ArcNet *(version 4.0)*

Revision History

1/8/2010	Edited by AECOM Consult, Inc.
4/20/2010	Edited by RSG, Inc.

The **ArcNet** program is used to:

1. Create ArcView shapefiles from TRANSIMS nodes, links, shape, lane-use, activity locations, parking, process links, pocket lanes, lane connectivity, turn prohibition, unsignalized nodes, signalized nodes, detector, transit routes, transit stop, transit driver, route header, and route nodes files.
2. Draw links and link-related attributes using individual lanes.
3. Draw transit routes using a different offset for each route.
4. Draw the network attributes associates with a specific time period.

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

ArcNet [-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:

-Q[uiet]	= execute without screen messages
-H[elp]	= show program syntax and control keys
-K[eyCheck]	= list unrecognized control file keys
-P[ause]	= pause before exiting
-N[oPause]	= never pause before exiting
-B[atch]	= execute in batch processing mode

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 **ArcNet** control file is shown below:

```
       TITLE                             Synthetic Network to ArcView Format
       ARCVIEW_DIRECTORY                 /u1/tdot/tdottech/projects/MPO3/ArcView
       NET_DIRECTORY                     /u1/tdot/tdottech/projects/MPO3/network
       NET_NODE_TABLE                    Node
       NET_LINK_TABLE                    Link
       NET_SHAPE_TABLE                   Shape
       NET_LANE_USE_TABLE                Lane_Use
       NET_ACTIVITY_LOCATION_TABLE       Activity_Location
       NET_PARKING_TABLE                 Parking
       NET_PROCESS_LINK_TABLE            Process_Link
       NET_POCKET_LANE_TABLE             Pocket_Lane
       NET_UNSIGNALIZED_NODE_TABLE       Unsignalized_Node
       NET_SIGNALIZED_NODE_TABLE         Signalized_Node
       NET_DETECTOR_TABLE                Detector
       NET_TRANSIT_STOP_TABLE            Transit_Stop
       NET_TRANSIT_ROUTE_TABLE           Transit_Route
       NET_TRANSIT_SCHEDULE_TABLE        Transit_Schedule
       NET_TRANSIT_DRIVER_TABLE          Transit_Driver
       INPUT_COORDINATE_SYSTEM           UTM, 10N, METERS
       INPUT_COORDINATE_ADJUSTMENT       0.0, 0.0, 1.0, 1.0
       OUTPUT_COORDINATE_SYSTEM          STATEPLANE, 3601, FEET
       OUTPUT_COORDINATE_ADJUSTMENT      0.0, 0.0, 1.0, 1.0
       LINK_DIRECTION_OFFSET             0.0
       POCKET_LANE_SIDE_OFFSET           2.0
       PARKING_SIDE_OFFSET               5.0
       ACTIVITY_LOCATION_SIDE_OFFSET     15.0
       UNSIGNALIZED_NODE_SIDE_OFFSET     10.0
       UNSIGNALIZED_NODE_SETBACK         25.0
       TRANSIT_STOP_SIDE_OFFSET          5.0
       TRANSIT_DIRECTION_OFFSET          0.0
       TRANSIT_TIME_PERIODS              9:00,15:00,18:00,24:00
```

This example generates the ArcView files from all the TRANSIMS network files. 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.

REPORT_FILE

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

REPORT_FLAG

The report flag key is optional. If it is specified as Yes or True, the report file or default printout file will be opened in "Append" mode rather than "Create" mode. This permits the user to consolidate the output of several programs into a signal report file.

MAX_WARNING_MESSAGES

When the program generates a warning message, a counter is incremented and the total number of warning messages is reported and a warning return coded (2) is set at the end of the execution. By default the program prints up to 100,000 warning messages to the print-out file. If more than 100,000 warning messages are sent, the program stops printing additional messages to the file or terminates the program with an error message based on the MAX_WARNING_EXIT_FLAG. This parameter enables the user to modify the default warning limit.

MAX_WARNING_EXIT_FLAG

If the maximum number of warning messages is exceeded, this flag directs the program in what to do. If the flag is TRUE (the default), the program is terminated with an error message about the warning messages. If the flag is FALSE, the program continues execution, but no additional warning messages are sent to the screen or written to the printout file. The warning message counter continues to count the messages and reports the total at the end of the execution.

PROJECT_DIRECTORY

The project directory key is not required. If it is specified, it is added to all non-network file names required by the program. If it is not specified, all non-network file names should fully specify the file path.

ARCVIEW_DIRECTORY

The ArcView directory key is required and specifies the path and name of the directory for all the output ArcView Shape files.

NET_DIRECTORY

The network directory key is not required. If it is specified, it is added to all network table names. If it is not specified, the network table names should fully specify the file path.

NET_NODE_TABLE

The node table key specifies the name of the TRANSIMS 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.

NET_ZONE_TABLE

The zone table key specifies the name of the TRANSIMS zone file within the network directory. The full path and file name for the zone table is constructed by appending the value of this key to the value of the NET_DIRECTORY key.

NET_LINK_TABLE

The link table key specifies the name of the TRANSIMS 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.

NET_SHAPE_TABLE

The shape table key is optional. If specifies the name of the TRANSIMS shape file within the network directory. The full path and file name for the shape table is constructed by appending the value of this key to the value of the NET_DIRECTORY key. If the shape table is provided, all links and link related offsets are drawn based on the shape of the link.

NET_POCKET_LANE_TABLE

The pocket lane table key is optional. It specifies the name of the TRANSIMS pocket lane file within the network directory. The full path and file name for the pocket lane table is constructed by appending the value of this key to the value of the NET_DIRECTORY key.

NET_LANE_USE_TABLE

The network lane use table key is optional. It specifies the name of the TRANSIMS lane-use file within the network directory. The full path and file name for the lane-use table is constructed by appending the value of this key to the value of the NET_DIRECTORY key.

NET_ACTIVITY_LOCATION_TABLE

The activity location table key is optional. It specifies the name of the TRANSIMS activity location file within the network directory. The full path and file name for the activity location table is constructed by appending the value of this key to the value of the NET_DIRECTORY key.

NET_PARKING_TABLE

The parking table key is optional. It specifies the name of the TRANSIMS parking file within the network directory. The full path and file name for the parking table is constructed by appending the value of this key to the value of the NET_DIRECTORY key.

NET_PROCESS_LINK_TABLE

The process link table key is optional. It specifies the name of the TRANSIMS process file within the network directory. The full path and file name for the process link table is constructed by appending the value of this key to the value of the NET_DIRECTORY key. If it is provided the NET_ACTIVITY_LOCATION_TABLE and NET_PARKING_TABLE are required.

NET_LANE_CONNECTIVITY_TABLE

The lane connectivity table key is optional. It specifies the name of the TRANSIMS lane connectivity file within the network directory. The full path and file name for the lane connectivity table is constructed by appending the value of this key to the value of the NET_DIRECTORY key.

NET_TURN_PROHIBITION_TABLE

The turn prohibition table key is optional. It specifies the name of the TRANSIMS turn prohibition file within the network directory. The full path and file name for the turn prohibition table is constructed by appending the value of this key to the value of the NET_DIRECTORY key.

NET_UNSIGNALIZED_NODE_TABLE

The unsignalized node table key is optional. It specifies the name of the TRANSIMS unsignalized node file within the network directory. The full path and file name for the unsignalized node table is constructed by appending the value of this key to the value of the NET_DIRECTORY key.

NET_SIGNALIZED_NODE_TABLE

The signalized node table key is optional. It specifies the name of the TRANSIMS signalized node file within the network directory. The full path and file name for the signalized node table is constructed by appending the value of this key to the value of the NET_DIRECTORY key.

NET_DETECTOR_TABLE

The network detector table key is optional. It specifies the name of the TRANSIMS detector file within the network directory. The full path and file name for the detector table is constructed by appending the value of this key to the value of the NET_DIRECTORY key.

NET_TRANSIT_STOP_TABLE

The transit stop table key is optional. It specifies the name of the TRANSIMS transit stop file within the network directory. The full path and file name for the transit stop table is constructed by appending the value of this key to the value of the NET_DIRECTORY key.

NET_TRANSIT_ROUTE_TABLE

The transit route table key is optional. It specifies the name of the TRANSIMS transit route file within the network directory. The full path and file name for the transit route table is constructed by appending the value of this key to the value of the NET_DIRECTORY key. If the key is provided, a NET_TRANSIT_STOP_TABLE is required.

NET_TRANSIT_SCHEDULE_TABLE

The transit schedule table key is optional. It specifies the name of the TRANSIMS transit schedule file within the network directory. The full path and file name for the transit schedule table is constructed by appending the value of this key to the value of the NET_DIRECTORY key. A NET_TRANSIT_STOP_TABLE and a NET_TRANSIT_ROUTE_TABLE are required to display this information. A separate ArcView file is not generated. The information is stored on the Arcview Route file.

NET_TRANSIT_DRIVER_TABLE

The transit driver table key is optional. It specifies the name of the TRANSIMS transit driver file within the network directory. The full path and file name for the transit driver table is constructed by appending the value of this key to the value of the NET_DIRECTORY key. If a NET_TRANSIT_ROUTE_TABLE is provided, this information will be used to draw the routes with link shapes. In this case a separate ArcView file is not generated. If the NET_TRANSIT_ROUTE_TABLE is not provided, an ArcView driver file is created. This file will display the links associated with the transit route, but does not attach the transit stops to the route or show the route starting and stopping at the first and last transit stops.

ROUTE_HEADER_FILE

The network route header key is optional. It specifies the name of the input route header file used by the TransitNet program to synthetically generate the TRANSIMS transit network. This file is created by the user or generated by the Emme2Route or TPPlusRoute programs. The full path and file name for the route header file is constructed by appending the value of this key to the value of the NET_DIRECTORY key. This key is only read if the transit route file is NULL.

ROUTE_NODES_FILE

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

SUBZONE_DATA_FILE

The subzone data key specifies the name of a TRANSIMS subzone data file. The full path and file name for the subzone data file is constructed by appending the value of this key to the value of the PROJECT_DIRECTORY key.

DRAW_NETWORK_LANES

This key controls the way that links are drawn. If the key is "false", links are drawn as a single centerline or as two parallel shapes if the link direction offset is non-zero. If the first character of the key is "0", "N", "n", "F", or "f", the key is interpreted as "false". Anything else is interpreted as "true". If this key is true, each thru traffic lane is drawn as a parallel shape and appropriate space is provided for the right and left side pocket lanes. The width between the lane lines is controlled by the lane width parameter. If the link is a one-way link, the relationship of the lane lines to the centerline is controlled by the center oneway links key.

LANE_WIDTH

The lane width parameter defines the space between lane lines on links and for lane connectivity at intersections. The default value is 1.0, which means that lane lines will be drawn one meter apart. Values between zero and 25.0 meters are permitted unless the draw network lanes key is true. In this case the value must be greater than 0.0.

CENTER_ONEWAY_LINKS

This key controls the way that links are drawn. If the key is "false", one-way links are drawn relative to a centerline on the left side of the direction of travel (i.e., like the centerline of a two-way street). If the first character of the key is "0", "N", "n", "F", or "f", the key is interpreted as "false". Anything else is interpreted as "true". If this key is true, the centerline is interpreted as the center of the one-way pavement. This affects how lane lines and link direction offsets are drawn. It has no affect on links drawn with a single centerline.

LINK_DIRECTION_OFFSET

If the draw network lanes key is true, this key is ignored. If it is false, the link direction offset determines how two way links are draw. The default value is 0.0, which means that only the link centerline is written to the ArcView Link file. If the value is greater than 0.0, it represents the number of meters to the right or left that a two way link is drawn. In this case, one shape and dBase records are saved in the ArcView Link file, but the shape will have two parts. Each part is drawn in the direction of travel. If the link is a one-way link, only one shape record is draw using the centerline of the roadway in the direction of travel.

POCKET_LANE_SIDE_OFFSET

If the draw network lanes key is true, this key is ignored. If it is false, the pocket lane side offset is used to display right side pocket lanes differently from left side pocket lanes. The default value is 2.0 meters. In this case the pocket lane is draw two meters to the right of the centerline of the link shape at the appropriate offset and length.

PARKING_SIDE_OFFSET

The parking side offset is used to display parking lots on the appropriate side of the street. The default value is 5.0 meters. In this case the parking lot is draw five meters to the right or left of the centerline of the link shape at the appropriate offset.

ACTIVITY_LOCATION_SIDE_OFFSET

The activity location side offset is used to display activity location on the appropriate side of the street. The default value is 15.0 meters. In this case the activity location is draw fifteen meters to the right or left of the centerline of the link shape at the appropriate offset.

UNSIGNALIZED_NODE_SIDE_OFFSET

The unsignalized node side offset is used to display signs on the right side of the street. The default value is 10.0 meters. In this case the sign is draw ten meters to the right of the centerline of the link shape at the offset specified by the UNSIGNALIZED_NODE_SETBACK key.

UNSIGNALIZED_NODE_SETBACK

The unsignalized node setback is used to display signs on the right side of the street. The default value is 25.0 meters. In this case the sign is draw twenty five meters from the end of the link to the right of the centerline of the link shape at the offset specified by the UNSIGNALIZED_NODE_SIDE_OFFSET key.

TRANSIT_STOP_SIDE_OFFSET

The transit stop side offset is used to display transit stops on the right side of the street. The default value is 5.0 meters. In this case the transit stop is draw five meters to the right of the centerline of the link shape at the appropriate offset.

TRANSIT_DIRECTION_OFFSET

The transit direction offset determines how transit routes on two way links are draw. The default value is 0.0, which means that the route is drawn on the centerline of the link. If the value is greater than 0.0, it represents the number of meters to the right of a two way link the transit route is drawn. If the link is a one-way link, the route is drawn on the centerline of the roadway in the direction of travel.

TRANSIT_TIME_PERIODS

The time periods parameter enables the user to specify the times of day that are considered for processing the route information. The number of runs starting in each time period are saved on the Transit Route file. This parameter is optional. If this key is not provided, no additional time period fields are added to the Transit Route file. If the key is provided, it is interpreted as a comma-delimited list of time period break points (i.e., the start time of the next time period). The time values can be provided as integer seconds (e.g., 10000, 20000), as decimal hours (e.g., 15.5, 19.5), or in standard clock format (e.g., 15:30, 24:00). Time periods will automatically start at 0:00 and end as 24:00.

TRANSIT_OVERLAP_FLAG

Transit routes will be drawn with a separate offset for each route on each link if this key is TRUE. The default is FALSE. The offset increments are based on the transit direction offset key.

VISUALIZER_GUIDEWAY_FILE

The visualizer guideway key specifies the name of a special shape point file used by the Balfour Output Visualizer to draw travel lanes. The full path and file name for the file is constructed by appending the value of this key to the value of the PROJECT_DIRECTORY key.

DRAW_ONEWAY_ARROWS

Direction of travel arrows will be drawn at the end of each link or lane if this key is TRUE. The default is FALSE. The size and shape of the arrow is controlled by the one-way arrow length and side offset keys. Since the arrow points are added to the link shape, a shape file that includes arrows can not be used in GISNet to create a TRANSIMS shape file.

ONEWAY_ARROW_LENGTH

This key specifies the length of the arrow symbol drawn at the end of one-way links in meters. The default is two times the lane width key. Arrows are only drawn if the draw one-way arrows key is TRUE.

ONEWAY_ARROW_SIDE_OFFSET

This key specifies the width of the arrow symbol drawn at the end of one-way links in meters. The default is one half the lane width key. Arrows are only drawn if the draw one-way arrows key is TRUE.

SELECT_TIME_PERIOD

This key specifies the hours of the day for which network attributes will be displayed. For example, 6:00..9:00 specifies that only the signals, lane-use, parking, tolls, turns, and transit routes that are active during the AM Peak period are drawn.

INPUT_COORDINATE_SYSTEM

The input coordinate system determines how the Easting and Northing data fields in the Node and Activity Location files are translated into generic Latitude and Longitude values. This key is optional. It is only needed if coordinate conversions are desired and then only if the input coordinates are not in degrees of Latitude and Longitude. By default, TRANSIMS data files store coordinate data in UTM coordinates in meters.

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 identified 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.

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. By default, TRANSIMS data files store coordinate data in meters that don’t require any adjustments.

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

OUTPUT_COORDINATE_SYSTEM

The output coordinate system determines how the internal Latitude and Longitude values are converted into X-Y coordinates in the output ArcView shape file. This key is optional. It is only needed if coordinate conversions are desired and then only if the output coordinates are not in degrees of Latitude and Longitude. If both the input coordinate system and the output coordinate system keys are NULL, no coordinate conversion takes place. The output coordinates will be the same as the input coordinates. In TRANSIMS, this means that the output ArcView shape file will be in UTM coordinates and 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 identified 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.

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:

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

OUTPUT_XYZ_SHAPES

By default, the ArcView shapefiles are generated with X and Y coordinates. If this key is specified as TRUE, the output shapefile will be constructed with X, Y, and Z coordinates. (The ArcView shapefile will also include M (measure) values and each M value will be equal to the corresponding Z value). If the TRANSIMS network does not include Z coordinates, the output Z coordinates will be zero.

OUTPUT_XYM_SHAPES

By default, the ArcView shapefiles are generated with X and Y coordinates. If this key is specified as TRUE, the output shapefile will be constructed with X, Y, and M coordinates where the M (measure) value will be equal to the Z coordinate in the TRANSIMS network. If this key is TRUE, it overrides the OUTPUT_XYZ_SHAPES key. This file structure generates a smaller shapefile than the XYZ structure.

Algorithm Notes

The **ArcNet** program generates the ArcView shape files for the TRANSIMS network files specified by the user. The program automatically creates three files in the output ArcView directory using the TRANSIMS network file specified. These are the ArcView shape file with a ".shp" extension, the ArcView index file with a ".shx" extension, and the ArcView data file with a ".dbf" extension. All three files are required for ArcView or ArcMap to read and display the data.

The conversion process does not change the values of any of the data fields included in the input data files. Coordinate conversion only impacts the ArcView shape files. The coordinate, length, and speed fields stored in the ArcView dBase file will have the same values as the input TRANSIMS file. This mean that the user can make manual changes to the data fields and then use the dBase file directly or convert it to a Tab delimited ASCII file (with Excel or some other conversion tool). Due to the multiple dependencies within the TRANSIMS network file structure, it is advisable to limit manual edits to stand-alone fields.

Note that all files need not be provided. The program will only convert the files specified by a non-NULL key. A number of the files are, however, dependent on data included in other files. In order to convert these files, the dependent files must be provided (and converted). The following table lists the dependences for each input file. Note that the Route Header and Route Node files will only be processed if the Transit Route key is NULL.

Conversion File	File Dependencies
Node	None
Zone	None
Link	Node (and Shape)
Shape	Node and Link
Pocket Lane	Node and Link (and Shape)
Lane Use	Node and Link (and Shape)
Activity Location	Node and Link (and Shape)
Parking	Node and Link (and Shape)
Process Link	Activity Location and Parking (and Transit Stop)
Lane Connectivity	Node and Link (and Shape)
Turn Prohibition	Node and Link (and Shape)
Unsignalized Node	Node and Link (and Shape)
Signalized Node	Node
Detector	Node and Link (and Shape)
Transit Stop	Node and Link (and Shape)
Transit Route	Transit Stop (and Transit Driver)
Transit Schedule	Transit Route
Transit Driver	Node and Link (and Shape)
Route Header	None
Route Node	Node and Link (and Shape)
Subzone Data	None

Sample Printout

The printout file generated by **ArcNet** 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.

```
********************************************
|                                          |
|         ArcNet - Version 4.0.22          |
|   Copyright (c) 2009 by AECOM Consult    |
|         Mon Apr 19 13:11:46 2010         |
|                                          |
********************************************

Control File = ArcNet.ctl
Report_File  = ArcNet.prn (Create)

Network Conversion Utility

Network Directory = d:\software\test\case3\network
Node File = d:\software\test\case3\network\Node.txt
Link File = d:\software\test\case3\network\Link.txt
Pocket Lane File = d:\software\test\case3\network\Pocket_Lane.txt
Parking File = d:\software\test\case3\network\Parking.txt
Activity Location File = d:\software\test\case3\network\Activity_Location.txt
Process Link File = d:\software\test\case3\network\Process_Link.txt
Unsignalized Node File = d:\software\test\case3\network\Unsignalized_Node.txt
Signalized Node File = d:\software\test\case3\network\Signalized_Node.txt
Transit Stop File = d:\software\test\case3\network\Transit_Stop.txt
Transit Route File = d:\software\test\case3\network\Transit_Route.txt
Transit Schedule File = d:\software\test\case3\network\Transit_Schedule.txt
Transit Driver File = d:\software\test\case3\network\Transit_Driver.txt

ArcView Output Directory = d:\software\test\case3\network\arcview

ArcView Node File = d:\software\test\case3\network\arcview\Node.shp

ArcView Link File = d:\software\test\case3\network\arcview\Link.shp
Link Direction Offset = 0.00 meters

ArcView Pocket Lane File = d:\software\test\case3\network\arcview\Pocket_Lane.shp
Pocket Lane Side Offset = 2.00 meters

ArcView Activity Location File = ...test\case3\network\arcview\Activity_Location.shp
Activity Location Side Offset = 15.00 meters

ArcView Parking File = d:\software\test\case3\network\arcview\Parking.shp
Parking Side Offset = 5.00 meters

ArcView Process Link File = d:\software\test\case3\network\arcview\Process_Link.shp

ArcView Unsignalized Node File = ...test\case3\network\arcview\Unsignalized_Node.shp
Unsignalized Node Side Offset = 10.00 meters
Unsignalized Node Setback = 25.00 meters

ArcView Signalized Node File = ...are\test\case3\network\arcview\Signalized_Node.shp

ArcView Transit Stop File = d:\software\test\case3\network\arcview\Transit_Stop.shp
Transit Stop Side Offset = 5.00 meters

ArcView Transit Route File = ...oftware\test\case3\network\arcview\Transit_Route.shp
Transit Time Period Breaks = 0:10, 0:15, 6:00, 9:00, 15:00, 18:00
Transit Direction Offset = 0.00 meters


Number of Node Records = 21

Number of Link Records = 24
Number of Directional Links = 48

Number of Pocket Lane Records = 72
Number of Parking Records = 96

Number of Activity Location Records = 96

Number of Transit Stop Records = 16

Number of Process Link Records = 224

Number of Unsignalized Node Records = 20
Number of Traffic Control Signs = 0
```