Opt	Bool	FALSE	P

Projection, Smooth Data, Difference Data Keys

Control File Keys:	Req/Opt	Type	Default	I/O/P
INPUT_COORDINATE_SYSTEM	Opt	List		P
INPUT_COORDINATE_ADJUSTMENT	Opt	List		P
OUTPUT_COORDINATE_SYSTEM	Opt	List		P
OUTPUT_COORDINATE_ADJUSTMENT	Opt	List		P
OUTPUT_XYZ_SHAPES	Opt	Bool	FALSE	P
OUTPUT_XYM_SHAPES	Opt	Bool	FALSE	P

Control Keys

Control File Keys:	Req/Opt	Type	Default	I/O/P
INPUT_LINK_FILE	Opt	File		I
INPUT_LINK_FORMAT	Opt	Text	TAB_DELIMITED	I
INPUT_NODE_FILE	Opt	File		P
INPUT_NODE_FORMAT	Opt	Text	TAB_DELIMITED	P
INPUT_ZONE_FILE	Opt	File		I
INPUT_ZONE_FORMAT	Opt	Text	TAB_DELIMITED	P
INPUT_SPDCAP_FILE	Opt	File		I
INPUT_SPDCAP_FORMAT	Opt	Text	TAB_DELIMITED	P
INPUT_UNITS_OF_MEASURE	Opt	Text	METRIC	P
FACILITY_INDEX_FIELD	Opt	Text	METRIC	P
AREA_TYPE_INDEX_FIELD	Opt	Text		P
CONVERSION_SCRIPT	Opt	File		I
CONVERSION_STACK ¹	Opt	File		I
INTERNAL_ZONE_RANGE	Opt	List	0	P
KEEP_ZONE_CONNECTORS_AS_LOCALS	Opt	Bool	FALSE	P
FIRST_NODE_NUMBER	Opt	Int.	1	P
FIRST_LINK_NUMBER	Opt	Int.	1	P

SPEED_FACTOR_BY_FACILITY	Opt	List	1.0	P
SPEED_ROUNDING_INCREMENT	Opt	Dec.	7.5 mps	P
COORDINATE_RESOLUTION	Opt	Dec.	1.0 meters	P
MAXIMUM_LENGTH_TO_XY_RATIO	Opt	Dec.	1.2	P
MAXIMUM_SHAPE_ANGLE	Opt	Int.	90 degrees	P
MINIMUM_SHAPE_LENGTH	Opt	Int.	10 meters	P
DROP_DEAD_END_LINKS	Opt	Int.	0 meters	P
DROP_SHORT_LINKS	Opt	Dec.	37.5 meters	P
SPLIT_LARGE_LOOPS	Opt	Int.	100 meters	P
COLLAPSE_SHAPE_NODES	Opt	Bool	FALSE	P
COLLAPSE_DIVIDED_ARTERIALS	Opt	Bool	FALSE	P
LOCAL_THRU_SEGMENT_LENGTHS_*	Opt	List		P
LOCAL_SELECTION_SPACING_*	Opt	List		P
KEEP_NODE_RANGE	Opt	List	ALL	P
KEEP_NODE_FILE	Opt	File		I
KEEP_LINK_RANGE	Opt	List	ALL	P
KEEP_LINK_FILE	Opt	File		I
DELETE_NODE_RANGE	Opt	List	NONE	P
DELETE_NODE_FILE	Opt	File		I
DELETE_LINK_RANGE	Opt	List	NONE	P
DELETE_LINK_FILE	Opt	File		I
NEW_LINK_DETAIL_FILE ²	Opt	File		O
NEW_LINK_NODE_LIST_FILE ²	Opt	File		O

Notes

Each '_FILE' key has a corresponding '_FORMAT' key. The following file formats can be used for input and output files: TEXT, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, SQLITE3, VERSION3

¹-Control key is listed in the -h command line listing generated by NetPrep, but the key is not listed in the associated Quick Reference document.

²-Control key is listed in the NetPrep Quick Reference document, but the key is not listed in the associated -h command line listing generated by this program.

Control Key Notes for NetPrep Version 5

TransimsNet 4.0 to NetPrep 5.0 File Name and Control Key Mappings for TRANSIMS 5.0

- NET_NODE_TABLE → NODE_FILE
- NET_ZONE_TABLE → ZONE_FILE
- NET_SHAPE_TABLE → SHAPE_FILE
- NET_LINK_TABLE → LINK_FILE
- NEW_NODE_TABLE → NEW_NODE_FILE
- NEW_ZONE_TABLE → NEW_ZONE_FILE

- NEW_SHAPE_TABLE → NEW_SHAPE_FILE
- NEW_LINK_TABLE → NEW_LINK_FILE
- CREATE_NOTES_AND_NAMES_FIELDS → NOTES_AND_NAME_FIELDS
- NET_DIRECTORY → [No equivalent key in Version 5.0]
- MINIMUM_LINK_LENGTH → DROP_SHORT_LINKS
- COLLAPSE_NODES_FLAG → COLLAPSE_SHAPE_NODES
- MAXIMUM_LENGTH_TO_XY_RATIO → MAXIMUM_LENGTH_TO_XY_RATIO
- KEEP_NODE_LIST → KEEP_NODE_FILE
- CELL_SIZE (Microsimulator V4.0) → SPEED_ROUNDING_INCREMENT

New File and Parameter Control Keys in the TRANSIMS 5.0 Software Suite (NetPrep 5.0)

- DROP_DEAD_END_LINKS
- DROP_SHORT_LINKS
- SPLIT_LARGE_LOOPS
- COLLAPSE_SHAPE_NODES
- COLLAPSE_DIVIDED_ARTERIALS
- LOCAL_THRU_SEGMENT_LENGTHS_*
- LOCAL_SELECTION_SPACING_*
- TRANSIT_TIME_PERIODS
- PERIOD_TRAVEL_TIMES
- FIRST_ROUTE_NUMBER
- INPUT_ROUTE_FORMAT
- ROUTE_MODE_MAP
- INPUT_ROUTE_FILE
- ROUTE_PERIOD_MAP
- FLIP_ROUTE_FLAG
- SPEED_FACTOR_BY_FACILITY

TransimsNet Version 5 Network Conversion Process

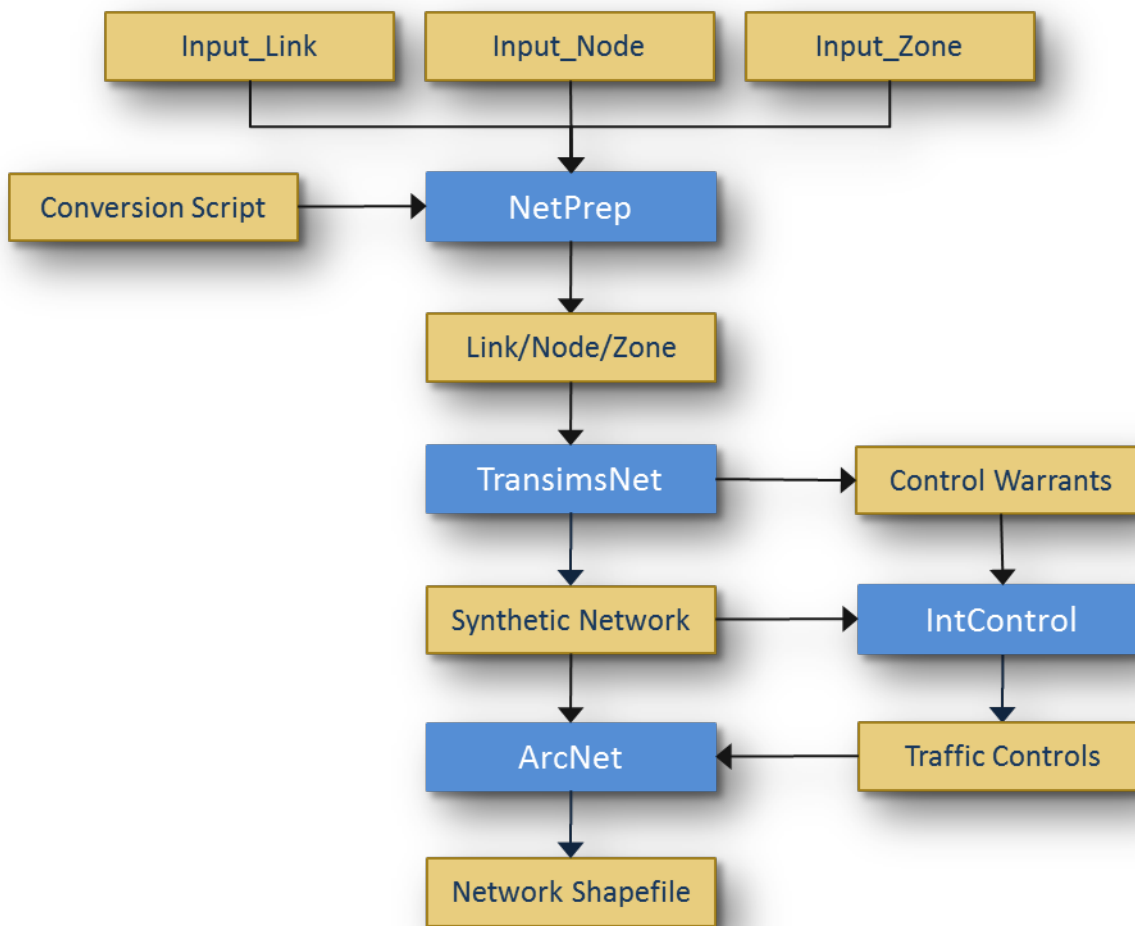


Figure 1: Network Conversion Process (High-Level)

The above flowchart shows the typical execution sequence of TRANSIMS 5.0 network programs and file inputs and outputs relative to each program, and relative to the overall process. Starting with TRANSIMS 5.0, NetPrep (a new part of the TRANSIMS program suite) is used instead of TransimsNet to convert input network files (nodes, links, and zones) into TRANSIMS 5.0 network format compatible files.

The raw input files may be from an existing TRANSIMS 4.0 network, from raw network files exported from commercial traffic modeling software such as TransCAD®, or may be manually created given that the file names, file types/formats/extensions, location, layout (header row in particular), and contents are consistent with what is expected by TRANSIMS 5.0 and the relevant control files for each program noted above.

The initial network input files and the TRANSIMS 5.0 network format files that result from the process are shown in Figure 1 above at a relatively high level. The combined file output of NetPrep, TransimsNet, and IntControl comprise the majority of the synthetic network files which will be used for subsequent trip table generation, trip routing, and finally trip simulation. The use of NetPrep is optional while TransimsNet and IntControl must be used. However, many TRANSIMS 5 users will benefit from the multiple functions performed by NetPrep. Since TRANSIMS 5 field names have been updated

or moved in many instances, the use of NetPrep with a simple conversion script will reduce the need for numerous manual file operations.

For a more detailed look at this process, refer to the Highway Network How-To document. Also refer to the Network Edit How-To document for additional details on using NetPrep in tandem with TransimsNet, IntControl, and ArcNet to synthesize and verify/validate a TRANSIMS 5 network.

Some associated tasks and notes not readily apparent from Figure 1 are the iterative nature of the process, subsequent review, manipulation, and generation of the remaining TRANSIMS network files. The process depicted in Figure 1 indicates the proper order for sequentially synthesizing the network. However, the overall process of network generation actually employed (within certain constraints) to develop a TRANSIMS 5.0 implementation varies based on the available network input data as well as the complexity of the implementation and its objectives.

Additional methods will typically be needed in order to generate a valid and verifiable TRANSIMS 5.0 synthetic network suitable for trip table generation, and eventually, for accurate simulation of traffic flows. This includes both pre- and post-processing methods to review, edit, and reiterate a process step or steps as needed. Errors and warnings should be addressed as they arise, and resolved whenever feasible. Warnings do not necessarily prevent subsequent creation and population of or use of a file or group of files by another program. Batch files can be used to good effect in iterating some repetitive tasks such as regenerating the network from scratch using updated files. Additionally, scripting methods (e.g., Visual Basic®, VBA®, Python, Perl, etc.) can also prove quite useful for automating many repetitive tasks (iterative program execution, file manipulation, file input/output, etc.), especially with larger scale efforts.

These methods are discussed in greater detail in the TransimsNet, IntControl, and ArcNet program references and quick references and training documents, as well as in the Highway Network How-To and Network Edit How-To documents. The Highway Network How-To document also provides a more detailed discussion of the aforementioned supply/network-side TRANSIMS 5.0 programs in the context of the sequential, iterative, network synthesis process.

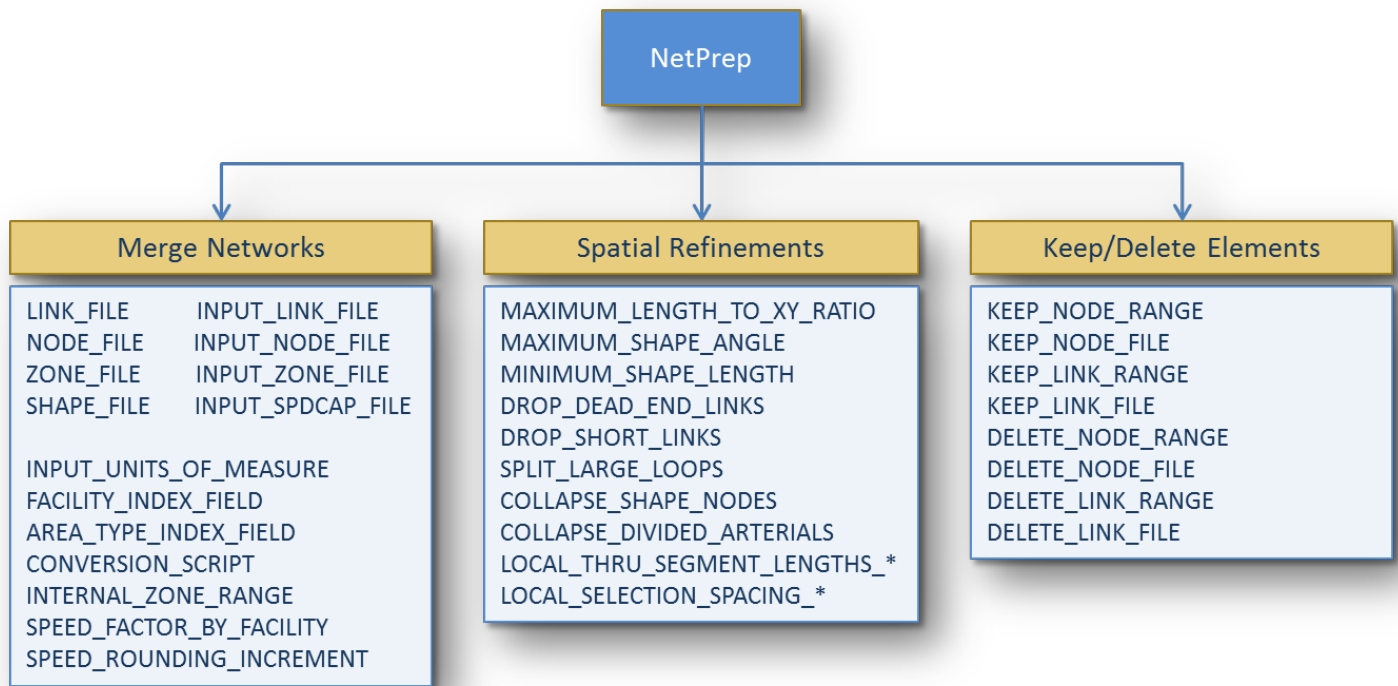


Figure 2: NetPrep Control Keys Grouped by Associated Function

Update and Delete Applications

The NetPrep program is also used to update or delete records in an existing TRANSIMS 5 network. This was previously accomplished using TransimsNet 4. The update and delete node and/or link ranges or files specify the records to be processed. Input network table keys are needed to define the existing network for TransimsNet 5. NetPrep 5 can be used to generate these necessary input files using a network shape file if one exists. Alternatively, these tasks can still be accomplished manually similar to TRANSIMS 4, taking into account the changes in file and control key names and locations introduced in TRANSIMS 5. The records in the input file are read and copied to the output network if not included in one of the update or delete lists. If the record is associated with a link or a node that is updated or deleted, the record is not copied to the output file. The new records generated by the update option are added to the end of the output file. As a result, records for a given location (link or node) will have different record numbers in the output file than in the input file.

The link-related data types include the pocket lanes (POCKET_FILE), activity locations (LOCATION_FILE), parking lots (PARKING_FILE), and process links (ACCESS_FILE). The reader is referred to the TRANSIMS 5 Training Material and the TransimsNet 5 Program Reference for additional information on the changes in naming conventions present in version 5, as well as the functional significance of these link-related data type changes. Since the input link file may include changes to the number of lanes for an updated link, the lane connectivity (CONNECTION_FILE) records at both ends of the link are also regenerated. In all cases, the methods used to regenerate pocket lanes, locations, and other link attributes utilized the parameters specified for the application. In other words, the rules for generating synthetic data may be different than the rules used to generate the original network. This enables the user to customize the synthetic generate process for specific links or subareas within the network. The node-related data types include the lane connections and the

traffic controls (sign and signal warrants (SIGN_FILE, SIGNAL_FILE)). The node update option is typically used for applying different signal and sign warrant (SIGN_FILE, SIGNAL_FILE) parameters for different areas or subareas within the region or to simply remove signals or signs that are not wanted.

The delete options are intended to physically remove link and nodes from the system. It is not possible to remove a node without also removing all of the links attached to the node. Deleting a link will automatically update the lane connections at the nodes at both ends of the deleted link.

Creating and Using Conversion Scripts

The conversion script key is a file name that includes a TRANSIMS User Program script. The programming language for the script is described in the User Programs documentation and the reader is referred there for more comprehensive discussion of writing user programs or scripts for TRANSIMS. By default, the data field names in the shape file are copied to the corresponding field names in the TRANSIMS link file. If a shape file (.shp) of the network exists and 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., Version 4 vs. Version 5)). If the GIS 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 GIS link fields are referenced as "Link.*field*" and the TRANSIMS link fields are referenced as "NewLink.*field*". The conversion script itself is typically a standalone ANSI text file named "Script.txt" or "GIS.cfg". The path to the location of the script file must be appended to the front of the associated control key value in the "NetPrep.ctl" control file.

An example of a conversion script appears below:

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

NewLink.LENGTH = 1609 * Link.LENGTH

NewLink.LANES_AB = Link.AB_LANE
NewLink.LANES_BA = Link.BA_LANE

NewLink.SPEED_AB = Link.AB_PKSPD
NewLink.SPEED_BA = Link.BA_PKSPD

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

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

IF (Link.FT == 1) THEN
    NewLink.TYPE = "FREEWAY"
ELSE IF (Link.FT == 2) THEN
    NewLink.TYPE = "Expressway"
ELSE IF (Link.FT == 3) THEN
    NewLink.TYPE = "Principal"
ELSE IF (Link.FT == 4) THEN
```

```

NewLink.TYPE = "Major"
ELSE IF (Link.FT == 5) THEN
  NewLink.TYPE = "Minor"
ELSE IF (Link.FT == 6) THEN
  NewLink.TYPE = "Collector"
ELSE IF (Link.FT == 7) THEN
  NewLink.TYPE = "Local"
ELSE IF (Link.FT == 8) THEN
  NewLink.TYPE = "Local"
ELSE IF (Link.FT == 9) THEN
  NewLink.TYPE = "Frontage"
ELSE IF (Link.FT == 20) THEN
  NewLink.TYPE = "External"
ENDIF ENDIF ENDIF ENDIF ENDIF ENDIF ENDIF ENDIF ENDIF ENDIF

RETURN (1)
END

```

Example 1–Collapse Divided Arterials Using NetPrep

It is often useful to collapse divided arterials in a highway network early in the network synthesis and subsequent simulation process. Raw network data provided by MPOs and other project stakeholders is often provided in a divided arterial format. For simplicity, it is recommended that the decision to collapse divided arterials be made and implemented early in the TRANSIMS modeling effort process as subsequent file processing will in turn be simplified.

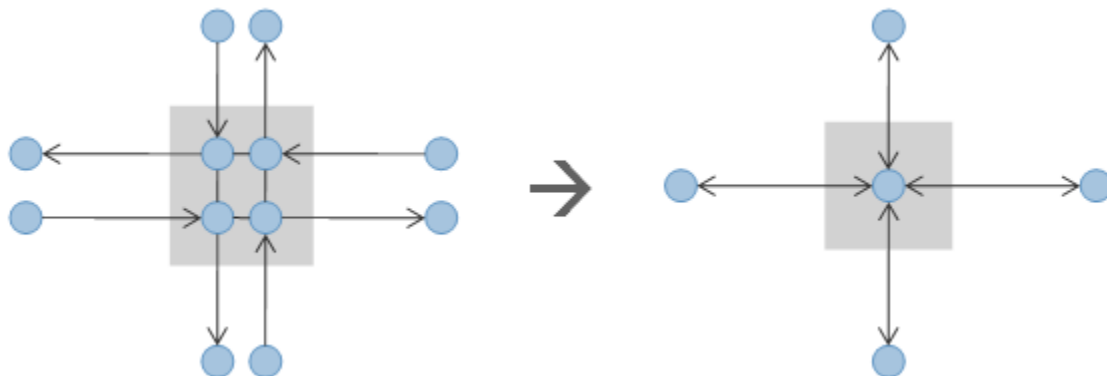


Figure 3: Collapsing Divided Arterials-Conceptual Diagram

Example Input File / Control File (NetPrep.ctl) – Alexandria V5 Network

//---- Use NetPrep to implement Node Keep/Collapse and Link Length adjustments from TransimsNet 4.0 ----

```

TITLE                               Network Preparation
DEFAULT_FILE_FORMAT                 TAB_DELIMITED
PROJECT_DIRECTORY                   ../

```

//---- Input Files ----

Input System Network Files:

Node File = ../inputs/Input_Node.txt

Zone File = ../inputs/Input_Zone.txt

Shape File = ../inputs/Input_Shape.txt

Link File = ../inputs/Input_Link.txt

Output System Network Files:

New Node File = ../inputs/Input_Node2.txt

New Zone File = ../inputs/Input_Zone2.txt

New Shape File = ../inputs/Input_Shape2.txt

New Link File = ../inputs/Input_Link2.txt

Notes And Name Fields = TRUE

Data Service Controls:

NetPrep Control Keys:

Speed Factor By Facility = 1, 1, 1.05, 1.05, 1.08, 1.1, 1.1, 1.1, 1.05, 1.05, 1.05, 1

Speed Rounding Increment = 7.50 mps

Maximum Length to XY Ratio = 1.2

Drop Short Links = 7.50 meters

Split Large Loops = 100 meters

Collapse Shape Nodes = TRUE

Keep Node File = ../inputs/Keep_Node_List.txt

Number of Node File Records = 2620

Number of Zone File Records = 85

Highest Zone Number = 92

Number of Shape File Records = 18072

Number of Link Shape Records = 2369

Number of Link File Records = 3653

Number of Directional Links = 6852

Number of Collapsed Nodes = 46

Number of Short Links Dropped = 2

Number of Collapsed Nodes = 1

New Node File Records = 2572

New Zone File Records = 85

Highest Zone Number = 92

New Shape File Records = 18285

New Link File Records = 3606

Fri Jan 21 08:22:32 2011 -- Process Complete with 1 Warning (0:00:00)

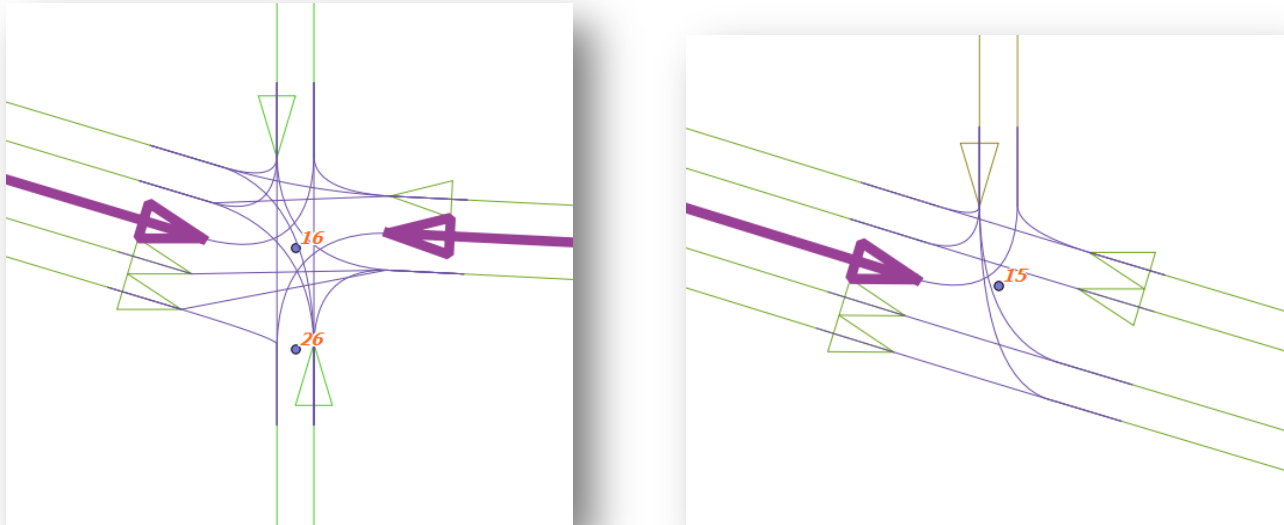


Figure 4: QGIS Visual Comparison of Divided Arterials and Collapsed Arterials in TRANSIMS 5.0

The top-left figure (Figure 4) illustrates a divided or 2-node arterial intersection in TRANSIMS 5. In contrast, the top-right figure (Figure 4) shows another intersection from the same network model that utilizes a single node/collapsed node arterial intersection. These images were produced by creating shape files from the TRANSIMS 5 network files using ArcNet 5 and loading the shape files as layers into a standalone GIS program. The reader is referred to the ArcNet Program Reference and Highway Network How-To Guide for more in-depth discussion of generating shape files using ArcNet 5.

Example 2–Using NetPrep to Convert Shape Files to TRANSIMS Network Input Files

Note that NetPrep 5.0 replaces GISNet 4.0 entirely. GISNet is no longer a part of the TRANSIMS 5.0 traffic microsimulation suite. In addition, NetPrep 5.0 includes many features that were previously incorporated into TransimsNet 4.0. The example below illustrates how to convert a GIS shape file into Input_Link.txt, Input_Node.txt, and Input_Shape.txt files which can be used as inputs into TransimsNet 5.0. Additional functionality has been incorporated into TransimsNet 5.0, and the reader is referred to the TransimsNet documentation for additional information.

Example Input File / Control File (NetPrep.ctl) – A Large Network

TITLE NetPrep – Shape File to Network Files Conversion

PROJECT_DIRECTORY ../
UNITS_OF_MEASURE METRIC

//---- The INPUT_LINK_FILE key requires a shape file with a .shp extension to create TRANSIMS network input files ----
INPUT_LINK_FILE inputs/arcview/Large.shp

//---- A conversion script is required for this and is similar to the type used by GISNet in TRANSIMS 4.0 ----

CONVERSION_SCRIPT	RTE/Script.txt
NEW_NODE_FILE	network/Input_Node.txt
NEW_LINK_FILE	network/Input_Link.txt
NEW_SHAPE_FILE	network/Input_Shape.txt

```
//---- The correct INPUT_COORDINATE_SYSTEM key value (UTM meters, LATLONG, etc.) corresponding to the shape ----
//---- file included above must be used to avoid NetPrep processing errors ----
```

INPUT_COORDINATE_SYSTEM	UTM, 15N, METERS
OUTPUT_COORDINATE_SYSTEM	UTM, 15N, METERS

```
NETPREP_REPORT_1      CONVERSION_SCRIPT
NETPREP_REPORT_2      CONVERSION_STACK
```

Example Output File / Results of Running NetPrep (NetPrep.prn) – A Large Network

```

*****
|
|           NetPrep - Version 5.0.9
|      Copyright 2012 by TRANSIMS Open-Source
|           Tue May 15 07:38:16 2012
|
*****

```

Control File = ../control/NetPrep.ctl
Report File = ../control/NetPrep.prn (Create)

NetPrep – Shape File to Network Files Conversion

Project Directory = ../
Units of Measure = METRIC
Random Number Seed = 1337081896

Output System Network Files:
 New Node File = ../network/Input_Node.txt
 New Shape File = ../network/Input_Shape.txt
 New Link File = ../network/Input_Link.txt

Data Service Controls:
Input Coordinate System = UTM, 15N, METERS
Output Coordinate System = UTM, 15N, METERS

NetPrep Control Keys:
Input Link Shapefile = ../inputs/arcview/Large.shp

Conversion Script = ../RTE/Script.txt

NetPrep Reports: 1. CONVERSION_SCRIPT
2. CONVERSION_STACK

Conversion Script

```
NewLink.USE = "ANY"
IF (Link.DIR == 0) THEN
    NewLink.LANES_BA = Link.LANES * 0.5
    NewLink.LANES_AB = Link.LANES * 0.5
ELSE IF (Link.DIR == 1) THEN
    NewLink.LANES_BA = 0
    NewLink.LANES_AB = Link.LANES
ELSE IF (Link.DIR == -1) THEN
    NewLink.LANES_AB = 0
    NewLink.LANES_BA = Link.LANES
ELSE IF (Link.DIR == 2) THEN
    NewLink.LANES_AB = Link.PERMLANESB
    NewLink.LANES_BA = Link.PERMLANESA
ENDIF ENDIF ENDIF ENDIF

NewLink.SPEED_AB = Link.SPEED_AB
NewLink.SPEED_BA = Link.SPEED_BA
NewLink.FSPD_AB = 0.8 * NewLink.SPEED_AB
NewLink.FSPD_BA = 0.8 * NewLink.SPEED_BA

IF (Link.FUNCL == 1) THEN
    NewLink.TYPE = "FREEWAY"
ELSE IF (Link.FUNCL == 2) THEN
    NewLink.TYPE = "Expressway"
ELSE IF (Link.FUNCL == 3) THEN
    NewLink.TYPE = "Principal"
ELSE IF (Link.FUNCL == 4) THEN
    NewLink.TYPE = "Major"
ELSE IF (Link.FUNCL == 5) THEN
    NewLink.TYPE = "Minor"
ELSE IF (Link.FUNCL == 6) THEN
    NewLink.TYPE = "Collector"
ELSE IF (Link.FUNCL == 7) THEN
    NewLink.TYPE = "Local"
ELSE IF (Link.FUNCL == 8) THEN
    NewLink.TYPE = "Local"
ELSE IF (Link.FUNCL == 9) THEN
    NewLink.TYPE = "Frontage"
ELSE IF (Link.FUNCL == 10) THEN
    NewLink.TYPE = "Ramp"
ELSE IF (Link.FUNCL == 20) THEN
    NewLink.TYPE = "External"
ENDIF ENDIF ENDIF ENDIF ENDIF ENDIF ENDIF ENDIF ENDIF ENDIF

RETURN (1)
END
```

Conversion Stack

- 1) String "ANY"
- 2) Assign =
- 3) String NewLink.USE
- 4) Integer Link.DIR
- 5) Integer 0
- 6) Relation EQ
- 7) Logical If False, Jump to 19
- 8) Integer Link.LANES
- 9) Real 0.500000
- 10) Math *
- 11) Assign =
- 12) Integer NewLink.LANES_BA
- 13) Integer Link.LANES
- 14) Real 0.500000
- 15) Math *
- 16) Assign =
- 17) Integer NewLink.LANES_AB
- 18) Logical Jump to 52
- 19) Integer Link.DIR
- 20) Integer 1
- 21) Relation EQ
- 22) Logical If False, Jump to 30
- 23) Integer 0
- 24) Assign =
- 25) Integer NewLink.LANES_BA
- 26) Integer Link.LANES
- 27) Assign =
- 28) Integer NewLink.LANES_AB
- 29) Logical Jump to 52
- 30) Integer Link.DIR
- 31) Integer 1
- 32) Math Negative
- 33) Relation EQ
- 34) Logical If False, Jump to 42
- 35) Integer 0
- 36) Assign =
- 37) Integer NewLink.LANES_AB
- 38) Integer Link.LANES
- 39) Assign =
- 40) Integer NewLink.LANES_BA
- 41) Logical Jump to 52
- 42) Integer Link.DIR
- 43) Integer 2
- 44) Relation EQ
- 45) Logical If False, Jump to 52
- 46) Real Link.PERMLANESB
- 47) Assign =
- 48) Integer NewLink.LANES_AB
- 49) Real Link.PERMLANESA
- 50) Assign =

51) Integer NewLink.LANES_BA
 52) Integer Link.SPEED_AB
 53) Assign =
 54) Real NewLink.SPEED_AB
 55) Integer Link.SPEED_BA
 56) Assign =
 57) Real NewLink.SPEED_BA
 58) Real 0.800000
 59) Real NewLink.SPEED_AB
 60) Math *
 61) Assign =
 62) Real NewLink.FSPD_AB
 63) Real 0.800000
 64) Real NewLink.SPEED_BA
 65) Math *
 66) Assign =
 67) Real NewLink.FSPD_BA
 68) Integer Link.FUNCL
 69) Integer 1
 70) Relation EQ
 71) Logical If False, Jump to 76
 72) String "FREEWAY"
 73) Assign =
 74) String NewLink.TYPE
 75) Logical Jump to 155
 76) Integer Link.FUNCL
 77) Integer 2
 78) Relation EQ
 79) Logical If False, Jump to 84
 80) String "Expressway"
 81) Assign =
 82) String NewLink.TYPE
 83) Logical Jump to 155
 84) Integer Link.FUNCL
 85) Integer 3
 86) Relation EQ
 87) Logical If False, Jump to 92
 88) String "Principal"
 89) Assign =
 90) String NewLink.TYPE
 91) Logical Jump to 155
 92) Integer Link.FUNCL
 93) Integer 4
 94) Relation EQ
 95) Logical If False, Jump to 100
 96) String "Major"
 97) Assign =
 98) String NewLink.TYPE
 99) Logical Jump to 155
 100) Integer Link.FUNCL
 101) Integer 5

102) Relation EQ
103) Logical If False, Jump to 108
104) String "Minor"
105) Assign =
106) String NewLink.TYPE
107) Logical Jump to 155
108) Integer Link.FUNCL
109) Integer 6
110) Relation EQ
111) Logical If False, Jump to 116
112) String "Collector"
113) Assign =
114) String NewLink.TYPE
115) Logical Jump to 155
116) Integer Link.FUNCL
117) Integer 7
118) Relation EQ
119) Logical If False, Jump to 124
120) String "Local"
121) Assign =
122) String NewLink.TYPE
123) Logical Jump to 155
124) Integer Link.FUNCL
125) Integer 8
126) Relation EQ
127) Logical If False, Jump to 132
128) String "Local"
129) Assign =
130) String NewLink.TYPE
131) Logical Jump to 155
132) Integer Link.FUNCL
133) Integer 9
134) Relation EQ
135) Logical If False, Jump to 140
136) String "Frontage"
137) Assign =
138) String NewLink.TYPE
139) Logical Jump to 155
140) Integer Link.FUNCL
141) Integer 10
142) Relation EQ
143) Logical If False, Jump to 148
144) String "Ramp"
145) Assign =
146) String NewLink.TYPE
147) Logical Jump to 155
148) Integer Link.FUNCL
149) Integer 20
150) Relation EQ
151) Logical If False, Jump to 155
152) String "External"

```
153) Assign    =  
154) String    NewLink.TYPE  
155) Integer   1  
156) Return    Integer  
157) End       1
```

Warning: Duplicate Link Direction Number = 25602
Warning: Duplicate Link Direction Number = 25625
Warning: Duplicate Link Direction Number = 25682
Warning: Duplicate Link Direction Number = 25687
Warning: Duplicate Link Direction Number = 25696
Warning: Duplicate Link Direction Number = 25859
Warning: Duplicate Link Direction Number = 26110

Number of Input Link Shapefile Records = 26244

Number of Short Loops Deleted = 1

New Node File Records = 20043

New Shape File Records = 74182

New Link File Records = 26228

Tue May 15 07:38:34 2012 -- Process Complete with 7 Warnings (0:00:18)



Figure 5: QGIS View of the Shape File used as Input in the Network Conversion Example