LocationData (version 4.0.12)

Revision History
11 April 2011 Created by Volpe Center

The LocationData program:
1. Creates or updates fields in the activity location file.
2. Assigns activity locations to a zone number based on the point-in-polygon equivalence to an ArcView zone boundary file.
3. Creates transit accessibility weights based on the number of transit runs within a specified distance of each activity location.
4. Creates trip distribution flags based on the use codes of the link attached to the activity location.
5. Creates trip distribution weights based on the location of subzone centroids and a subzone data field.
6. Copies data fields from a zone file based on a zone number in the activity location file.
7. Applies custom data processing scripts to manipulate and calculate fields in the activity location file based on inputs from several related files.
8. Accesses fields in an ArcView polygon boundary file based on a point-in-polygon match to the activity location coordinates.

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

```
LocationData [-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, 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.

Known Gaps in this Document

Only purposes 1, 2, 7, 8 are covered herein.

Control File Examples

Example 1 Assign Zones to Activity Locations (see Case Study 1, below)

```
    TITLE        Performs a Location Data Update to Activity Location File
    DEFAULT_FILE_FORMAT        TAB_DELIMITED

#---- Input Files ----
    NET_DIRECTORY        ./
    NET_NODE_TABLE        Node.txt
    NET_ZONE_TABLE       Zone.txt

    NET_LINK_TABLE        Link.txt
    NET_ACTIVITY_LOCATION_TABLE        Activity_Location.txt
    NET_PROCESS_LINK_TABLE        Process_Link.txt

    ZONE_FIELD_NAME    Taz
    ZONE_UPDATE_RANGE    ALL
    ZONE_BOUNDARY_POLYGON    Arcview/PolyZone.shp

#---- Output Files ----
    NEW_ACTIVITY_LOCATION_TABLE        Activity_Location2.txt
    LOCATIONDATA_REPORT_1    CHECK_ZONE_COVERAGE

#---- Parameters ----
    INPUT_COORDINATE_SYSTEM    UTM, 18N, METERS
    INPUT_COORDINATE_ADJUSTMENT    0.0,0.0,1.0,1.0
    OUTPUT_COORDINATE_SYSTEM    UTM, 18N, METERS
    OUTPUT_COORDINATE_ADJUSTMENT    0.0,0.0,1.0,1.0
```

Example 2 Add Fields to the Activity Location File

```
  TITLE        Performs a Location Data Update to Activity Location File

    DEFAULT_FILE_FORMAT        TAB_DELIMITED

#---- Input Files ----

    NET_DIRECTORY        ./

    NET_ACTIVITY_LOCATION_TABLE        Activity_Location.txt

    CONVERSION_SCRIPT  LocationData_Script.txt
    NEW_LOCATION_FIELD_1   ORIG_COEF, I, 2
    NEW_LOCATION_FIELD_2   DEST_COEF, I, 2

#---- Output Files ----
    NEW_ACTIVITY_LOCATION_TABLE        Activity_Location2.txt
    LOCATIONDATA_REPORT_1    CONVERSION_SCRIPT
```

Example 3 Activity Location Weights (see Case Study 2, below)

```
    TITLE        Performs a Location Data Update to Activity Location File
    DEFAULT_FILE_FORMAT        TAB_DELIMITED

#---- Input Files ----

    NET_DIRECTORY        ./
    NET_NODE_TABLE        Node.txt
    NET_ZONE_TABLE       Zone.txt

    NET_LINK_TABLE        Link.txt
    NET_ACTIVITY_LOCATION_TABLE        Activity_Location.txt
    NET_PROCESS_LINK_TABLE        Process_Link.txt

    ZONE_FIELD_NAME    Taz
    ZONE_UPDATE_RANGE    ALL
    ZONE_BOUNDARY_POLYGON    Arcview/PolyZone.shp

    BOUNDARY_POLYGON     Arcview/LandUses.shp

    CONVERSION_SCRIPT  LocationData_Script.txt
    NEW_LOCATION_FIELD_1   ORIG_COEF, I, 2
    NEW_LOCATION_FIELD_2   DEST_COEF, I, 2

#---- Output Files ----
    NEW_ACTIVITY_LOCATION_TABLE        Activity_Location3.txt
    LOCATIONDATA_REPORT_1    CONVERSION_SCRIPT
```

Case Study 1: Ensuring Zones are Covered by Activity Locations

An issue in any trip-based TRANSIMS implementation is ensuring that each internal zone is associated with at least one activity location. This case study illustrates how LocationData, along with a Python script developed at Arizona State University can be used to ensure this.

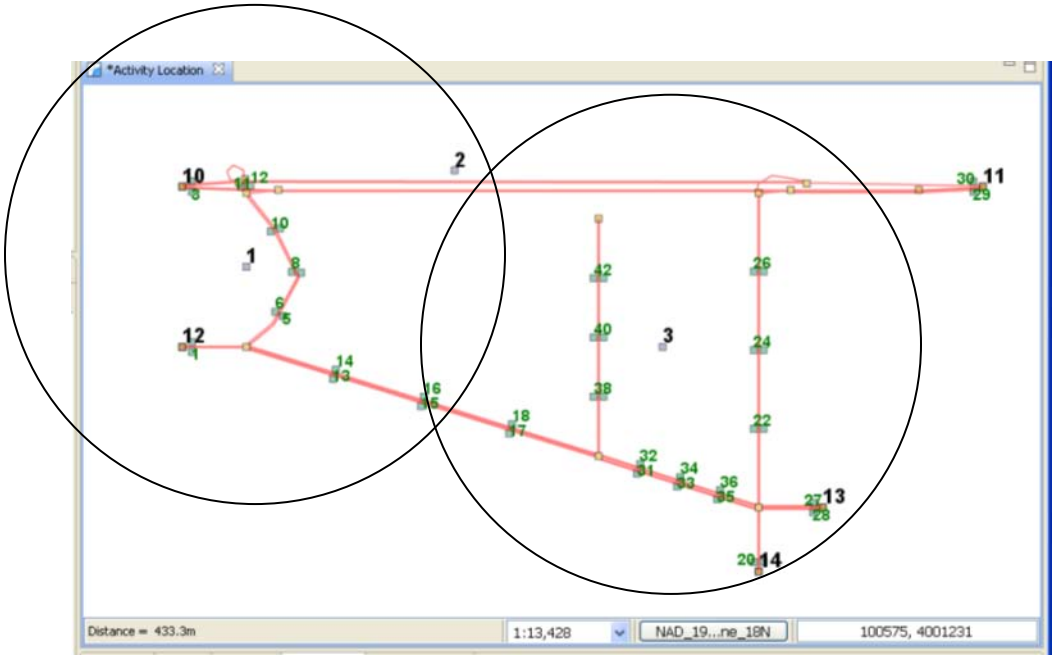

Figure 1 Example Network

Figure 1 illustrates an example network. It has three internal zones, with centroids numbered 1, 2, and 3. It has some external zones numbered 11 through 14. TransimsNet assigns each activity location (the pairs of points along each link) to its nearest zone. Activity locations 5 through 16 are assigned to zone 1, while internal activity locations 17 to 42 are assigned to zone 3. No activity locations are assigned to zone 2.

The LocationData program can be used to assign activity locations to zones based on the zone polygons (Figure 2).

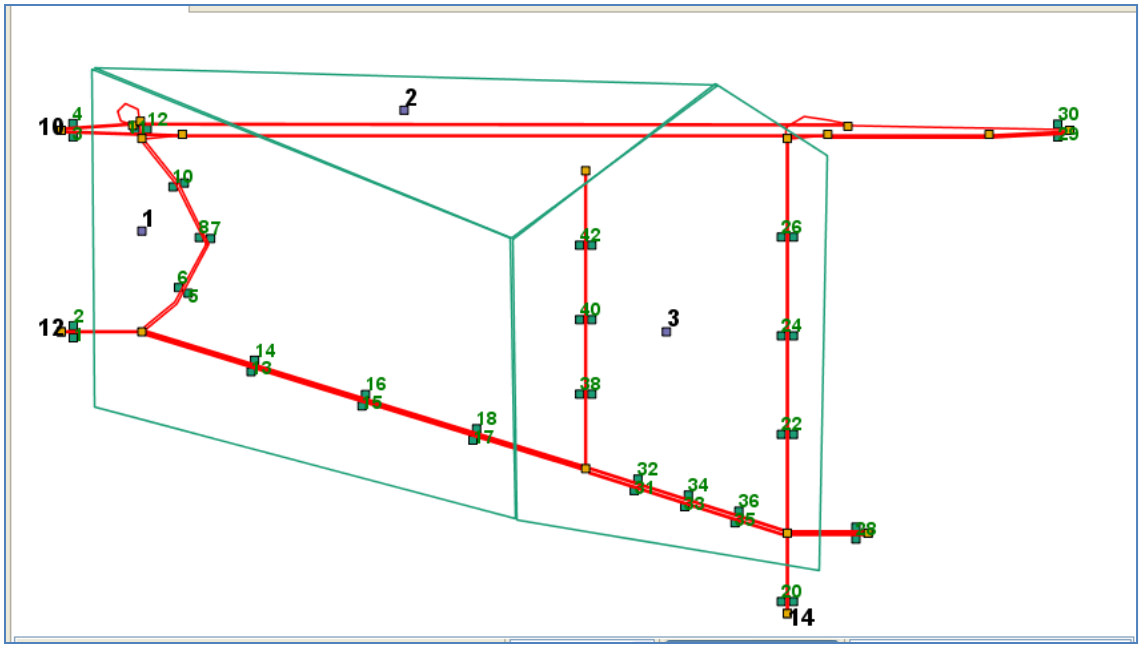

Figure 2 Example Network with Zone Polygons

Figure 2 illustrates three polygons, corresponding to the three internal zones. Each polygon is an element in a shape file that includes the integer field TAZ. The following parameters in the LocationData.ctl file cause these polygons to be associated with the proper zones.

```
ZONE_FIELD_NAME     Taz
ZONE_BOUNDARY_POLYGON     Arcview/PolyZone.shp
```

After running LocationData, with the control file given in Example 1, above, a new Activity_Location file is generated, similar to the old one, except that locations 17 and 18 are now associated with zone 1. No locations are associated with zone 2.

At this point, one could make manual adjustments to the activity_location file (for example, reassigning locations 41 and 42 to zone 2), or one could run a script that automates the reassignments. One example is the Python script *asu_reassign_zone.py*, developed at Arizona State University. This script identifies the zones that have no associated activity location, and then reassigns the nearest activity location to each of the zones. For example, the nearest activity locations to the Zone 2 centroid are locations 41 and 42, so the script will reassign these locations to zone 2. A copy of this script is available at http://transims.googlecode.com/svn/reassign_zone.py

Case Study 2: Different Activity Weights

This second case study implements the control file shown above in Example 3. It makes use of two shapefiles. The first is the shapefile of traffic analysis zones, used in the previous case study. The second is a shapefile of land use intensities (Figure 3). It contains three areas:

- The first, Zone_1_1, has medium intensity land use. It is a subset of TAZ 1.
- The second, Zone_Multi_1, has low intensity land use. It covers parts of all three TAZs.
- The third, Zone_3_1, has the highest intensity land use. It is a subset of TAZ 3.

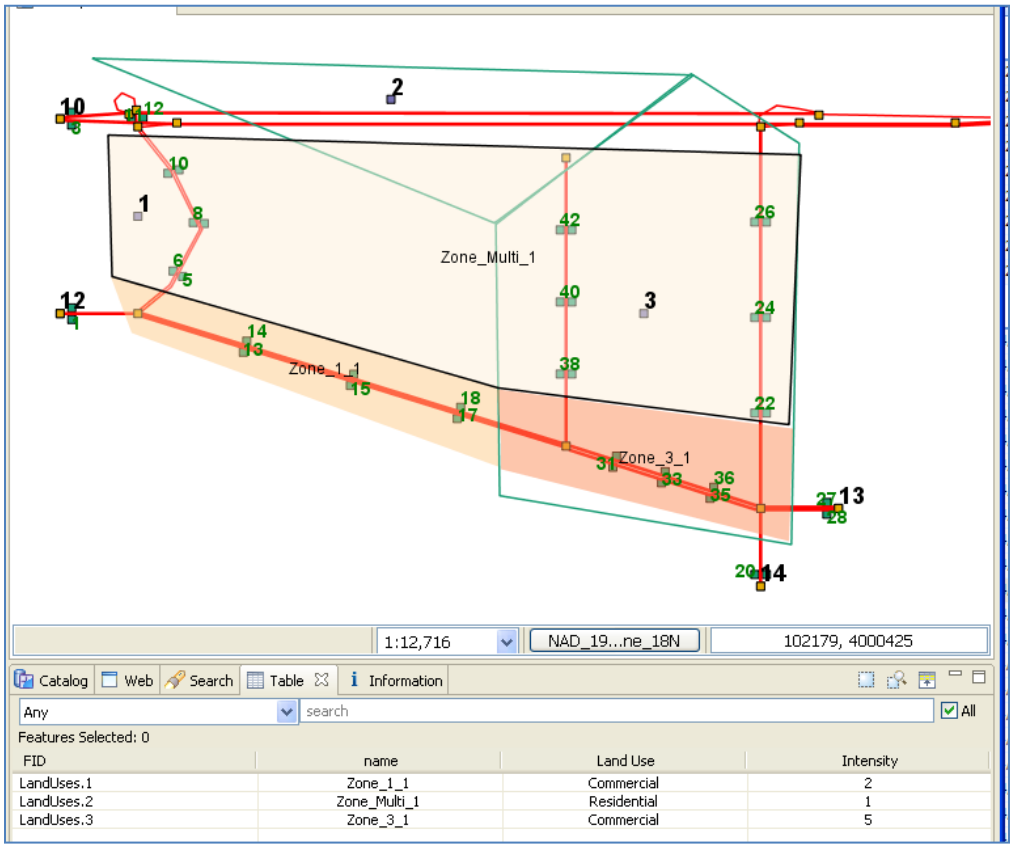

Figure 3 Land Uses

The conversion script that assigns these intensities, as well as dealing with the external origins and destinations, is as follows:

```
OUT.ORIG_COEF = POLYGON.Intensity
OUT.DEST_COEF = POLYGON.Intensity
IF (IN.NOTES == "External Destination") THEN
        OUT.ORIG_COEF = 0
        OUT.DEST_COEF = 1
ENDIF
IF (IN.NOTES == "External Origin") THEN
        OUT.ORIG_COEF = 1
        OUT.DEST_COEF = 0
ENDIF
RETURN (1)
```

The resultant origin and destination coefficients are as follows:

LOCATION	LINK	NODE	ZONE	ORIG_COEF	DEST_COEF	Note
1	1	22	12	1	0	External Origin
2	1	12	12	0	1	External Dest.
3	2	24	10	1	0	External Origin
4	3	10	10	0	1	External Dest.

LOCATION	LINK	NODE	ZONE	ORIG_COEF	DEST_COEF	Note
5	4	23	1	1	1	Zone_Multi_1
6	4	22	1	1	1	Zone_Multi_1
7	4	23	1	1	1	Zone_Multi_1
8	4	22	1	1	1	Zone_Multi_1
9	4	23	1	1	1	Zone_Multi_1
10	4	22	1	1	1	Zone_Multi_1
11	6	23	1	0	0	Not in LandUses
12	6	25	1	0	0	Not in LandUses
13	7	15	1	2	2	Zone_1_1
14	7	22	1	2	2	Zone_1_1
15	7	15	1	2	2	Zone_1_1
16	7	22	1	2	2	Zone_1_1
17	7	15	1	2	2	Zone_1_1
18	7	22	1	2	2	Zone_1_1
19	10	26	14	1	0	External Origin
20	10	14	14	0	1	External Dest.
21	11	27	3	1	1	Zone_Multi_1
22	11	26	3	1	1	Zone_Multi_1
23	11	27	3	1	1	Zone_Multi_1
24	11	26	3	1	1	Zone_Multi_1
25	11	27	3	1	1	Zone_Multi_1
26	11	26	3	1	1	Zone_Multi_1
27	14	26	13	1	0	External Origin
28	14	13	13	0	1	External Dest.
29	15	11	11	0	1	External Dest.
30	16	29	11	1	0	External Origin
31	19	26	3	5	5	Zone_3_1
32	19	15	3	5	5	Zone_3_1
33	19	26	3	5	5	Zone_3_1
34	19	15	3	5	5	Zone_3_1
35	19	26	3	5	5	Zone_3_1
36	19	15	3	5	5	Zone_3_1
37	22	17	3	1	1	Zone_Multi_1
38	22	15	3	1	1	Zone_Multi_1
39	22	17	3	1	1	Zone_Multi_1
40	22	15	3	1	1	Zone_Multi_1
41	22	17	3	1	1	Zone_Multi_1
42	22	15	3	1	1	Zone_Multi_1

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.

The keys recognized by the **LocationData** program are listed below. These keys can be defined in a variety of different ways to perform different tasks. The first 2 keys specify the input and output activity location tables. They are required; other keys are optional.

Required Keys

NET_ACTIVITY_LOCATION_TABLE

The activity location table key is required. 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 optional NET_DIRECTORY key.

NEW_ACTIVITY_LOCATION_TABLE

The activity location table key is required. It specifies the name of the new TRANSIMS activity location file within the new 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 optional NEW_DIRECTORY key.

Optional Keys

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. Its default is FALSE. 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 single 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 code (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.

DEFAULT_FILE_FORMAT

Default format for files other than network files. Default is VERSION3. Other possible values include BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, LANL and SQLITE3.

COPY_EXISTING_FIELDS

Indicates whether existing fields in the activity location file are copied to the new file. Defaults to FALSE. Possible values are {true/false/yes/no/1/0}. If existing fields are not copied, only the basic activity location fields are included (LOCATION, NODE, LINK, OFFSET, X_COORD, Y_COORD, and ZONE)

NEW_WALK_ACCESS_FIELD

This is a field name. Field names can be any unique combination of numbers, letters, and underscores. Note that ArcView or dBase field names are limited to 10 characters.

The new walk access field and maximum walk distance keys are used to calculate the relative accessibility of a given activity location to near-by transit stops. This calculation requires the link, node, process link, and transit network files. The number of runs at each stop and the distance between the stop and the activity location determine the accessibility weight.

MAX_WALK_DISTANCE

Defaults to 1000 meters, with a range of 10 to 3000 meters.

WALK_ACCESS_TIME_RANGE

Defaults to 0..24:00. The time ranges can be used to created multiple transit weights for different times of day. They are expressed as 0:00..6:00, 18:00..23:00, etc.

TIME_OF_DAY_FORMAT

Format for the time of day. Possible values are HOURS, SECONDS, 24_HOUR_CLOCK, 12_HOUR_CLOCK, with a default of 24_HOUR_CLOCK

NEW_USE_FLAG_FIELD, OR NEW_USE_FLAG_FIELD_#

This is a field name. The new use flag field and link use types keys are used to set the field value to 0 or 1 depending on how the link related to the activity location can be used. If the link use code permits any of the options included in the use types key, the field value is set to 1. One or more fields can be set using the _# key variations. This key is typically used to identify auto or truck access restrictions.

LINK_USE_FLAG_TYPES OR LINK_USE_FLAG_TYPES_#

Any combination of use codes separated by a slash (/) {ANY, WALK, BICYCLE, AUTO, TRUCK, BUS, RAIL, SOV, HOV2, HOV3, HOV4, LIGHTTRUCK, HEAVYTRUCK, RESTRICTED, CAR, BIKE, TAXI, TROLLEY, STREETCAR, LIGHTRAIL, RAPIDRAIL, REGIONRAIL}. Defaults to ANY.

NEW_SUBZONE_FIELD OR NEW_SUBZONE_FIELD_#

Each subzone key group consists of up to five keys. The new subzone field will include an activity location weight based on the proximity of the activity location to the subzone centroid found in the subzone file and the value of the subzone field. The average of the weights to the two best subzone centroids is saved to the new activity location field. This option is typically used to assign trip distribution weights to activity locations based on subzone population or employment data.

MAX_SUBZONE_DISTANCE OR MAX_SUBZONE_DISTANCE_#

10 to 10000 meters, with a default of 1000

SUBZONE_DATA_FILE OR SUBZONE_DATA_FILE_#

See above

SUBZONE_DATA_FORMAT OR SUBZONE_DATA_FORMAT_#

See above

SUBZONE_DATA_FIELD OR SUBZONE_DATA_FIELD_#

See above

NEW_LOCATION_FIELD OR NEW_LOCATION_FIELD_#

This key defines new fields to add to the activity location file. The values assigned to these fields are initialized to zero or blank and are typically set using a conversion script. The key can include up to three comma separated values. The first is the field name. This is followed by the field type and the field size. The type options include integer (default, I, INTEGER), floating point (R, REAL, D, DOUBLE), or string (S, STRING, C, CHARACTER). The default size is 10. Floating point fields can be defined with decimal points (e.g., 10.2). Two decimal points are assumed by default.

CONVERSION_SCRIPT

The conversion script key is a file name that includes a TRANSIMS User Program script. Any field in the input activity location file can be referenced using the file label IN (e.g., IN.*field*). Any field in the output activity location file (including all newly created fields) can be referenced using the field label OUT (e.g., OUT.*field*). All fields in each Data File are referenced using DATA and the key group number. For example, a field in DATA_FILE_2 is accessed as DATA2.*field*. An additional field called "AL_COUNT" is added to each data file and is set to the number of activity locations with the same join field.

An example of a script that sets up external stations fields (ORIG_COEF and DEST_COEF) is shown below:

```
#---- check for external stations ----
OUT.ORIG_COEF = 1
OUT.DEST_COEF = 1
IF (IN.NOTES == "External Destination") THEN
        OUT.ORIG_COEF = 0
        OUT.DEST_COEF = 1
ENDIF

IF (IN.NOTES == "External Origin") THEN
        OUT.ORIG_COEF = 1
        OUT.DEST_COEF = 0
ENDIF

RETURN (1)
```

DATA_FILE OR DATA_FILE_#

This is a filename. Each data file group consists of up to four keys. The two join fields must exist in their respective files. The appropriate data record from each data file is passed to the conversion script for each activity location. The program counts the number of activity locations with the same join field value and saves this value to the AL_COUNT field added to each data file. This field can be used to proportionally distribute data items to activity locations based on the number of activity locations associated with the data record. For example, population and employment data from traffic analysis zones can be distributed equally to each activity location within the zone by dividing the data by the value in the AL_COUNT field.

DATA_FORMAT OR DATA_FORMAT_#

Format for the Data File. Defaults to VERSION3. Options include VERSION3, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, LANL, SQLITE3

DATA_JOIN_FIELD OR DATA_JOIN_FIELD_#

This is the field name used for the data join

LOCATION_JOIN_FIELD OR LOCATION_JOIN_FIELD_#

This is the field name used for the location join.

ZONE_BOUNDARY_POLYGON

The zone boundary polygon is an arcview shapefile that contains polygons record defining the zones. If a zone boundary polygon key is provided, an ArcView shapefile with one boundary polygon for each zone is read. The zone field name key is the field name in the shapefile that defines the value posted in the ZONE field of the activity location file. The zone update range defines the range of zone values in the activity location file that will be eligible for update if they fall within a zone polygon. A warning message is generated for each zone in range that does not fall within a zone boundary. These activity location zone numbers are not changed.

The polygons can be created in any GIS. Here are some steps, taken from the TRANSIMS training course developed at Argonne National Labs, for creating a polygon using uDig:

- Zoom in to an area slightly bigger than the polygon to be created
- Choose “Layer” and then “Create Layer” from the menu bar
  - Under “geometry”, change the type from “LineString” to “Polygon”
  - Change the coordinate system from “WGS84” (or some other possible default) to the appropriate UTM zone of the TRANSIMS model (TRANSIMS does not convert projections automatically):
    - NAD83 / UTM zone 16N (EPSG:26916) for Chicago
    - NAD83 / UTM zone 18N (EPSG:26918) for Alexandria
- From the tool bar, select the “Create Polygon Tool” (make sure that the newly created layer stays selected when using the polygon tool)
  - Click once for each shape point of the polygon, and double-click to finish polygon creation
- In the layer list, right-click on the newly created polygon layer and choose “Rename”, then choose a meaningful name (e.g. “PolyZone”)
- In the layer list, right-click on the newly created polygon layer and choose “Export”, then “Layer Export”, then choose file name “PolyZone.shp”

ZONE_FIELD_NAME

The name of the field that contains the zone number, e.g., TAZ

ZONE_UPDATE_RANGE

The range of zone numbers are defined as a comma separated list (e.g., 1000..1200, 3000..3100). Default is ALL.

BOUNDARY_POLYGON OR BOUNDARY_POLYGON_#

If one or more boundary polygon keys are provided, ArcView shapefiles with one boundary polygon for each data record are read. The coordinates of the activity locations are used in a point-in polygon search to identify the best boundary record for each location. The data fields from the corresponding polygon are passed to the user program script for processing. The fields are referenced in the script using the file name Polygon or Polygon#.

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 Activity_Location 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 Activity_Location 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

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 is required. It 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_LINK_TABLE

The link table key is required. It 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_PROCESS_LINK_TABLE

The process link table key is required. 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. The process link data are used to assign vehicles to parking lots attached to activity locations.

NET_TRANSIT_STOP_TABLE

The transit stop table is optional. If the stop table is not provided, transit paths cannot be built. This key 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 is required if the transit stop file is provided. This key 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.

NET_TRANSIT_SCHEDULE_TABLE

The transit schedule table is required if the transit stop file is provided. This key 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.

NEW_DIRECTORY

The new 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_DEFAULT_FORMAT

Default format for network files. The default file format is set by DEFAULT_FILE_FORMAT. Other options include VERSION3, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, LANL..

NET_*_FORMAT

The file format key enables the user to specify the input format for an input network file. Replace the * with any of the network file types: node, link, etc. The default file format is set by NET_DEFAULT_FORMAT. Other options include VERSION3, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, LANL.

NEW_DEFAULT_FORMAT

Default format for new output (activity_location) files. The default file format is set by DEFAULT_FILE_FORMAT. Other options include VERSION3, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, LANL.

.

NEW_ACTIVITY_LOCATION_FORMAT

The file format key enables the user to specify the input format for a new activity_location file. The default file format is set by NEW_DEFAULT_FORMAT. The format options include VERSION3, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, LANL.

Sample Printouts

Sample printout files generated by the **LocationData** program are shown below. Each printout 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.

Example 1

```
*****************************************
|                                       |
|      LocationData - Version 4.0.12    |
|   Copyright (c) 2009 by AECOM Consult |
|         Wed Mar 23 09:26:10 2011      |
|                                       |
*****************************************

Control File = LocationData.ctl
Report_File  = LocationData.prn (Append)

Performs a Location Data Update to Activity Location File

Default File Format = TAB_DELIMITED

Network Directory = ./
Activity Location File = ./Activity_Location.txt

New Activity Location File = Activity_Location2.txt
Warning: No New Location Fields

Input Coordinate System = UTM, 18N, METERS
Input Coordinate Adjustment = 0.0,0.0,1.0,1.0
Output Coordinate System = UTM, 18N, METERS
Output Coordinate Adjustment = 0.0,0.0,1.0,1.0

Zone Boundary Polygon = Arcview/PolyZone.shp
Zone Field Name = Taz
Zone Update Range = ALL

LocationData Reports:  1. CHECK_ZONE_COVERAGE

Number of Zone Boundary Polygon Records = 3
Warning: Location 1 was not within a Zone Polygon
Warning: Location 2 was not within a Zone Polygon
Warning: Location 3 was not within a Zone Polygon
Warning: Location 4 was not within a Zone Polygon
Warning: Location 19 was not within a Zone Polygon
Warning: Location 20 was not within a Zone Polygon
Warning: Location 27 was not within a Zone Polygon
Warning: Location 28 was not within a Zone Polygon
Warning: Location 29 was not within a Zone Polygon
Warning: Location 30 was not within a Zone Polygon

Number of Activity Location Records Read = 42
Number of Activity Location Records Written = 42

Highest Traffic Analysis Zone = 14

No Activity Locations for Zones...
```

```
      2 4 5 6 7 8 9
Warning: 7 Zones have No Activity Locations

Wed Mar 23 09:26:10 2011 -- Process Complete with 12 Warnings (0:00:00)
```

Example 2

```
*****************************************
|                                       |
|      LocationData - Version 4.0.12    |
|   Copyright (c) 2009 by AECOM Consult |
|         Thu Apr 14 10:20:37 2011      |
|                                       |
*****************************************

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

Performs a Location Data Update to Activity Location File

Default File Format = TAB_DELIMITED

Network Directory = ./
Activity Location File = ./Activity_Location.txt

New Activity Location File = Activity_Location2.txt

New Location Field #1 = ORIG_COEF, I, 2
New Location Field #2 = DEST_COEF, I, 2

Conversion Script = LocationData_Script.txt

LocationData Reports:  1. CONVERSION_SCRIPT

Compiling Conversion Script

Conversion Script

      #---- check for external stations ----

      OUT.ORIG_COEF = 1
      OUT.DEST_COEF = 1
      IF (IN.NOTES == "External Destination") THEN
              OUT.ORIG_COEF = 0
              OUT.DEST_COEF = 1
      ENDIF

      IF (IN.NOTES == "External Origin") THEN
              OUT.ORIG_COEF = 1
              OUT.DEST_COEF = 0
      ENDIF

      RETURN (1)

Number of Activity Location Records Read = 42
Number of Activity Location Records Written = 42

Thu Apr 14 10:20:37 2011 -- Process Complete (0:00:00)
```

Example 3

```
*****************************************
|                                       |
|      LocationData - Version 4.0.12    |
|   Copyright (c) 2009 by AECOM Consult |
|         Thu Apr 14 15:09:41 2011      |
|                                       |
*****************************************

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

Performs a Location Data Update to Activity Location File

Default File Format = TAB_DELIMITED

Network Directory = ./
Activity Location File = ./Activity_Location.txt

New Activity Location File = Activity_Location3.txt

New Location Field #1 = ORIG_COEF, I, 2
New Location Field #2 = DEST_COEF, I, 2

Conversion Script = LocationData_Script.txt


Zone Boundary Polygon = Arcview/PolyZone.shp
Zone Field Name = Taz
Zone Update Range = ALL

Boundary Polygon File = Arcview/LandUses.shp

LocationData Reports:  1. CONVERSION_SCRIPT

Compiling Conversion Script

Conversion Script

      #---- check for external stations ----

      OUT.ORIG_COEF = POLYGON.Intensity
      OUT.DEST_COEF = POLYGON.Intensity
      IF (IN.NOTES == "External Destination") THEN
              OUT.ORIG_COEF = 0
              OUT.DEST_COEF = 1
      ENDIF

      IF (IN.NOTES == "External Origin") THEN
              OUT.ORIG_COEF = 1
              OUT.DEST_COEF = 0
      ENDIF

      RETURN (1)
```

```
Number of Zone Boundary Polygon Records = 3

Number of Boundary Polygon File Records = 3
Warning: Location 1 was not within a Zone Polygon
Warning: Location 2 was not within a Zone Polygon
Warning: Location 3 was not within a Zone Polygon
Warning: Location 4 was not within a Zone Polygon
Warning: Location 19 was not within a Zone Polygon
Warning: Location 20 was not within a Zone Polygon
Warning: Location 27 was not within a Zone Polygon
Warning: Location 28 was not within a Zone Polygon

Conversion Script

Warning: Location 29 was not within a Zone Polygon
Warning: Location 30 was not within a Zone Polygon

Number of Activity Location Records Read = 42
Number of Activity Location Records Written = 42

Thu Apr 14 15:09:41 2011 -- Process Complete with 10 Warnings (0:00:00)
```