TransimsNet *(version 1.9)*

The TransimsNet program is used to:

1. Synthesize TRANSIMS network files from generic node and link information. Node, Link, Activity Location, Parking, Process Link, Lane Connectivity, and Pocket Lane files are generated by the program.
2. The program also generates Sign and Signal Warrant files that could be used as input to the IntControl program to synthesize the TRANSIMS Unsignalized Node, Signalized Node, Timing Plan, Phasing Plan, Detector and Signal Coordinator files.

TransimsNet is a consoled-based program that runs in a command window on either Windows or Linux. The command syntax is:

TransimsNet [-Q] [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 'Q' parameter is also optional. If it is provided, the program runs in "quiet" mode. This suppresses all screen output. The 'Q' parameter is most often used in application scripts.

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 ".prn" is added. The printout file will be created in the current working directory and will overwrite an existing file with the same name.

Control File Parameters

Control parameters are defined using a control key followed by a string or number. The control parameters can be specified in any order. If a given key is defined more than once, the last instance of the key is used. The default value for each key is 0 or "Null". Null parameters do not need to be included in the file. Note that comment lines or extraneous keys can be included in the file. They will be ignored by the program.

A typical TransimsNet control file is shown below. These keys can be defined in a variety of different ways to perform different tasks.

```
TITLE                          Synthetic TRANSIMS Network
NET_DIRECTORY                  d:\Portland\Convert\Emme2
NET_NODE_TABLE                 Node
NET_LINK_TABLE                 Link
INPUT_ZONE_FILE                d:\Portland\Convert\TransimsNet\Zone_File.txt
KEEP_NODE_LIST                 d:\Portland\Convert\TransimsNet\Keep_Nodes.txt
TURN_PROHIBITIONS              d:\Portland\Convert\TransimsNet\Turn.out
TURN_FILE_FORMAT               AT-FROM-TO
LINK_NODE_EQUIVALENCE          d:\Portland\Convert\TransimsNet\Link_Node.txt
NEW_DIRECTORY                  d:\Portland\Convert\TransimsNet
```

```
NEW_NODE_TABLE                  Node
NEW_LINK_TABLE                  Link
NEW_ACTIVITY_LOCATION_TABLE     Activity_Location
NEW_PARKING_TABLE               Parking
NEW_PROCESS_LINK_TABLE          Process_Link
NEW_POCKET_LANE_TABLE           Pocket_Lane
NEW_LANE_CONNECTIVITY_TABLE     Lane_Connectivity
NEW_UNSIGNALIZED_NODE_TABLE     Sign_Warrants
NEW_SIGNALIZED_NODE_TABLE       Signal_Warrants
POCKET_RANGE_FOR_FACILITY_1     100, 400
POCKET_RANGE_FOR_FACILITY_2     60, 200
POCKET_RANGE_FOR_FACILITY_3     40, 100
POCKET_RANGE_FOR_FACILITY_4     30, 60
POCKET_RANGE_FOR_FACILITY_8     30, 60
SIGNAL_WARRANT_FOR_AREA_TYPE_1  COLLECTOR, LOCAL
SIGNAL_WARRANT_FOR_AREA_TYPE_2  COLLECTOR, COLLECTOR
SIGNAL_WARRANT_FOR_AREA_TYPE_3  MINOR, COLLECTOR
SIGNAL_WARRANT_FOR_AREA_TYPE_4  MINOR, MINOR
SIGNAL_WARRANT_FOR_AREA_TYPE_5  MAJOR, MINOR
SIGNAL_WARRANT_FOR_AREA_TYPE_6  MAJOR, MAJOR
SIGNAL_WARRANT_FOR_AREA_TYPE_7  PRINCIPAL, MAJOR
SIGNAL_WARRANT_FOR_AREA_TYPE_8  PRINCIPAL, PRINCIPAL
MAXIMUM_ACCESS_POINTS           4
MINIMUM_SPLIT_LENGTHS           100, 200, 200, 300, 300, 300, 300, 300
MINIMUM_LINK_LENGTH             37.5
MAXIMUM_LENGTH_TO_XY_RATIO      1.2
INTERSECTION_SETBACK_DISTANCE 0.0
```

This example generates the TRANSIMS network files from the generic link and the node files generated by the Emme2Net program. The keys recognized by the TransimsNet program are listed below. These keys can be defined in a variety of different ways to perform different tasks.

TITLE

Any text string can be used on this line. This text is printed on the top of each output page.

NET_DIRECTORY

The network directory key is required. It specifies the path to the input network link and the node file.

NET_NODE_TABLE

The network node table key is required. It specifies the name of the input node file within the network directory. The full path and file name for the node table is constructed by appending the value of this key to the value of the NET_DIRECTORY key. This file can be created by running the Emme2Net program to convert the EMME/2 highway card images into the input node file format.

The input node file is a comma, space, or tab delimited ASCII file. At a minimum, the header line should include three fields in any order: ID, EASTING, and NORTHING. These fields correspond to the node number, x-coordinate, and y-coordinate for the node. If the coordinates are not in UTM meters, the input and output coordinate system parameters should be specified to convert the coordinates to UTM meters.

NET_LINK_TABLE

The network link table key is required. It specifies the name of the input link file within the network directory. The full path and file name for the link table is constructed by appending the value of this key to the value of the NET_DIRECTORY key. This file can be created by running the Emme2Net program to convert the EMME/2 highway card images into the input link file.

The input link file is a comma, space, or tab delimited ASCII file. At a minimum, the header line should include nine fields in any order: NODEA, NODEB, LENGTH, PERMLANESA, PERMLANESB, FREESPDA, FREESPDB, FUNCTCLASS, and VEHICLE. Two additional fields are processed if provided. They include CAPACITYA and CAPACITYB. The meaning and definition of these fields is consistent with the TRANSIMS network link table. This means that a standard TRANSIMS link and node file can be used as input to this program.

The length and free speed fields should be defined in the same units as the input coordinate system. The following table shows the relationship between the coordinates in the input node file and the expected length and speed units.

Coordinates	Length	Speed
Meters	Meters	Meters per Second
Kilometers	Kilometers	Kilometers per Hour
Feet	Feet	Miles per Hour
Miles	Miles	Miles per Hour

If no coordinate conversion is performed, length should be defined in meters and speed in meters per second.

The PERMLANESA and PERMLANESB fields represent the number of full travel lanes in each direction. PERMLANESA is interpreted as the number of lanes available from NODEB to NODEA. PERMLANESB is the number of lanes from NODEA to NODEB. Similarly FREESPDA and CAPACITYA represent the free flow speed and hourly capacity in the NODEB to NODEA direction.

TRANSIMS uses text strings to define the FUNCTCLASS and VEHICLE fields. The functional class options include: FREEWAY, XPRESSWAY, PRIARTER, SECARTER, FRONTAGE, COLLECTOR, LOCAL, RAMP, ZONECONN, OTHER, WALKWAY, BIKEWAY, BUSWAY, LIGHTRAIL, HEAVYRAIL, and FERRY. Refer to the TRANSIMS documentation for a description of each functional class.

The VEHICLE field is used to define the use permissions for the link. The use options include: WALK, AUTO, TRUCK, BICYCLE, TAXI, BUS, TROLLEY, STREETCAR, LIGHTRAIL, RAPIDRAIL, and REGIONRAIL. Multiple use options are defined by creating a VEHICLE string with "/" separators. For example, WALK/AUTO/TRUCK/BICYCLE/BUS defines a link that can be used for walk, auto, truck, bicycle, and bus travel options.

INPUT_ZONE_FILE

The input zone file key is required and specifies the full path and file name of the input zone file. The zone file is a comma, space, or tab delimited ASCII file. At a minimum, the header line should include four fields in any order: ZONE, EASTING, NORTHING, and AREATYPE. These fields correspond to the zone number, the x-coordinate and y-coordinate of the zone centroid, and the area type of the zone. The centroid coordinates should be in the same units as the input node coordinates. The area type information is used for generating signal warrants. It is also added to the Activity Location file. Area type should be between 1 and 8.

KEEP_NODE_LIST

The keep node list 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. The TransimsNet 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 list file even when they are not required for the highway simulation. A sample keep node list file is shown below.

```
15467
11634
11617
11625
11759
11601
11598
14489
11008
```

TURN_PROHIBITIONS

The turn prohibitions key is optional and specifies the full path and filename of the input turn prohibition file. The TransimsNet program does not generate lane connections for the node combinations specified in this file. A sample turn prohibition file is shown below.

```
8512   8520   8534
8512   8520   8536
8512   8523   8534
8512   8523   8536
8512   8534   8520
8512   8534   8523
8512   8536   8520
8512   8536   8523
8701   8705   9856
8705   8712   9864
```

TURN_FILE_FORMAT

The turn file format key defines the format of the turn prohibitions file. The two options are FROM-THRU-TO and AT-FROM-TO. The default value for this key is FROM-THRU-TO. The code instructs the program in how to interpret the three node numbers provided for each record in the turn prohibition file. In FROM-THRU-TO format the intersection node is the

second node in each record. In the AT-FROM-TO format the intersection node is the first node in each record.

LINK_NODE_EQUIVALENCE

The Link Node Equivalence file is optional and specifies the full path and file name of the output link-node equivalency file. A sample equivalency file generated by the TransimsNet program is shown below. This file displays the sequence of nodes associated with each link ID. If links in the input link file are combined by the network collapsing process, the Link-Node Equivalence file provided a mechanism for associating the original node sequence with the newly created TRANSIMS link.

```
1 = 1245 9089
2 = 1246 8761
3 = 1247 14945
4 = 1248 15418
7 = 1251 30125
18 = 8000 8056
19 = 8000 8716 8667
20 = 8045 8001 8011
22 = 8002 8041
23 = 8002 8048
24 = 8002 9883
25 = 8003 8004 8042
26 = 8003 8007
27 = 8003 9787 8048
29 = 8044 8005 8009
```

NEW_DIRECTORY

The new directory key is required. It specifies the path to the output TRANSIMS network directory.

NEW_NODE_TABLE

The new node table key is required and specifies the filename of the output TRANSIMS node file generated by the TransimsNet program. The full path and file name for the output node table is constructed by appending the value of this key to the value of the NEW_DIRECTORY key.

NEW_LINK_TABLE

The new link table key is required and specifies the filename of the output TRANSIMS link file generated by the TransimsNet program. The full path and file name for the output link table is constructed by appending the value of this key to the value of the NEW_DIRECTORY key.

NEW_ACTIVITY_LOCATION_TABLE

The new activity location table key is required and specifies the filename of the output TRANSIMS activity location file generated by the TransimsNet program. The full path and file name for the output activity location table is constructed by appending the value of this key to the value of the NEW_DIRECTORY key.

NEW_PARKING_TABLE

The new parking table key is required and specifies the filename of the output TRANSIMS parking file generated by the TransimsNet program. The full path and file name for the output parking table is constructed by appending the value of this key to the value of the NEW_DIRECTORY key.

NEW_PROCESS_LINK_TABLE

The new process link table key is required and specifies the filename of the output TRANSIMS process link file generated by the TransimsNet program. The full path and file name for the output process link table is constructed by appending the value of this key to the value of the NEW_DIRECTORY key.

NEW_POCKET_LANE_TABLE

The new pocket lane table key is required and specifies the filename of the output TRANSIMS pocket lane file generated by the TransimsNet program. The full path and file name for the output pocket lane table is constructed by appending the value of this key to the value of the NEW_DIRECTORY key.

NEW_POCKET_LANE_TABLE

The new pocket lane table key is required and specifies the filename of the output TRANSIMS pocket lane file generated by the TransimsNet program. The full path and file name for the output pocket lane table is constructed by appending the value of this key to the value of the NEW_DIRECTORY key.

NEW_LANE_CONNECTIVITY_TABLE

The new lane connectivity table key is required and specifies the filename of the output TRANSIMS lane connectivity file generated by the TransimsNet program. The full path and file name for the output lane connectivity table is constructed by appending the value of this key to the value of the NEW_DIRECTORY key.

NEW_UNSIGNALIZED_NODE_TABLE

The new unsignalized node table key is required and specifies the filename of the output sign warrants generated by the TransimsNet program. The full path and file name for the output sign file is constructed by appending the value of this key to the value of the NEW_DIRECTORY key. The sign warrants can be used as an input to the IntControl program to generate a complete TRANSIMS Unsignalized Node file. It can also be read as an Unsignalized Node file by the ArcNet program to create an ArcView Shape file for reviewing and editing the sign warrants prior to final conversion with the IntControl program. Note that this file only includes records where a sign is recommended. A complete TRANSIMS Unsignalized Node file also identifies intersection approaches that are not controlled. A sample sign warrants file is shown below.

```
NODE   INLINK SIGN   NOTES
8078   185    S      STOP SIGN
8082   194    S      STOP SIGN
8085   201    S      STOP SIGN
8085   202    S      STOP SIGN
```

```
8087   185    Y      YIELD SIGN
8094   201    S      STOP SIGN
8130   261    S      STOP SIGN
8155   310    Y      YIELD SIGN
8165   324    S      STOP SIGN
8166   328    S      STOP SIGN
8178   286    S      STOP SIGN
8178   352    Y      YIELD SIGN
8178   354    S      STOP SIGN
8179   353    S      STOP SIGN
8179   356    S      STOP SIGN
8183   364    S      STOP SIGN
```

NEW_SIGNALIZED_NODE_TABLE

The new signalized node table key is required and specifies the filename of the output signal warrants file generated by the TransimsNet program. The full path and file name for the output signal table is constructed by appending the value of this key to the value of the NEW_DIRECTORY key. The signal warrants generated by the TransimsNet program could be used as an input to the IntControl program to generate complete TRANSIMS Signalized Node, Timing Plan, Phasing Plan, Detector and Signal Coordinator files. It can also be read as a Signalized Node file by the ArcNet program to create an ArcView Shape file for reviewing and editing the signal warrants prior to final conversion with the IntControl program. A sample signal warrants file is shown below.

```
NODE  TYPE    PLAN    OFFSET  STARTTIME      COORDINATR     RING    ALGORITHM      NOTES
8002  A       0       0       ALL00:00       8002           S       B              SIGNAL WARRANT
8003  A       0       0       ALL00:00       8003           S       B              SIGNAL WARRANT
8006  A       0       0       ALL00:00       8006           S       B              SIGNAL WARRANT
8007  A       0       0       ALL00:00       8007           S       B              SIGNAL WARRANT
8009  A       0       0       ALL00:00       8009           S       B              SIGNAL WARRANT
8010  A       0       0       ALL00:00       8010           S       B              SIGNAL WARRANT
8011  A       0       0       ALL00:00       8011           S       B              SIGNAL WARRANT
8015  A       0       0       ALL00:00       8015           S       B              SIGNAL WARRANT
8016  A       0       0       ALL00:00       8016           S       B              SIGNAL WARRANT
8017  A       0       0       ALL00:00       8017           S       B              SIGNAL WARRANT
```

INPUT_COORDINATE_SYSTEM

The input coordinate system key is optional. If it is not provided, the coordinates should be in UTM meters, the link length should be in meters, and the speeds should be in meters per second.

If the key is provided it defines how the coordinate fields on the input node and zone files are translated into generic Latitude and Longitude values. The Output Coordinate System is then used to convert the Latitude and Longitude data to the units required by TRANSIMS.

The input coordinate command includes three parts separated by a comma. The first part is the coordinate system description. The options include UTM, STATEPLAN, and LATLONG. The second part identifies the code number within the coordinate system that relates to the local conversion parameters. For UTM coordinates these codes range from 1N to 23N. Stateplane coordinates are defined using four digit FIPS codes (e.g., Oregon North = 3601). A code is not needed for the Latitude/Longitude system. The third parameter defines the coordinate units. By default, UTM is in meters, Stateplane is in feet, and Latitude/Longitude is in degrees. The user

can override these assumptions using the following keywords: FEET, METERS, MILES, KILOMETERS, DEGREES, and MILLION_DEGREES.

For example, coordinates from the Oregon North Stateplane system defined in miles should be specified with the following key:

```
INPUT_COORDINATE_SYSTEM          STATEPLANE, 3601, MILES
```

INPUT_COORDINATE_ADJUSTMENT

The input coordinate adjustment enables the user to manipulate the coordinates before they are sent to the input coordinate conversion calculation. This key is optional. It is only needed if the coordinates are not in the units expected by the conversion algorithm.

The adjustment command includes four floating-point numbers separated by commas. The first two numbers are the X and Y offsets. The last two numbers are X and Y adjustment factors. The process adds the offset value to the coordinate and then applies the adjustment factor. In other words:

```
X = (X + X_offset) * X_factor
Y = (Y + Y_offset) * Y_factor
```

OUTPUT_COORDINATE_SYSTEM

The output coordinate system determines how the internal Latitude and Longitude values are converted into EASTING and NORTHING coordinates in the output node file. This key is required if an Input Coordinate System is defined. If the node data file is to be used as input to TRANSIMS, this key should define how the Latitude and Longitude values are converted to the local UTM coordinate system in meters.

The output coordinate command includes three parts separated by a comma. The first part is the coordinate system description. The options include UTM, STATEPLAN, and LATLONG. The second part identifies the code number within the coordinate system that relates to the local conversion parameters. For UTM coordinates these codes range from 1N to 23N. Stateplane coordinates are defined using four digit FIPS codes (e.g., Oregon North = 3601). A code is not needed for the Latitude/Longitude system. The third parameter defines the coordinate units. By default, UTM is in meters, Stateplane is in feet, and Latitude/Longitude is in degrees. The user can override these assumptions using the following keywords: FEET, METERS, MILES, KILOMETERS, DEGREES, and MILLION_DEGREES.

The output key corresponding to the Oregon input parameter discussed above would be:

```
OUTPUT_COORDINATE_SYSTEM         UTM, 10N, METERS
```

OUTPUT_COORDINATE_ADJUSTMENT

The output coordinate adjustment enables the user to manipulate the coordinates after they are returned from the output coordinate conversion calculation. This key is optional. It is only

needed if the output coordinates should be in units that are different from the conversion algorithm.

The adjustment command includes four floating-point numbers separated by commas. The first two numbers are the X and Y offsets. The last two numbers are X and Y adjustment factors. The process adds the offset value to the coordinate and then applies the adjustment factor. In other words:

```
EASTING = (X + X_offset) * X_factor
NORTHING = (Y + Y_offset) * Y_factor
```

POCKET_RANGE_FOR_FACILITY_#

This key is optional and when provided specifies the minimum and the maximum lengths (in meters) of the pocket lanes for different roadway facilities. The # at the end of facility keyword specifies the facility type. The number code associated with different roadway facility types is shown below

Number Code	Facility Type
1	Freeway
2	Expressway
3	Primary Arterial
4	Secondary Arterial
5	Frontage
6	Collector
7	Local
8	Ramps
9	Zone Connectors

The limits for the minimum and maximum values for the pocket lane lengths are 0 and 2000 meters respectively. The values are separated by a comma as shown in the examples below.

```
POCKET_RANGE_FOR_FACILITY_1      100, 400
POCKET_RANGE_FOR_FACILITY_2      60, 200
POCKET_RANGE_FOR_FACILITY_3      40, 100
POCKET_RANGE_FOR_FACILITY_4      30, 60
```

SIGNAL_WARRANT_FOR_AREA_TYPE_#

This key is optional and when provided specifies the combinations of the roadway facilities that warrant a traffic signal in a given area type. The # at the end of area type keyword specifies the area type code. For example the SIGNAL_WARRANT_FOR_AREA_TYPE_1 specifies the roadway facility combination that warrants a traffic signal in area type 1. Each intersection is assigned to the area type of the closest zone centroid.

The facility codes are limited to the following key words: PRINCIPAL, MAJOR, MINOR, COLLECTOR, LOCAL, and NONE. These keys correspond to the TRANSIMS functional class codes XPRESSWAY, PRIARTER, SECARTER, FRONTAGE/COLLECTOR, and LOCAL.

Freeways, ramps, zone connectors, and transit modes are not considered by the signal warrant algorithm.

The keys are in hierarchical order and should be interpreted as the minimum facility type where a signal is warranted. For example, MINOR means that all minor arterial and above (i.e., major and principal arterials) are included in the warrant. A warrant for a given area type is defined using two key words. The first value corresponds to the primary facility entering the intersection and the second value corresponds to the secondary facility entering the intersection. For example, "MAJOR, MINOR" means that all intersections that join a major arterial with a minor arterial will be assigned a traffic signal.

A typical set of signal warrants for area types that represent decreasing land-use densities (e.g., CBD thru rural) would look something like the example below.

```
SIGNAL_WARRANT_FOR_AREA_TYPE_1      COLLECTOR, LOCAL
SIGNAL_WARRANT_FOR_AREA_TYPE_2      COLLECTOR, COLLECTOR
SIGNAL_WARRANT_FOR_AREA_TYPE_3      MINOR, COLLECTOR
SIGNAL_WARRANT_FOR_AREA_TYPE_4      MINOR, MINOR
SIGNAL_WARRANT_FOR_AREA_TYPE_5      MAJOR, MINOR
SIGNAL_WARRANT_FOR_AREA_TYPE_6      MAJOR, MAJOR
SIGNAL_WARRANT_FOR_AREA_TYPE_7      PRINCIPAL, MAJOR
SIGNAL_WARRANT_FOR_AREA_TYPE_8      PRINCIPAL, PRINCIPAL
```

MAXIMUM_ACCESS_POINTS

This maximum access points key is required and specifies the maximum number of activity locations that can be generated on each side of a given link. The actual under of activity locations will depend on the link length. This value can range from 1 to 20. Note that activity locations are not created on freeways, ramps, zone connectors, "other", and transit only links.

MINIMUM_SPLIT_LENGTHS

This minimum split lengths key is required and specifies the minimum distance between activity locations assigned to a link. A minimum distance value specified in meters should be provided for each area type. For example, the following command:

```
MINIMUM_SPLIT_LENGTHS      100, 200, 300, 300, 300, 300, 300, 300
```

means that links in area type 1 will have activity locations every 100 meters, and links in area type 2 will have activity locations every 200 meters, etc.

MINIMUM_LINK_LENGTH

This minimum link length key is optional. This key defaults to 37.5 meters when the value is not provided. This key specifies the minimum length of the network links. When the network link length is less than this value the length is reset to this value. The value can range from 7.5 to 100 meters. Note that TRANSIMS requires a link to be at least as long as the longest vehicle that can use the link. Vehicle sizes are defined in the vehicle prototype file and are defined as multiples of TRANSIMS cells (7.5 meters).

MAXIMUM_LENGTH_TO_XY_RATIO

The maximum length to X-Y ratio key is optional. This key defaults to 1.2 when the value is not specified. This key is used to compare the link length in the input network file to the straight-line distance between the coordinates of the nodes at either end of the link. If the length is less than the coordinate-based length, the output length is set equal to the coordinate-based length. If the ratio between the length and the coordinate-based length is greater than the value specified by this key, the program reduces the link length to the ratio times the coordinate-based length. For example, if the length field indicates the link is 1.3 miles long and the straight-line distance between the nodes is 1.0 miles, and the maximum ratio is 1.2, the program will reset the link length to 1.2 miles.

Note that all of these calculations are made after the coordinate conversions are performed. It is important to note that the conversion process also adjusts the length data. The program assumes that the length units are compatible with the input coordinate units. In other words, if the input coordinates are in miles, the length is interpreted as miles, and if the output coordinates are in meters, the output length is in meters. The output printout file reports the number of links that were changed as a result of the distance checks.

INTERSECTION_SETBACK_DISTANCE

This intersection setback distance key is optional and specifies the setback distance from the center of intersection. The default value for this key is 0.0 meters. TRANSIMS reduces the link length by the setback distance at each end of the link when defining cells in the Microsimulator.

Algorithm Notes

The TransimsNet program synthesizes TRANSIMS node, link, activity locations, parking, process link, lane connectivity and pocket lane files from generic input node and link files. The software also generates signal and sign warrant files for input to the IntControl program.

The program using the ZONECONN functional class to identify external access links. A special external access link with a boundary parking and activity location is generated for each zone connector included in the input link file. The program assumes that the lower node number attached to a zone connector represents the external station zone number. Unless the user wants to include external access points for internal zone centroids, all zone connectors attached to internal zone centroids should be deleted from the input link file prior to running this program.

If the original network attached zone connectors to mid-block nodes, the input link file will include a significant number of mid-block nodes that are no longer needed. Since TRANSIMS has difficulty simulating short links, it is desirable to remove these extra nodes from the network. The TransimsNet program includes a collapsing algorithm to remove unnecessary nodes. This algorithm considers the number of links in to and out of a given node, the compass orientation of these links, and the functional class, lanes, and speeds of the links. The program will keep nodes that are in the keep node list file, represent a significant change in direction (i.e., a major shape point), or define a change in network attributes.

The intersection logic also examines the number of links in to and out of a given node and determines how to construct turn pockets, thru links, lane connectivity, and traffic controls. If the node includes freeway links, special algorithms are used to connect multiple freeways and/or ramps. Merge and diverge lanes are added to the right or left side of a freeway depending on the relative orientation of the links. Arterial intersections examine the relative orientation or each movement and the functional class of each link to determine when and where to include turn pockets and signals or signs. In general, if an approach has opposing traffic, a turn pocket is added to accommodate the movement. The signal warrants are determined based on the number of legs and the functional class by area type signal warrant parameters.

Parking lots, activity locations, and process links are added to links using the split length and access point parameters. The link must also permit auto traffic to be assigned parking lot access points. Activity locations are defined on each side of the link at the spacing specified by the split length. A parking lot and process links are included for each activity location. The zone number and area type of the closest zone centroid will be included as user fields on the activity location file. If trip table disaggregation or population synthesis and activity generation are to be performed, the zone number assigned to an activity location should probably be refined. The coordinates on the activity location file can be used to load the data into a GIS software package. The GIS software can be used to locate each activity location more accurately within zone boundary polygons and update the zone value. The ArcNet and TAZUpdate utilities are provided to assist with this process.

Sample Printout

The printout file generated by TransimsNet will look something like the example below. It is an ASCII text file with a maximum of 95 characters per line and 65 lines per page. The file can be viewed or printed using a variety of text editors. For best results in a word processor, use a 10-point Courier font and 0.5 inch margins on all sides.

```
*****************************************
|                                       |
|        TransimsNet - Version 1.9      |
|    Copyright (c) 2005 by AECOM Consult   |
|         Fri Jan 28 09:44:35 2005      |
|                                       |
*****************************************


Control File = TransimsNet.ctl
Report File  = TransimsNet.prn
       Number of Keys Kept/Read = 37/37

Synthetic TRANSIMS Network

Input Network Directory = d:\Portland\Convert\Emme2

Input Node File = d:\Portland\Convert\Emme2\Node
Input Link File = d:\Portland\Convert\Emme2\Link
Input Zone File = d:\Portland\Convert\TransimsNet\Zone_File.txt

Keep Node List File = d:\Portland\Convert\TransimsNet\Keep_Nodes.txt

Turn Prohibition File = d:\Portland\Convert\TransimsNet\Turn.out
Turn File Format = AT-FROM-TO
```

```
Link Node Equivalence File = d:\Portland\Convert\TransimsNet\Link_Node.txt

Network Directory = d:\Portland\Convert\TransimsNet

TRANSIMS Node File = d:\Portland\Convert\TransimsNet\Node
TRANSIMS Link File = d:\Portland\Convert\TransimsNet\Link
TRANSIMS Activity Location File = d:\Portland\Convert\TransimsNet\Activity_Location
TRANSIMS Parking File = d:\Portland\Convert\TransimsNet\Parking
TRANSIMS Process Link File = d:\Portland\Convert\TransimsNet\Process_Link
TRANSIMS Pocket Lane File = d:\Portland\Convert\TransimsNet\Pocket_Lane
TRANSIMS Lane Connectivity File = d:\Portland\Convert\TransimsNet\Lane_Connectivity
TRANSIMS Unsignalized Node File = d:\Portland\Convert\TransimsNet\Sign_Warrants
TRANSIMS Signalized Node File = d:\Portland\Convert\TransimsNet\Signal_Warrants

Pocket Lengths for Facility Type 1 = 100 to 400 meters
Pocket Lengths for Facility Type 2 = 60 to 200 meters
Pocket Lengths for Facility Type 3 = 40 to 100 meters
Pocket Lengths for Facility Type 4 = 30 to 60 meters
Pocket Lengths for Facility Type 8 = 30 to 60 meters

Signal Warrant for Area Type 1 = COLLECTOR, LOCAL
Signal Warrant for Area Type 2 = COLLECTOR, COLLECTOR
Signal Warrant for Area Type 3 = MINOR, COLLECTOR
Signal Warrant for Area Type 4 = MINOR, MINOR
Signal Warrant for Area Type 5 = MAJOR, MINOR
Signal Warrant for Area Type 6 = MAJOR, MAJOR
Signal Warrant for Area Type 7 = PRINCIPAL, MAJOR
Signal Warrant for Area Type 8 = PRINCIPAL, PRINCIPAL

Maximum Number of Access Points Per Link = 4
Minimum Distance between Access Points in Area Type 1 = 100 meters
Minimum Distance between Access Points in Area Type 2 = 200 meters
Minimum Distance between Access Points in Area Type 3 = 200 meters
Minimum Distance between Access Points in Area Type 4 = 300 meters
Minimum Distance between Access Points in Area Type 5 = 300 meters
Minimum Distance between Access Points in Area Type 6 = 300 meters
Minimum Distance between Access Points in Area Type 7 = 300 meters
Minimum Distance between Access Points in Area Type 8 = 300 meters


Minimum Link Length = 37.5 meters
Maximum Length to XY Ratio = 1.20
Intersection Setback Distance = 0.0 meters


Number of Deleted Nodes = 2617

Number of Turn Prohibitions = 399

Number of Input Node Records = 8691
Number of Input Link Records = 10992

Number of Node Records = 5840
Number of Link Records = 8375
Number of Activity Location Records = 20838
Number of Parking Lot Records = 20838
Number of Process Link Records = 41676
Number of Pocket Lane Records = 3735
Number of Lane Connectivity Records = 39772
Number of Unsignalized Node Records = 2892
Number of Signalized Node Records = 1418

Number of External Connections = 17

Number of Short Links Increased in Length = 79

Number of Coordinate-Based Length Adjustments = 2846

Fri Jan 28 09:44:36 2005  Processing Complete.
```