

Location Data Version 5 – Program Reference

Version 5.0.3

Revision History:

March 2013 – Created by Volpe Center

The **LocationData V5** program is used to:

1. Create or update fields in the activity location file.
2. Assign activity locations to a zone number based on the point-in-polygon equivalence to an ArcView zone boundary file.
3. Create transit accessibility weights based on the number of transit runs within a specified distance of each activity location.
4. Create trip distribution flags based on the use codes of the link attached to the activity location.
5. Create trip distribution weights based on the location of subzone centroids and a subzone data field.
6. Copy data fields from a zone file based on a zone number in the activity location file.
7. Apply custom data processing scripts to manipulate and calculate fields in the activity location file based on inputs from several related files.
8. Access fields in an ArcView polygon boundary file based on a point-in-polygon match to the activity location coordinates.

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:

Optional Flags:

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

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

Version 5 Features

Control Key List

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

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

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

Configuration Keys

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

System File Keys

Control File Keys:	Req/Opt	Type	Default	I/O/P
NODE_FILE	Req	File		I
NODE_FORMAT	Opt	Text	TAB_DELIMITED	P
LINK_FILE	Req	File		I
LINK_FORMAT	Opt	Text	TAB_DELIMITED	P
LOCATION_FILE	Req	File		O
LOCATION_FORMAT	Opt	Text	TAB_DELIMITED	P
NEW_LOCATION_FILE	Req	File		O
NEW_LOCATION_FORMAT	Opt	Text	TAB_DELIMITED	P
ZONE_FILE	Req	File		I
ZONE_FORMAT	Opt	Text	TAB_DELIMITED	P
SHAPE_FILE	Opt	File		I
SHAPE_FORMAT	Opt	Text	TAB_DELIMITED	P
ACCESS_FILE	Opt	File		I

ACCESS_FORMAT	Opt	Text	TAB_DELIMITED	P
TRANSIT_STOP_FILE				
TRANSIT_ROUTE_FILE				
TRANSIT_SCHEDULE_FILE				

File Service Keys

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

Projection, Smooth Data, Difference Data Keys

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

Control Keys

Control File Keys:	Req/Opt	Type	Default	I/O/P
COPY_EXISTING_FIELDS	Opt	Bool	FALSE	P
NEW_WALK_ACCESS_FIELD	Opt	Text		P
MAX_WALK_DISTANCE	Opt	Int.	1000 meters	P
WALK_ACCESS_TIME_RANGE	Opt	Text		P
NEW_USE_FLAG_FIELD_*	Opt	Text		P
LINK_USE_FLAG_TYPES_*	Opt	Text		P
SUBZONE_ZONE_FACTOR_FILE	Opt	File		I?
NEW_SUBZONE_FIELD_*	Opt	Text		P
MAX_SUBZONE_DISTANCE_*	Opt	Int.		P
SUBZONE_DATA_FILE_*	Opt	File		I
SUBZONE_DATA_FORMAT_*	Opt	Text		P
SUBZONE_DATA_FIELD_*	Opt	Text		P
SUBZONE_ZONE_FIELD_*	Opt	Text		P
NEW_LOCATION_FIELD_*	Opt	Text		P
CONVERSION_SCRIPT	Opt	File		I
DATA_FILE_*	Opt	File		I
DATA_FORMAT_*	Opt	Text		P
DATA_JOIN_FIELD_*	Opt	Text		P
LOCATION_JOIN_FIELD_*	Opt	Text		P
BOUNDARY_POLYGON_FILE_*	Opt	File		I
NEW_ZONE_LOCATION_MAP_FILE	Opt	File		O
MINIMUM_ZONE_LOCATIONS	Opt	Int.	4	P
ZONE_BOUNDARY_FILE	Opt	File		I
ZONE_FIELD_NAME	Opt	Text		P
ZONE_UPDATE_RANGE	Opt	List	ALL	P

Notes

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

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

Control Key Changes in LocationData Version 5

New Key NEW_ZONE_LOCATION_MAP_FILE

This is the name of a file that contains a list of zones along with nearby activity locations that are not currently assigned to the zones. The zones that are listed include those zones that are currently assigned to fewer activity locations than the number specified in MINIMUM_ZONE_LOCATIONS. For each such zone, one or more locations, near to but not currently assigned to the zone, are listed. An example appears below.

```
ZONE  LOCATIONS
2      48, 9
5      46
10     13, 56
11     57, 14
12     51, 1
13     29, 37
14     58, 41
```

New Key MINIMUM_ZONE_LOCATIONS

This key is an integer, with a default value of 4, and a valid range of 2 to 20. It indicates the minimum number of locations that should be associated with each zone. It is used with the NEW_ZONE_LOCATION_MAP_FILE key to suggest additional location assignment to those zones that are currently assigned to fewer than MINIMUM_ZONE_LOCATIONS.

Possibly Unused Keys

The following keys are inherited from the Execution Service, but do not appear to be used:

MODEL_START_TIME	Opt	Time	0:00	P
MODEL_END_TIME	Opt	Time	24:00:00	P
UNITS_OF_MEASURE	Opt	Text	METRIC	P
RANDOM_NUMBER_SEED	Opt	Int.	0	P
MAX_PROBLEM_COUNT	Opt	Int.	0	P
NUMBER_OF_THREADS	Opt	Int.	1	P
OUTPUT_XYZ_SHAPES	Opt	Bool	FALSE	P
OUTPUT_XYM_SHAPES	Opt	Bool	FALSE	P

Keys Needing Definition

The following keys need to be defined:

SUBZONE_ZONE_FACTOR_FILE

BOUNDARY_POLYGON -> BOUNDARY_POLYGON_FILE_*

Network File Name and Control Key Name Changes (TransimsNet 4.0 → 5.0)

Additional control key changes are present in TransimsNet 5.0 that are not described in this section. Changes have been made to select control key and file and parameter names in many instances as can be seen from the list below. Additionally, file structures have been modified in some instances. New conceptual constructs have been introduced to improve overall performance and ease of use and robustness of TRANSIMS 5.0 as well. Lastly, new data fields have been added within some input and output files (e.g., SubArea). Refer to the File Reference and Parameter Reference documents for additional details. Of particular importance to LocationData is the Location_File:

ACTIVITY_LOCATION_TABLE → LOCATION_FILE

The ACTIVITY_LOCATION_TABLE file control key (and associated file format key) from LocationData 4.0 is replaced by the LOCATION_FILE key in LocationData 5.0. Refer to the File Reference mentioned above for additional details on the programmatic changes made to the LOCATION_FILE control key in TRANSIMS 5.0. These changes also apply to the NEW_ACTIVITY_LOCATION_FILE control key and associated file output format control key.

Other specific examples include the following control key and file name changes (V4 → V5):

- NET_NODE_TABLE → NODE_FILE
- NET_ZONE_TABLE → ZONE_FILE
- NET_SHAPE_TABLE → SHAPE_FILE
- NET_LINK_TABLE → LINK_FILE
- NET_POCKET_LANE_TABLE → POCKET_FILE
- NET_LANE_USE_TABLE → LANE_USE_FILE
- NET_TOLL_TABLE → LANE_USE_FILE
- NET_LANE_CONNECTIVITY_TABLE → CONNECTION_FILE
- NET_TURN_PROHIBITION_TABLE → TURN_PENALTY_FILE
- NET_PARKING_TABLE → PARKING_FILE
- NET_ACTIVITY_LOCATION_TABLE → LOCATION_FILE
- NET_PROCESS_LINK_TABLE → ACCESS_FILE
- NET_UNSIGNALIZED_NODE_FILE → SIGN_FILE
- NET_SIGNALIZED_NODE_TABLE → SIGNAL_FILE

Examples

1. Using a TAZ shape file to assign locations to zones

This example reads a list of zones (Input_Zone.txt), and a zone shape file (TAZ.shp) where the id field contains the zone number. It generates a new location file with location-zone assignments as well as a location map file (the location map file is optional). The control file is as follows.

```
TITLE                               Tiny Example - Zones from Shape File

LINK_FILE                          ../network/Link.txt
NODE_FILE                          ../network/Node.txt
SHAPE_FILE                         ../network/Shape.txt
LOCATION_FILE                       ../network/Location.txt
ZONE_FILE                          ../input/Input_Zone.txt

NEW_LOCATION_FILE                  ../network/Location1.txt

ZONE_FIELD_NAME    id
ZONE_UPDATE_RANGE  ALL
ZONE_BOUNDARY_FILE ../input/TAZ.shp

NEW_ZONE_LOCATION_MAP_FILE    zonemap1_8.txt
MINIMUM_ZONE_LOCATIONS       8
```

Outputs include a .prn file, a new location file and a zonemap file.

```
*****
|                                     |
|      LocationData - Version 5.0.3   |
| Copyright 2012 by TRANSIMS Open-Source |
|      Fri Mar 15 14:04:48 2013       |
|                                     |
|                                     |
*****
```

```
Control File = LocationData1.ctl
Report File  = LocationData1.prn (Create)
```

Tiny Example - Zones from Shape File

```
Project Directory = ./
Default File Format = TAB_DELIMITED
Time of Day Format = HOUR_CLOCK
Model Start Time = 0:00
Model End Time = 27:00
Units of Measure = METRIC
Random Number Seed = 1363370688
Number of Threads = 1
```

```
Input System Network Files:
Node File = ../../network/Node.txt
Zone File = ../../input/Input_Zone.txt
Shape File = ../../network/Shape.txt
Link File = ../../network/Link.txt
Location File = ../../network/Location.txt
```

```
Output System Network Files:
New Location File = ../../network/Location1.txt
```

Notes And Name Fields = TRUE

Data Service Controls:

LocationData Control Keys:

Warning: No New Location Fields

New Zone Location Map File = ./zonemap1_8.txt

Minimum Zone Locations = 8

Zone Boundary File = ../../input/TAZ.shp

Zone Field Name = id

Zone Update Range = ALL

Number of Node File Records = 23

Number of Zone File Records = 10

Highest Zone Number = 14

Number of Shape File Records = 13

Number of Link Shape Records = 3

Number of Link File Records = 24

Number of Directional Links = 37

Number of Zone Boundary File Records = 5

Warning: Location 21 was not within a Zone Polygon

Warning: Location 51 was not within a Zone Polygon

Warning: Location 52 was not within a Zone Polygon

Warning: Location 53 was not within a Zone Polygon

Warning: Location 54 was not within a Zone Polygon

Warning: Location 55 was not within a Zone Polygon

Warning: Location 56 was not within a Zone Polygon

Warning: Location 57 was not within a Zone Polygon

Warning: Location 58 was not within a Zone Polygon

Warning: Location 59 was not within a Zone Polygon

Warning: Location 60 was not within a Zone Polygon

Number of Location File Records = 60

New Zone Location Map File contains 7 Zones and 18 Locations

Fri Mar 15 14:04:48 2013 -- Process Complete with 13 Warnings (0:00:00)

The zonemap file is as follows:

ZONE	LOCATIONS
2	32, 20, 9
5	50, 40, 9
10	13, 2
11	31
12	51, 8, 26
13	54, 38, 29
14	21, 58, 19

See Figure 1 and Table 1 (later in this document), for a map and a zone-location assignment.

2. Adding fields to the Location table

```
TITLE                                LocationData Tiny Example

LINK_FILE                           ../network/Link.txt
NODE_FILE                           ../network/Node.txt
SHAPE_FILE                           ../network/Shape.txt
LOCATION_FILE                           ../network/Location.txt
ZONE_FILE                           ../input/Input_Zone.txt

NEW_LOCATION_FILE                     ../network/Location2.txt
CONVERSION_SCRIPT                     ../input/LocationData_Script2.txt
NEW_LOCATION_FIELD_1                  ORIG_WGT, I, 2
NEW_LOCATION_FIELD_2                  DEST_WGT, I, 2

LOCATIONDATA_REPORT_1                  CONVERSION_SCRIPT
```

The script, LocationData_Script2.txt, is as follows:

```
NewLocation.Orig_Wgt = 1
NewLocation.Dest_Wgt = 1

if (Location.Notes == "External Origin") then
    NewLocation.Orig_Wgt = 1
    NewLocation.Dest_Wgt = 0
endif
if (Location.Notes == "External Destination") then
    NewLocation.Orig_Wgt = 0
    NewLocation.Dest_Wgt = 1
endif
return(1)
```

The output .prn file is as follows:

```
*****
|                                     |
|      LocationData - Version 5.0.3  |
|  Copyright 2012 by TRANSIMS Open-Source  |
|      Fri Mar 15 14:21:57 2013      |
|                                     |
*****
```

```
Control File = LocationData2.ct1
Report File  = LocationData2.prn (Create)
```

LocationData Tiny Example

```
Project Directory = ./
Default File Format = TAB_DELIMITED
Time of Day Format = HOUR_CLOCK
Model Start Time = 0:00
Model End Time = 27:00
Units of Measure = METRIC
Random Number Seed = 1363371717
Number of Threads = 1
```

Input System Network Files:


```
Node File = ../../network/Node.txt
Shape File = ../../network/Shape.txt
Link File = ../../network/Link.txt
Location File = ../../network/Location.txt

Output System Network Files:
New Location File = ../../network/Location2.txt

Notes And Name Fields = TRUE

Data Service Controls:

LocationData Control Keys:

New Location Field #1 = ORIG_WGT, I, 2
New Location Field #2 = DEST_WGT, I, 2

Conversion Script = ../../input/LocationData_Script2.txt

LocationData Reports:  1. CONVERSION_SCRIPT

Compiling Conversion Script

Conversion Script

NewLocation.Orig_Wgt = 1
NewLocation.Dest_Wgt = 1
if (Location.Notes == "External Origin") then
    NewLocation.Orig_Wgt = 1
    NewLocation.Dest_Wgt = 0
endif
if (Location.Notes == "External Destination") then
    NewLocation.Orig_Wgt = 0
    NewLocation.Dest_Wgt = 1
endif
return(1)

Number of Node File Records = 23

Number of Shape File Records = 13
Number of Link Shape Records = 3

Number of Link File Records = 24
Number of Directional Links = 37

Number of Location File Records = 60

Fri Mar 15 14:21:57 2013 -- Process Complete (0:00:00)
```

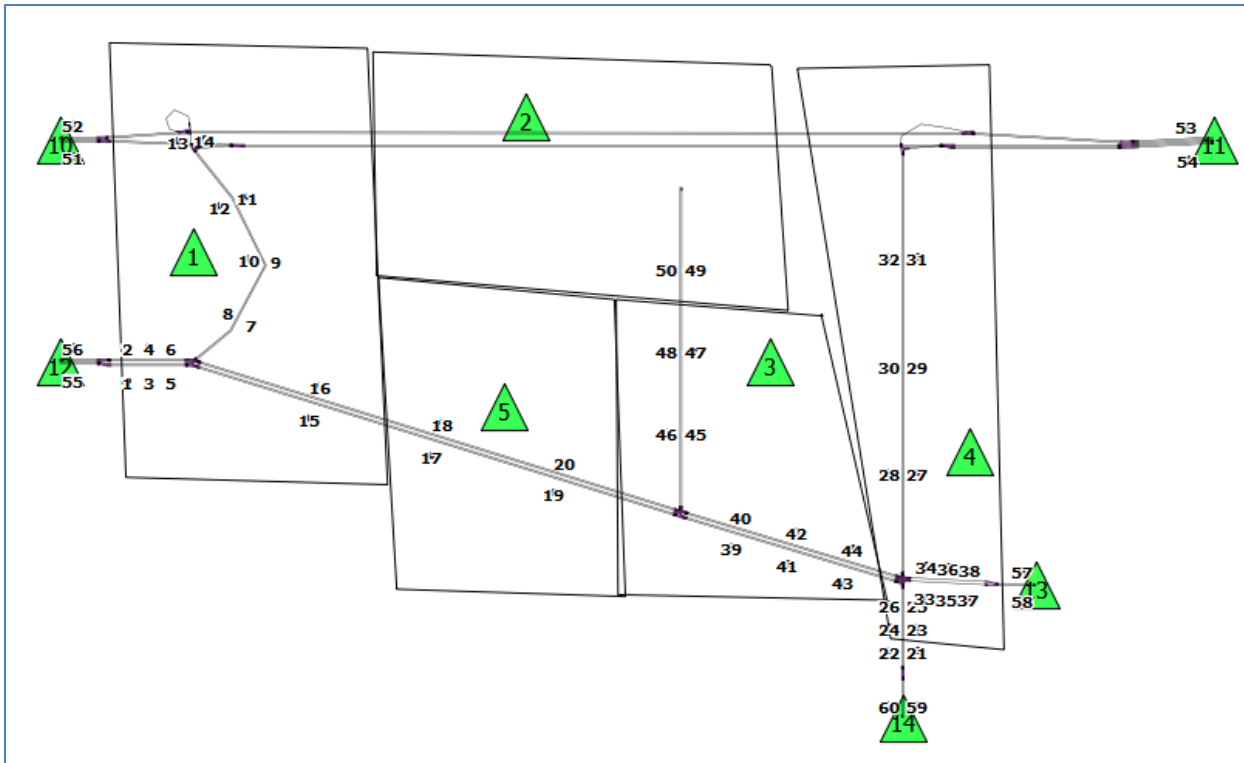


Figure 1 Zones and Locations for the Two Examples

Table 1 Locations and zone assignments for the two examples

	INPUT	After running example 1	After running example 2			
LOCATION	ZONE	ZONE	ZONE	ORIG_WGT	DEST_WGT	NOTES
1	1	1	1	1	1	Activity Location
2	1	1	1	1	1	Activity Location
3	1	1	1	1	1	Activity Location
4	1	1	1	1	1	Activity Location
5	1	1	1	1	1	Activity Location
6	1	1	1	1	1	Activity Location
7	1	1	1	1	1	Activity Location
8	1	1	1	1	1	Activity Location
9	1	1	1	1	1	Activity Location
10	1	1	1	1	1	Activity Location
11	1	1	1	1	1	Activity Location
12	1	1	1	1	1	Activity Location
13	1	1	1	1	1	Activity Location
14	1	1	1	1	1	Activity Location
15	5	1	5	1	1	Activity Location
16	5	1	5	1	1	Activity Location
17	5	5	5	1	1	Activity Location
18	5	5	5	1	1	Activity Location
19	5	5	5	1	1	Activity Location
20	5	5	5	1	1	Activity Location
21	4	4	4	1	1	Activity Location
22	4	4	4	1	1	Activity Location
23	4	4	4	1	1	Activity Location
24	4	4	4	1	1	Activity Location
25	4	4	4	1	1	Activity Location

	INPUT	After running example 1	After running example 2			
LOCATION	ZONE	ZONE	ZONE	ORIG_WGT	DEST_WGT	NOTES
26	4	4	4	1	1	Activity Location
27	4	4	4	1	1	Activity Location
28	4	4	4	1	1	Activity Location
29	4	4	4	1	1	Activity Location
30	4	4	4	1	1	Activity Location
31	3	4	3	1	1	Activity Location
32	3	4	3	1	1	Activity Location
33	4	4	4	1	1	Activity Location
34	4	4	4	1	1	Activity Location
35	4	4	4	1	1	Activity Location
36	4	4	4	1	1	Activity Location
37	4	4	4	1	1	Activity Location
38	4	4	4	1	1	Activity Location
39	3	3	3	1	1	Activity Location
40	3	3	3	1	1	Activity Location
41	3	3	3	1	1	Activity Location
42	3	3	3	1	1	Activity Location
43	4	3	4	1	1	Activity Location
44	4	3	4	1	1	Activity Location
45	3	3	3	1	1	Activity Location
46	3	3	3	1	1	Activity Location
47	3	3	3	1	1	Activity Location
48	3	3	3	1	1	Activity Location
49	3	2	3	1	1	Activity Location
50	3	2	3	1	1	Activity Location
51	10	10	10	1	0	External Origin
52	10	10	10	0	1	External Destination
53	11	11	11	1	0	External Origin
54	11	11	11	0	1	External Destination
55	12	12	12	1	0	External Origin
56	12	12	12	0	1	External Destination
57	13	13	13	1	0	External Origin
58	13	13	13	0	1	External Destination
59	14	14	14	1	0	External Origin
60	14	14	14	0	1	External Destination

3. Using Data files to allocate supplemental information to locations

Suppose you have a data file that contains population by zone, and you want to calculate population by location, and create a new field called “population” in the location file. The control file is as follows:

```

TITLE                               LocationData Tiny Example

LINK_FILE                          ../network/Link.txt
NODE_FILE                          ../network/Node.txt
SHAPE_FILE                         ../network/Shape.txt
LOCATION_FILE                       ../network/Location.txt
ZONE_FILE                          ../input/Input_Zone.txt

ZONE_FIELD_NAME    id
ZONE_UPDATE_RANGE  ALL
ZONE_BOUNDARY_FILE ../input/TAZ.shp

```

```

NEW_LOCATION_FILE          ../network/Location3.txt
CONVERSION_SCRIPT          ../input/LocationData_Script3.txt
NEW_LOCATION_FIELD_1      ORIG_WGT
NEW_LOCATION_FIELD_2      DEST_WGT
NEW_LOCATION_FIELD_3      POPULATION

DATA_FILE                  ../input/Zone_Population.txt
DATA_JOIN_FIELD            TAZ
LOCATION_JOIN_FIELD          ZONE

```

The conversion script (LocationData_Script3.txt) is

```

NewLocation.POPULATION = DATA.POPULATION / DATA.AL_COUNT
NewLocation.POPULATION = DATA.POPULATION / DATA.AL_COUNT
NewLocation.ORIG_WGT = 1
NewLocation.DEST_WGT = 1
IF (Location.NOTES == "External Destination") THEN
    NewLocation.ORIG_WGT = 0
    NewLocation.DEST_WGT = 1
ENDIF
IF (Location.NOTES == "External Origin") THEN
    NewLocation.ORIG_WGT = 1
    NewLocation.DEST_WGT = 0
ENDIF
RETURN (1)

```

AL_COUNT is a built-in TRANSIMS variable that refers to the number of activity locations in a zone.

The input Zone_Population file is

```

TAZ    POPULATION
1      2500
2      400
3      1300
4      1200
5      800

```

The LocationData program does three things:

- Assigns locations to zones in accordance with the TAZ.shp file (as in Example 1)
- Assigns ORIG_WGT and DEST_WGT to the external zones (as in Example 2)
- Adds a field called POPULATION to the output location table, and fills it in.

In this example, values for POPULATION range from 66 (zone 4 where a population of 1200 is split among 18 activity locations) to 200 (zone 2 where a population of 400 is split among 2 activity locations).

The important file and parameter names for this example are

DATA_FILE

The name of the file (typically a text file) that contains supplemental information for each zone. It is used by LocationData in conjunction with a conversion script to add supplemental information to each activity location.

DATA_FORMAT OR DATA_FORMAT_#

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

DATA_JOIN_FIELD

The name of the field in the DATA_FILE (above) that identifies the zone.

LOCATION_JOIN_FIELD

The name of the field in the Location file that identifies the zone.