ConvertTrips *(version 4.0)*

The ConvertTrips program converts traditional zone-to-zone trip tables to TRANSIMS trips by activity location and time of day. The ConvertTrips program is used to:

1. Convert multiple trip tables to Trip, Vehicle, Household, and Population files
2. Assign each trip table to a different trip purpose and vehicle type
3. Apply different diurnal distribution curves to each trip table or zone range.
4. Apply district-to-district correction factors to the input trips by trip purpose.
5. Append additional trips to existing Trip, Vehicle, Household, and Population files.
6. A period field can be included in the input trip tables to limit the time-of-day distribution of the trips.

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

ConvertTrips [-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.

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 ConvertTrips control file is shown below:

```
PROJECT_DIRECTORY             d:\software\test\case1\demand
NET_DIRECTORY                 d:\software\test\case1\network
NET_ACTIVITY_LOCATION_TABLE   Activity_Location.txt
NET_PROCESS_LINK_TABLE        Process_Link.txt

VEHICLE_TYPE_FILE             Vehicle_Type.txt

#---- Output ----

NEW_TRIP_FILE                 Trip.txt
NEW_POPULATION_FILE           Population.txt
NEW_HOUSEHOLD_FILE            Household.txt
NEW_VEHICLE_FILE              Vehicle.txt

STARTING_HOUSEHOLD_ID         1
STARTING_VEHICLE_ID           1
TIME_OF_DAY_FORMAT            SECONDS
ADDITIONAL_TRAVEL_TIME        180
RANDOM_NUMBER_SEED            0

#---- table 1 ----

TRIP_TABLE_FILE_1             TripTable1.txt
TRIP_TIME_FILE_1              TripTime1.txt
TIME_CONTROL_POINT_1          ORIGIN
ORIGIN_WEIGHT_FIELD_1         USER1
DESTINATION_WEIGHT_FIELD_1    USER2
TRIP_PURPOSE_CODE_1           1
TRAVEL_MODE_CODE_1            2
AVERAGE_TRAVEL_SPEED_1        15
VEHICLE_TYPE_1                1
VEHICLE_SUBTYPE_1             0

#---- table 2 ----

TRIP_TABLE_FILE_2             TripTable2.txt
TRIP_TIME_FILE_2              TripTime2.txt
TIME_CONTROL_POINT_2          ORIGIN
ORIGIN_WEIGHT_FIELD_2         USER1
DESTINATION_WEIGHT_FIELD_2    USER2
TRIP_PURPOSE_CODE_2           1
TRAVEL_MODE_CODE_2            2
AVERAGE_TRAVEL_SPEED_2        15
VEHICLE_TYPE_2                1
VEHICLE_SUBTYPE_2             0


#---- table 3 ----

TRIP_TABLE_FILE_3             TripTable3.txt
TRIP_TIME_FILE_3              TripTime.txt
TRIP_TIME_FORMAT_3            TAB_DELIMITED
TRIP_TIME_SCRIPT_3            Script.txt
TIME_PERIOD_RANGE_3           0:00..6:00, 9:00..15:00, 19:00..24:00
TIME_CONTROL_POINT_3          ORIGIN
ORIGIN_WEIGHT_FIELD_3         USER1
DESTINATION_WEIGHT_FIELD_3    USER2
TRIP_PURPOSE_CODE_3           1
TRAVEL_MODE_CODE_3            2
AVERAGE_TRAVEL_SPEED_3        15
VEHICLE_TYPE_3                1
VEHICLE_SUBTYPE_3             0

#---- table 4 ----

TRIP_TABLE_FILE_4             TripTable4.txt
TRIP_TIME_FILE_4              TripTime.txt
TRIP_TIME_FORMAT_4            TAB_DELIMITED
TRIP_TIME_SCRIPT_4            Script.txt
```

```
TIME_PERIOD_RANGE_4           6:00..9:00
TIME_CONTROL_POINT_4          ORIGIN
ORIGIN_WEIGHT_FIELD_4         USER1
DESTINATION_WEIGHT_FIELD_4    USER2
TRIP_PURPOSE_CODE_4           1
TRAVEL_MODE_CODE_4            2
AVERAGE_TRAVEL_SPEED_4        15
VEHICLE_TYPE_4                1
VEHICLE_SUBTYPE_4             0
```

The keys recognized by the ConvertTrips program are listed below. These keys can be defined in a variety of different ways to perform different tasks.

TITLE

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

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 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 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.

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_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 NET_DIRECTORY key. The activity location file is a primary input file for the ConvertTrips process.

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_ZONE_TABLE

The zone table key is optional. If provided, it is used to check the zone numbers in the activity location file against zones in the zone file. Warning messages about zones without activity locations will focus on zones in the zone file. 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 is optional. If provided, it is used to check the warning messages about activity locations that do not have parking lots. If the link the activity location is assigned to does not permit auto access, no warning message about a missing parking lot is generated. 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.

DEMAND_FILE_FORMAT

The demand file format key can be used to change the default output file format. The default format is VERSION3; a tab delimited file compatible with the TRANSIMS Version 3.x software. Other options include BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, and DBASE.

TRIP_FILE

The trip file key is appended to the PROJECT_DIRECTORY key to specify the file name for the input trip file copied to the output trip file by the program.

TRIP_FORMAT

The trip format key enables the user to specify the input format for the trip file. The default file format is set by DEMAND_FILE_FORMAT. The format options include VERSION3, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, and DBASE.

HOUSEHOLD_FILE

The household file key is appended to the PROJECT_DIRECTORY key to specify the file name for the input household file copied to the output household file by the program..

HOUSEHOLD_FORMAT

The household format key enables the user to specify the input format for the household file. The default file format is set by DEMAND_FILE_FORMAT. The format options include VERSION3, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, DBASE, and SQLITE3.

POPULATION_FILE

The population file key is appended to the PROJECT_DIRECTORY key to specify the file name for the input population file copied to the output population file by the program. .

POPULATION_FORMAT

The population format key enables the user to specify the input format for the population file. The default file format is set by DEMAND_FILE_FORMAT. The format options include VERSION3, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, DBASE, and SQLITE3.

VEHICLE_FILE

The vehicle file key is appended to the PROJECT_DIRECTORY key to specify the file name for the input vehicle file copied to the output vehicle file by the program.

VEHICLE_FORMAT

The vehicle format key enables the user to specify the input format for the vehicle file. The default file format is set by DEMAND_FILE_FORMAT. The format options include VERSION3, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, DBASE, and SQLITE3.

NEW_TRIP_FILE

The new trip file key is appended to the PROJECT_DIRECTORY key to specify the file name for the output trip file created by the program. The program generates one trip record for each trip in the input trip tables.

NEW_TRIP_FORMAT

The new trip format key enables the user to specify the output format for the trip file. The default file format is set by DEMAND_FILE_FORMAT. The format options include VERSION3, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, DBASE, and SQLITE3.

NEW_HOUSEHOLD_FILE

The new household file key is appended to the PROJECT_DIRECTORY key to specify the file name for the output household file created by the program. One household is generated for each trip in the input trip tables.

NEW_HOUSEHOLD_FORMAT

The new household format key enables the user to specify the output format for the household file. The default file format is set by DEMAND_FILE_FORMAT. The format options include VERSION3, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, DBASE, and SQLITE3.

NEW_POPULATION_FILE

The new population file key is appended to the PROJECT_DIRECTORY key to specify the file name for the output population file created by the program. One person is generated for each trip in the input trip tables.

NEW_POPULATION_FORMAT

The new population format key enables the user to specify the output format for the population file. The default file format is set by DEMAND_FILE_FORMAT. The format options include VERSION3, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, DBASE, and SQLITE3.

NEW_VEHICLE_FILE

The new vehicle file key is appended to the PROJECT_DIRECTORY key to specify the file name for the output vehicle file created by the program. The program generates one vehicle record for each vehicle trip in the input trip tables.

NEW_VEHICLE_FORMAT

The new vehicle format key enables the user to specify the output format for the vehicle file. The default file format is set by DEMAND_FILE_FORMAT. The format options include VERSION3, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, DBASE, and SQLITE3.

VEHICLE_TYPE_FILE

The vehicle type file is optional. The key value is appended to the PROJECT_DIRECTORY key to specify the file name for the input vehicle type file read by the program. The vehicle type file is used to validate the Vehicle Type and Subtype values entered for a given trip group.

STARTING_HOUSEHOLD_ID

This key specifies the integer number used to begin the household ID numbering. The default value is one or the highest household ID in the input household file. If the results of this application are to be combined with the results of the Activity Generator or other ConvertTrips applications, the user must define an appropriate offset to ensure unique Household IDs in the combined file.

STARTING_VEHICLE_ID

This key specifies the integer number used to begin the vehicle ID numbering. The default value is one or the highest vehicle ID in the input vehicle file. If the results of this application are to be combined with the results of the Activity Generator or other ConvertTrips applications, the user must define an appropriate offset to ensure unique Vehicle IDs in the combined file.

TIME_OF_DAY_FORMAT

The time of day format defines how the trip start and end times are written to the trip file. The default format will display values in 24 hour clock time. The format options include HOURS, SECONDS, 24_HOUR_CLOCK, and 12_HOUR_CLOCK.

RANDOM_NUMBER_SEED

This key specifies the random number seed used by the ConvertTrips program. If the key is not provided or the key value is zero, the random number seed will be set by the computer clock.

ADDITIONAL_TRAVEL_TIME

The ConvertTrips program estimates the travel time between the trip origin and destination using the user provided average speed by trip group and the straight line distance between the origin and destination activity locations. A constant value is added to this result to account for vehicle access, parking, and overall uncertainty in the travel time estimate. This key is used to define the additional travel time added to each trip. The default value is 600 seconds (10 minutes).

ZONE_EQUIVALENCE_FILE

The zone equivalence file is required for the trip adjustment factors. The key specifies the name of the file that defines a group of zones. Zone Groups typically represent large geographic areas or governmental entities (i.e., cities and counties). Each zone may only be associated with one Zone Group. The software generates warning messages if a zone is used more than once or appears to be missing from the sequence of zone numbers.

The zone equivalence file is a tab, space, or comma-delimited ASCII file with special format rules. A sample equivalence file is shown below.

```
1  0  Portland CBD - 1
1  1  1..16
2  0  West Suburbs - 2
2  1  79..307, 1248..1253
3  0  Southwest Suburbs - 3
3  1  308..403, 931..933
4  0  Southeast Suburbs - 4
4  1  404..557, 934..943, 1254..1258
5  0  East Portland - 5
5  1  561..563, 714..721, 731..738, 763..929, 949..961, 963..969
6  0  East Suburbs - 6
6  1  558..560, 564..713, 722..730, 739..762, 1259..1260
7  0  West Portland - 7
7  1  17..78, 930, 944..948, 962, 1247
8  0  Clark County - 8
```

```
    8  1  970..1246
```

If the file contains a header record, it is ignored by the software. The first integer on each subsequent record is the district or zone group number. This number is followed by an index number that is used to associate multiple records with a given district. If the index number is zero, the software interprets everything that follows the index number as the district label. The first 25 characters of the label are printed in reports.

If the index number is not zero, the values that follow are interpreted as a range of zone numbers. Individual zone numbers and ranges of zone numbers can be specified on a given record. A range of zone numbers is specified using the first and last number in the sequence connected by two or more periods. For example, "79..307" represents all of the zone numbers between 79 and 307.

TRAVEL_TIME_FILE

The travel time file key is appended to the PROJECT_DIRECTORY key to specify the file name for the optional input travel time file. This is a zone-to-zone travel time matrix that, if provided, overrides the travel time calculation for interchange with non-zero values. Otherwise the average speed and straight line distance is used.

Trip Groups

The following 17 keys make up a trip group. The "#" at the end of the each key identifies the trip group number. All keys with the same trip group number combine to define how the trips are converted from zone-to-zone trip tables to trips between two activity locations at a specific time of day. Any number of trips groups can be defined. They are processed in group number order.

TRIP_TABLE_FILE_#

The trip table file key is appended to the PROJECT_DIRECTORY key to specify the file name for the input trip table file for the trip group. If the trip table format is not specified and a Definition file is not found, the program assumes the file is in Version 3 format. The default Version 3 format is a tab-delimited text file with three integer data fields and no header record. The first field is the origin zone number, the second field is the destination zone number, and the third field is the number of trips.

TRIP_TABLE_FORMAT_#

The trip table format key enables the user to specify the input format for the trip table file. The default file format is VERSION3. Other options include BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, DBASE, and SQLITE3. This key is only used if a Definition file is not found. It is primarily used to override the default Version 3 processing and have the program construct a Definition file based on the file header and field types.

TRIP_SCALING_FACTOR_#

The trip scaling factor key enables the user to factor the input trips by a scaling factor. This could be used to select of subset of the full trip table or grow the trips to a future year estimate.

The factor is a floating point number, but the result of applying the factor will be an integer number of trips for each origin-destination pair. A bucket rounding process is applied to minimize the impact of integer rounding as much as possible. The default value is 1.0.

TRIP_TIME_FILE_#

The trip time file key is appended to the PROJECT_DIRECTORY key to specify the file name for the input trip time file for the trip group. If the trip time format is not specified and a Definition file is not found, the program assumes the file is in Version 3 format. The default Version 3 format is a tab-delimited text file with three floating point data fields and no header record. The first field is the start time in hours, the second field is the end time in hours, and the third field is the relative share of trips assigned to the period between the start time and end time. The shares are automatically normalized to 1.0.

TRIP_TIME_FORMAT_#

The trip time format key enables the user to specify the input format for the trip time file. The default file format is VERSION3. Other options include BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, DBASE, and SQLITE3. This key is only used if a Definition file is not found. It is primarily used to override the default Version 3 processing and have the program construct a Definition file based on the file header and field types.

TRIP_TIME_SCRIPT_#

The trip time script is optional. If provided, the key is appended to the PROJECT_DIRECTORY key to specify the file name for a user program script. This script enables the user to apply a different set of diurnal distribution shares to selected origin or destination zone numbers. Refer to the Algorithm section of this document for more details about how the script file is used.

TRIP_TIME_FIELD_#

The trip time field is optional and is only processed when a trip time script is not provided. By default, the program selects the third field in the trip time file as the diurnal distribution values for a given trip group. The user may, however, include multiple diurnal distribution fields in a given trip time file. This key enables the user to specify the field name that contains the diurnal distribution values used for this trip group.

TIME_PERIOD_RANGE_#

The time period range is optional. If provided, this key specifies the range of time periods that will be selected from the trip time file to construct the diurnal distribution probabilities. By default the program uses all time periods in the trip time file in the distribution. If for example, the trip time file includes a diurnal distribution for all day and the trips in the trip table should only be assigned to off peak time periods, the user can use the time period range to specify the off peak time periods. As a result, none of the trips will be assigned to the peak periods.

A time range is a comma separated list of start and end time pairs. The time values in each pair are separated by two periods (e.g., 6:30..9:00). The values can be specified in seconds, decimal hours, or 24 hour clock times. An off peak time period range can be specified with a key like:

0:00..6:30, 9:00..15:30, 18:30..24:00

TIME_CONTROL_POINT_#

The diurnal distribution produced by the trip time file is used to define the probability that a given trip will take place at a given time. The time assigned to a trip is based on the offset of a random number within the cumulative probability distribution. By default, this time value (in seconds) is assumed to be the time at the origin of the trip. This key enables the user to specify how the time value is applied. The options include “ORIGIN”, “DESTINATION”, and “MID-TRIP”. If “DESTINATION” is selected, the program interprets the diurnal distribution as an arrival time distribution. The start time for the trip is then calculated by subtracting the estimated travel time from the selected arrival time. If “MID-TRIP” is selected, the program interprets the diurnal distribution as a time-in-motion distribution. The start time and the end time for the trip are calculated by subtracting and adding one half of the estimated travel time from the selected mid-trip time. The time control point is saved in the CONSTRAINTS field in the output trip file. The constraint codes are:

0	Origin
1	Destination
2	Mid-Trip

RETURN_TRIP_OFFSET_#

The return trip offset key is optional. The default value is “0” which indicates that return trip processing is disabled. If the key is greater than zero, it defines the duration of the activity at the destination zone. The program then assumes that the input trip table is provided in Production-Attraction format and that half of the trips should be assigned in the production-attraction direction and half of the trips should be oriented in the attraction-production direction. The diurnal distribution provided by the trip time file is used to set the time for the outbound trip (production-attraction orientation). The return trip starts at a time equal to the arrival time of the outbound trip plus the activity duration specified by this key. If the return time is after midnight, the time is reset by subtracting 24 hours.

ORIGIN_WEIGHT_FIELD_#

This key identifies the user-defined field name in the Activity Location file that is used to determine the probability that an activity location in the zone will be chosen as the origin location. The probability of selecting a particular activity location is equal to the value of this field divided by the sum of field values of all activity locations assigned to the origin zone number. If a field name is not provided, the ConvertTrips program allocates an equal probability to each activity location in the zone.

DESTINATION_WEIGHT_FIELD_#

This key identifies the user-defined field name in the Activity Location file that is used to determine the probability that an activity location in a given zone will be chosen as the destination location. The probability of selecting a particular activity location is equal to the value of this field times the distance between the origin activity location and this activity location divided by the sum of the value-distance products of all activity locations assigned to the destination zone number. If a field name is not provided, the ConvertTrips program allocates an equal probability to each activity location in the zone.

TRIP_PURPOSE_CODE_#

The trip purpose code is optional. Any value between 0 and 100 can be specified. The default value is 1. In some applications this value is used to define the relative priority and scheduling flexibility of a given activity. The original TRANSIMS documentation includes the following values as a suggested list of trip purposes or activity types.

0	Home
1	Work
2	Shop
3	Visit
4	Social/Recreation
5	Other
6	Serve Passenger
7	School
8	College

TRAVEL_MODE_CODE_#

The travel mode code is optional. The default value is “2” which corresponds to the drive mode. The mode code options include:

1	Walk
2	Drive
3	Transit
4	Transit with Rail Bias
5	Park-&-Ride Outbound
6	Park-&-Ride Inbound
7	Bicycle
8	Magic Move
9	School Bus
10	Two Person Carpool
11	Three Person Carpool
12	Four Person Carpool
13	Kiss-&-Ride Outbound
14	Kiss-&-Ride Inbound

Separate trip tables should not be provided for the Park-&-Ride and Kiss-&-Ride Inbound modes. Since the return trip for a park-&-ride and kiss-&-ride trip needs to retrieve the vehicle at a remote park-&-ride or kiss-&-ride lot (i.e., a parking lot not attached to the home location), the program needs to build park-&-ride or kiss-&-ride trips as round trip tours rather than independent trips. Park-&-ride and kiss-&-ride trip tables should therefore be provided in production-attraction format and the RETURN_TRIP_OFFSET key should be used to specify the activity duration. This will generate an origin-destination trip and a destination-origin trip from the same activity locations using the same household ID, person ID, and vehicle ID. This enables the Router to park the vehicle at the park-&-ride or kiss-&-ride lot during the outbound trip and retrieve the vehicle from that same lot for the inbound trip. Note that parking penalties and parking costs are not added to kiss-&-ride vehicles.

AVERAGE_TRAVEL_SPEED_#

The average travel speed is used to estimate the travel time between activity locations. The straight-line distance between the coordinates of the two points is divided by the average travel speed to estimate the travel time. The additional travel time value is added to the estimated travel time to calculate the trip duration. Default value for this key is 10 meters per second (about 22 mph). This calculation is used if a TRAVEL_TIME_FILE is not provided or the origin-destination cell in this matrix is zero.

VEHICLE_TYPE_#

The vehicle type key specifies the type of vehicle that will be assigned to trips from the corresponding trip table. Different values can be specified to distinguish auto and truck trips. The vehicle type value is an integer that corresponds to the vehicle type and subtype code defined in the Vehicle Type file. This key defaults to 1 when the value is not specified.

1	Car
2	Truck
3	Taxi
4	Local Bus
5	Express Bus
6	Trolley
7	Street Car
8	Light Rail Transit
9	Rapid Rail
10	Commuter Rail
11	Two Person Carpool
12	Three Person Carpool
13	Four Person Carpool
14	Light Duty Truck
15	Heavy Duty Truck
16	Restricted Use Vehicle

VEHICLE_SUBTYPE_#

The vehicle subtype key specifies the subtype of the vehicle assigned to trips from the corresponding trip table. It works in conjunction with the Vehicle Type key to specify a subclass of vehicle as defined in the Vehicle Type file. The vehicle subtype is an integer and defaults to 0 when the value is not provided.

TRIP_ADJUSTMENT_FACTORS_#

Trip adjustment factors are options. The key is appended to the PROJECT_DIRECTORY key to specify the file name for the input factor file for the trip group. If a factor file is provided, the Zone Equivalence file must also be specified. The default factor file should contain at least three fields:

ORG	the origin district or zone group number
DES	the destination district or zone group number
FACTOR	a floating point correction factor to be applied to the number of trips

Since the default adjustment factor is 1.0, only interchanges that need to be increased or decreased should to be included in the file. The program reports the number of trips that were added and deleted for each trip group and all trip groups.

TRIP_ADJUSTMENT_FORMAT_#

The trip adjustment format key enables the user to specify the input format for the trip adjustment factors file. The default file format is VERSION3 which assumes the three fields listed above. The format options include VERSION3, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, and DBASE.

TIME_PERIOD_EQUIVALENCE_#

Time period equivalence file is optional. By default, the trip adjustment factors apply to all times of day. If the factors differ for different times of day, the trip adjustment factor file can include a PERIOD field to specify the time period code for the adjustment factor. These codes are translated into actual seconds of the day using the time period equivalence file. This file includes one or more time period range strings for each time period code.

CONVERTTRIPS_REPORT_#

Reports are optional. The "#" at the end of the report keyword represents the report number (e.g., CONVERTTRIPS_REPORT_1). The key can be provided with additional numbers to specify additional reports. The reports are generated in numerical order (i.e., 1, 2, 3…) for each trip group.

The string parameter associated with a report keyword is limited to the following options:

TRIP_TIME_SCRIPT
TRIP_TIME_STACK
PRINT_ZONE_EQUIVALENCIES
TIME_PERIOD_EQUIVALENCE

The above reports are printed in the "*.prn" file that is generated in the same directory as the control file used to run the ConvertTrips program. Each of above reports is described below:

Trip Time Script

If a trip group includes a trip time script, this report lists the commands found in the script file. An example is shown below.

```
Trip Time Script #1

      IF (Matrix.ORG == 1) THEN
           RETURN (Diurnal.SHARE1)
      ELSE IF (Matrix.ORG == 3) THEN
           RETURN (Diurnal.SHARE3)
      ELSE
           RETURN (Diurnal.SHARE2)
      ENDIF
      ENDIF
```

Trip Time Stack

The program compiles the trip time script into a command processing stack for execution. This report shows how the script was converted to processing commands. The stack commands corresponding to the script listed above are shown below.

```
Trip Time Stack #1

  1) Integer      Matrix.ORG
  2) Integer      1
  3) Relation     EQ
  4) Logical      If False, Jump to 8
  5) Integer      Diurnal.SHARE1
  6) Return       Integer
  7) Logical      Jump to 17
  8) Integer      Matrix.ORG
  9) Integer      3
 10) Relation     EQ
 11) Logical      If False, Jump to 15
 12) Integer      Diurnal.SHARE3
 13) Return       Integer
 14) Logical      Jump to 17
 15) Integer      Diurnal.SHARE2
 16) Return       Integer
 17) End
```

Print Zone Equivalencies

The Zone Equivalence file is used to define Zone Groups. If this report option is selected, a list of the zones associated with each Zone Group is printed after the control keys are processed.

```
Zone Equivalence

[      Portland CBD - 1]  1 = 1..16
[      West Suburbs - 2]  2 = 79..307, 1248..1253
[ Southwest Suburbs - 3]  3 = 308..403, 931..933
[ Southeast Suburbs - 4]  4 = 404..557, 934..943, 1254..1258
[     East Portland - 5]  5 = 561..563, 714..721, 731..738, 763..929, 949..961
                              963..969
[      East Suburbs - 6]  6 = 558..560, 564..713, 722..730, 739..762, 1259..1260
[     West Portland - 7]  7 = 17..78, 930, 944..948, 962, 1247
[      Clark County - 8]  8 = 970..1246
```

Algorithm Notes

The ConvertTrips program reads each trip table in trip group number order and processes each record as they are read. The process begins by reading the trip time file, selecting the specified time period records, and normalizing the resulting share fields to a cumulative distribution between 0 and 1. The specific second assigned to the trip is set by interpolating the start and end time of the selected time period by the cumulative share at the beginning and end of the time period. For example, if the random probability is 0.55 and the cumulative distribution is as follows:

Start Time	End Time	Cumulative Probability
0	10000	0.10
10000	20000	0.35
20000	40000	0.60
40000	50000	0.80
50000	86400	1.00

The time assigned to the trip is:

$$\text{Time} = 20000 + (40000 - 20000 - 1) * (0.55 - 0.35) / (0.60 - 0.35) = 35999$$

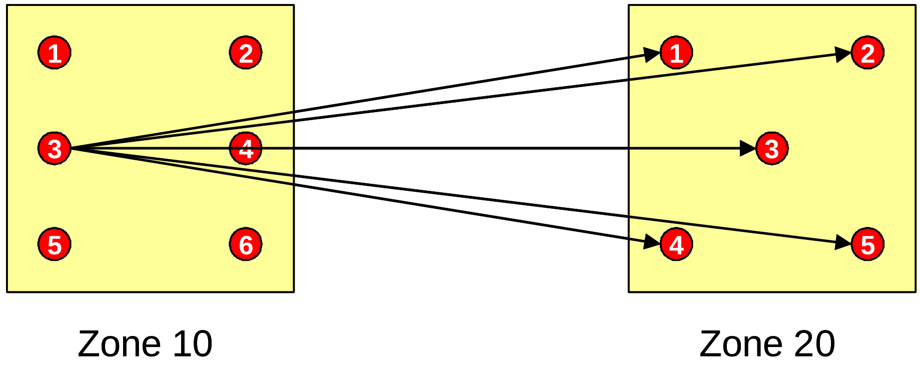
This interpolation works well if the time increments are relatively small. The example shown above would produce a distribution that is not very smooth. That is way the software converts the user-provided distribution to a smoothed one-minute probability distribution curve. A multi-iteration moving average procedure is used to estimate the probability of each minute of the day given the control points provided by the user. The initial probability assigned to each minute is based on the user-provided probability divided by the number of minutes included in the time period. For example, the time period 40000 to 50000 in the sample above has a probability of 0.20. The program initialized the probability for each minute between 40000 and 50000 to 0.0012 (i.e., 0.20 / ((50000 – 40000) / 60)). The program then smoothes these values to create uniform transitions between each time period.

In addition to smoothing the distribution, the program maintains and monitors trip targets for each minute. Floating point targets for each minute are accumulated by applying the smoothed distribution to the total number of trips on each zone interchange. If the probability distribution attempts to assign more than the next rounded integer number of trips to a given minute, it re-tries the allocation with a new random number. Up to 10 re-tries as attempted before is over assigns trips to a given minute of the day.

The process then assigns the trip to an origin and destination activity location. It first assigns the origin of the trip to an activity location within the origin zone and then based on the origin location selects the destination location.

The origin location is randomly selected from the activity locations with non-zero origin weights assigned to the origin zone for this particular trip table. A cumulative distribution of the weights is generated for the zone and a random probability is used to select a location from the normalized distribution. Once the origin is selected, the cumulative distribution for the destination zone is generated. The weight assigned to each activity location within the destination zone is the product of the non-zero destination weights and the straight-line distance between the specific origin location and each location assigned to the destination zone.

The figure below provides a simplified example of the destination weighting process. In this example, zone 10 is the origin zone and activity location number 3 within zone 10 was selected as the origin activity location. Zone 20 is the destination zone and it contains five activity locations. The probability of selecting a destination activity location is the product of the distance from activity location 3 in zone 10 to each activity location in zone 20 and the destination weight for the activity location. Table below shows some typical calculations.



			Example 1			Example 2		
Zone 20	**Zone 20**	**Weight**	**Distance**	**Destination**	**Percent**	**Distance**	**Destination**	**Percent**
Location	**Weight**	**Distribution**	**to Origin**	**Weight**	**Probability**	**to Origin**	**Weight**	**Probability**
1	10	16.7%	500	5000	6.8%	50000	500000	16.4%
2	15	25.0%	1950	29250	40.1%	51450	771750	25.4%
3	10	16.7%	1175	11750	16.1%	50675	506750	16.7%
4	15	25.0%	500	7500	10.3%	50000	750000	24.6%
5	10	16.7%	1950	19500	26.7%	51450	514500	16.9%
Total	60	100.0%		73000	100.0%		3043000	100.0%

Notice the impact of distance on the relative probability of selecting a given destination based on the examples in the table above. Example 1 represents trips between near-by zones. In this case the probabilities are significantly affected by the relative distance between the origin and destination activity locations. Example 2 shows how the distance impact is almost non-existent for zones that are far apart. In other words, the distance weighting function favors destinations that are farther away when the relative trip length is short. Since this algorithm is primarily applied to vehicle trips, the implication is that very short vehicle trips are less likely. Another way to think about this is that people are more likely to walk if the distance is very short.

The impact of the distance weighting function is most significant when allocating intrazonal trips. The procedure tends to pick a destination as far away from the origin as possible. Notice also that the probability of selecting a single location as the origin and destination of the trip is zero. This is why it is important to include at least two activity locations within each zone. If the zone has only one activity location, all of the intrazonal trips within that zone will be lost. Of course, if the zone has no activity locations, no trips to or from that zone can be converted.

Given the origin and destination activity location and the trip time, the trip record can be constructed. The trip start time and the trip end time is calculated using the trip time, the distance between the origin and destination, the average travel speed for the trip group, the additional travel time, and the time control point. The travel time is estimated using as follows:

Travel time = (OD distance / average travel speed) + additional travel time

The following calculations are then used to set the start and end time by time control type.

ORIGIN:
Start time = trip time
End time = trip time + travel time

DESTINATION:
Start time = trip time – travel time
End time = trip time

MID-TRIP:
Start time = trip time – (travel time / 2)
End time = trip time + (travel time / 2)

If a RETURN_TRIP_OFFSET key is provided, a return trip is generated for the same person and vehicle with the same activity locations and a trip start time equal to the original trip end time plus the offset value specified by the RETURN_TRIP_OFFSET key.

Trip Time Scripts

By default, the algorithm uses a single diurnal distribution to set the trip time for all trips in a trip group. In many cases a single diurnal distribution does not adequately capture the time of day variations of travel in different parts of the region. The user can split these trips into different trip tables and provided different diurnal distributions for each sub-category. Alternatively, the user can provide a trip time file with multiple diurnal distributions and define a trip time script to select the appropriate distribution for a given trip interchange.

The script references the zone numbers in the trip table using the syntax

Matrix.*Org_Field_Name*
Matrix.*Des_Field_Name*

where *Org_Field_Name* and *Des_Field_Name* are replaced with the actual field name included in the trip table file for the origin and destination zone numbers. In the default Version 3 format these fields are labeled “ORG” and “DES”. The program also recognizes “FROM_ZONE”, “FROM”, and “ORIGIN” for the origin zone field and “DESTINATION”, “TO_ZONE”, “TO”, and for the destination zone field.

Standard user program syntax (see the User Program documentation for more details) can be used to establish if-then-else statements that identify which range of zones should be assigned to a given distribution field. The distribution field is identified using the following syntax:

Diurnal.*Share_Field_Name*

where *Share_Field_Name* is replaced with the actual field name included in the trip time file. The script must return the value of this field to the ConvertTrips program in order to select the appropriate set of distribution factors for the trip interchange.

A simple example script is shown below:

```
IF (Matrix.DES >= 1 AND Matrix.DES <= 50) THEN
     RETURN (Diurnal.SHARE1)
ELSE IF (Matrix.ORG > 1000) THEN
     RETURN (Diurnal.SHARE3)
ELSE
     RETURN (Diurnal.SHARE2)
ENDIF
ENDIF
```

Sample Printout

A sample printout file generated by ConvertTrips program is shown 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.

```
********************************************
|                                          |
|       ConvertTrips - Version 4.0.4       |
|   Copyright (c) 2006 by AECOM Consult    |
|         Sun Jul 23 16:17:46 2006         |
|                                          |
********************************************

Control File = ConvertTrips1.ctl
Report_File  = ConvertTrips1.prn (Create)

Trip Table Conversion Utility

Project Directory = d:\software\test\case1\demand

Network Directory = d:\software\test\case1\network
Activity Location File = d:\software\test\case1\network\Activity_Location.txt
Process Link File = d:\software\test\case1\network\Process_Link.txt

New Household File = d:\software\test\case1\demand\Household.txt

New Population File = d:\software\test\case1\demand\Population.txt

Vehicle Type File = d:\software\test\case1\demand\Vehicle_Type.txt

New Vehicle File = d:\software\test\case1\demand\Vehicle.txt
Vehicle File is Sorted by Vehicle ID

New Trip File = d:\software\test\case1\demand\Trip.txt

Starting Household ID = 1
Starting Vehicle ID = 1

Time of Day Format = SECONDS

Additional Travel Time = 180 seconds

Random Number Seed = 1153685866

Trip Table File #1 = d:\software\test\case1\demand\TripTable1.txt
Trip Time File #1 = d:\software\test\case1\demand\TripTime.txt
Time Control Point = ORIGIN
Time Period Range = 0:00..1:00
Origin Weight Field Name = USER1, Number = 13
Destination Weight Field Name = USER2, Number = 14
Trip Purpose Code = 1
Travel Mode Code = 2
Average Travel Speed = 15.0 (meters/second)
Vehicle Type = 1, Subtype = 0

Trip Table File #2 = d:\software\test\case1\demand\TripTable2.txt
Trip Time File #2 = d:\software\test\case1\demand\TripTime.txt
Time Control Point = ORIGIN
Time Period Range = 2:00..3:00
Origin Weight Field Name = USER1, Number = 13
Destination Weight Field Name = USER2, Number = 14
Trip Purpose Code = 1
Travel Mode Code = 2
Average Travel Speed = 15.0 (meters/second)
Vehicle Type = 1, Subtype = 0

Trip Table File #3 = d:\software\test\case1\demand\TripTable3.txt
Trip Time File #3 = d:\software\test\case1\demand\TripTime.txt
Time Control Point = ORIGIN
Time Period Range = 4:00..5:00
Origin Weight Field Name = USER1, Number = 13
Destination Weight Field Name = USER2, Number = 14
Trip Purpose Code = 1
Travel Mode Code = 2
Average Travel Speed = 15.0 (meters/second)
Vehicle Type = 1, Subtype = 0

Trip Table File #4 = d:\software\test\case1\demand\TripTable4.txt
Trip Time File #4 = d:\software\test\case1\demand\TripTime.txt
```

```
Time Control Point = ORIGIN
Time Period Range = 6:00..7:00
Origin Weight Field Name = USER1, Number = 13
Destination Weight Field Name = USER2, Number = 14
Trip Purpose Code = 1
Travel Mode Code = 2
Average Travel Speed = 15.0 (meters/second)
Vehicle Type = 1, Subtype = 0

Highest Trip Group Number = 4
Number of Trip Table Groups = 4

Number of Unique Activity Location Weight Fields = 1


Number of Activity Location Records = 8

Number of Process Link Records = 16

Number of Vehicle Type Records = 8

Highest Traffic Analysis Zone = 4

Processing Trip Table File #1
Trip Table File #1 has 6 Records and 3100 Trips

Processing Trip Table File #2
Trip Table File #2 has 6 Records and 6500 Trips

Processing Trip Table File #3
Trip Table File #3 has 6 Records and 7700 Trips

Processing Trip Table File #4
Trip Table File #4 has 6 Records and 10000 Trips

Total Number of Trips Read     = 27300
Total Number of Trips Written  = 27300
Number of Trips Not Allocated  = 0

Number of Households Generated = 27300
Number of Persons Generated    = 27300
Number of Vehicles Generated   = 27300
```