Highway Network Conversion How-To

This document provides basic information on preparing and converting network data to the format required by TRANSIMS programs. Data typically available from a travel demand forecasting model or a geographic information system (GIS) are used to synthetically generate the detailed information needed for a regional microscopic traffic simulation.

Revision History

9/15/2006	Created by AECOM Consult, Inc.
2/12/2007	Edited by AECOM Consult, Inc.
4/1/2010	Edited by RSG, Inc.

Table of Contents

1.0 Assumptions and Prerequisites

This document assumes you have installed TRANSIMS Version 4.0.3 on a Windows or Linux computer system and that you understand the basic procedures and terminology for executing TRANSIMS programs.

The TRANSIMS software and documentation can be downloaded from http://sourceforge.net/projects/transims/files/ → software. Basic TRANSIMS procedures and terminology are addressed in the Installation and Testing How-To available in the documentation set.

You should have a basic understanding of traditional transportation planning networks and link-node network concepts. A rudimentary understanding of traffic signal systems is desirable, but not necessary.

Text files are used to store the input and output information. You need to be able to review and edit these files using a standard text editor (e.g., vi, Pico, WordPad) or other software that can manipulate tab-delimited files (e.g., Excel).

This document also describes how to generate ArcView shapefiles for displaying and editing network data in map format. Familiarity with software that can read and display ArcView shapefiles is desirable, but not necessary. A brief description about how this can be done using ArcGIS 9.1 is provided.

1.1 Download Network Data

This How-To document uses network information from Alexandria, Virginia to demonstrate the procedures, discuss outcomes, and describe concepts.

To download the Alexandria data to your computer or local area network, select http://sourceforge.net/projects/transims/files/ → test data → 4.0.3a Test Cases → Alexandria_4.0.3a.zip

You should create a directory with a name such as

c:\TRANSIMS\Alexandria	(Windows)
/home/TRANSIMS/Alexandria	(Linux)

and then unzip the downloaded file to that directory.

1.2 Network Data Source

The network was derived from an ArcGIS roadway centerline file developed by the City of Alexandria. To access additional information related to this network, visit the City of Alexandria's web page at

http://alexandriava.gov/GIS

Network data can also be derived from a variety of other sources. For example, traditional regional travel demand forecasting networks have been used as basic input to the TRANSIMS process. A number of utility programs are available to help convert data in other formats to the standard input network format described in this document.

2.0 TRANSIMS Network Overview

A TRANSIMS network is a collection of 20 or more files that define various aspects of the highway and transit facilities and operational characteristics. Although a given application may not require all of these files, it will require most of them in order to perform a detailed microscopic simulation. These files are typically identified within TRANSIMS using the following names:

Node
Zone
Link
Shape
Pocket Lane
Parking
Activity Location
Process Link
Lane Use
Lane Connectivity
Turn Prohibition
Unsignalized Node
Signalized Node
Timing Plan
Phasing Plan
Detector
Signal Coordinator
Transit Stop
Transit Fare
Transit Route
Transit Schedule
Transit Driver

Most transportation modelers are familiar with node, zone, and link file types from traditional travel demand forecasting software packages. Many of the other file types will likely be new to them. It is possible to create the data required for each file through a manual coding process, similar to the way regional modeling networks have been created in the past. For a large network, however, this can be a very time consuming and data-collection intensive process.

This document presents an alternate approach that starts from network data that are generally available from regional travel demand forecasting models or geographic information systems and applies general traffic engineering principles and operational assumptions to synthesize a complete set of TRANSIMS network files. The synthetic data can then be refined as necessary to more accurately replicate localized conditions.

3.0 How to Prepare Highway Network Input Files

The synthetic network development process starts with two basic types of information: network node and link files. The information can be enhanced using link shape points and zone centroid data. This section identifies how this information should be formatted for effective input to the network conversion utilities.

3.1 Node Data

The node file contains information about the location of each network node or intersection. The sample node file provided in the inputs directory is called "Input_Node.txt." This file includes three fields: NODE, X_COORD, and Y_COORD. TRANSIMS expects coordinates to be provided as UTM coordinates, in meters. These coordinates are rounded to the nearest decimeter (i.e., one decimal point, or the equivalent of four inches) within the TRANSIMS programs.

Node numbers must be unique positive integers. They do not need to be sequential or sorted. If zone centroids are included in the node file, they must be the lowest node numbers. If external stations are included, they should be the highest zone numbers.

3.2 Link Data

The link file contains information about roadways, walkways and transit guideways; in a two-directional coding convention. The beginning of the link is identified by an "A" node number, and the end of the link is identified by a "B" node number. The number of lanes, speeds, and other directional attributes are then identified as "AB" or "BA." TRANSIMS permits the user to code separate links in each direction if necessary, but this tends to complicate the coding and data processing. Links should not have the same "A" node and "B" node.

The sample link file provided in the inputs directory is called "Input_Link.txt." This file includes 11 fields: LINK, STREET, ANODE, BNODE, LENGTH, TYPE, LANES_AB, SPEED_AB, LANES_BA, SPEED_BA, and USE. All but the LINK and STREET fields are required. It is recommended that hourly capacities (CAP_AB, CAP_BA) be provided, if the data are available. If they are not, capacity will be estimated based on the facility type and the number of lanes. Capacities are typically used to allocate green time in the signal generation process and to estimate travel times based on traditional volume-delay functions prior to the first simulation. The Microsimulator does not use the capacity data to estimate network performance or throughput.

The link length is defined in meters, and the speed values are defined in meters per second. The facility type and vehicle use codes are provided as text strings. The options for facility type include:

```
FREEWAY, EXPRESSWAY, PRINCIPAL, MAJOR, MINOR, COLLECTOR, LOCAL,
FRONTAGE, RAMP, BRIDGE, EXTERNAL, XPRESSWAY, PRIARTER, SECARTER,
ZONECONN, OTHER, WALKWAY, BIKEWAY, BUSWAY, LIGHTRAIL, HEAVYRAIL,
FERRY
```

The vehicle use code is a combination of the following values, entered in any order and separated by slashes (e.g., CAR/TRUCK/BUS).

```
ANY, WALK, BIKE, CAR, TRUCK, BUS, RAIL, SOV, HOV2, HOV3, HOV4,
LIGHTTRUCK, HEAVYTRUCK, RESTRICTED, AUTO, BICYCLE, TAXI, TROLLEY,
STREETCAR, LIGHTRAIL, RAPIDRAIL, REGIONRAIL
```

Note that use codes represent the default access permissions for a link over the full simulation period. Access restrictions that only affect certain times of day or certain directions (e.g., peak period HOV restrictions) are identified in the lane use file. The meaning of each of these codes can be found in the TRANSIMS network documentation (Network Data Tables.pdf)

3.3 Zone Data

A zone file is optional, but recommended for most applications that cover a diverse geographic area or that need to be linked to zone-based trip tables. If a zone file is provided, every activity location generated in the conversion process will be associated with the closest zone centroid. This information is used to distribute zone-to-zone trips to appropriate activity locations.

The sample zone file provided in the inputs directory is called "Input_Zone.txt." This file includes four fields: ZONE, X_COORD, Y_COORD, and AREATYPE. The X and Y coordinates represent the zone centroid and are defined in UTM meters. AREATYPE is a user-defined integer code used by the synthetic network generation process to control signal warrants and intersection configurations. The density and type of development in a given area (i.e., area type) often influences signal density and turn pocket construction.

3.4 Shape Data

A shape file is optional. If provided, it includes any additional shape points in UTM meters between the "A" node and the "B" node of the link (in the A→B direction). The sample shape file provided in the inputs directory is called "Input_Shape.txt." This file contains two nested records for selected links. The first record has the link number and the number of shape points that follow. The second record includes the X and Y coordinates for each shape point. Link shapes are used in the synthetic network generation process to accurately identify relationships between entering and exiting links at an intersection. They are also useful in generating network maps. Shape points do not have a major impact on the TRANSIMS modeling process.

(Note: the TRANSIMS shape file is different from an ArcView Shapefile. An ArcView Shapefile representation of the network links can be converted to TRANSIMS link, node, and shape files using the GISNet program. The ArcNet program can be used to convert TRANSIMS link, node, and shape files to ArcView Shapefiles.)

4.0 How to Synthesize a TRANSIMS Highway Network

Using a set of basic input files, the first step in the synthetic network generation process can be executed. A utility program called **TransimsNet** is provided for this purpose. This section describes how to set up and run the **TransimsNet** program.

4.1 Overview of the TransimsNet Program

The **TransimsNet** program uses basic link and node information to synthesize the data fields and network files needed by TRANSIMS. The additional network files contain information about the number and location of turn pockets, the location of parking lots (i.e.,the start and end points of vehicle trips) and activity sites and the process links that connect the two, the lane connections at intersections, and traffic control warrants. The program also removes unnecessary nodes (those which don't represent intersections), updates the shape points, and converts external station zone connectors to roadways with the appropriate number of lanes and travel speeds.

In the TRANSIMS network design, trips start and end at activity locations. Unlike zones in traditional modeling systems, activity locations are directly related to links. Each side of a link (or block face) can contain one or more activity locations. Activity locations are accessed by walking on the associated link or utilizing a process link. Process links are used to connect activity locations directly to parking lots or transit stops. Parking lots are also associated with links, and all vehicle trips must start and end at a parking lot. The **TransimsNet** program automates the generation of activity location–process link–parking lot combinations at regular intervals on both sides of streets that permit auto access (and that are not freeways, ramps, or bridges).

"Pocket lane" refers to a turn lane entering an intersection, a merge or diverge lane on a freeway, and pull-out lanes at mid-block locations. The **TransimsNet** program adds pocket lanes based on the intersection configuration and length criteria defined by facility and area types. For example, a freeway merge lane is likely to be longer in suburban areas than in dense urban areas; freeway pocket lanes will be much longer than turn pockets on local streets; and turn pockets in the CBD are likely to be very different from turn pockets on suburban arterials.

Pocket lanes and regular lanes are then numbered and linked to lanes on other links using lane connectivity records. These connections differ if the movement represents a through movement, left turn, right turn, or U-turn. A given entry lane can be assigned one or more connections to exit links and lanes. A turn prohibition file can be provided to eliminate these connections at specific intersections.

The program then uses the intersection configuration and lane connectivity to assign signal or sign warrants. The modeler defines the signal warrants based on the primary facility types that come together at the intersection in a given area type. The resulting signal and sign warrants are typically reviewed and edited prior to executing the **IntControl** program to generate the full set of traffic control data files required by TRANSIMS.

4.2 The TransimsNet Control File

A sample control file for the **TransimsNet** program is provided in the control directory. The file "Alex.2005.Net.TransimsNet.ctl" is a text file that can be reviewed and edited using a standard text editor. The file records are listed below.

```
 TITLE                              Synthetic TRANSIMS Network
DEFAULT_FILE_FORMAT                TAB_DELIMITED
PROJECT_DIRECTORY                  ../

#---- Input Files ----

NET_DIRECTORY                      ../inputs
NET_NODE_TABLE                     Input_Node.txt
NET_ZONE_TABLE                     Input_Zone.txt
NET_LINK_TABLE                     Input_Link.txt
NET_SHAPE_TABLE                    Input_Shape.txt

NET_TURN_PROHIBITION_TABLE         Turn_Prohibition.txt
KEEP_NODE_LIST                     Keep_Node_List.txt

#---- Output Files ----

NEW_DIRECTORY                      ../network
NEW_NODE_TABLE                     Node
NEW_ZONE_TABLE                     Zone
NEW_LINK_TABLE                     Link
NEW_SHAPE_TABLE                    Shape
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

NEW_TURN_PROHIBITION_TABLE         Turn_Prohibition

LINK_NODE_EQUIVALENCE              Link_Node

#---- Parameters -----

POCKET_LENGTHS_FOR_FACILITY_1      100, 150, 150, 150, 300, 350, 400, 500
//---- meters ----
POCKET_LENGTHS_FOR_FACILITY_2      60, 75, 75, 75, 150, 200, 250, 300
POCKET_LENGTHS_FOR_FACILITY_3      40, 50, 50, 50, 80, 100, 125, 150
POCKET_LENGTHS_FOR_FACILITY_4      30, 40, 40, 40, 70, 80, 90, 100
POCKET_LENGTHS_FOR_FACILITY_5      30, 40, 40, 40, 70, 80, 90, 100
POCKET_LENGTHS_FOR_FACILITY_8      60, 75, 75, 75, 150, 200, 250, 300
POCKET_LENGTHS_FOR_FACILITY_9      60, 75, 75, 75, 150, 200, 250, 300
POCKET_LENGTHS_FOR_FACILITY_10     60, 75, 75, 75, 150, 200, 250, 300

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

STOP_WARRANT_FOR_AREA_TYPE_1       LOCAL
STOP_WARRANT_FOR_AREA_TYPE_2       COLLECTOR
STOP_WARRANT_FOR_AREA_TYPE_3       COLLECTOR
STOP_WARRANT_FOR_AREA_TYPE_4       COLLECTOR
STOP_WARRANT_FOR_AREA_TYPE_5       COLLECTOR

ACTIVITY_LOCATION_SIDE_OFFSET      15               //---- meters ----
MAXIMUM_ACCESS_POINTS              3
```

```
MINIMUM_SPLIT_LENGTHS   60, 60, 60, 60, 60, 60, 60, 60      //---- meters ----
MINIMUM_LINK_LENGTH                    7.5            //---- meters ----
MAXIMUM_LENGTH_TO_XY_RATIO             1.2
INTERSECTION_SETBACK_DISTANCE          7.5            //---- meters ----
FIRST_EXTERNAL_ZONE_NUMBER             70
COLLAPSE_NODES_FLAG                    TRUE
ADD_UTURN_TO_DEADEND_LINKS             YES
CREATE_NOTES_AND_NAME_FIELDS           YES
```

This application reads four input data files and creates ten TRANSIMS network files plus a link-node equivalence file. The `POCKET_LENGTHS_FOR_FACILITY_#` keys control pocket lane lengths (in meters) by facility type and area type. The # value relates to the following facility type numbering scheme:

```
1      FREEWAY
2      EXPRESSWAY
3      PRINCIPAL
4      MAJOR
5      MINOR
6      COLLECTOR
7      LOCAL
8      FRONTAGE
9      RAMP
10     BRIDGE
```

The number of length values relate to the area type codes defined in the input zone file. The first value corresponds to area type #1, the second to area type #2, etc.

The signal warrants also depend on the area type codes in the zone file. Each warrant is followed by up to four parameters. The first two parameters define the lowest facility type for the primary and secondary cross streets that warrant a signal. The next two parameters define the type of signal that is installed. For example, the key:

```
SIGNAL_WARRANT_FOR_AREA_TYPE_3   MINOR, COLLECTOR, ACTUATED, SINGLE_RING
```

states that intersections in area type #3 with a primary street classified as minor arterial or better and a secondary street classified as collector or better will be assigned a single-ring, demand-actuated signal (i.e., a signal with up to four phases controlled by a series of detectors that sense the presence of traffic in through lanes and left-turn lanes).

The `MAXIMUM_ACCESS_POINTS` and `MINIMUM_SPLIT_LENGTHS` keys control the number of activity locations and parking lots that are added to links in a given area type. The length parameter defines the minimum spacing between access points. The number of access points is a function of the link length, the split length, and the maximum number of points.

`The MINIMUM_LINK_LENGTH, MAXIMUM_LENGTH_TO_XY_RATIO`, and `INTERSECTION_SETBACK_DISTANCE` keys control and check the link lengths and intersection configurations. The length to XY ratio compares the length coded in the LENGTH field against the length calculated from the XY coordinates of the nodes and shape points. The parameter above permits the difference in the coordinate calculation and the length field to be up to 20 percent off before it resets the length field to the coordinate distance. If the length field is less than the coordinate distance it is reset to the coordinate distance. The length must also be longer than the minimum length value. This is to ensure that each link has the minimum number of cells required to accommodate the intersection setbacks and at least one vehicle.

The COLLAPSE_NODES_FLAG and FIRST_EXTERNAL_ZONE_NUMBER keys control how zone centroids, zone connectors, and extraneous nodes are handled. The **TransimsNet** program deletes zones and zone connectors that are not external traffic generators and nodes that do not mark a change in the link characteristics or an intersection.

TransimsNet will add U-turn connectivity to dead-end links and other intersections based on the values coded in the MAXIMUM_CONNECTION_ANGLE and ADD_UTURN_TO_DEADEND_LINKS keys. The values specified above limit U-turns to dead-end links.

4.3 Reviewing the Results

The TransimsNet program can be executed using a batch file included in the batch directory:

Alex.2005.Net.ConvertNet.bat (Windows)

Note, the Alex.2005.Net.ConvertNet.bat runs TransimsNet, IntControl, TransitNet, and ArcNet as part of a full batched network conversion process.

The printout file “Alex.2005.Net.TransimsNet.prn” will be created by the process in the control directory, along with the new data files stored in the network directory. The printout file will include warning messages about nodes that have entry links but no exit links and other issues. It is advisable to check each of these warnings to ensure that a coding error has not been made. For example, a one-way link might have been coded in the wrong direction. The relative approach angles could also be the cause of the warning message. In this case, the user could adjust the MAXIMUM_CONNECTION_ANGLE parameter to permit sharper connection angles or could manually correct the coding.

You will also want to review the intersections where the program recommends including signals and signs. Adjusting the area types assigned to various zones is one way to make area wide adjustments to the signal warrants. Site specific corrections can be made by adding or deleting records in the sign or signal files. Most of this review and editing can be accomplished relatively easily in map format. The ArcView shapefiles described in Section 6 are a convenient way to generate network maps.

5.0 How to Synthesize Intersection Traffic Controls

The second step in synthesizing a TRANSIMS network converts the signal and sign warrants to the full set of traffic control files required by the software. A utility program called **IntControl** is provided for this purpose. This section describes how to set up and run the **IntControl** program.

5.1 Overview of the IntControl Program

The **IntControl** program uses intersection configuration and lane connectivity data to populate the traffic control files required by TRANSIMS. The sign and signal warrant files identify the intersections and approach links that require traffic controls. The program validates and completes the unsignalized node file. It then builds signal controls for each signalized node. This includes the timing plan, phasing plan, detectors, and signal coordinator files. The number of phases and the green time allocated to each phase is based on the intersection configuration and the number of lanes or lane capacity assigned to each phase. Fixed timed and demand-actuated signals with one, two, or three rings and block groups can be synthesized based on the input signal warrant.

5.2 IntControl Control File

A sample control file for the **IntControl** program is provided in the control directory. The file "Alex.2005.Net.IntControl.ctl" is a text file that can be reviewed and edited using a standard text editor. The file records are listed below.

```
TITLE                                  Alexandria Intersection Controls
DEFAULT_FILE_FORMAT                    TAB_DELIMITED
PROJECT_DIRECTORY                      ../

#---- Input Files ----

NET_DIRECTORY                          ../network
NET_NODE_TABLE                         Node
NET_LINK_TABLE                         Link
NET_POCKET_LANE_TABLE                  Pocket_Lane
NET_ACTIVITY_LOCATION_TABLE            Activity_Location
NET_LANE_CONNECTIVITY_TABLE            Lane_Connectivity

INPUT_SIGNAL_FILE                      network/Signal_Warrants
INPUT_SIGN_FILE                        network/Sign_Warrants

#---- Output Files ----

NEW_DIRECTORY                          ../network
NEW_UNSIGNALIZED_NODE_TABLE            Unsignalized_Node
NEW_SIGNALIZED_NODE_TABLE              Signalized_Node
NEW_TIMING_PLAN_TABLE                  Timing_Plan
NEW_PHASING_PLAN_TABLE                 Phasing_Plan
NEW_DETECTOR_TABLE                     Detector
NEW_SIGNAL_COORDINATOR_TABLE           Signal_Coordinator

#---- Parameters ----

TIME_PERIOD_BREAKS                     6:00, 9:30, 16:00, 19:00

SIGNAL_CYCLE_LENGTH                    100             //---- seconds ----
MINIMUM_PHASE_TIME                     5               //---- seconds ----
YELLOW_PHASE_TIME                      3               //---- seconds ----
RED_CLEAR_PHASE_TIME                   1               //---- seconds ----
SIGNAL_DETECTOR_LENGTH                 30              //---- meters ----
POCKET_LANE_FACTOR                     0.5
GENERAL_GREEN_FACTOR                   0.5
EXTENDED_GREEN_FACTOR                  0.6
SIGNAL_SPLIT_METHOD                    CAPACITY
```

```
    MINIMUM_LANE_CAPACITY               500   //---- vehicles / lane / hour ----
    MAXIMUM_LANE_CAPACITY               1500  //---- vehicles / lane / hour ----
-
    FIXED_TIME_SIGNAL_FIELD             AREATYPE
    FIXED_TIME_FIELD_RANGE              1..2
    PRINT_WARNING_MESSAGES              TRUE

    ADD_NO_CONTROL_RECORDS              NO
    PRINT_SIGN_WARNINGS                 FALSE
    CREATE_NOTES_AND_NAME_FIELDS        YES
```

There are two basic groups of control parameters provided by the **IntControl** program. The first group defines how a timing plan is developed for each phase. This group includes `MINIMUM_PHASE_TIME`, `YELLOW_PHASE_TIME`, `RED_CLEAR_PHASE_TIME`, and `EXTENDED_GREEN_FACTOR`. These parameters define how the minimum, maximum, and extended green times are computed for fixed-timed and demand-actuated phases.

The second group determines how the total cycle length is allocated to different phases. This group includes `POCKET_LANE_FACTOR`, `GENERAL_GREEN_FACTOR`, `SIGNAL_SPLIT_METHOD`, `MINIMUM_LANE_CAPACITY`, and `MAXIMUM_LANE_CAPACITY`. These parameters define how the number of lanes or lane capacity assigned to one phase is balanced against similar values from other phases to allocate a share of the total cycle time to each phase.

IntControl generates isolated signal control plans. If signals need to be coordinated, an appropriate offset value must be added to each signal. This can be done manually to link two or three signals. The Progression program is typically used to coordinate the signal progression for a large group of signals.

5.3 Reviewing the Results

The **IntControl** program can be executed using a batch files included in the batch directory:

Alex.2005.Net.ConvertNet.bat (Windows)

Note, the Alex.2005.Net.ConvertNet.bat runs TransimsNet, IntControl, TransitNet, and ArcNet as part of a full batched network conversion process.

The printout file “Alex.2005.Net.IntControl.prn” should be created by the process in the control directory along with the new data files stored in the network directory. The printout file will contain a number of warning messages about missing traffic controls and problematic intersections. It is advisable to review these warnings to ensure that a coding error has not been made.

6.0 How to Create ArcView Shapefiles from a TRANSIMS Network

You should review and edit the synthetic network to confirm the coding logic and correct site specific issues. The best way to do this review is typically through a network map. The **ArcNet** utility converts the TRANSIMS network to a series of ArcView shapefiles that can be displayed and edited in ArcGIS or other mapping software. This section describes how to set up and run the **ArcNet** program.

6.1 Overview of the ArcNet Program

The **ArcNet** program can generate an ArcView shapefile representation of most of the TRANSIMS network files. Since the dBase file included in the ArcView shapefile is a complete copy of the data fields included in the corresponding TRANSIMS file, it is possible to edit or manipulate the ArcView shapefile data fields and use the edited data in TRANSIMS programs. The dBase file can be converted back to a standard text file using the **FileFormat** utility or other software such as Excel. Alternatively, TRANSIMS programs can be configured to read the dBase file directly.

The **ArcNet** program also provides ways of controlling the relative placement and offset of data items in map format. This includes generating parallel shapes for each direction of travel and user-controlled offset distanced for displaying parking lots, activity locations, and traffic control features. The utility can also convert coordinates from UTM meters to the state plane coordinate system or the latitude-longitude system for integration with other data sources.

6.2 The ArcNet Control File

A sample control file for the **ArcNet** program is provided in the control directory. The file "Alex.2005.Net.ArcNet_TransimsNet.ctl" is a text file that can be reviewed and edited using a standard text editor. The file records are listed below.

```
TITLE                                  Convert the Network to Shapefiles
DEFAULT_FILE_FORMAT                    TAB_DELIMITED
PROJECT_DIRECTORY                      ../

#---- Input Files ----

NET_DIRECTORY                          ../network
NET_NODE_TABLE                         Node
NET_LINK_TABLE                         Link
NET_SHAPE_TABLE                        Shape
NET_ACTIVITY_LOCATION_TABLE            Activity_Location
NET_PARKING_TABLE                      Parking
NET_PROCESS_LINK_TABLE                 Process_Link
NET_POCKET_LANE_TABLE                  Pocket_Lane
NET_LANE_CONNECTIVITY_TABLE            Lane_Connectivity

NET_SIGNALIZED_NODE_TABLE              Signal_Warrants
NET_UNSIGNALIZED_NODE_TABLE            Sign_Warrants

#---- Output Files ----

ARCVIEW_DIRECTORY                      ../network/arcview

#---- Parameters -----

DRAW_NETWORK_LANES                     YES
LANE_WIDTH                             3.5            //---- meters ----
LINK_DIRECTION_OFFSET                  0.0            //---- meters ----
ACTIVITY_LOCATION_SIDE_OFFSET          15             //---- meters ----
```

```
PARKING_SIDE_OFFSET                5              //---- meters ----

OUTPUT_COORDINATE_SYSTEM           UTM, 18N, METERS
```

ArcNet can be applied in a number of ways for a number of purposes. It does not need to convert all of the TRANSIMS network files at one time. Most conversions will need to include at least the link and node files to provide geographic references for displaying data objects that are defined based on link-offsets. Map representations of the initial input files and the output of TransimsNet are good ways of reviewing intermediate products and addressing warning and error messages generated along the way.

6.3 Visualizing the Results

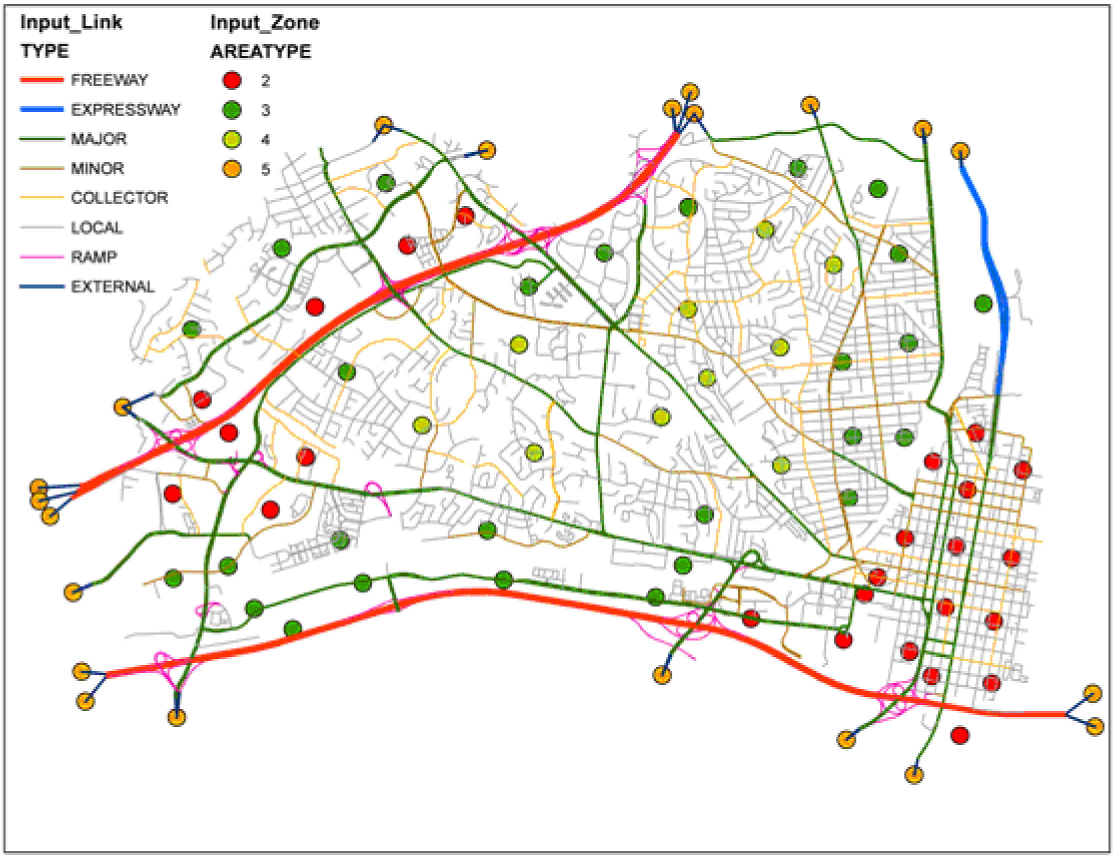
The **ArcNet** program can be executed using a batch file included in the batch directory:

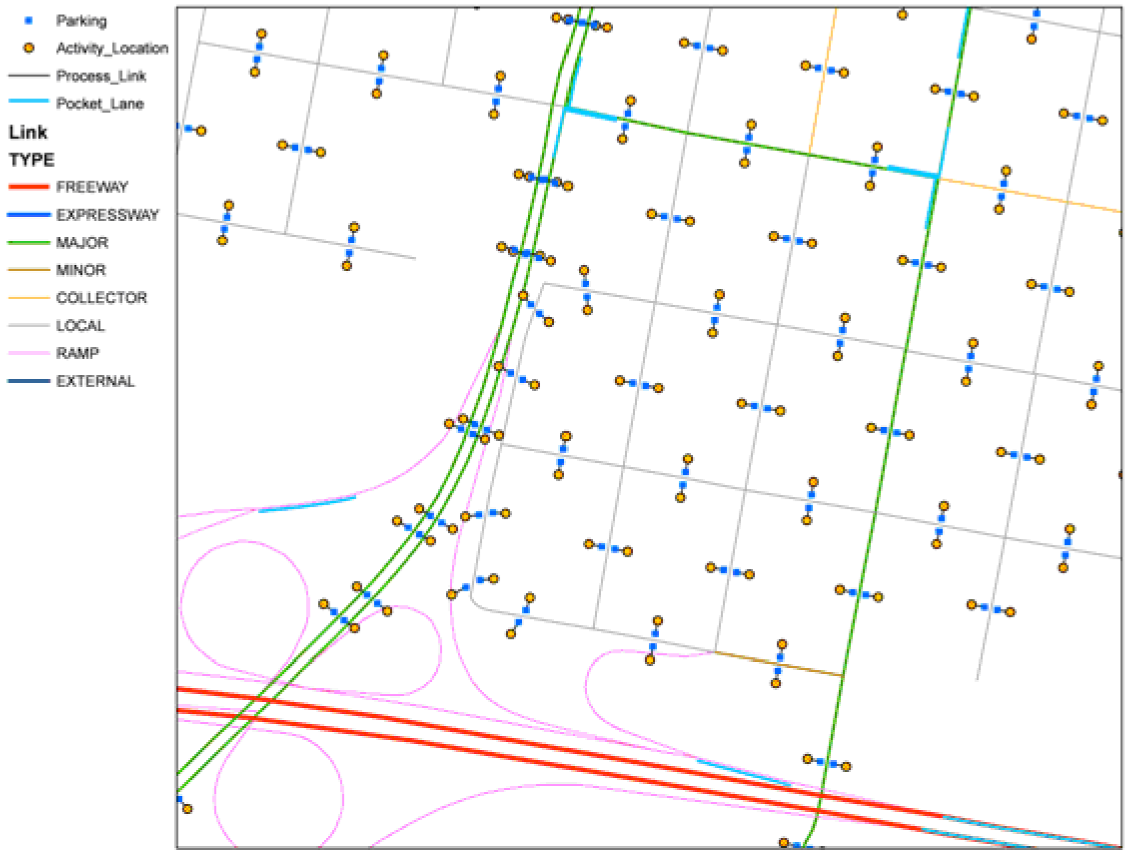
Alex.2005.Net.ConvertNet.bat (Windows)

Note, the Alex.2005.Net.ConvertNet.bat runs TransimsNet, IntControl, TransitNet, and ArcNet as part of a full batched network conversion process.

The printout file "Alex.2005.Net.ArcNet_TransimsNet.prn" should be created by the process in the control directory along with the new ArcView shapefiles in the arcview subdirectory of the network directory. The ArcView shapefile will have the same name as the input network files with extensions "shp," "shx," and "dbf." The *.shp and *.shx files are binary files that include the shape coordinates and index data. The *.dbf file is the dBase file that includes the data fields from the TRANSIMS data file.

A program such as ArcGIS/ArcMap can display these files in map format. The following graphics show some of the ways TRANSIMS portrays the Alexandria data. The first graph depicts the network using different colors for roads of different facility type. On top of the road network, colored circles represent the different area types defined at zone centroids. The second graph shows activity locations, parking lots, process links, and pocket lanes in downtown Alexandria.


Input_Link
TYPE
FREEWAY
EXPRESSWAY
MAJOR
MINOR
COLLECTOR
LOCAL
RAMP
EXTERNAL
Input_Zone
AREATYPE
2
3
4
5



7.0 Troubleshooting

Traditional travel demand forecasting models tend to use link capacity rather than lanes to define the performance of a roadway. This means that many of the networks used as input to TRANSIMS will not include accurate or consistent lane information. Because the number of lanes is much more important in TRANSIMS, these inaccuracies can cause problems or serious distortions in the way lane connectivity is implemented at intersections and link transitions.

Traditional models also do not pay much attention to the approach and departure angles at intersections or merge points. This often results in very sharp angles that have no impact on the connection within the traditional model, but are not connected within TRANSIMS. In such situations, it is often desirable to set the `MAXIMUM_CONNECTION_ANGLE` parameter in **TransimsNet** to 180 degrees to permit connections in all directions at a given intersection.

The way freeway interchanges are coded can have a significant impact on the performance of the Microsimulator. We recommend coding freeways with one-way directional links and accurate ramp connections. The user should also verify that the **TransimsNet** program placed merge and diverge lanes in the proper places.

The way arterials are coded can also cause problems for TRANSIMS. It is important that all of the links entering a signalized intersection come together at a single point. Directional coding with multiple nodes representing the intersection will create lane connectivity and traffic control problems within the Microsimulator.

8.0 Frequently Asked Questions

Are there utility programs to convert files from other software packages to a TRANSIMS input network?

> Utilities for EMME/2, TP+, and ArcView shapefiles have been written to translate data from these packages to TRANSIMS link-node format. TransCAD's interactive mode can also be used to generated the need input file.

Do we need to have every roadway in the network for TRANSIMS to work properly?

> TRANSIMS is designed to simulate networks that include all roadways, but it is not limited to these applications. Applications based on MPO-level networks (i.e., collectors and above) are more typical and have been successfully implemented thus far. In fact, it is often easier to work with less dense networks because this enables the user to focus on coding the significant facilities more accurately and to not get bogged down with the connections to all of the local streets.