AccessDocs Documentation

Release 0.0.1

Renaissance Planning

Jun 21, 2022

Contents:

MMA is an anlytical lens and spatial/network analysis toolkit developed by Renaissance Planning Group to facilitate accessibility analysis for multimodal transportation and land use planning applications.

The first half of this guide provides an overview of the key terms and concepts behind accessibility analysis, the basics of analyzing accessibility, and the details of working with the MMA geoprocessing toolbox.

The Maryland Chapter 30 scoring process is a legislatively-mandated scoring system for evaluating transportation projects. The Chapter 30 scoring model evaluates projects across nine goals and twenty-three measures that were established in statute, using a combination of project data, modeling analysis, and qualitative questionnaires.

The second half of this guide focuses on the step-by-step process for developing accessibility scores for Chapter 30 transit project applications.

Note: This documentation details steps used to score projects in 2019. Documention on procedures used during 2018 scoring can be accessed here.

RENAISSANCE PLANNING
http://www.citiesthatwork.com

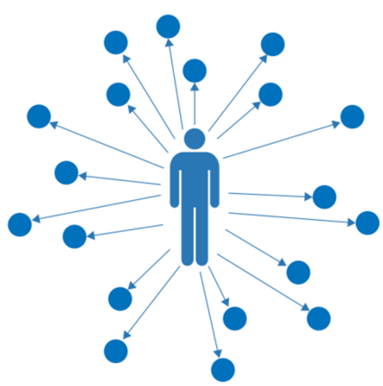
CHAPTER 1

MMA Key Terms and Concepts

This section focuses on the fundamentals of MMA, including the concept of accessibility, why it is important and what it reveals about places. It includes a glossary of key terms for reference when exploring MMA workflows and geoprocessing tools, and provides a brief overview of the basic steps in combining land use data and OD matrices (skims) to produce accessibility scores.

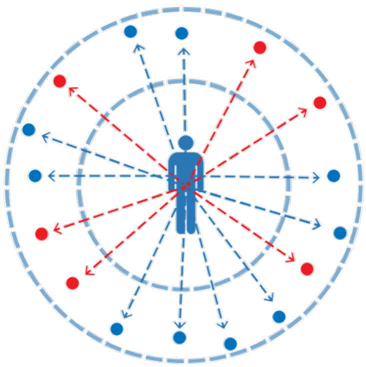
1.1 Multimodal Accessibility: Concept and Purposes

1.1.1 MMA in Concept

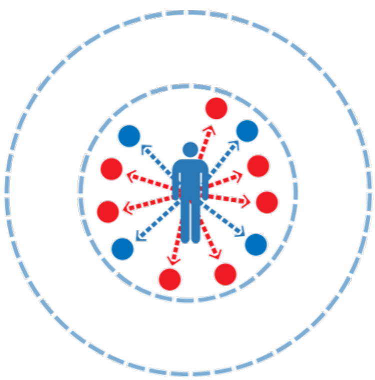


We all have important destinations to reach in our daily lives. Multimodal accessibility (MMA) analysis measures our ability to reach these destinations. It summarizes the number of activities within reach by different travel modes and compares results across modes, times of day, and scenarios to understand the universe of opportunity available from a given location. These opportunities define the possibilities for movement and inform many aspects of our daily travel patterns, such as when and how we travel, for what purposes and to what destinations.

Combined Land Use and Transportation Analysis



Accessibility can be increased through mobility solutions, such as transportation projects that allow people to reach distant locations OR through land use solutions that enhance destination proximity



MMA provides an analytical lens to understand the impacts of changes in land use and transportation on travel choices. It is sensitive to new developments and how they are designed as well as transportation system design and performance. By offering measures that address each of these critical urban systems, MMA has many uses in urban planning and performance measurement applications. Examples of the kinds of questions MMA can help answer include:

- How many jobs can I reach by transit from my house?
- On average, how many jobs can workers in a city reach by transit, by walking, or by driving?
- A proposed highway project promises to improve mobility for many commuters. To what extent does it increase their ability to access key destinations?
- Where does regional transit service offer competitive access to jobs or essential goods and services relative to driving?
- Does any segment of the population have better or worse access to key destinations than another?
- Will a transportation or land use project help create more equitable access to jobs, education, or health care for disadvantaged groups?
- Will increased investment in sidewalks and bike lanes result in a higher proportion of trips using non-motorized modes?

1.1.2 Purposes of MMA Analysis

The simple concept behind MMA analysis – measuring the activities that can be reached by different modes – reveals important information about the structure of a place and its connections to its surroundings. These attributes of a place can provide insight into travel behaviors and marketability, making accessibility measures a key component in transportation and land use forecasting, economic modeling, and project prioritization approaches.

Structural analysis of neighborhoods, cities, and regions.

MMA offers measures that describe how urban development patterns and transportation system design and performance define travel options. Simple MMA cumulative accessibility scores show how many jobs, shopping destinations, recreational opportunities, or rooftops are reachable from a given location, accounting for travel time and multimodal options. Advanced analyses show how access to opportunity varies by different populations or how connectivity and land use diversification limits or enhances access. Comparisons among peers are natural, revealing structural differences across a town or across the country.

- Accessibility Observatory
- Brookings "Moving to Access"
- Smart Location Database

Modal Competitiveness

By measuring the accessibility offered by different modes, MMA provides natural comparisons across modes, describing their relevance and competitiveness for meeting the travel needs of an area. How many jobs are reachable by transit versus by driving? Or how many shopping and dining destinations are reachable by walking? The MMA-based answers to these questions help establish the "lift" required to provide multimodal transportation options that meet the needs of travelers. Paired with the structural analyses described above, they can also show where and how land use can be part of the solution.

- Transit:Auto accessibility ratio (TAR) mapping
- US 15-501 Travel Profile

Travel Behavior

Urban structures and modal interaction are the key factors influencing accessibility scores and the competitiveness of various modes. These factors, in turn, influence travel behaviors. Trips in urban places - with walkable destinations, easily accessible transit service, parking constraints and traffic congestion - generate different types of trips than suburban or rural locations with different structural characteristics and travel options. All facets of trip-making may be affected, including trip generation rates, diurnal trip-making patterns, trip-chaining, mode choice, and trip length. This means that accessibility measures can be useful in understanding travel behaviors at an aggregate scale without the need for running more complex travel models. In transportation planning applications, accessibility impacts are thus able to differentiate among alternatives at an early stage, enhancing the efficiency of transportation decision-making.

- NCHRP 770 (also see here)
- TCRP H-51 (see also here)
- Central Maryland Mode Choice Models
- West Palm Beach Sub-Area Study

Project Prioritization

In their daily work, planners analyze land use and transportation systems to understand travel demand and development trends, identify needed improvements, and prioritize investments. MMA measures can provide import information for differentiating and ranking projects or alternatives. They allow projects affecting different modes to be compared side-by-side in common terms? For example, a highway project and a transit project both are expected to increase access to jobs, but which offers the greatest impact and to what groups of residents? Or a company is planning to open new offices in a city and is considering three alternative sites. Which one will be the most accessible to its employees and by which modes? How will the new offices impact travel in the area surrounding each site?

- Maryland Chapter 30
- Virginia SmartScale
- Smart Location Calculator

1.2 Multimodal Accessibility: Key Terms

1.2.1 Origins and Destinations

Any location from which accessibility is measured is referred to as an **"origin"** location in MMA analysis. Put simply, origins are places where trips begin. Usually, many origin locations are analyzed across a neighborhood, city, or region.

MMA analysis summarizes what is reachable from a given origin. Thus, for each origin location, there may be numerous **"destination"** locations. Destinations are places where trips end.

Origins and destinations are often referred to as "Os" and "Ds."

1.2.2 Activities

Each destination reachable from a given origin is characterized by its own mix and intensity of **activities**. In MMA analyses, the term "activities" is a catch-all referring to anything a traveler may want to reach. Examples of activities to which accessibility is measured include jobs, shopping and dining, educational resources, health care services, healthy food, parks acreage, etc.

1.2.3 Population Groups

"Population groups" are analogous to "activities" at the origin location. Similar to activities at destinations, each origin analyzed has its own distinct population composition. Usually, they are segments of the population, such as all residents, transportation-disadvantaged residents, or hotel visitors, but they can also include employees or any other group that varies by origin depending on the goals of the analysis.

1.2.4 Zones

In reality, origins and destinations are discrete locations. That is, travel takes place from "door to door." When measuring accessibility, it is useful to work in more general terms. Origin population groups and destination activities are dealt with in aggregations called **"zones."** Each zone represents a geographic area in which many population groups or activities may be located. Common readily-available zonal aggregation datasets include census blocks and block groups, traffic analysis zones (TAZs), and parcels. Which zonal system is right for a given analysis depends on a variety of factors, including the mode being analyzed, the spatial and temporal scopes of an analysis, and the richness of the available data.

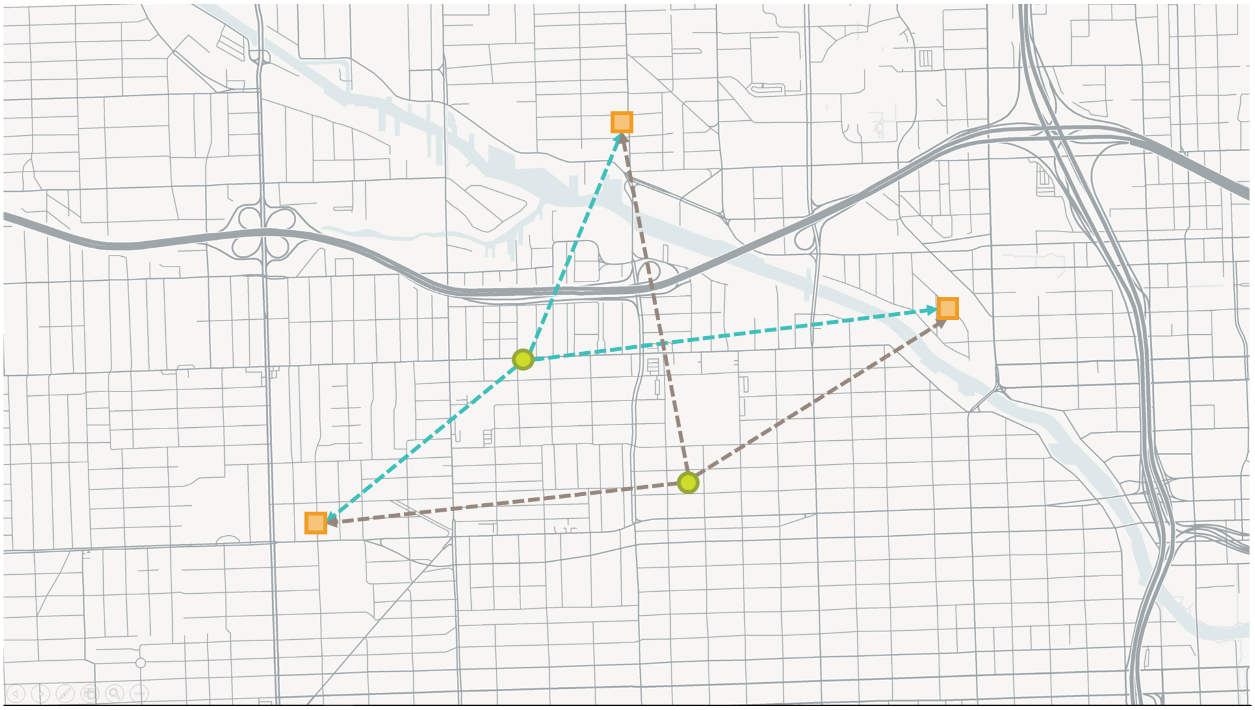


Fig. 1: *Green circles are origins. Orange squares are destinations. In MMA analysis, travel opportunity is analyzed from origins to destinations. As shown by the dashed lines, each origin is analyzed based on its connections to the destinations. For most MMA analyses, thousands of origins and destinations are analyzed.*

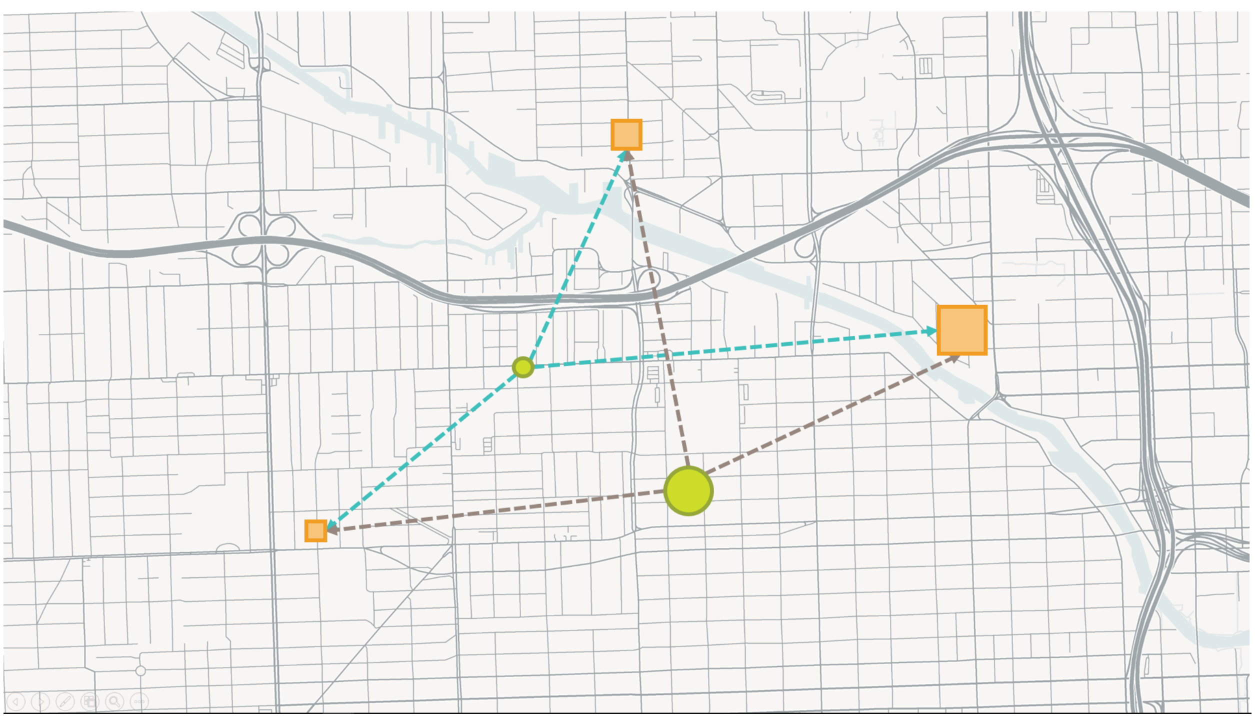


Fig. 2: *Each origin and each destination has different population groups and activities (in terms of type and quantity). In the illustration above, larger symbols indicate larger numbers of activity or population.*

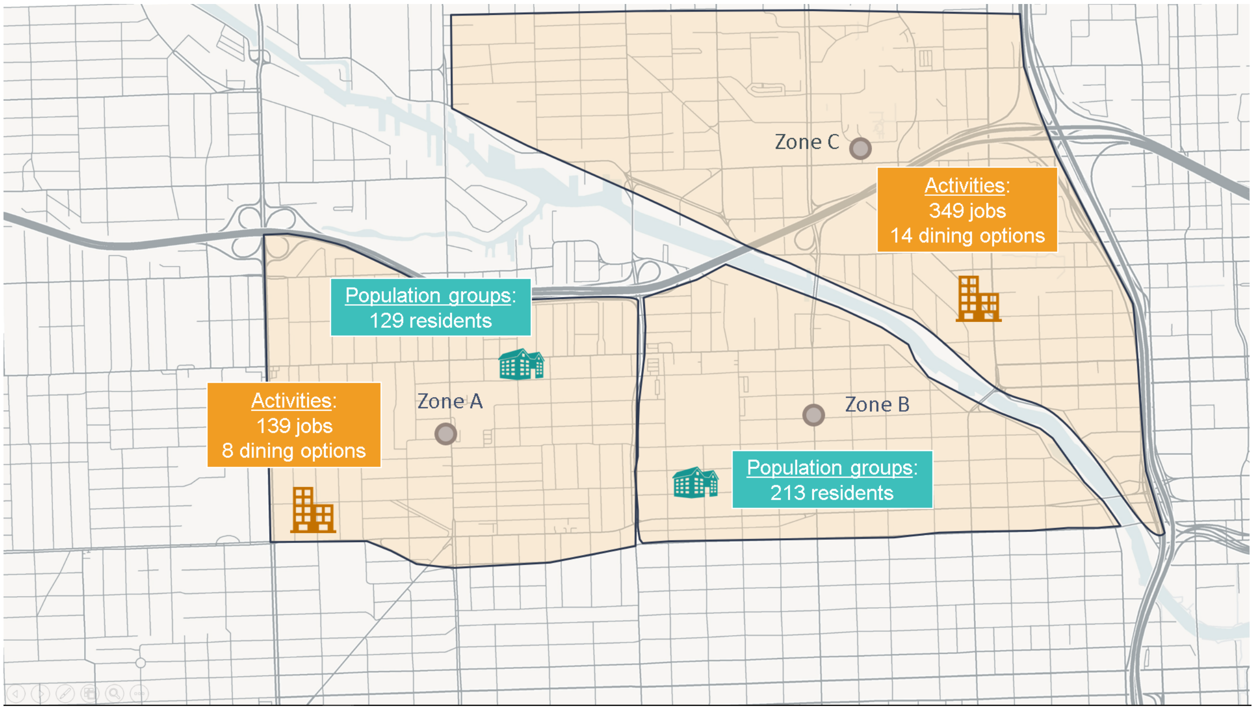


Fig. 3: *To simplify analysis, discrete origin and destination locations – along with their population groups and activities – are aggregated into zones. Travel impedances are analyzed between zone centroids, shown in the grey circles above. In this example, the zonal geography would be too coarse for walk analysis, but may be suitable for auto analysis.*

- *Coming soon: Determinants of Zone Size*

1.2.5 Centroids

For MMA analysis purposes, zones condense all activities and population groups within the zonal boundaries to a single point called a **"centroid."** Centroids generally represent the approximate center of activity within each zone. The use of centroids simplifies processing by representing each zone and its activities and population groups as a single point rather than as a complex polygon. Centroids work best when it is reasonable to suppose that the perceived impedance of travel is similar for all locations within each zone.

1.2.6 Impedance

Accessibility depends on how easy it is to reach destination zones from a given origin zone. Some destinations are nearer than others, and travel conditions - such as congested highways or infrequent transit service - can sometimes make nearby destinations hard to reach in a timely manner. In MMA analysis, the term **"impedance"** refers to any measure of the ease of traveling from an origin zone to a destination zone. Impedance is usually measured in travel time or distance, but it can also be measured in cost, such as fuel expenses and parking costs for personal vehicle travel or fares for shared mobility (transit, taxi, Uber, e.g.) or through generalized cost functions that take into account a wide variety of factors.

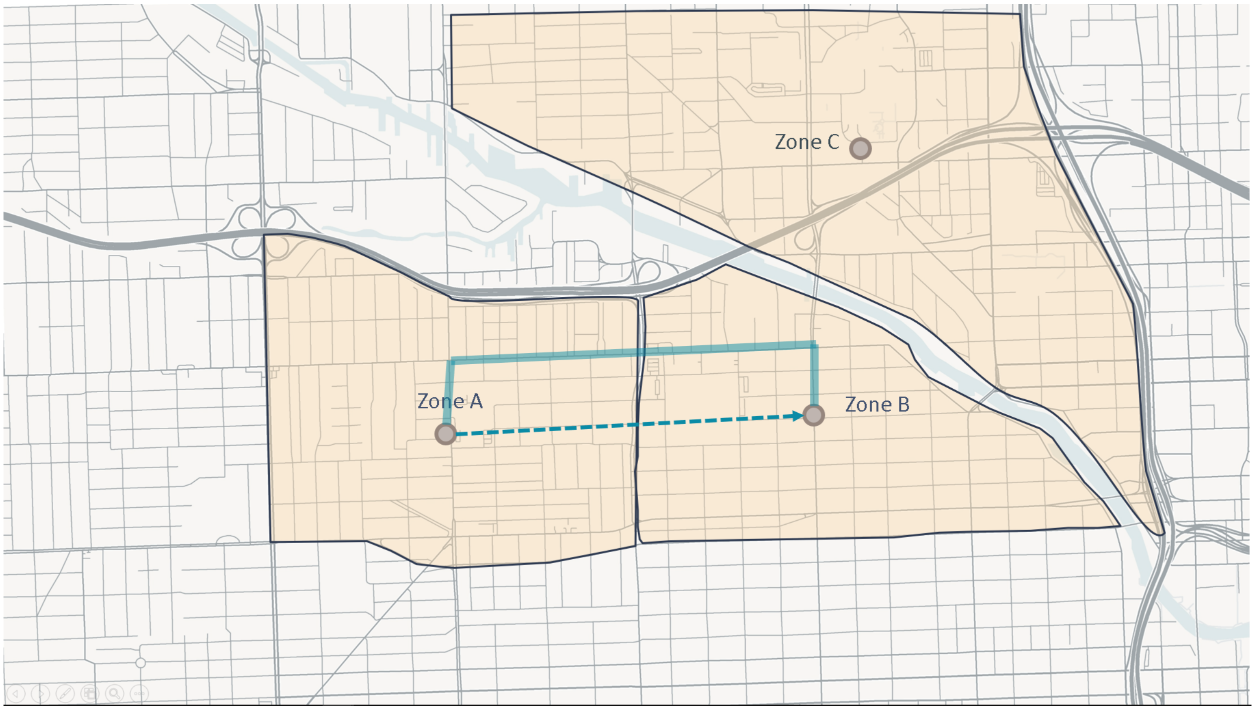


Fig. 4: *Impedance can be analyzed based on simple spatial relationships or based on network analysis. The example above illustrates impedance estimation from Zone A to Zone B. Using spatial analysis, the distance between the zonal centroids is measured and used as the basis for impedance estimation. Using network analysis, the lowest-cost (shortest travel time, e.g.) route is found based on network connectivity and attributes, such as average travel speeds.*

Spatial Analysis of Impedance

One way of understanding the impedance between an origin and destination zone is to consider the distance between them (usually between their centroid points). It is generally reasonable to assume that nearby destinations are easier to reach than those far away. Using **spatial analysis** to estimate impedances between origin and destination zones can provide a useful means of quickly estimating accessibility with minimal data requirements. It can also offer a benchmark for evaluating how well- connected places are based on the networks that serve them (see "Network Analysis of Impedance").

Network Analysis of Impedance

Determining the impedances between origin and destination zones is best accomplished through **network analysis**. Networks approximate real-world conditions on the transportation system and bring greater precision to accessibility analysis than can be achieved through simple spatial estimates. Network datasets have strict rules for determining where and how locations connect to each other. There are numerous algorithms used to determine the shortest path between two zones and for analyzing many origin-destination pairs at a time.

- The MMA geoprocessing tools rely on ESRI's ArcGIS Network Analyst Extension

1.2.7 Skims

The impedance values between origin zones and reachable destination zones are recorded in a matrix called a **"skim."** In MMA processing, the skims are stored as tables in which each row represents an origin-destination pair. Columns in the skim table identify the specific O-D pair and the impedance of the shortest path from the origin zone to the destination zone.

Example of a skim table

Origin Zone	Destination Zone	Impedance (minutes)
A	A	0.0
A	B	12.3
A	C	19.6
B	A	10.8
B	B	0.0
B	C	5.2
C	A	21.1
C	B	6.4
C	C	0.0

A skim is a table that records the impedance associated with traveling between each origin-destination pair. A skim is also sometimes called an "OD Matrix."

See also:

- Geoprocessing Toolbox - Create Skim
- Geoprocessing Toolbox - Create Average Matrix
- mma.Skim

1.2.8 Decay Rates

As impedance to a destination increases, it is reasonable to suggest that the destination's relevance to the origin's accessibility diminishes. For example, suppose zone j has 100 jobs in its area and is reachable from zones i and k. In simple terms, those 100 jobs are accessible from both zone i and zone k. However, it takes 35 minutes to reach those jobs from zone k, and just 12 minutes to reach them from zone i. Which origin zone has the greater accessibility?

Decay rates allow accessibility results to account for the value of time. They provide a formula to translate impedance into discount factors that can then be applied to activities at destinations when summarizing accessibility for each origin. In the example above, the 100 jobs at zone j might be discounted so that they are effectively equivalent to 88 jobs from zone i and 46 jobs from zone k, taking into account the time it takes to reach them from each origin zone.

Decay rates are an optional component of MMA analysis, but they can significantly impact results and enhance their relevance and explanatory power.

See also:

- Geoprocessing Toolbox - Manage Decay Rates
- mma.Decay

1.2.9 Weighted Averages

All of the elements of accessibility analysis described in this section yield estimates of access to activities at a zonal level. When the aim of the analysis is to describe accessibility for an area consisting of multiple origin zones, averages based on the zones' population groups must be calculated. This approach to calculating averages for aggregated data (zones) based on the distribution of values (population groups) across each record is called a **weighted average.**

Example of a weighted average calculation

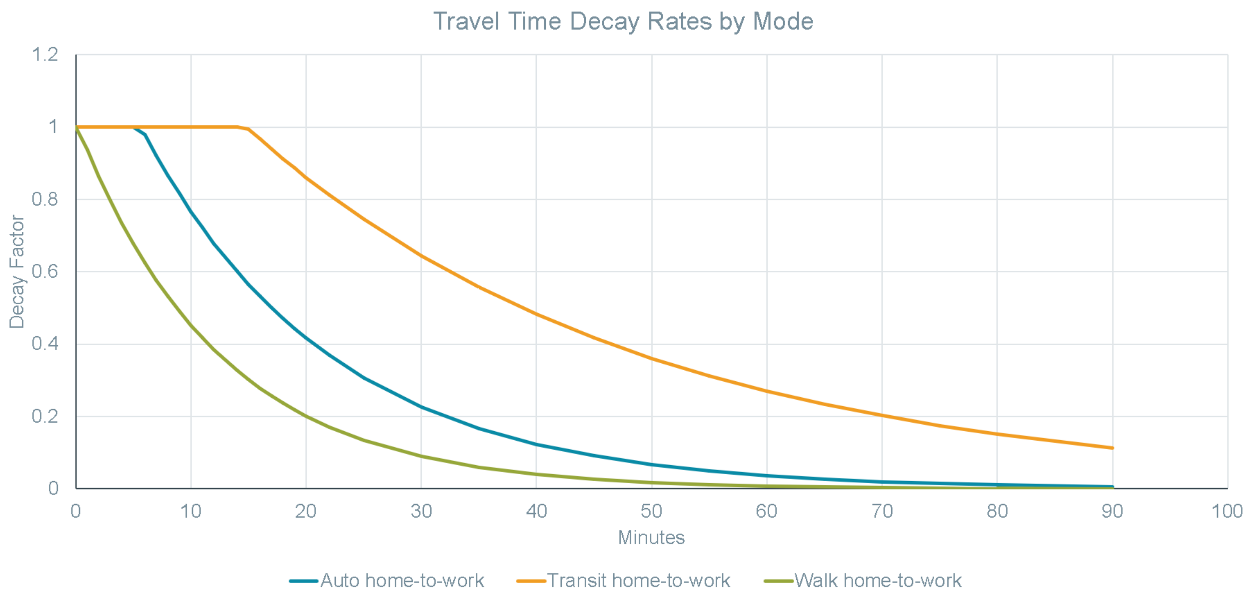


Fig. 5: *Decay rates define how to discount destination-end activities based on the impedance between the origin and the destination. They often vary by mode and travel purpose. A collection of curves modeling decay based on travel time for the auto, walk, and transit modes for home-to-work trips is shown in this illustration.*

Zone	AccessScore	Population	Disadvantaged Population	Access Score * Population	Access Score * Disadvantaged Population
A	5,000	550	325	2,750,000	1,625,000
B	3,000	1,630	150	4,890,000	450,000
C	10,500	920	630	9,660,000	6,615,000
SUM	(NA)	3,100	1,105	17,300,000	8,690,000

The weighted average *AccessScore* for combined zones *A*, *B*, and *C* depends on which population group is being considered. For the general population (*Population* field), the weighted average is the sum of the product of each zone's *Population* and *AccessScore* values, divided by the total *Population* in all three zones. A similar approach is taken for the *Disadvantaged Population*, but the resulting value will be different because the distribution of population across the three zones is different for each population group.

- Average *AccessScore* for *Population* = 17,300,000/3,100 = **5,581**
- Average *AccessScore* for *Disadvantaged Population* = 8,690,000/1,105 = **7,864**

In this example, the disadvantaged population has a higher average access score than the general population.

See also:

Geoprocessing Toolbox - Calculate Weighted Average

1.3 MMA Process Fundamentals

The process of developing multimodal accessibility scores is simple in concept but challenging in practice. There are many decisions to make along the way, and processing data for numerous origin-destination pairs can be computationally cumbersome. For this reason, a set of geoprocessing tools for ArcGIS have been developed to guide analysts through the process. The geoprocessing toolbox is documented here. This section provides insight into the major phases and components of an MMA analysis.

The basic procedures for MMA processing are presented in the diagram below:

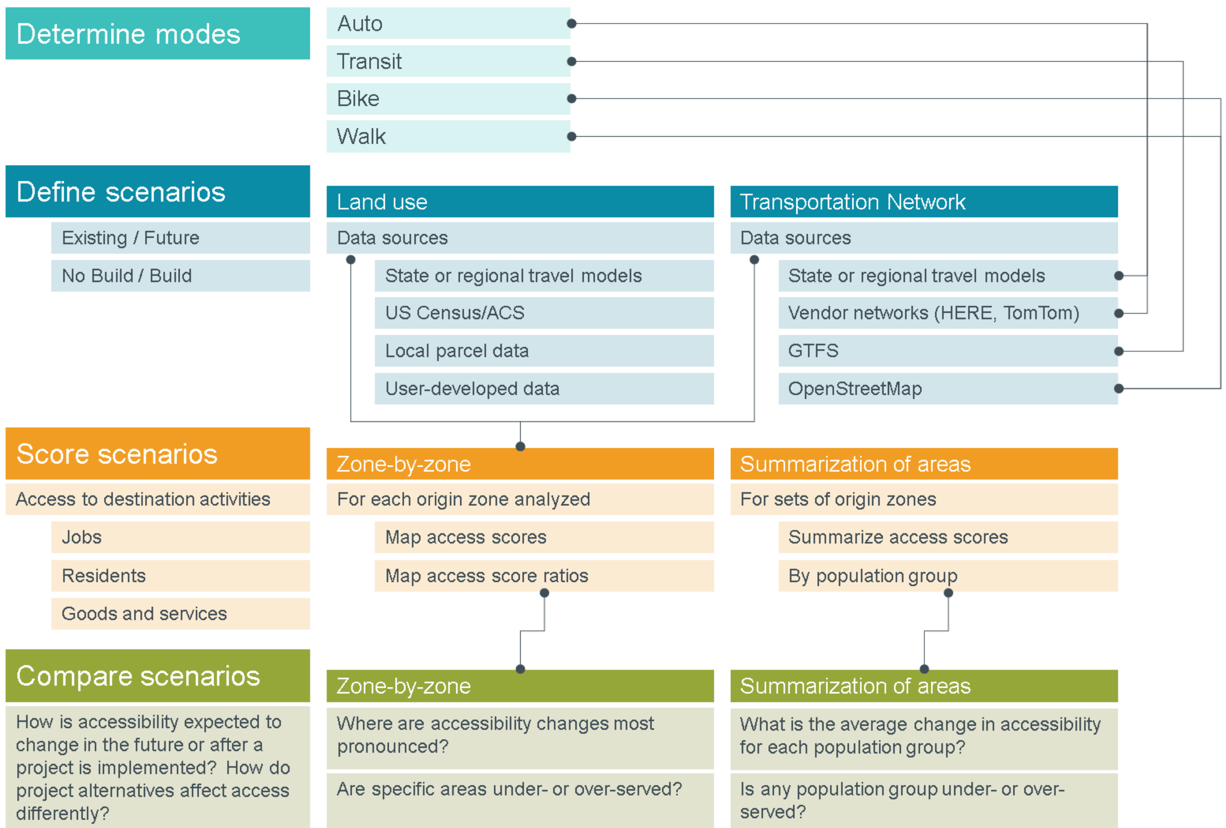


Fig. 6: *The process of developing accessibility scores starts by determining the modes to be analyzed and defining scenarios by selecting appropriate land use and network data. Examples of potential data sources are listed in the figure above. Scores are developed for each scenario by using network data to evaluate travel times and summarize the number of activities reachable by each population group. Comparisons across different scenarios can provide useful insight for analyzing a variety of planning issues.*

1.3.1 Determine which modes will be analyzed

Depending on the focus of the analysis, you may only need to calculate accessibility for a single travel mode, such as walking or transit. In other cases, complete multimodal analysis may be required. The most commonly evaluated modes are walking, biking, transit (walk access), and auto. The modes selected will determine the data used in the analysis.

1.3.2 Define scenarios

What conditions will you analyze and what comparisons are desired? Scenarios include combinations of land use and network data. Thus, the selection of data sources is critical in scenario definition. Important considerations include the temporal and spatial scope of the analysis, the modes to be analyzed, budget for obtaining vendor data, and availability of open data sources such as GTFS feeds. Each scenario can blend alternative land use and network data. For example, suppose you want to assess future accessibility based on proposed transportation improvements and in light of potential changes in land use. You may choose to define four scenarios as shown below:

	Network	
Land Use	Existing	Future
Existing	"Base"	"Transportation-only"
Future	"Land use-only"	"Combined"

Comparing the "Transportation-only" and "Land use-only" scenarios to the "Base" scenario provides insight into how much each component (transportation improvements or land development) can be expected to change accessibility over existing conditions. Comparing the "Combined" scenario against the others shows how synergies between transportation and land use interact to enhance accessibility above what can be accomplished through focusing only on transportation or land use.

1.3.3 Score scenarios

For a given scenario and for each mode, calculate accessibility scores. The calculation of scores itself is relatively simple, consisting of simple table operations, such as calculating a decay factor in a new column; joining activity data based on destination zone IDs; and summarizing accessibility activity, grouping by origin ID. See the figure below for a diagram illustrating these steps for three zones. Optionally, accessibility scores can be summarized for groups of origins, with averages weighted by population groups (to keep things simple, this is not shown in the diagram). Using the MMA geoprocessing toolbox, these steps are automated.

Accessibility scores, once developed, can be mapped to show heatmaps highlighting the most and least accessible zones in the study area. Comparisons across modes using ratios can also be mapped to show the contours of modal competitiveness within the study area.

See also:

- Geoprocessing Toolbox - Summarize Accessibility

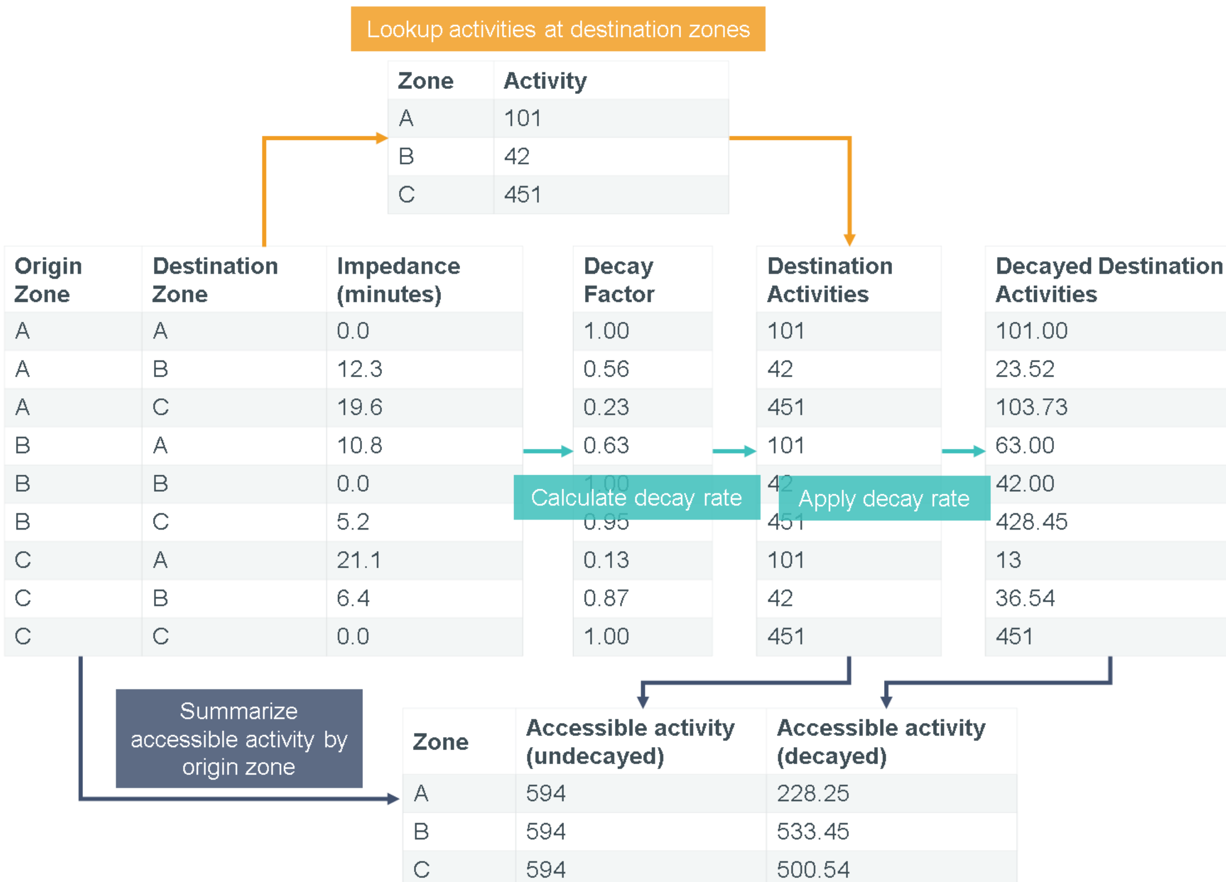
1.3.4 Compare scenarios

Once each scenario has been scored, comparisons across scenarios can be made. These comparisons may reveal how combined land use and transportation projects enhance accessibility, as described in the four-scenario example above. They may provide insight into how alternative project configurations or site locations impact accessibility and travel behavior. Or they may produce scores for ranking projects on a case-by-case basis to prioritize investments.

In all cases, comparisons among scenarios can be made for multi-zone areas to understand the average changes in accessibility that would be experienced by different population groups. Ideally, projects will benefit all populations and help connect disadvantaged groups to greater opportunity.

See also:

- Geoprocessing Toolbox - Calculate Change in Accessibility
- Geoprocessing Toolbox - Calculate Weighted Average



Zone	Activity
A	101
B	42
C	451

Origin Zone	Destination Zone	Impedance (minutes)	Decay Factor	Destination Activities	Decayed Destination Activities
A	A	0.0	1.00	101	101.00
A	B	12.3	0.56	42	23.52
A	C	19.6	0.23	451	103.73
B	A	10.8	0.63	101	63.00
B	B	0.0	1.00	42	42.00
B	C	5.2	0.95	451	428.45
C	A	21.1	0.13	101	13
C	B	6.4	0.87	42	36.54
C	C	0.0	1.00	451	451

Zone	Accessible activity (undecayed)	Accessible activity (decayed)
A	594	228.25
B	594	533.45
C	594	500.54

Fig. 7: *Once a skim has been created, the calculation of access scores is accomplished through a series of simple table joins and field calculations, as shown above. Results are summarized for each origin, making analyses and comparisons by population group possible.*

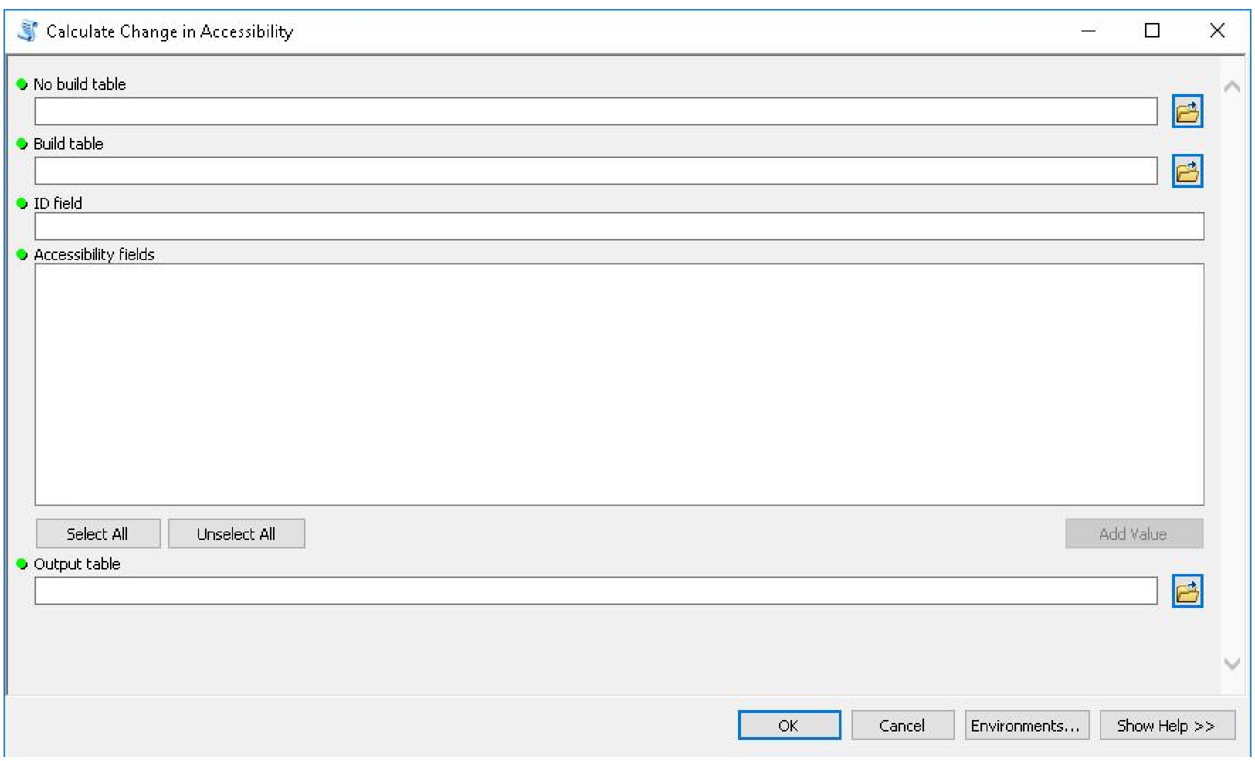
CHAPTER 2

MMA Geoprocessing Toolbox

2.1 Calculate Change in Accessibility

See also:

- MMA Process Fundamentals - Compare Scenarios



This is the geoprocessing interface for calculating the change in accessibility between two tables containing accessibility summaries for a set of zones under two alternative conditions or scenarios.

To understand how accessibility is modified by a project altering the transportation system and/or land uses, for example, provide a *no build table* reflecting baseline accessibility scores and a *build* table reflecting new accessibility scores assuming the project is implemented. For each zone affected by the project, subtract the no build scores from the build scores to determine the changes wrought by the project.

No build table [ArcGIS Table or Table View] The table containing accessibility scores for a set of zones in the "no build" or "baseline" condition.

Build table [ArcGIS Table or Table View] The table containing accessibility scores for a set of zones (matching those in the *no build table*) in the "build" or "change" condition.

ID field [Field] The field - present in both the *no build table* and the *build table* - containing zone IDs allowing records in the two tables to be related to each other.

Accessibility fields [[Field,. . .]] The fields containing accessibility scores for which the differences between the *build* and *no build* conditions will be calculated and stored in the *output table*. Field names must be the same in both input tables and will carry over to the *output table*.

Output table [ArcGIS Table] The output table storing the differences between the *build table* and the *no build table*.

`gp_calcChangeInAccessibility.`**`tableDifference`**(*table_1*, *table_2*, *id_field*, *diff_fields*, *output_table*, *skip_nulls=True*, *null_value=0*)

Given two tables of identical structure and similar content, calculate the differences (table_2 minus table_1) between values in a selection of fields for records identified by a common ID value in each table.

This function assumes two tables of identical structure and similar content as shown in the example below. They each have an ID field with at least some common ID values in both tables. They also share common numerical fields, for which the analyst wants to know the differences between values in each table for each distinct ID value. If an ID value is present in one table and not the other, it will be treated based on the provided *null_value* (default is 0).

Example table_1:

ID	Field 1	Field 2
1	1000	900
2	2000	600
9	3000	300

Example table_2:

ID	Field 1	Field 2
1	1200	750
2	2100	1019
3	3500	25

Example result table (table_2 minus table_1):

ID	Field 1	Field 2	*notes*
1	200	-150	*presnt in both tables*
2	100	419	*present in both tables*
3	3500	25	*present in table_2, not table_1*
9	-3000	-300	*present in table_1, not table_2*

Parameters

- **`table_1`** (*`ArcGIS Table or TableView`*) – A table that organizes data by distinct values in an *id_field* and containing measures in one or more numerical fields.

- **table_2** (*ArcGIS Table or Table View*) – A second table identical in structure to *table_1* with similar distinct values in the *id_field* and different values for measures, representing an alternative condition or scenario, e.g.
- **id_field** (*String*) – The field in *table_1* that is also in *table_2* that uniquely and consistently identifies records such that each table's rows can be directly joined and compared to each other.
- **diff_fields** (*[String, ..]*) – The list of numerical fields in *table_1* that are also in *table_2*, representing measures for which the differences between the tables are to be calculated (by distinct values in *id_field*).
- **output_table** (*String*) – The full path to the output table storing the differences (*table_2* minus *table_1*).
- **skip_nulls** (*Boolean, optional*) – If True, null (missing) values in each table will not be considered in the calculated of differences between *table_1* and *table_2*. If False, null (missing) values in each table will be included in the difference calculation and
- **null_value** (*Float, optional*) – The value to assume whenever a null (missing) value is found in a table. Default is 0. Onyl applies if skip_nulls is False.

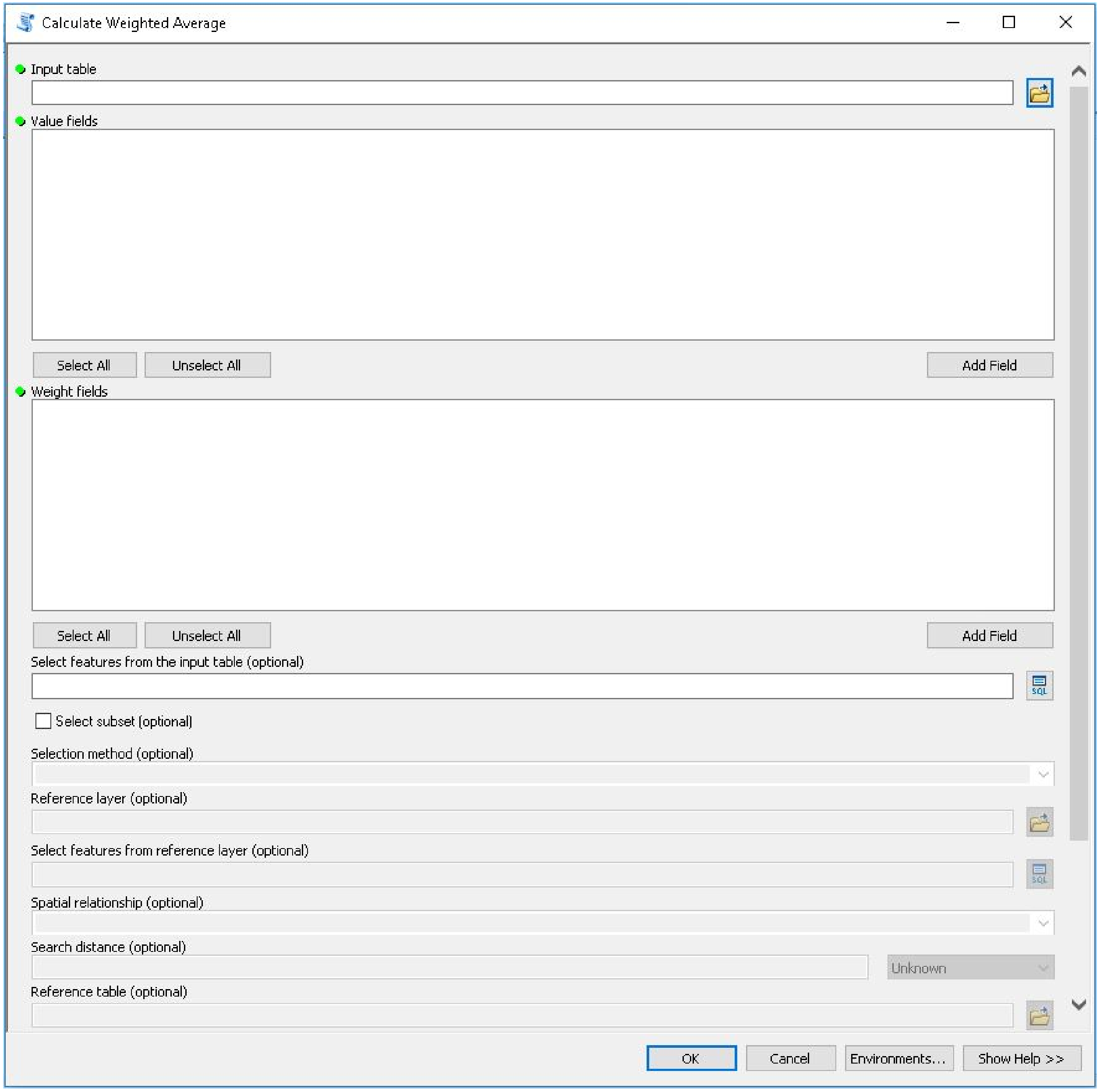
Returns Writes an output table with the differences between *table_1* and *table_2* .

Return type None

2.2 Calculate Weighted Average

See also:

- MMA Key Terms - Weighted Averages



This is the geoprocessing interface for calculating the weighted average of value(s) of a (set of) column(s) in a table based on the distribution of values in another (set of) column(s) in the table.

Determine an areawide average score or value based on sub-area distributions of activity (population-weighted average access to jobs, e.g.).

Input table [ArcGIS Table/Table View/Feature Class/Feature Layer] The table containing values for which averages are desired and weight fields for calculating the appropriate average values.

Value fields [[Field,. . .]] The fields in the *input table* for which the average value based on values in each *weight field* will be reported.

Weight fields [[Field,. . .]] The fields in the *input table* that will be applied to each *value field* as weights to determine the *value field's* average value across all rows in the table. Weighted average values for each *value field* are reported in separate rows for each *weight field*.

Select features from the input table [SQL Expression, optional] A SQL expression applied to the *input table* to focus the weighted average calculation on records matching the criteria defined by the expression.

Select subset [Boolean, optional] If True, a "secondary" table or feature class may be used to select records from the *input table* to focus the weighted average calculation on records matching criteria in the secondary table or having a specified spatial relationship to the secondary feature class. (Default is False.)

Selection method [{"SPATIAL", "TABULAR"}, optional] If *select subset* is True, specify whether the selection of records in the *input* table will be based on a spatial relationship to a secondary feature class ("SPATIAL") or on a lookup relationship to a secondary table ("TABULAR"). If "SPATIAL", *input table* must be a feature class or feature layer.

Reference layer [ArcGIS Feature Class or Feature Layer, optional] If *selection method* is "SPATIAL," the secondary feature class or feature layer to use as the basis for spatial selection of features in *input table*.

Select features from reference layer [SQL Express, optional] If *selection method* is "SPATIAL," optionally provide a SQL expression to limit the features in *reference layer* used for the spatial selection of features in *input table*.

Spatial relationship [String, optional] If *selection method* is "SPATIAL," define the spatial relationship to use when selecting features in *input table*. All ArcGIS overlap_types are valid.

Search distance [Linear Unit, optional] If *selection method* is "SPATIAL," define the search tolerance to guide the selection of features from the *input table* based on their *spatial relationship* to features in the *reference layer*. If blank, a strict spatial selection is applied.

Reference table [ArcGIS Table or Table View, optional] If *selection method* is "TABULAR," the secondary table or table view to use as the basis for tabular selection (through lookup) of records in *input table*.

Refrence table key field [Field, optional] If *selection method* is "TABULAR," the field in *reference table* containing values to lookup in *input table* (based on its values in the *input table lookup field*) for inclusion in the weighted average calculation.

Input table lookup field [Field, optional] If *selection method* is "TABULAR," the field in *input table* containing values corresponding to those in the *reference table key field* (in the *reference table*). Only *input table* records having values in this field that match those listed in the *reference table key field* will be included in the weighte average calculation.

Select records from reference table [SQL Expression, optional] If *selection method* is "TABULAR," optionally provide a SQL expression to limit the records in *reference table* used for the tabular lookup of records in *input table*.

Output table [ArcGIS Table] The output table storing the weighted averages. The output table is organized in to rows representing each *weight field* and columns containing the weighed average values for each *value field* for that *weight field*.

`gp_calcWeightedAverage.`**`weightedAverage`**(*table*, *value_fields*, *weight_fields*)

Calculate the average value(s) of a (list of) column(s) in a table, weighted by the values in another column or list of columns in the table.

Parameters

- **`table`** (*`ArcGIS Table or ArcGIS Table View`*) – The table with *value_fields* and *weight_fields* from which to calculate and tabulate weighted averages.
- **`value_fields`** (*`[String, ..]`*) – A list of field names whose values will be averaged.
- **`weight_fields`** (*`[String, ..]`*) – A list of field names whose values will provide weights that influence the averages calculated from *value_fields*.

Returns **out_array** – Returns an output array with weighted averages reported such that each *weight_field* is in its own row and each column provides the weighted average of each *value_field* as weighted by the *weight_field* reflected in that row.

Return type Numpy array

Notes

When values in a table are recorded in an aggregated manner such that a single record may represent a common condition or value for a collective, a simple average of column values for that table offers limited insight into typical conditions for members of that collective. In these cases, a weighted average is needed to describe the average value for the collective.

An example of this is geographic aggregation into zones. Each zone may have multiple people and different population groups residing within it. For measures calculated at the zone level, a simple average of those measures across zones will not reflect typical conditions for the people living in those zones. A weighted average takes into account the conditions as they apply on a *per person* basis rather than on a *per zone* basis.

See the example below showing commute times and distaces for male and female populations in three zones. The average commute time for males in the combined three-zone area cannot be calculated as the sum of the "commute_time" column divided by 3 (the zonal average commute time) because the distribution of the male population is not uniform across all zones. Most males live in Zone 1 with an estimated commute time of 26 minutes. Thus, the average for all males in the three-zone area will be closer to 26 minutes than to the 16 minutes shown for zone 2, where only 5 males reside. The weighted average commute time for males is actually 24.2, as shown in the example result array.

The average commute time for females differs from that for males because the distribution of females across the zones differs from the male population distribution.

Example input table

Zone_ID	Male_pop	Female_pop	Commute_time	Commute_dist
1	100	90	26	8.5
2	5	9	16	5.3
3	27	40	19	7.1

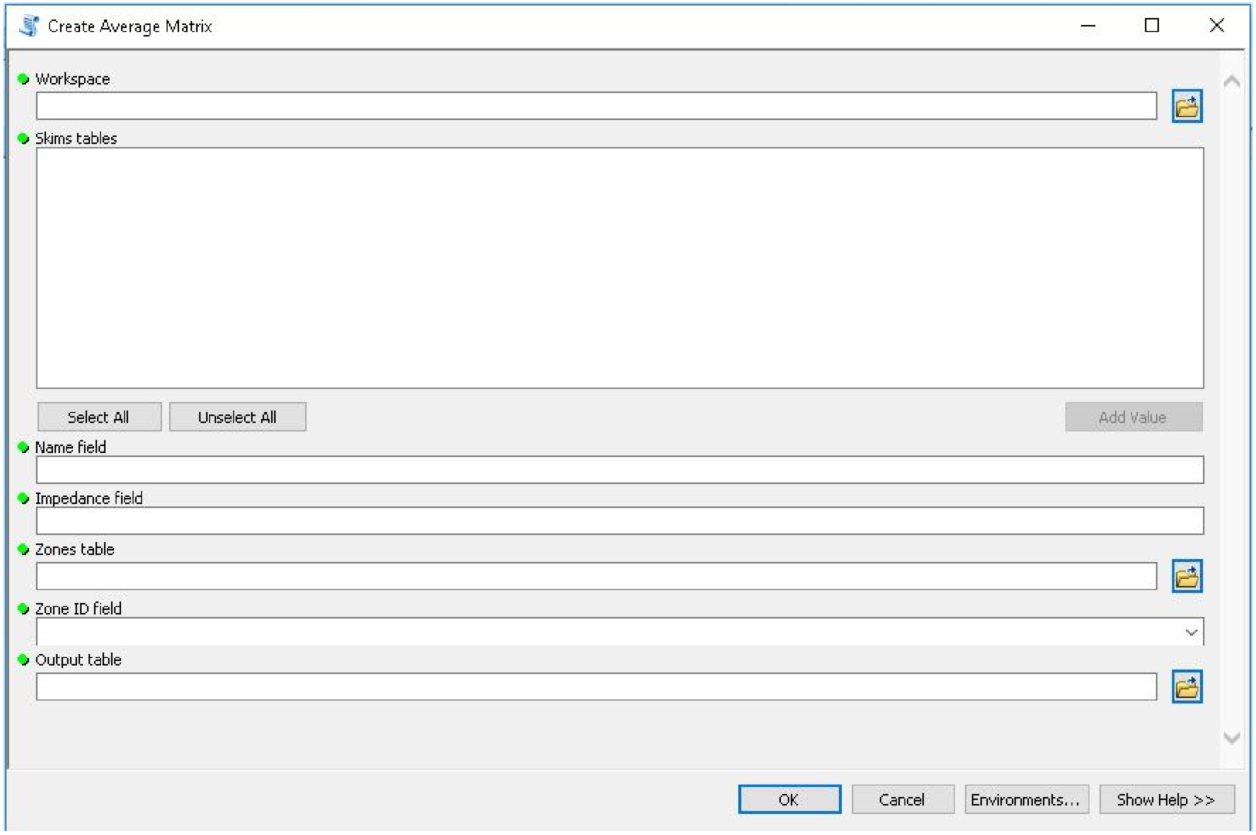
Example result array

WeightField	Commute_time	Commute_dist
Male_pop	24.2	8.1
Female_pop	23.3	7.9

2.3 Create Average Matrix

See also:

- Geoprocessing Toolbox - Create Skim (Time Window)



This is the geoprocessing interface for the createAverageMatrix function in the mma module.

If multiple skims are developed representing a consistent set of potential O-D pairs (using a travel time window or alternative network parameters, e.g.), it may be desirable to summarize the average impedance between each O-D pair describing a single "typical" impedance. The *createAverageMatrix* function facilitates such an analysis.

Workspace [ArcGIS workspace] ArcGIS Workspace object (file folder, geodatabase, etc.) or string representing the path to the workspace location where input skim tables are stored.

Skims tables [[String. . .]] A selection of file names within *workspace* to reference in developing average impedance values between each potential O-D pair listed in *zones_table*.

Name field [Field] The name of the field in each *skims table* file containing O-D zone ID information. The field must by a string field formatted as "{origin_id} - {destiantion_id}". It must be present in all *skims table* files to be analyzed.

Impedance field [Field] The name of the field in each *skims table* file containing impedance information. The field must have a numeric type. The field must have the same name in all *skims table* files to be analyzed.

Zones table [String] Path to an ArcGIS table or name of an ArcGIS table view in the active data frame containing the set of zones expected to be found in the *skims tables* files.

Zone ID field [String] The name of the field in *zones table* containing zone ID information. Unique values in this field are used to construct an O-D matrix that is used to calculated average travel times.

Output table [String] Full path to an ArcGIS table where the average O-D impedances will be stored.

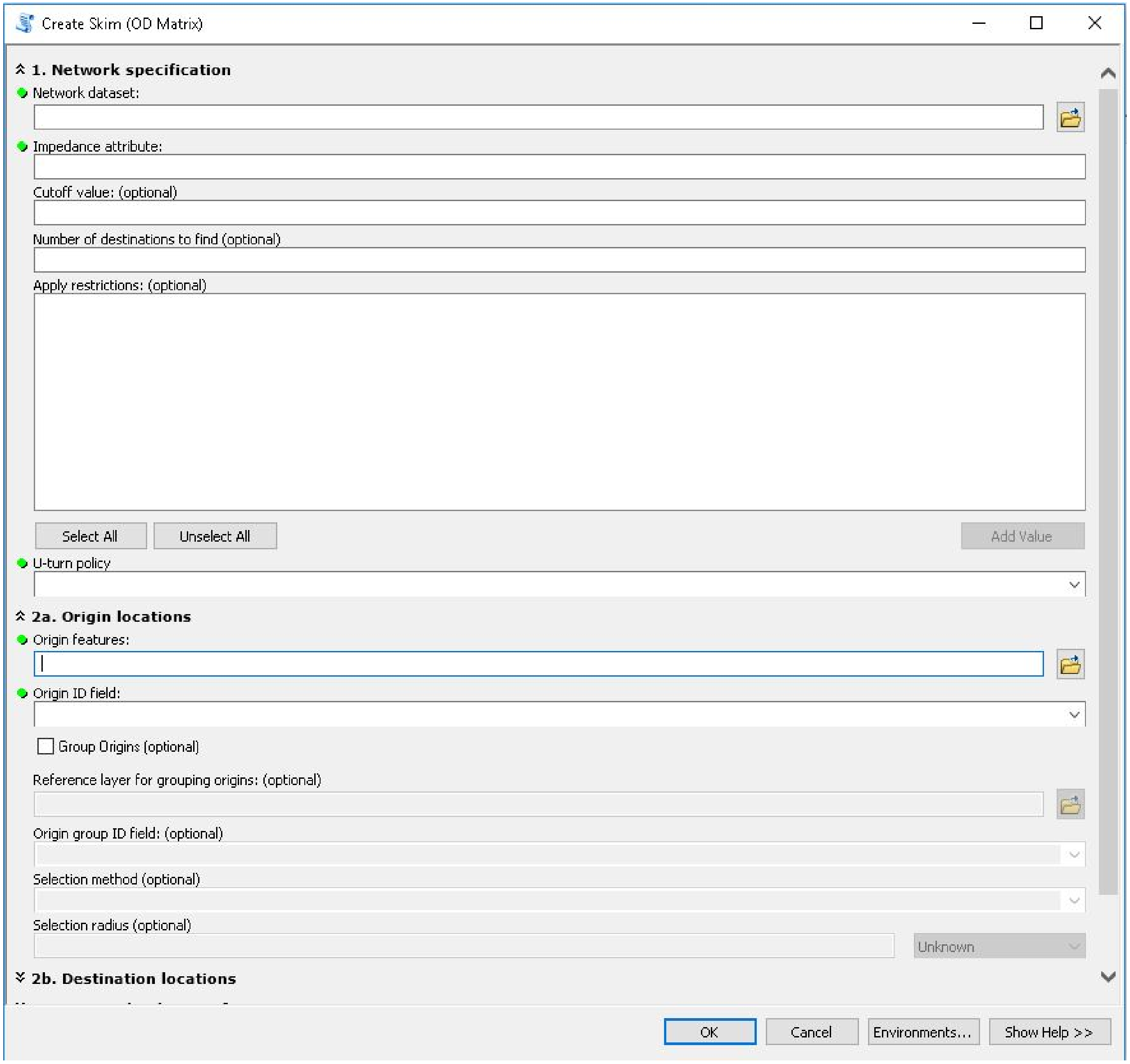
See also:

`mma.createAverageMatrix`

2.4 Create Skim

See also:

- MMA Key Terms - Skims



This is the geoprocessing interface for the summarizeAccessibility function in the mma module.

Network dataset [ArcGIS Network Dataset or Network Dataset Layer] The network dataset used to find shortest paths between *origin locations* and *destination locations*.

Impedance attribute [String] The name of the impedance attribute in *network dataset* to be used in determining shortest paths between *origin locations* and *destination locations* (options are shown with units listed alongside each impedance attribute's name).

Cutoff value [Float, optional] The maximum *impedance attribute* value from *origin locations* beyond which *destination locations* will not be tabulated in the skim. If no value is provided, no cutoff is applied. Applying a cutoff can reduce run times and focus the skim content on relevant destinations.

Number of destinations to find [Integer, Optional] The maximum number of *destination locations* to find for each *origin location.* If no value is provided, all destinations (within the *cutoff value*) will be found.

Apply restrictions [[String,. . .]] The restriction attributes in *network dataset* to honor when finding shortest paths ("Oneway;PedesetrianOnly" e.g.). If no restriction attributes exist in the *network dataset* this field in the geoprocessing interface will be empty.

U-turn policy [{"ALLOW_UTURNS", "NO_UTURNS", "ALLOW_DEAD_ENDS_ONLY", "ALLOW_DEAD_ENDS_AND_INTERSECTIONS_ONLY"}] The u-turn policy to honor when finding shortest paths.

Origin locations [ArcGIS Feature Class or Feature Layer] An ArcGIS point feature class or point feature layer in the active data frame representing origin locations to be recorded in the skim(s).

Origin ID field [Field] Field in *origin locations* to use as the origin ID value when tabulating travel times in the skim(s).

Group Origins [Boolean, optional] (Default is False.) If checked (true), origins will be grouped for processing. Grouping limits the number of features included in a given OD matrix tabulation to manage memory and output file sizes. If unchecked (false), all origins will be evaluated as a single group.

Reference layer for grouping origins [ArcGIS Feature Class or Feature Layer, optional] If *group origins* is checked (true), origins will be grouped based on the spatial relationship of features in *origin features* to features in this layer.

Origin group ID field [Field, optional] Name of the field in *reference layer for grouping origins* that organizes the grouping of *origin locations*. Distinct values in this field will be included in output file names to relate each skim table to its origin group.

Selection method [String, optional] The spatial relationship to apply when grouping *origin locations* based on *reference layer for grouping origins*. All ArcGIS overlap_types are valid.

Selection radius [Linear Unit, optional] The distance to search around *reference layer for grouping origins* for testing their spatial relationship to *origin locations*. If no value is provided a strict spatial relationship among featues will be applied (i.e., no search radius).

Destination features [ArcGIS Feature Class or Feature Layer] An ArcGIS point feature class or point feature layer in the active data frame representing destination locations to be recorded in the skim(s).

Destination ID field [Field] Field in *destination locations* to use as the destination ID value when tabulating travel times in the skim(s).

Use network locations [Boolean, optional] If checked (true), *origin locations* and *destination locations* will load on to the *network dataset* using pre-calculated values stored in various fields stored in their respective attribute tables. If unchecked (false), *origin locations* and *destination locations* will load on to the *network dataset* based on spatial criteria (this takes longer and can lead to inconsistencies in loading locations).

Origin SourceID/SourceOID/PosAlong/SideOfEdge/SnapX/SnapY/Distance field [Field, optional] If *use network locations* is checked (true), provide the names of the network location fields in the *origin locations* layer's attributes table. Each field specifies a portion of the pre-calculated network location.

Destination SourceID/SourceOID/PosAlong/SideOfEdge/SnapX/SnapY/Distance field [Field, optional] If *use network locations* is checked (true), provide the names of the network location fields in the *destination locations* layer's attributes table. Each field specifies a portion of the pre-calculated network location.

Search tolerance units [Linear unit, optional] If *use network locations* is unchecked (false), specify the maximum distance from a *network dataset* source listed in *network location search criteria* to search for *origin locations* and *destination locations* for loading. Features beyond the *search tolerance units* will be ignored during loading. (Default is "5000 Meters".)

Network location search criteria [[String,. . .]] If *use network locations* is unchecked (false), list the network sources and snapping points on which *origin locations* and *destination locations* may load.

Match to closest [Boolean, optional] If *use network locations* is unchecked (false), specify how to select loading locations based on *network dataset* sources listed in *network location search criteria*. If *match to closest* is checked (true), features will load on the closest valid source. If *match to closest* is unchecked (false), features will honor the priority of *network dataset* sources implied by the order in which they are listed in *network location search criteria*, loading on the closest in a given priority group.

Exclude restricted portions of the network [Boolean, optional] If *use network locations* is unchecked (false), specify whether *origin locations* and *destination locations* can be loaded on excluded *network* features. Excluded features are those honored as restricted as listed in *apply restrictions*. If *exclude restricted portions…* is checked (true), excluded features will be ignored during network loading. If *exclude restricted portions…* is unchecked (false), some locations may load on restricted features.

Additional criteria for loading on edges [String (edge source feature), optional] If *use network locations* is unchecked (false), optionally specify which *network dataset* features are available for loading. This field points to a particular edge source feature class for additional querying (see *search criteria* below).

Search criteria [SQL Expression, optional] If *use network locations* is unchecked (false), optionally specify which *network dataset* features are available for loading. This is an expression string to be applied to *additional criteria…* above that further constrains loading beyond the limits set by *network location search criteria*, *match to closest*, and *exclude restricted…* parameters.

Output workspace [ArcGIS workspace] ArcGIS Workspace (file folder, geodatabase, etc.) where output skim tables will be stored.

Analysis name [String] A string of characters to include in the names of output files to differentiate them from other files produced in the same *output workspace.* Short strings of 7 characters or fewer are recommended.

Use time of day [Boolean, optional] If checked (true) the *network dataset* is time-enabled and the user desires skims for a specific day and time(s). If unchecked (false), no differentiation by time of day will be considered.

Day of week [{"Today", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"}, optional] If *use_time_of_day* is checked (true), time of day differences will be based on the day of week specified here.

Time window start [Date/time, optional] If *use time of day* is checked (true), the first time on *day of week* to be analyzed. Multiple skims can be produced based on the *time window end* and *time window increment…* values.

Time window end [Date/time, optional] If *use time of day* is checked (true), the last time on *day of week* to be analyzed. Multiple skims can be produced based on the *time window end* and *time window increment…* values. If only a single time of day skim is required, set *time window end* equal to *time window start*.

Time window increment in minutes [Float, optional] If *use time of day* is checked (true), the interval at which to increment the time so that multiple skims will be produced for every interval of *time window increment* between *time window start* and *time window end*.

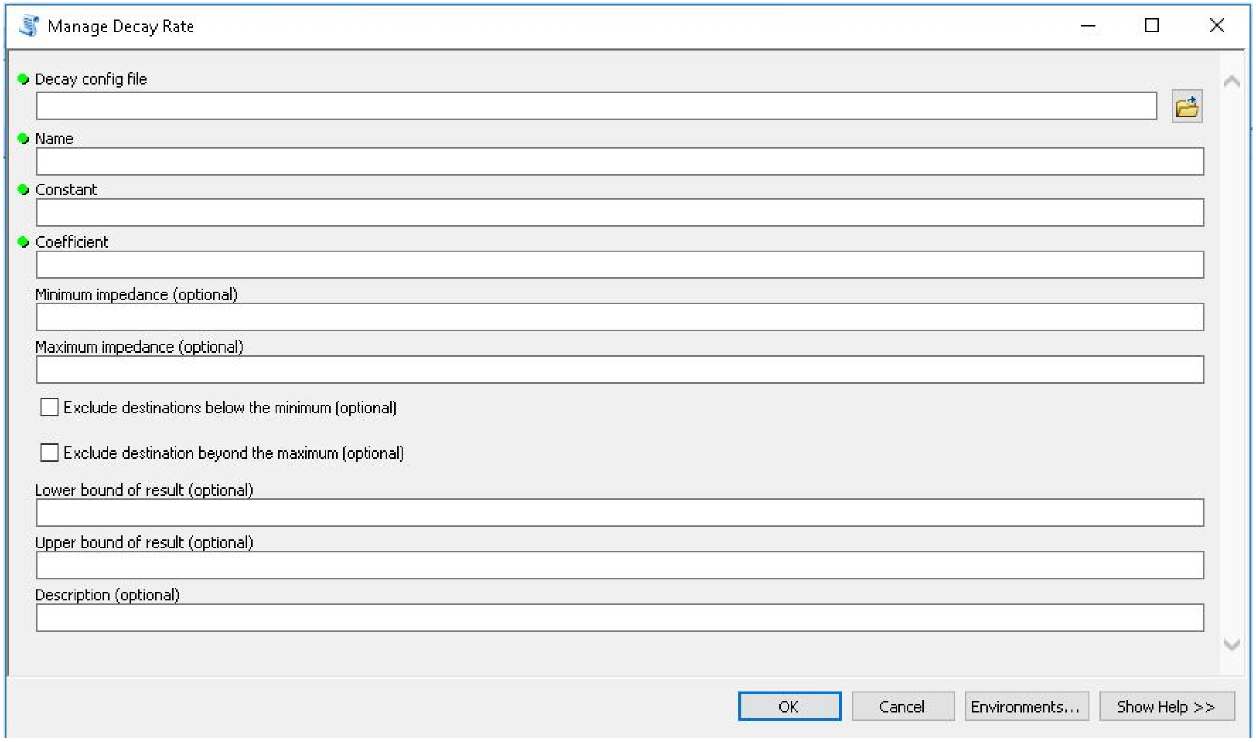
See also:

`mma.createSkims`, `mma.Skim`

2.5 Manage Decay Rate

See also:

- MMA Key Terms - Decay Rates



This is the geoprocessing interface for creating, storing, and modifying decay rate configuration files for use in MMA analyses.

Decay rates define how to discount activities at different destinations based on the cumulative impedance (usually travel time) from the origin.

Decay config file [File (.json)] The .json config file in which to store decay rate details for later processing during accessibility summarization.

Name A short name to identify the decay rate. The name is added to fields in the tables generated by the *Summarize Accessibility* tool, so a short string of a few characters is recommended.

Constant [Float] The constant term in the decay expression. Typical values are generally near or slightly above 1.0.

Coefficient [Float] The coefficient term in the decay expression. Typical values are less than zero.

Minimum impedance [Float, optional] The minimum impedance value to be evaluated as the *impedance* term in the decay formula. If blank, zero is assumed as the minimum impedance value.

Maximum impedance [Float, optional] The maximum impedance value to be evaluated as the *impedance* term in the decay formula. If blank, infinity is assumed as the maximum impedance value.

Exclude destinations below the minimum [Boolean, optional] Flag defining how to treat *impedance* values less than *minimum impedance*. If True, values less than *minimum impedance* are ignored. If False, values less than *minimum impedance* are treated as *minimum impedance*.

Exclude destinations beyond the maximum [Boolean, optional] Flag defining how to treat *impedance* values greater than *maximum impedance*. If True, values greater than *maximum impedance* are ignored. If False, values greater than *maximum impedance* are treated as *maximum impedance*.

Lower bound of result [Float, optional] Minimum value to return when evaluating the decay funtion specified by the *decay config file* with respect to a given impedance value. If no *lower bound* is provided, no lower bound constraint is placed on the decay formula (the fomula will typicall assymptotically approach zero when *constant* and *coefficient* values are in normal ranges.)

Upper bound of result [Float, optional] Maximum value to return when evaluating the decay funtion specified by the *decay config file* with respect to a given impedance value.

Description [String, optional] A longer description of the decay curve being defined. The *description* offers more detail than can be conveyed in the *name* parameter.

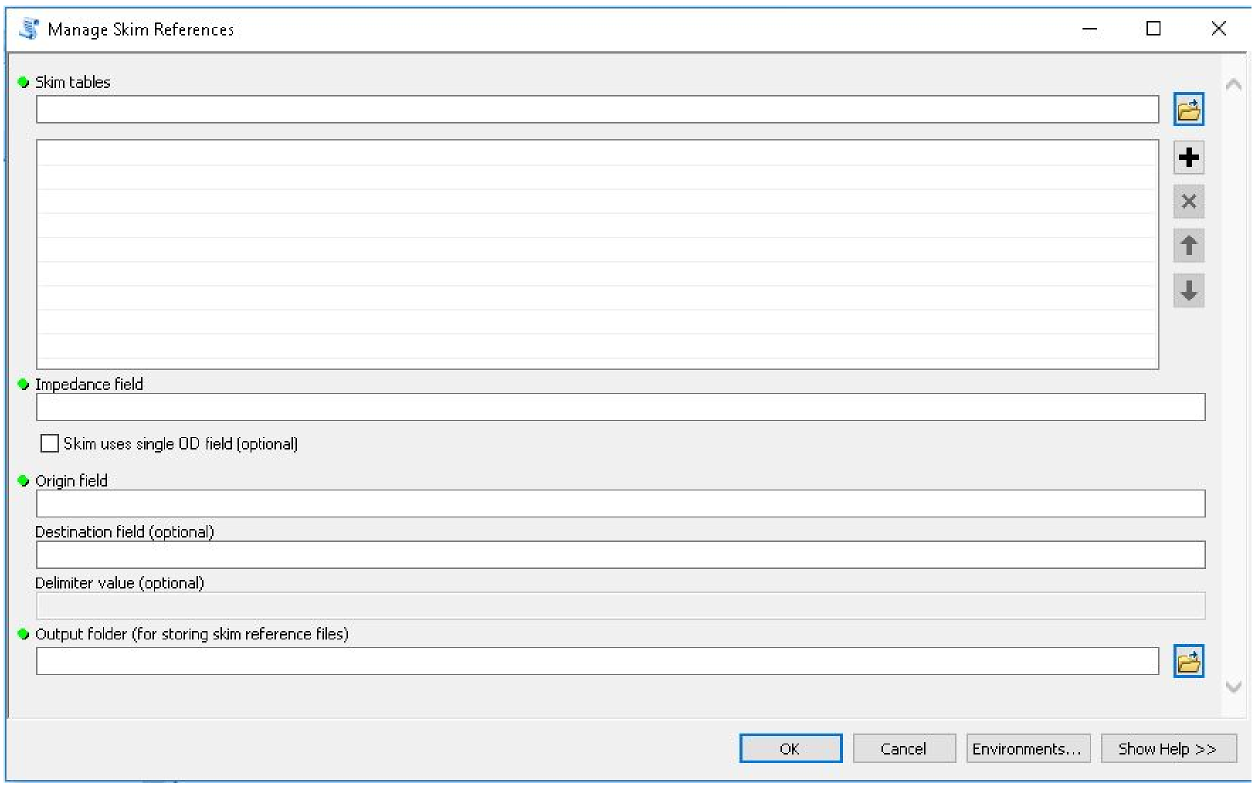
See also:

`mma.Decay`, `mma.decayToJson`, `mma.jsonToDecay`, `gp_summarizeAccessibility`

2.6 Manage Skim References

See also:

- MMA Key Terms - Skims



This is the geoprocessing interface for creating, storing, and modifying skim reference configuration files for use in MMA analyses.

Skim reference config files define how to create a *mma.Skim* object, detailing the key fields in a skim table and how to parse them when running an accessibility analysis. Skim reference config files are required to run the *summarizeAccessibility* geoprocessing tool.

Batches of skim reference config files may be created by running this tool, but all *skim tables* must have identical structures. For multiple tables with variable structure, the tool must be run multiple times.

Skim tables [[ArcGIS Table or Table View,. . .]] A list of skim tables for which skim reference config files will be created. All tables in the list must have identical structures.

Impedance field [Field] The field in each *skim table* containing the operative cumulative impedance (usually travel time) value for calculating accessibility.

Skim uses single OD field [Boolean, optional] If checked (True), the *skim tables* have origin-destination ID's organized in a single field. Skim tables produced by the *gp_createSkims* geoprocessor, for example, include a "Name" field which containts both origin and destination ID values, separated by the character string " - ". If unchecked (False), the *skim tables* have separate columns for origin and destination ID's.

Origin field [Field] If *skim uses single OD field* is false, the field in each *skim table* that identifies the origin zone associated with each record. If *skim uses single OD field* is true, this is the name of the field containing combined origin and destination ID values.

Destination field [Field, optional] If *skim uses single OD field* is false, the field in each *skim table* that identifies the destination zone associated with each record. If *skim uses single OD field* is true, this parameter is ignored.

Delimiter value [String, optional] If *skim uses single OD field* is true, the character string used to appropriately parse origin and destination IDs from the combined values stored in *origin field*. If *skim uses single OD field* is false, this parameter is ignored.

Output folder [Folder] The folder where all skim reference config files generated by the tool will be stored. A skim reference config file (.json) will be created for each table listed in *skim tables*. Each output skim reference config file takes its name from the corresponding *skim table* it describes and includes an absolute path reference to that *skim table*.

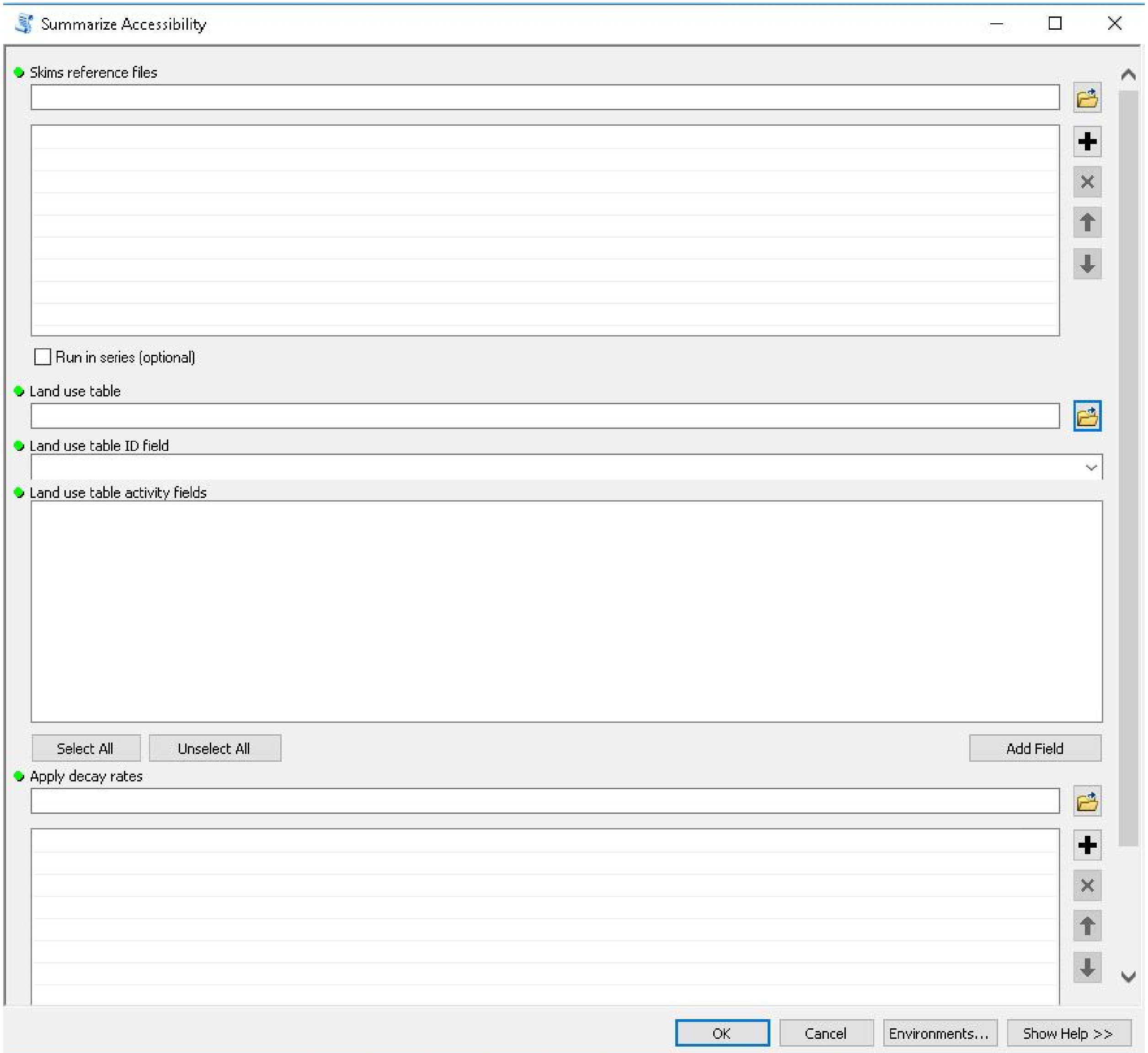
See also:

`mma.Skim`, `mma.jsonToSkim`, `mma.skimToJson`, `gp_summarizeAccessibility`

2.7 Summarize Accessibility

See also:

- MMA in Concept
- MMA Process Fundamentals - Score Scenarios



This is the geoprocessing interface for the summarizeAccessibility function in the mma module.

This is the final step in developing accessibility scores by zone using a skim file (or set of skim files) and a table of land use data.

Skims reference files [File (.json)] A (list of) skim reference file(s)

Run in series [Boolean, optional] If checked, the accessibility results for each *skim reference file* will be saved to a distinct output table bearing that *skim reference file*'s name in the *output workspace*. If unchecked, accessibility results for all *skim reference files* will be consolidated in a single *output table*

Land use table [ArcGIS Table or Table View] A table that organizes land use data by zones with an ID field that corresponds to values in each skim table's (*skim reference file*'s) destination ID values.

Land use table ID field [Field] The field in *Land use table* that identifies each zone. Values in this field should correspond to values in each skim table's (*skim reference file*'s) destination ID values.

Land use table activity fields [[Field,. . .]] A list of field names representing the activities at each (destination) zone to summarize in the accessibility tabulation.

Apply decay rates [File (.json)] A (list of) decay file(s) defining decay rates to be applied in the summarization of activities in *lu_table_activity_fields*. Decay rates define how destination-end activities should be discounted based on the impedance from the origin.

Output table [ArcGIS Table, optional] If *Run in series* is unchecked, a single *Output table* will be generated summarizing accessibility from all origins listed in the *skim reference files* based on the travel times recorded in the skims, the *land use activity table fields*, and any *decay rates* applied.

Output workspace [ArcGIS workspace, optional] If *Run in series* is checked, an *Output table* will be generated for each *skim refernce file*, summarizing accessibility from all origines listed in that skim based on the travel times recorded in the skim, the *land use activity table fields*, and any *decay rates* applied.

See also:

`mma.summarizeAccessibility`, `mma.Skim`, `mma.Decay`

2.8 mma python module

Note: Dependencies

- arcpy (installs with ArcGIS) v. 10.2 or later
- numpy v. 1.11.2 or later
- pandas v. 0.18.1 or later

This module facilitates many procedures required to develop multimodal accessibility (mma) scores in ArcGIS.

The MMA module allows users to generate network analysis problems to create (series of) travel time skims and process those skims against related zone data and decay curves to generate accessibility scores for each zone.

Key functions:

- createAverageMatrix()
- createSkims()
- summarizeAccessibility()

Key classes:

- Decay
- Skim
- SkimSet

Notes

Accessibility scores describe the quantity of "activities" reachable within a given impedance (usually travel time) by a given mode, discounted by impedance tolerances for that mode. Key steps in generating accessibility scores include:

- Configure decay rates
- Create skims (OD matrices)
- Summarize accessibility for a table of land uses

The mma module relies on the following dependencies:

- arcpy (ArcGIS Network Analyst extension must be installed, licensed, and enabled)

- numpy
- pandas

2.8.1 Classes and Functions

class mma.**Decay**(*name*, *const*, *coef*, *min_impedance=0*, *max_impedance=inf*, *excl_less_than_min=False*, *excl_greater_than_max=True*, *lbound=0.0*, *ubound=1.0*, *desc=''*)

The Decay class defines how to discount activities at different destinations based on the cumulative impedance (usually travel time) from the origin.

The Decay object assumes a (negative) exponential formula that defines a curve for discounting activities at destinations based on their cumulative impedance (usually travel time) from the origin. The basic decay formula is:

$$self.const * (e^{self.coef*impedance})$$

It can be modified such that the value of *impedance* and/or the results of the formula can be constrained by minimum/maximum values as specified in various attributes.

Parameters

- **name** (*String*) – A short name to identify the decay rate.
- **const** (*Float*) – The constant term in the decay expression. Typical values are generally near or slightly above 1.0.
- **coef** (*Float*) – The coefficient term in the decay expression. Typical values are less than zero.
- **min_impedance** (*Float, optional*) – Default is 0, which implies no minimum value.
- **max_impedance** (*Float, optional*) – Default is float('inf'), which implies no maximum value.
- **excl_less_than_min** (*Boolean, optional*) – Default is False (values less than *min_impedance* are treated as *min_impedance*)
- **excl_greater_than_max** (*Boolean, optional*) – Default is True (values greater than *max_impedance* are ignored).
- **lbound** (*Float, optional*) – Default is None.
- **ubound** (*Float, optional*) – Default is None.
- **desc** (*String, optional*) – Default is None.

name

A short name to identify the decay rate. The name is added to fields in the tables generated by the *summarizeAccessibility* function, so a short string of a few characters is recommended.

Type String

const

The constant term in the decay expression.

Type Float

coef

The coefficient term in the decay expression.

Type Float

min_impedance
The minimum impedance value to be evaluated as the *impedance* term in the Decay formula. Values less than *min_impedance* are ignored if *excl_less_than_min* is True. Otherwise, values less than *min_impedance* are treated as the *min_impedance* value.

Type Float

max_impedance
The maximum impedance value to be evaluated as the *impedance* term in the Decay formula. Values greater than *max_impedance* are ignored if *excl_greater_than_max* is True. Otherwise, values greater than *max_impedance* are treated as the *max_impedance* value.

Type Float

excl_less_than_min
Flag defining how to treat *impedance* values less than *min_impedance*. If True, values less than *min_impedance* are ignored. If False, values less than *min_impedance* are treated as *min_impedance*.

Type Boolean

excl_greater_than_max
Flag defining how to treat *impedance* values greater than *max_impedance*. If True, values greater than *max_impedance* are ignored. If False, values greater than *max_impedance* are treated as *max_impedance*.

Type Boolean

lbound
Minimum value to return when evaluating the decay funtion specified by the Decay object with respect to a given value of *impedance*.

Type Float

ubound
Maximum value to return when evaluating the decay funtion specified by the Decay object with respect to a given value of *impedance*.

Type Float

desc
A longer description of the decay curve being defined. The *desc* parameter offers more detail than can be conveyed in the *name* parameter.

Type String

See also:

summarizeAccessibility, *decayToJson*, *jsonToDecay*

evaluate(*impedance*)
Estimates the decay weight to return based on the Decay object's attributes and a float value *impedance*

Parameters impedance (*Float*) – A non-negative numeric value representing cumulative impedance (usually travel time) between an O-D pair.

Returns decay_factor – A weighting factor that reflects the discount to apply when weighting destinations in accessibility analyses based on the Decay object's attributes and the cumulative impedance between an O-D pair.

Return type Float

class `mma.`**`Skim`**(*path*, *table*, *impedance_field*, *o_field*, *d_field=None*, *delimiter=None*)

The Skim class describes an OD matrix (skim) table to guide the mma module in parsing the table during accessibility processing.

Skims tabluate impedances (usually travel times) between pairs of zones. These data are stored in tables in which each row represents an O-D pair and an impedance field defines the impedance between the two zones. A Skim object exposes the path location, key fields, and type (arcpy vs text) of a skim table for parsing during accesibility analysis.

Parameters

- **path** (*String*) – The workspace path in which the skim table is stored.
- **table** (*String*) – The name of the skim table (with file extension, if any).
- **impedance_field** (*String*) – The field in the skim table containing the operative cumulative impedance (usually travel time) value.
- **o_field** (*String*) – The field in the skim table that identifies the origin zone **or** the field that provides the full origin- destination pair names.
- **d_field** (*String, optional*) – The field in the skim table that identifies the destination zone. (Default is None, which implies that *o_field* gives the full origin-destination pair names.)
- **delimiter** (*String, optional*) – If *o_field* gives the full origin-destination pair names, *delimeter* indicates the character sequance by which to split *o_field* values to get separate origin and destination ID's. (Default is None, implying o_field and d_field have been provided. Note that *delimiter* can't be None if *d_field* is also None.)

path

The workspace path in which the skim table is stored.

Type String

table

The name of the skim table (with file extension, if any).

Type String

impedance_field

The field in the skim table containint the operative cumulative impedance (usually travel time) value.

Type String

o_field

The field in the skim table that provides the origin value **or** the field that provides the full origin- destination pair names.

Type String

d_field

The field in the skim table that identifies the destination zone.

Type String

delimiter

Indicates the character sequance by which to split *o_field* values to get separate origin and destination ID's when *o_field* provides full O-D pair names.

Type String

o_idx

Expected index of *o_field* in a cursor navigating *table*

Type Integer

d_idx
Expected index of *d_field* in a cursor navigating *table*

Type Integer

imp_idx
Expected index of *impedance_field* in a cursor navigating *table*

Type Integer

table_type
Indicates the appropriate method for navigating the table for processing. –Probable deprecation.–

Type {"arcpy", "text"}

See also:

createSkims, *SkimSet*, *skimToJson*, *jsonToSkim*, *gp_manageSkimRef*

getSkimFields()
Returns a list of skim fields to pass to cursor for processing.

setDField(*d_field*)
Set the field in *table* that identifies destination zones. Can be None if *o_field* contains full O-D pair names.

setImpedanceField(*impedance_field*)
Set the field in *table* that represents the operative cumulative impedance (usually total travel time) between two zones.

setOField(*o_field*)
Set the field in *table* that identifies origin zones **or** that identifies the full O-D pair name.

setPathAndTable(*path*, *table*)
Set the the Skim object's *path* and *table* attributes. *table* must already existing in *path*.

class mma.**SkimSet**
The SkimSet class groups one or many Skim objects for use in batch accessibility summarization.

skims
The Skim objects that comprise the SkimSet

Type [Skim. . .]

See also:

Skim

addSkim(*skim_obj*)
Adds a skim_obj to the SkimSet's *skims* list.

removeSkim(*path*, *table*)
Remove a skim_obj from the SkimSet's *skims* list based on its *path* and *table* attributes.

mma.**createAverageMatrix**(*skims_ws*, *skims_files*, *name_field*, *impedance_field*, *zones_table*, *zone_id_field*, *output_table*)
Tabulates average O-D impedances (usually travel times) based on a set of skims representing variable O-D travel times (by time of day, for instance).

If multiple skims are developed representing a consistent set of potential O-D pairs (using a travel time window or alternative network parameters, e.g.), it may be desirable to summarize the average impedance between each O-D pair describing a single "typical" impedance.

Parameters

- **skims_ws** (*ArcGIS workspace or string*) – ArcGIS Workspace object (file folder, geodatabase, etc.) or string representing the path to the workspace. Location where input skim tables are stored.
- **skims_files** (*[String...]*) – List of strings representing file names (with extensions) within *skims_ws* to reference in developing average impedance values between each potential O-D pair listed in *zones_table*.
- **name_field** (*String*) – The name of the field in each *skims_file* containing O-D zone ID information. The field must by a string field formatted as "origin_id - destiantion_id". It must be present in all *skims_files* to be analyzed.
- **impedance_field** (*String*) – The name of the field in each *skims_file* containing impedance information. The field must have a numeric type. The field must have the same name in all *skims_files* to be analyzed.
- **zones_table** (*String*) – Path to an ArcGIS table or name of an ArcGIS table view in the active data frame containing the set of zones expected to be found in the *skims_files*.
- **zone_id_field** (*String*) – The name of the field in *zones_table* containing zone ID information. Unique values in this field are used to construct an O-D matrix that is used to calculated average travel times.
- **output_table** (*String*) – Full path to an ArcGIS table where the average O-D impedances will be stored.

Returns Writes an output skim table with average impedance values by OD pair to *output_table*.

Return type None

Notes

Skim tables to be averaged are assumed to reside in a single workspace.

Skim tables to be averaged are assumed to contain consistent field names. All are assumed to have *name* field containing values formatted as "origin_id - destination_id" and an *impedance* field containing non-negative numerical values.

The averaging process generally takes the sum of the impedance values recorded in each *skims_file* for each OD pair and divides it by the number of *skims_files* being analyzed.

In the event an OD pair is present in one *skims_file* and not another, it is assumed the impedance between the two zones is infinity in the file(s) for which the value is missing. This accomplished using inversion. All impedance values are stored as recipricals, such that OD pairs with no impedance are given a value of 1.0, missing OD pairs are given a value of 0.0, and all OD pairs' impedance values are stored as a value between 0.0 and 1.0. These values are then summed and normalized by the number of *skims_files* being analyzed. The resulting value is then re-inverted to obtain the average typical impedance between each OD pair.

See also:

`Skim()`, `gp_averageTravelTime()`

mma.**createSkims**(*network, impedance_attribute, o_features, o_name, d_features, d_name, output_workspace, analysis_name, cutoff=None, number_of_ds=None, restrictions=None, u_turns='ALLOW_UTURNS', group_origins=False, group_features=None, group_id_field=None, group_selection_method='INTERSECT', group_selection_radius='', use_network_locations=False, o_SourceID='', o_SourceOID='', o_PosAlong='', o_SideOfEdge='', o_SnapX='', o_SnapY='', o_Distance='', d_SourceID='', d_SourceOID='', d_PosAlong='', d_SideOfEdge='', d_SnapX='', d_SnapY='', d_Distance='', search_criteria='', tolerance='5000 Meters', match=True, exclude_restricted=True, query_layer='', search_query='', use_time_of_day=False, day_of_week='Today', time_window_start='', time_window_end='', time_window_increment=1*)

Create a (set of) OD travel time matrix tables.

Skims tabluate travel times between pairs of zones. This function generates output tables in which each row represents an O-D pair and an impedance column defines the travel time between the two zones. Numerous optional parameters are provided representing location loading preferences, time window settings, groupings of origins for data management needs, etc.

Parameters

- **network** (*ArcGIS Network Dataset or Network Dataset Layer*) – Path to an ArcGIS network dataset or name of an ArcGIS network dataset layer in the active data frame.
- **impedance_attribute** (*String*) – The name of the impedance attribute in *network* to be used in determining shortest paths between *o_features* and *d_features*.
- **o_features** (*ArcGIS Feature Class or Feature Layer*) – Path to an ArcGIS point feature class or name of an ArcGIS point feature layer in the active data frame representing origin locations to be recorded in the skim(s).
- **o_name** (*String*) – Field in *o_features* to use as the origin ID value when tabulating travel times in the skim(s).
- **d_features** (*ArcGIS Feature Class or Feature Layer*) – Path to an ArcGIS point feature class or name of an ArcGIS point feature layer in the active data frame (can be the same FC/FL as *o_features*).
- **d_name** (*String*) – Field in *d_features* to use as the destination ID value when tabulating travel times.
- **output_workspace** (*ArcGIS Workspace or string*) – ArcGIS Workspace object (file folder, geodatabase, etc.) or string representing the path to the workspace. Location where output skim tables will be stored.
- **analysis_name** (*String*) – A string of characters to include in the names of output files to differentiate them from other files produced in the same *output_workspace.* Short strings of 7 characters or fewer are recommended.
- **cutoff** (*Double, optional*) – The maximum *impedance_attribute* value from *o_features* beyond which *d_features* will not be tabulated in the skim. (Default is None, which implies no maximum *impedance_attribute* value.)
- **number_of_ds** (*Integer, optional*) – The maximum number of *d_features* to find for each *o_feature.* (Default is None, which finds all origins within *cutoff.*)
- **restrictions** (*String, optional*) – (Default is None.) The restriction attributes in *network* to honor when finding shortest paths as a semi-colon-separated string. *Example : "Oneway;PedesetrianOnly"*

- **u_turns** (*{"ALLOW_UTURNS", "NO_UTURNS", "ALLOW_DEAD_ENDS_ONLY", "ALLOW_DEAD_ENDS_AND_INTERSECTIONS_ONLY" }*) – The u-turn policy to honor when finding shortest paths.

Returns Nothing is retured by the function. It will output one or more skim tables to the *output_workspace* with names reflecting *analysis_name*, *group_id_field* (if any), and the time window (if any). For each output skim, a skim reference configuration file (.json) will also be created in {*output_workspace*_skim_references}

Return type None

Other Parameters

- **group_origins** (*Boolean, optional*) – (Default is False.) True indicates that origins should be grouped for processing. Grouping limits the number of features included in a given OD matrix tabulation to manage memory and output file sizes.
- **group_features** (*ArcGIS Feature Class or Feature Layer, optional*) – (Deafult is None.) If *group_origins* is True, origins will be grouped based on the relationship of features in *o_features* to features in *group_features*.
- **group_id_field** (*String, optional*) – (Default is None.) Name of the field in *group_features* that organizes the grouping of *o_features*. Distinct values in this field will be included in output file names to relate each skim table to its origin group.
- **group_selection_method** (*String, optional*) – The spatial relationship to apply when grouping *o_features* based on *group_features*. (Default is "INTERSECT", which relates *o_features* to *group_features* that they intersect. All ArcGIS overlap_types are valid.)
- **group_selection_radius** (*Linear Unit (String), optional*) – The distance to search around *group_features* for testing their spatial relationship to *o_features*. (Default is "", which implies a strict relationship among featues [i.e., no search radius].) *Example: "100 Feet"*
- **use_network_locations** (*Boolean, optional*) – (Default is False.) True indicates *o_features* and *d_features* will load on to the *network* using pre-calculated values stored in various fields stored in their respective attribute tables. False indicates *o_features* and *d_featues* will load on to the *network* based on spatial criteria (this takes longer and can lead to inconsistencies in loading locations).
- **o_SourceID** (*String, optional*) – (Default is "".) If *use_network_locations* is True, provide the name of the SourceID field in *o_features*
- **o_SourceOID** (*String, optional*) – (Default is "".) If *use_network_locations* is True, provide the name of the SourceOID field in *o_features*
- **o_PosAlong** (*String, optional*) – (Default is "".) If *use_network_locations* is True, provide the name of the PosAlong field in *o_features*
- **o_SideOfEdge** (*String, optional*) – (Default is "".) If *use_network_locations* is True, provide the name of the SideOfEdge field in *o_features*
- **o_SnapX** (*String, optional*) – (Default is "".) If *use_network_locations* is True, provide the name of the SnapX field in *o_features*
- **o_SnapY** (*String, optional*) – (Default is "".) If *use_network_locations* is True, provide the name of the SnapY field in *o_features*
- **o_Distance** (*String, optional*) – (Default is "".) If *use_network_locations* is True, provide the name of the Distance field in *o_features*
- **d_SourceID** (*String, optional*) – (Default is "".) If *use_network_locations* is True, provide the name of the SourceID field in *d_features*

- **d_SourceOID** (*String, optional*) – (Default is "".) If *use_network_locations* is True, provide the name of the SourceOID field in *d_features*
- **d_PosAlong** (*String, optional*) – (Default is "".) If *use_network_locations* is True, provide the name of the PosAlong field in *d_features*
- **d_SideOfEdge** (*String, optional*) – (Default is "".) If *use_network_locations* is True, provide the name of the SideOfEdge field in *d_features*
- **d_SnapX** (*String, optional*) – (Default is "".) If *use_network_locations* is True, provide the name of the SnapX field in *d_features*
- **d_SnapY** (*String, optional*) – (Default is "".) If *use_network_locations* is True, provide the name of the SnapY field in *d_features*
- **d_Distance** (*String, optional*) – (Default is "".) If *use_network_locations* is True, provide the name of the Distance field in *d_features*
- **search_criteria** (*String, optional*) – If *use_network_locations* is False, list the network sources and snapping points on which *o_features* and *d_features* may load in a semi-colon-separated string. *Example: "Streets Midpoint;Streets Endpoint"*
- **tolerance** (*Linear Unit (String), optional*) – If *use_network_locations* is False, specify the maximum distance from a *network* source listed in *search_citeria* to search for *o_features* and *d_features* for loading. Features beyond the *tolerance* will be ignored during analysis. (Default is "5000 Meters".)
- **match** (*Boolean, optional*) – If *use_network_locations* is False, specify how to select loading locations based on *network* sources listed in *search_criteria*. If *match* is True, features will load on the closest valid source. This is the deafult. If *match* is False, features will honor the priority of *network* sources implied by the order in which they are listed in *search_criteria*, loading on the closest in a given priority group.
- **exclude_restricted** (*Boolean, optional*) – If *use_network_locations* is False, specify whether *o_features* and *d_features* can be loaded on excluded *network* features. Excluded features are those honored as restricted as listed in *restrictions*. If *exclude_restricted* is True, excluded features will be ignored. This is the default. If *exclude_restricted* is False, some locations may load on restricted features.
- **search_query** (*String, optional*) – If *use_network_locations* is False, optionally specify which *network* features are available for loading. This is an expression string that further constrains loading beyond the limits set by *search_criteria*, *match*, and *exclude_restricted*. (Default is "".)
- **use_time_of_day** (*Boolean, optional*) – (Default is False.) True indicates that *network* is time-enabled and the user desires skims for a specific day and time. If False, no differentiation by time of day will be considered.
- **day_of_week** (*{"Today", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"}, optional*) – If *use_time_of_day* is True, indicate the day of week to analyze.
- **time_window_start** (*DateTime, optional*) – If *use_time_of_day* is True, indicate the initial time to analyze. Multiple skims may be produced based on the *time_window_end* and *time_window_increment* values.
- **time_window_end** (*DateTime, optional*) – If *use_time_of_day* is True, indicate the last time to analyze. Multiple skims may be produced based on the *time_window_start* and *time_window_increment* values. If only a single time of day skim is required, set *time_window_end* equal to *time_window_start*.

- **time_window_increment** (*Double, optional*) – If *use_time_of_day* is True, indicate the interval at which to increment the time so that multiple skims will be produced for every interval of *time_window_increment* between *time_window_start* and *time_window_end*.

See also:

Skim(), *skimToJson()*, *jsonToSkim()*, *SkimSet()*

mma.**decayToJson**(*decay_obj*, *output_file*)
Stores a Decay object in a JSON config file.

Parameters

- **decay_obj** (Decay) – A Decay object defines how to discount activities at different destinations based on the travel time from the origin.
- **output_file** (*String*) – The full path to the output file. If just the file name is passed, the file will be saved in the current working directory.

Returns Writes a decay config JSON file to the *output_file*.

Return type None

See also:

jsonToDecay(), *Decay()*

mma.**jsonToDecay**(*in_file*)
Creates a Decay object for accessibility processing from a JSON config file.

Parameters **in_file** (*string*) – The file name in the current working directory **or** full path to the JSON config file from which to generate the Decay object.

Raises KeyError – If *in_file* is not a valid Decay config file.

Returns

Return type *Decay*

See also:

decayToJson(), *Decay()*

mma.**jsonToSkim**(*in_file*)
Creates a Skim object for accessibility processing from a JSON config file.

Parameters **in_file** (*String*) – The file name in the current working directory **or** full path to the JSON config file from which to generate the Skim object.

Raises KeyError – If *in_file* is not a valid Skim config file.

Returns

Return type *Skim*

See also:

skimToJson(), *Skim()*

mma.**skimToJson**(*skim_obj*, *output_file*)
Stores a Skim object in a JSON config file.

Parameters

- **skim_obj** (Skim) – A Skim object desribes an OD matrix (skim) table to guide the mma module in parsing the table during accessibility processing.

- **output_file** (*String*) – The full path to the output file. (If just the file name is passed, the file is saved in the current working directory.)

Returns Writes a skim config JSON file to the *output_file*.

Return type None

See also:

jsonToSkim(), *Skim()*

mma.**summarizeAccessibility**(*skim_set_obj*, *lu_table*, *lu_table_id*, *lu_table_activity_fields*, *out_table*, *decays=[]*)

Summarizes access to activities at desinations based on skims in a SkimSet object, a land use table, and a list of Decay objects.

Parameters

- **skim_set_obj** (SkimSet) – A SkimSet object comprised of the Skim objects describing the skim tables to reference in accessibility summarization.
- **lu_table** (*ArcGIS Table or TableView*) – A table that organizes land use data by zones with an ID field that corresponds to values in each skim table's destination ID values.
- **lu_table_id** (*String*) – The field in *lu_table* that identifies each zone. Values in this field should correspond to values in each skim table's destination ID values.
- **lu_table_activity_fields** (*[String, ..]*) – A list of field names representing the activities at each (destination) zone to summarize in the accessibility tabulation.
- **out_table** (*ArcGIS Table*) – The output table containing accessiblity summarization results.
- **decays** (*[Decay, ..]*) – A list of Decay objects to be applied in the summarization of activities (in *lu_table_activity_fields*).

Returns Nothing is retured by the function. It generates an *output_table* containing accessibility summaries. The *output_table* will contain a row for each distinct origin ID listed in the skim tables included in *skim_set_obj*. For each origin, fields summarizing all activities listed in *lu_table_activity_fields* (having the same field names as those in that list) as well as sets of the same fields with decayed summaries with names pre- fixed by each decay rate's *name* attribute.

Return type None

See also:

Skim(), *SkimSet()*, *Decay()*

MMA analysis can be facilitated using a suite of geoprocessing tools for ArcGIS. The tools simplify the creation and processing of skims using Network Analyst to produce, summarize, and compare accessibility scores. Each tool can be accessed through the familiar ArcGIS geoprocessing interface. A brief description of each tool is provided below. Click on the tool heading for detailed information.

Calculate Change in Accessibility Evaluate the differences between the accessibility scores reported for an overlapping set of origin zones. Accessibility scores reported in the "no build" table are subtracted from those reported in the "build" table to return the change in accessibility. This tool is useful for mapping the differences between scenarios.

Calculate Weighted Average Calculate areawide average accessibility scores or change in accessibility scores based on the distribution of population groups in a collection of origin zones. Among other uses, the weighted average calculation is useful for equity analyses by comparing the accessibility experienced by different segments of the population.

Create Average Matrix If multiple skims are developed representing a consistent set of potential O-D pairs (using a travel time window or alternative network parameters, e.g.), it may be desirable to summarize the average impedance between each O-D pair describing a single "typical" impedance for MMA processing.

Create Skim Using network analysis to create skim tables is a critical component in MMA processing. This tool provides a single interface for setting up the details of a network analysis and creating output skim files.

Manage Decay Rates Decay rates define how to discount activities at different destinations based on the cumulative impedance (usually travel time) from the origin. Create decay rate configuration files and manage their parameters for MMA processing using this tool.

Manage Skim References When accessibility is summarized in the *Summarize Accessibility* tool, skims are processed based on skim reference configuration files. These files are automatically generated by the *Create Skim* tool, so this tool is only required if working with exogenous skims or if analysis paramters change such that the skim should be processed differently (using an alternative impedance field, e.g.).

Summarize Accessibility Generate accessibility scores for a set of skim files and a table of land use data.

2.9 Typical Geoprocessing Workflow

The figure below shows the typical geoprocessing workflow for generating accessibility scores and making comparisons among scenarios. The phasing aligns with that displayed in the MMA Process Fundamentals section. The hollow boxes represent the MMA geoprocessing tools, showing the specific aspects of the scoring process they support and depend on.

Analysis is undertaken for each mode. Scenarios are defined by selecting data reflecting the combination of network and land use parameters to use for processing. A "base" scenario is likely to be created using existing land use and network data, for example. If analyzing the accessibility impacts of a potential transit service improvement project, the existing land use data may remain, but an alternative transit ntework will be needed reflecting the improved services.

For each scenario, land use centroid locations are loaded on to the network to produce one or more skims. In some cases, the resulting skims may require additional processing to reduce multiple skims into a single "typical" condition, or to properly configure the skim reference files. Use decay rates to properly discount destination-end activities based on travel impedance for the current mode and scenario.

Summarize accessibility scores to create data and map products for visualization and additional analysis as needed.

Finally, compare accessibility results - by zone or in aggregated study areas - to understand how changing conditions impact accessibility.

See also:

mma python module

Note: Dependencies

- arcpy (installs with ArcGIS) v. 10.2 or later
- numpy v. 1.11.2 or later
- pandas v. 0.18.1 or later

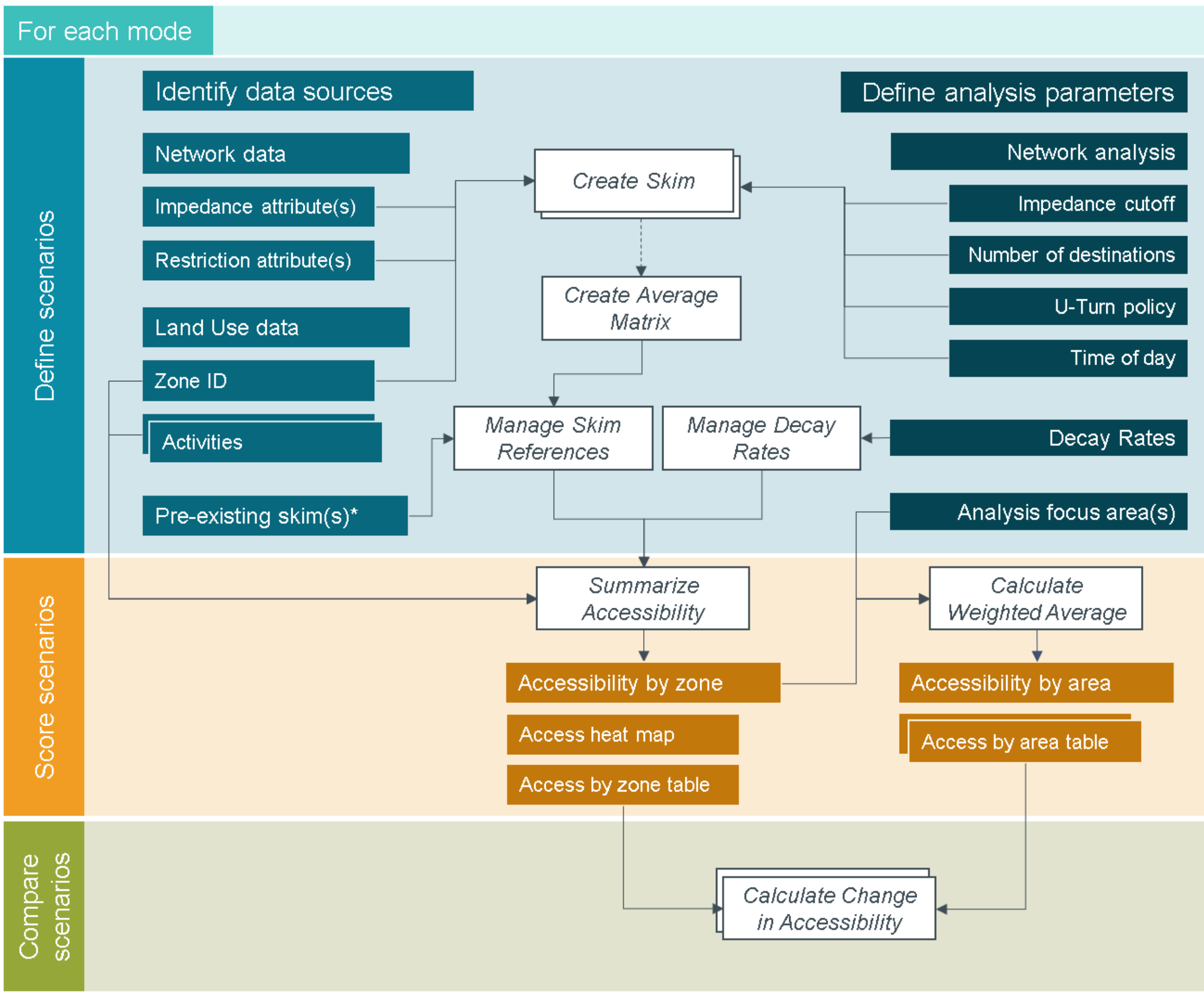


Fig. 1: *For each mode, define the land use and network data and parameters associated with a given scenario. produce and manage skims and decay rates before summarizing accessibility by zone. Aggregate results for study areas of special interest as needed. Finally, compare scenario results and report your findings.*

CHAPTER 3

Chapter 30 Transit Accessibility Scoring

Pursuant to Chapter 30 Acts of 2017 (Senate Bill 307) the Maryland Department of Transportation (MDOT) "shall, in accordance with federal transportation requirements, develop a project–based scoring system for major transportation projects using the goals and measures established under [Transportation Article 2-103.7(c)]. This process wil be used for all major projects being considered for inclusion in the Consolidated Transportation Program (CTP).) An amendment to this law in 2017 defines a "major transportation project" as a highway or transit capacity project that exceeds $5,000,000, and excludes any "projects that are solely for system preservation."

The Chapter 30 scoring model evaluates projects across nine goals and twenty-three measures that were established in statute, using a combination of project data, modeling analysis, and qualitative questionnaires. A project application process has been established requiring counties and municipalities to submit detailed project information when requesting funding for major transportation projects to ensure the necessary project information and priorities are provided to conduct the scoring.

Among the goals and measures articulated for Chapter 30 projects are "reducing congestion and improving commute times" (goal #3) and "equitable access to transportation" (goal #7). For each of these goals, the State has defined access to jobs as a key measure for project scoring, as shown in the following excerpts from the Chapter 30 Technical Guide.

Measures G3 M1 and G7 M1 both rely on assessing the change in access to jobs attributable to the project. This document focuses on the development of scores for these measures for transit capacity project applications.

Multimodal Accessibility Analysis (MMA) in Maryland

3.3. Goal 3: Reducing Congestion and Improving Commute Times

The Chapter 30 goal of Reducing Congestion and Improving Commute Times includes three measures that evaluate the mobility improvements of the project. The measures and their weights are given below in Table 3.3.

Table 3.3 Reducing Congestion & Improving Commute Times Measures and Weights

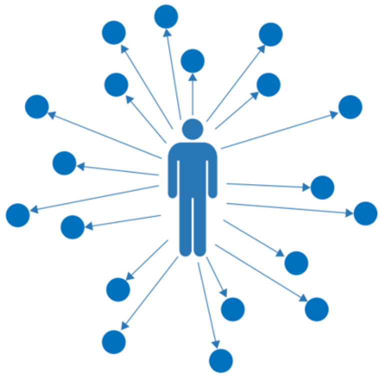
Measure ID	Description	Weight
G3 M1	The expected change in cumulative job accessibility within an approximately 60–minute commute for highway projects or transit projects.	11%
G3 M2	The degree to which the project has a positive impact on travel time and congestion.	64%
G3 M3	The degree to which the project supports connections between different modes of transportation and promotes multiple transportation choices.	25%

3.7. Goal 7: Equitable Access to Transportation

The Chapter 30 goal of Equitable Access to Transportation includes two measures that evaluate how the project will impact job accessibility and economic development for disadvantaged or low-income populations. The measures and their weights are given below in Table 3.7.

Table 3.7 Equitable Access to Transportation Measures and Weights

Measure ID	Description	Weight
G7 M1	The expected increase in job accessibility for disadvantaged populations within an approximately 60-minute commute for projects.	53%
G7 M2	The projected economic development impact on low-income communities.	47%



The Maryland Department of Transportation (MDOT) has for the past several years investigated emerging methods for estimating multimodal accessibility and applications of cumulative accessibility analyses for transportation planning. These efforts have yielded a planning framework referred to as the Multimodal Accessibility (MMA) framework. MMA relies on transportation network analysis and land use data at a variety of scales to measure access to activities of interest (e.g., jobs, essential services, education/training) by multiple travel modes.

In concept, MMA analysis is simple. The goal is to measure travel times from origin zones to destination zones and summarize the activities accessible from each origin zone. The resulting measure describes how well connected each zone is to other zones, accounting for the distribution of activities across all zones and the travel times expected for different system users to reach various destinations. In short, the measure is sensitive to changes in land uses and transportation system performance. It provides insight into travel behaviors such as mode choices and can reveal differences in experienced or expected accessibility for different population groups, such as low-income households and minorities.

Note: Assumed Knowledge

This portion of the guide, which reviews the step-by-step process for scoring Chapter 30 transit project appplications, assumes basic knowledge of the ArcMap interface and the following concepts, tools, and data sources:

- Multimodal Accessibility: Key Terms and Concepts
- MMA Geoprocessing Toolbox
- Add GTFS to a Network Dataset Toolbox
- GTFS Feed Specification

3.1 Recommended Working Directory Structure for Chapter 30 Transit Scoring

To organize the process of scoring multiple projects and streamline the workflow, it is helpful to follow a specific directory structure while shepherding Chapter 30 projects through the accessibility scoring process. The recommended directory structure consists of the following folders:

Decay_rates Contains one or more decay rate configuration files (.JSON) to pass to the Summarize Accessibility tool in the MMA MMA geoprocessing toolbox. The decay rates define how the value of a destination diminishes as travel impedance to it increases. Decay rate configuration files are provided as an a priori input to the Chapter 30 accessibility scoring process. They should not be edited during project scoring.

GTFS Contains all GTFS feeds to be used in the development of the statewide base transit network as well as updated/additional feeds representing specific projects. All feeds should be stored in this folder and appropriate feeds selected for the development of the base and project networks during the network development phases of analysis. Each base or project feed should be included in a separate subfolder to avoid confusion among feeds.

Land_use Contains feature classes representing zone and centroid features across the study area. For Chapter 30 scoring, the study area is the entire state of Maryland and portions of neighboring states. Level 2 zones from the Maryland Statewide Transportation Model (MSTM) and accompanying socio-economic/demographic data are utilized as the standard set of zones for Chapter 30 scoring. They are provided as a priori inputs to the Chapter 30 accessibility scoring process. Demographic and employment data generally should not be edited, unless a project application is accompanied by a project-specific land use forecast.

MMA_scores Contains tables that represent summarized MMA scores generated using the Summarize Accessibility geoprocessing tool.

Networks Contains base and project Network Datasets as developed following the Add GTFS to a Network Dataset Toolbox. See the "Network Setup" section below.

Project_specs Contains shape files for each project network that include proposed route alignment and stops.

Skims Contains skim tables and skim reference files (.JSON) generated by the Create Skims tool (and the Manage Skim References tool when working with exogenous skims - this is generally not necessary for Chapter 30 scoring purposes).

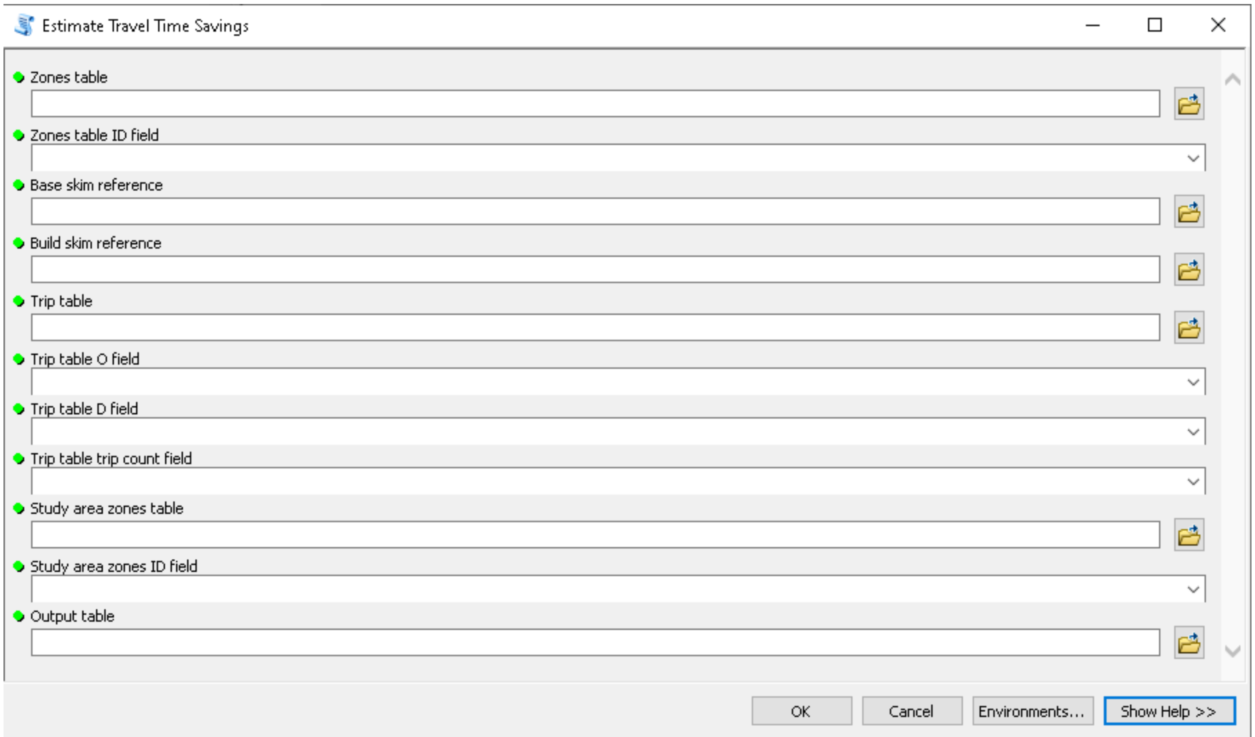
Tools Contains georpocessing toolboxes for use in ArcGIS. - MMA Geoprocessing Toolbox - Chapter 30 Project Scoring Toolbox

3.2 Chapter 30 Project Scoring Toolbox

3.2.1 Estimate Travel Time Savings

See also:

- Chapter 30 accessibility score development



This is the geoprocessing interface for the travelTimeSavings function in the Chapter 30 tools (Ch30Tools) module.

Zones table [ArcGIS Table or Table View] A table listing all distinct zones in the skims and trip tables (usually all level 2 zones in the MSTM).

Zones table ID field [Field] The field in *zones table* that uniquely identifies each zone.

Base skim reference [File (.json)] The skim reference file reflecting travel times in the base condition.

Build skim reference [File (.json)] The skim reference file reflecting travel times in the build condition.

Trip table [ArcGIS Table or Table View] The O-D table that records trips between OD pairs.

Trip table O field [Field] The field in the *trip table* that identifies the origin zone in each row.

Trip table D field [Field] The field in the *trip table* that identifies the destination zone in each row.

Trip table trip count field [Field] The field in the *trip table* that identifies the number of trips between each O-D pair.

Study area zones table [ArcGIS Table or TableView] The table listing all zones in a project study area.

Study area zones ID field [Field] The field in *study area zones table* that identifies each unique zone in the project study area.

Output table: ArcGIS Table The output table to be produced, listing average travel time savings from each origin zone in the *study are zones table* in the "AvgTTChg" field. It will also list the total number of trips from each zone ("SumTrips"), and the total travel time savings from each zone ("SumTTChg"). For study-area-wide average travel time savings, the column sum of "SumTTChg" may be divided by the column sum of "SumTrips".

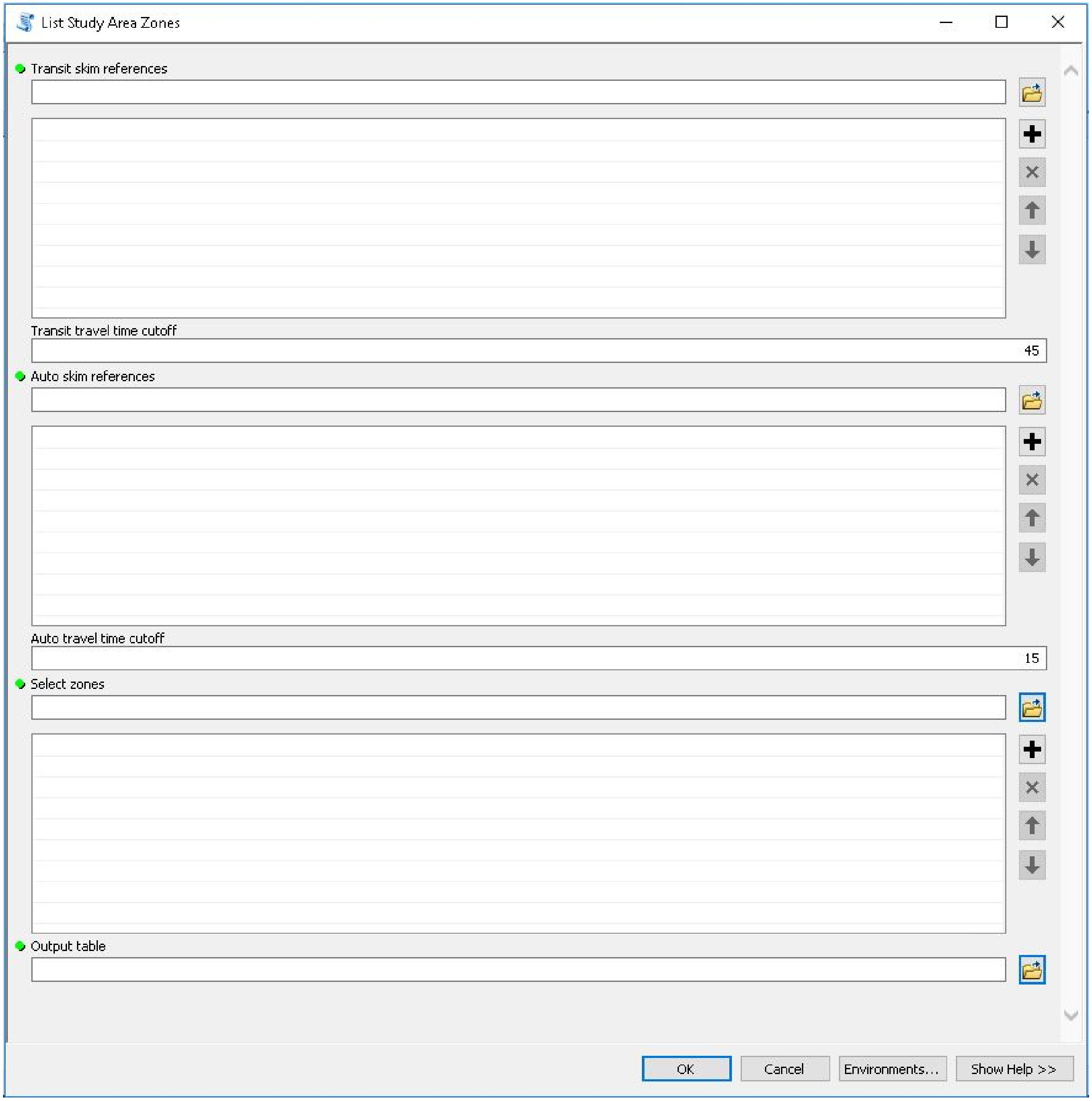
See also:

`Ch30Tools.travelTimeSavings`, `mma.skimReference`

3.2.2 List Study Area Zones

See also:

- Chapter 30 accessibility score development



This is the geoprocessing interface for the listStudyAreaZones function in the Chapter 30 tools (Ch30Tools) module.

Transit skim references [File (.json)] A (list of) skim reference configuration file(s) to search for study area zones reachable within *transit time cutoff* from zones included in *select zones*.

Transit travel time cutoff [Double] The transit travel time tolerance used to determine a zone's inclusion in the project study area.

Auto skim references [File (.json)] A (list of) skim reference configuration file(s) to search for study area zones reachable within *auto time cutoff* from zones included in *select zones*.

Auto travel time cutoff [Double] The auto (highway) travel time tolerance used to determine a zone's inclusion in the project study area.

Select zones [[Variant,. . .]] A list of values corresponding to Zone IDs. The list includes all zones considered to be "within the project limits."

Output table [ArcGIS Table] The output table to be produced by the tool, listing all zones in the project area. The table includes zones in *select zones* as well as those reachable by transit within *transit time cutoff* and by auto within *auto time cutoff*.

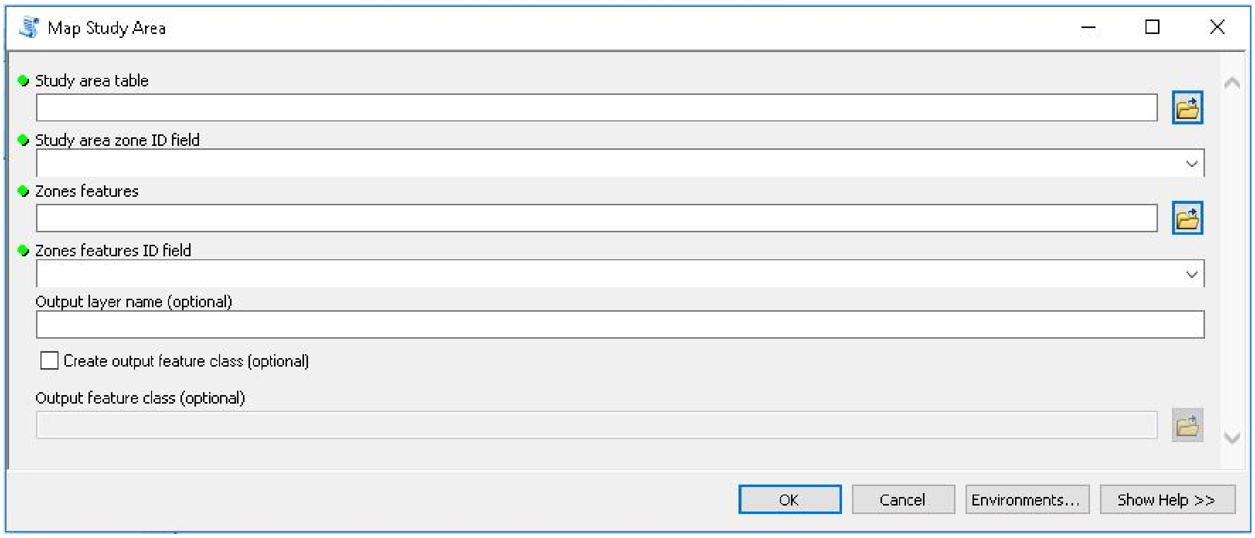
See also:

`Ch30Tools.listStudyAreaZones`, `gp_mapStudyArea`

3.2.3 Map Study Area

See also:

- Chapter 30 accessibility score development



This is the geoprocessing interface for the mapStudyArea function in the Chapter 30 tools (Ch30Tools) module.

Study area table [ArcGIS Table or Table View] The table listing all zones included in the project study area.

Study area zone ID field [Field] The field in *study area table* that identifies each zone in the project study area.

Zones features [ArcGIS Feature Class or Feature Layer] A polygon feature class having zones with ID values corresponding to those in *study area zone ID field*.

Zones features ID field [String] The field in *zones features* that contains zone ID values. It should be of the same data type as *study area zone ID field*.

Output layer name [String, optional] A recongizable name for the feature layer to be produced. Default is None, indicating an auto-generated unique name will be applied to the output feature layer.

Create output feature class: Boolean, optional If checked, features in the *output layer* are dissolved and saved in *Output feature class*. If unchecked, only a feature layer selecting *Zones features* in the *Study area table* is returned.

Output feature class [ArcGIS Feature Class, optional] The output feature class to be produced, outlining the project study area (dissolved zonal polygons).

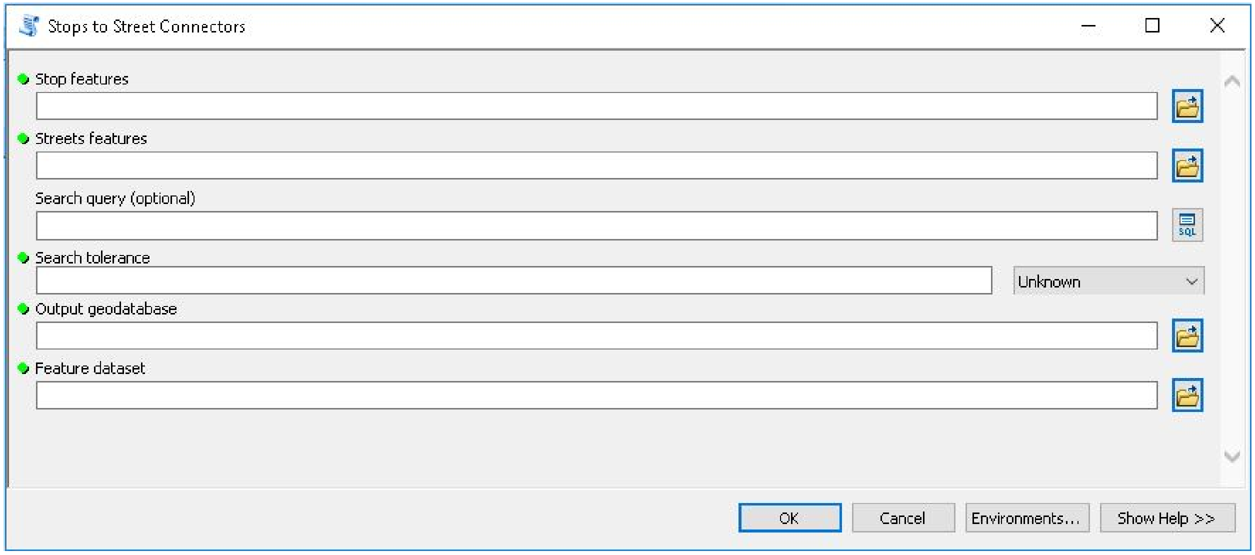
See also:

`Ch30Tools.mapStudyArea`, `gp_listStudyAreaZones`

3.2.4 Stops to Streets Connectors

See also:

- Chapter 30 transit network setup



This is the geoprocessing interface for the createStopsToStreetsConnectors function in the Chapter 30 tools (Ch30Tools) module.

Stops features [ArcGIS Feature Class or Feature Layer] Point features representing GTFS stop locations.

Streets features [ArcGIS Feature Class or Feature Layer] Line features representing the pedestrian network from which *stops features* will be accessed.

Search query [SQL Expression, optional] Criteria to apply to *streets features* to limit which streets stops will snap to.

Search tolerance [Linear Unit] The distance from *stops features* to search for potential *streets features* to snap stops to streets.

Output geodatabase [Geodatabase] The file or personal geodatabase in which output tables and will be stored.

Feature dataset [ArcGIS Feature Dataset] The feature dataset within *output_ws* in which output feature classes will be stored.

Nothing is returned by the function. Several tables and feature classes are generated in the *output_ws* and *feature_dataset* workspaces:

- Connectors_Stops2StreetsTable (Table, interim)
- Connectors_Stops2Streets (Feature class, final)
- Stops_Snapped2Streets (Feature class, final)

3.2.5 Chapter 30 Tools python module

Note: Dependencies

- arcpy (installs with ArcGIS) v. 10.2 or later
- numpy v. 1.11.2 or later

- mma

This module facilitates the designation of a project study area and the execution of travel time savings analysis procedures for Maryland Chapter 30 scoring purposes. It also provides support for transit network development for users following the "Add GTFS to Network Dataset" toolbox (ESRI) workflow in the ArcGIS Basic license level.

Project Study Areas

Project study areas are used to define the area in which impacts of projects are measured. They are dynamically determined based on the travel mode affected by the project, the expected usership of the facilities affected by the project, and travel time changes wrought by the project.

For transit projects, the study area is defined based on transit and highway travel time skims. Zones overlapping project limits are provided by the user as a list of zone IDs, and zones within user- specified travel time tolerances by each mode are included in the project's study area.

See also:

gp_listStudyAreaZones, *gp_mapStudyArea*

Travel Time Savings Calculation

Travel time savings summarizes the impact of a given transportation project by comparing travel times between a "build" condition (with the project) to a "base" condition (without the project). The difference in travel time between each O-D pair is weighted by the number of trips estimated between the zones in deriving average travel time savings. The procedures offered in this module assume a constant trip table and are used for estimating the travel time savings that accrue to current transit riders. The derivation and utilization of these estimates alongside travel time savings benefits accruing to motorists due to increased transit ridership are spelled out in the Chapter 30 Technical Guide.

The matrix processing procedures required for the travel time savings estimates are facilitated by the ZoneMatrixManager class and executed using the travelTimeSavings function.

See also:

gp_travelTimeSavings

Transit Network Development Support

For transit network development, the "Add GTFS to Network Dataset" toolbox workflow requires ArcGIS Standard or Advanced license levels to run step 2 (stops-to-streets connectors). An alternative function for generating connection features is provided for ArcGIS Basic users. - createStopsToStreetsConnectors

See also:

gp_StopsToStreetsConnectors

Classes and Functions

class Ch30Tools.**ZoneMatrixManager**(*zones_table*, *id_field*, *o_zones=None*, *d_zones=None*)

The ZoneMatrixManager class generates a matrix (2-dimensional numpy array) based on a provided table of zones and optional lists of zones representing a subset to consider as "origins" and/or "destinations."

Matrices generated by the ZoneMatrixManager has a number of rows equal to the length of the origin zones list and a number of columns equal to the length of the destination zones list. Accessing specific cells in the matrix through zone indexes or labels is facilitated through various methods.

Parameters

- **zones_table** (*ArcGIS TableView or string*) – A table listing all distinct zones potentially serving as O or D zones.
- **id_field** (*string*) – The field in *zones_table* that identifies each zone.
- **o_zones** (*array_like (1dim)*) – A list or array of origin zone names to focus on.
- **d_zones** (*array_like (1dim)*) – A list or array of destination zone names to focus on.

zones_table

A table listing all distinct zones potentially serving as O or D zones.

Type ArcGIS TableView or string

id_field

The field in *zones_table* that identifies each zone.

Type string

o_zones

A list or array of origin zones to focus on. Matrices generated by the ZoneMatrixManager object will have a number of rows equal to the length of o_zones. The array is retained as an attribute to facilitate row indexing.

Type array_like (1dim)

d_zones

A list or array of destination zones to focus on. Matrices generated by the ZoneMatrixManager object will have a number of columns equal to the length of d_zones. The array is retained as an attribute to facilitate column indexing.

Type array_like (1dim)

idx_array

A numpy array listing all zone id's from the zones table.

Type np.array(1dim)

o_mask

A boolean array that masks *idx_array* to facilitate row indexing.

Type np.array (1dim, boolean)

d_mask

A boolean array that masks *idx_array* to facilitate column indexing.

Type np.array (1dim, boolean)

ODTableToMatrix(*od_table*, *od_o_field*, *value_fields*, *od_d_field=None*, *delimiter=None*, *zeros=True*, *invert=False*)

Generates and O-D matrix and iterates over records in an O-D table to assign values to the new matrix based on the indexed row and column locations of O and D zone names.

Parameters

- **od_table** (*ArcGIS TableView or string (path)*) – The table containing O-D information.
- **od_o_field** (*String*) – The field identifying the origin zone in od_table. If the table has only a single O-D name field, use this field name as the od_o_field and provide a value for *delimiter*.

- **od_d_field** (*String*) – The field identifying the destination zone in od_table. If the table has only a single O-D name field, the od_d_field is not needed.
- **value_fields** (*[String, ..]*) – The O-D values to assign to the output array. The number of value_fields provided determines the number of panes in the output array.
- **delimiter** (*string*) – If od_table has only a single O-D name field, the delimiter defines the string on which to split values in that field to deterimine the O and D zone names for each row.
- **zeros** (*boolean (default=True)*) – If True, the output array will be initialized to zeros. Any O-D pairs in the array not found in od_table will show up as zeros. These cannot be differentiated from values of zero found in the od_table. If False, the output array will be initialized to ones. Any O-D pairs in the array not found in od_table will show up as ones. These cannot be differentiated from values of one found in the od table (but see invert below).
- **invert** (*boolean (default=False)*) – If True, the output array will be multiplied by -1 when initialized. This only applies when zeros=False. Any O-D pairs in the array not found in the od_table will show up as -1. These are easily distinguished from valid O-D values (which are typically non-negative numbers).

Returns np.array of shape [len(val_fields), len(self.o_zones), len(self.d_zones)] where the first pane (index=0) provides O-D values for the first value field, the second pane (index=1) provides O-D values for the second value fields, and so on.

Return type numpy array

createOnes(*num_panes=1*)

Create a 3-d matrix initialized to ones. The matrix shape is [num_panes, len(*self.o_zones*), len(*self.d_zones*)]. If no o_zones or d_zones have been specified, those dimensions are sized according to the total number zones in *self.index_array*.

The ZoneMatrixManager object generates matrices that have "panes". Each pane has a consistent number of rows and columns and represents distinct O-D data (a "time" pane, and "distance" pane, e.g.)

Parameters **num_panes** (*integer*) – The number of O-D panes to create.

Returns np.array of shape [num_panes, len(self.o_zones), len(self.d_zones)] with all values initialized to ones.

Return type numpy array

createZeros(*num_panes=1*)

Create a 3-d matrix initialized to zeros. The matrix shape is [num_panes, len(*self.o_zones*), len(*self.d_zones*)]. If no o_zones or d_zones have been specified, those dimensions are sized according to the total number zones in *self.index_array*.

The ZoneMatrixManager object generates matrices that have "panes". Each pane has a consistent number of rows and columns and represents distinct O-D data (a "time" pane, and "distance" pane, e.g.)

Parameters **num_panes** (*integer*) – The number of O-D panes to create.

Returns np.array of shape [num_panes, len(self.o_zones), len(self.d_zones)] with all values initialized to zero.

Return type numpy array

dZoneIndex(*zone_name*, *relative_to_ds=True*)

Given a zone_name or array of zone names, returns the column index(es) corresponding to the requested zones.

Parameters

- **zone_name** (*variable*) – The name of the zone (or list of names of zones) for which to find column indexes in a matrix generated by this ZoneMatrixManager object.
- **relative_to_ds** (*boolean (default=True)*) – If True, the column indexes returned will correspond to a matrix where the number of columns is equal to len(*self.d_zones*). If False, the column indexes returned will correspond to a matrix where the number of columns is equal to len(*self.index_array*)

Returns The column index locations for the named zone(s).

Return type [integer]

oZoneIndex(*zone_name*, *relative_to_os=True*)

Given a zone_name or array of zone names, returns the row index(es) corresponding to the requested zones.

Parameters

- **zone_name** (*variable*) – The name of the zone (or list of names of zones) for which to find row indexes in a matrix generated by this ZoneMatrixManager object.
- **relative_to_os** (*boolean (default=True)*) – If True, the row indexes returned will correspond to a matrix where the number of rows is equal to len(*self.o_zones*). If False, the row indexes returned will correspond to a matrix where the number of rows is equal to len(*self.index_array*)

Returns The row index locations for the named zone(s).

Return type [integer]

setDZonesArray(*array_1d*)

Define a new list of desitination zones. Once updated, matrices previously generated by this ZoneMatrix-Manager object cannot be reliably indexed.

Parameters array_1d (*array_like (1d)*) – A list or array of destination zone names to focus on.

setOZonesArray(*array_1d*)

Define a new list of origin zones. Once updated, matrices previously generated by this ZoneMatrixManager object cannot be reliably indexed.

Parameters array_1d (*array_like (1d)*) – A list or array of origin zone names to focus on.

setZonesTable(*zones_table*, *id_field=None*)

Sets the *zones_table* attribute and updates the *idx_array* attribute.

Parameters

- **zones_table** (*ArcGIS TableView or string (path)*) – The new table listing all distinc zones potentially serving as O or D zones. Once updated, matrices previously generated by this ZoneMatrixManager object cannot be reliably indexed.
- **id_field** (*string*) – The field in *zones_table* that identifies each zone. If *None*, it is assumed that the current *id_field* value is a valid field in the new *zones_table*. The *index_array* will be recreated by referencing the *id_field* in the *zones_table*.

Ch30Tools.**createStopToStreetConnectors**(*stops_layer*, *streets_layer*, *search_tolerance*, *output_ws*, *feature_dataset*)

Create a line feature connecting each stop feature to a street feature.

This function snaps transit stops to the street feature class, generates connector lines between the original stop location and the snapped stop location, and adds vertices to the street features at the locations of the snapped stops. These steps ensure good connectivity in the network dataset based on GTFS feeds. It mimics step 2 in ESRI's Add GTFS to Network Dataset toolbox for users having the ArcGIS Basic license.

Parameters

- **stops_layer** (*ArcGIS Feature Class or Feature Layer*) – Point features representing GTFS stop locations.
- **streets_layer** (*ArcGIS Feature Class or Feature Layer*) – Line features representing the pedestrian network from which *stops_layer* features will be accessed.
- **search_tolerance** (*String (Linear Unit)*) – The distance from *stops_layer* features to search for potential *streets_layer* features to snap stops to streets.
- **output_ws** (*Geodatabase*) – The file or personal geodatabase in which output tables and will be stored.
- **feature_dataset** (*ArcGIS Feature Dataset*) – The feature dataset within *output_ws* in which output feature classes will be stored.

Returns Nothing is returned by the function. Several tables and feature classes are generated in the *output_ws* and *feature_dataset* workspaces: - Connectors_Stops2StreetsTable (Table, interim) - Connectors_Stops2Streets (Feature class, final) - Stops_Snapped2Streets (Feature class, final)

Return type None

See also:

gp_StopsToStreetsConnectors()

Ch30Tools.**listStudyAreaZones**(*transit_skim_references*, *transit_time_cutoff*, *auto_skim_references*, *auto_time_cutoff*, *select_zones*, *output_table*)

Produces a table listing zones in the transit projet study area based on travel times to the project by transit and driving.

Parameters

- **transit_skim_references** (*File (json)*) – A (list of) skim reference configuration file(s) to search for study area zones reachable within *transit_time_cutoff* from zones included in *select_zones*.
- **transit_time_cutoff** (*Double*) – The transit travel time tolerance used to determine a zone's inclusion in the project study area.
- **auto_skim_references** (*File (json)*) – A (list of) skim reference configuration file(s) to search for study area zones reachable within *auto_time_cutoff* from zones included in *select_zones*.
- **auto_time_cutoff** (*Double*) – The auto (highway) travel time tolerance used to determine a zone's inclusion in the project study area.
- **select_zones** (*[Variant,..]*) – A list of values corresponding to Zone IDs. The list includes all zones considered to be "within the project limits.
- **output_table** (*ArcGIS Table*) – The output table to be produced by the tool, listing all zones in the project area. The table includes zones in *select_zones* as well as those reachable by transit within *transit_time_cutoff* and by auto within *auto_time_cutoff*.

Returns Nothing is returned by the function. A new output table is generated at the path specified by *output_table*.

Return type None

See also:

gp_listStudyAreaZones(), *mapStudyArea()*, *mma.Skim()*, *mma.jsonToSkim()*

Ch30Tools.**mapStudyArea**(*study_area_table*, *sa_zone_field*, *zones_fc*, *zone_id_field*, *layer_name=None*, *output_fc=None*)

Using a list of study area zones and a related zones feature class, produces a feature layer showing the project study area. Optionally, produces a feature class outlining the study area.

Parameters

- **study_area_table** (*ArcGIS Table or Table View*) – The table listing all zones included in the project study area.
- **sa_zone_field** (*String*) – The field in *study_area_table* that identifies each zone in the project study area.
- **zones_fc** (*ArcGIS Feature Class or Feature Layer*) – A polygon feature class having zones with ID values corresponding to those in *sa_zone_field*.
- **zone_id_field** (*String*) – The field in *zones_fc* that contains zone ID values. It should be of the same data type as *sa_zone_field*.
- **layer_name** (*String, optional*) – A recongizable name for the feature layer to be produced. Default is None, indicating an auto-generated unique name will be applied to the output feature layer.
- **output_fc** (*String, optional*) – The full path to the output feature class to be produced, outlining the project study area (dissolved zonal polygons). Default is None, indicating no output feature class will be produced.

Returns **out_layer** – A feature layer showing the study area limits based on matching *zones_fc* features included in *study_area_table*. Also, optionally outptus a feature class outlining the study area to the path specified by *output_fc*.

Return type ArcGIS feature layer

See also:

gp_mapStudyArea(), *listStudyAreaZones()*

Ch30Tools.**travelTimeSavings**(*zone_table*, *zone_id_field*, *base_skim_ref*, *build_skim_ref*, *trip_table*, *trip_o_field*, *trip_d_field*, *trip_val_field*, *o_zone_table*, *o_zone_field*, *out_table*)

Estimates average travel time savings in a build scenario relative to a base sceanrio using a consistent trip table.

When the build condition improves travel times between commmonly traveled O-D pairs (discerned from the trip table), the travel time savings will be high.

Parameters

- **zone_table** (*ArcGIS TableView or string*) – A table listing all distinct zones in the skims and trip tables.
- **zone_id_field** (*string*) – The field in *zone_table* that identifies each zone. Values in this field should correspond to values in the skim tables' and trip table's origin and destination zone columns.
- **base_skim_ref** (*string*) – The path to the skim reference .json file to use as the base skim. The skim reference points to a skim table and includes metadata for appropriately parsing origin, destination, and impedance fields.

- **build_skim_ref** (*string*) – The path to the skim reference .json file to use as the build skim. The skim reference points to a skim table and includes metadata for appropriately parsing origin, destination, and impedance fields.
- **trip_table** (*ArcGIS TableView or string*) – An origin-destination table containing the number of trips between OD pairs.
- **trip_o_field** (*string*) – The field in the *trip_table* that identifies the origin zone of the OD pair.
- **trip_d_field** (*string*) – The field in the *trip_table* that identifies the destination zone of the OD pair.
- **trip_val_field** (*string*) – The field in the *trip_table* that identifies the number of trips between the origin zone and the destination zone.
- **o_zone_table** (*ArcGIS TableView or string*) – A table listing the origin zones within the study area to limit the travel time savings analysis to this specific subset of zones.
- **o_zone_field** (*string*) – The field in *o_zone_table* that identifies each study area zone.
- **out_table** (*string*) – The path to the output file to be generated with travel time savings results summarized.

Returns Nothing is retured by the function. It will output a table listing all zones in the *o_zone_table* and the average travel time change for trips from each zone ("AvgTTChg"), the total number of trips from each zone ("SumTrips"), and the total travel time savings from each zone ("SumTTChg" = "AvgTTChg" * "SumTrips"). For study-area-wide average travel time savings, the column sum of "SumTTChg" may be divided by the column sum of "SumTrips".

Return type None

See also:

mma.jsonToSkim(), *mma.Skim()*

Notes

The zone table provides a consistent frame of reference for indexing O-D values in the base skim, build skim, and trip tables. This allows all O-D impedances and trip totals to be stored in a matrix for efficient processing.

The Chapter 30 Project Scoring Tools are comprised of four separate geoprocessing script tools for use in ArcGIS. These tools include:

1. **Estimate Travel Time Savings** - Create a table that lists all study area zones and the average and total change in travel time weighted by a trip table. Total trips from each zone are also reported. The table can be summarized to estimate overall changes in travel time for all trips originating in the study area.

2. **List Study Area Zones** - Create a table that lists all MSTM Level 2 TAZs in the project study area. In Chapter 30 transit project scoring, the study area is defined as all zones within 45 minutes transit travel time from the project or within 15 minutes driving time from the project.

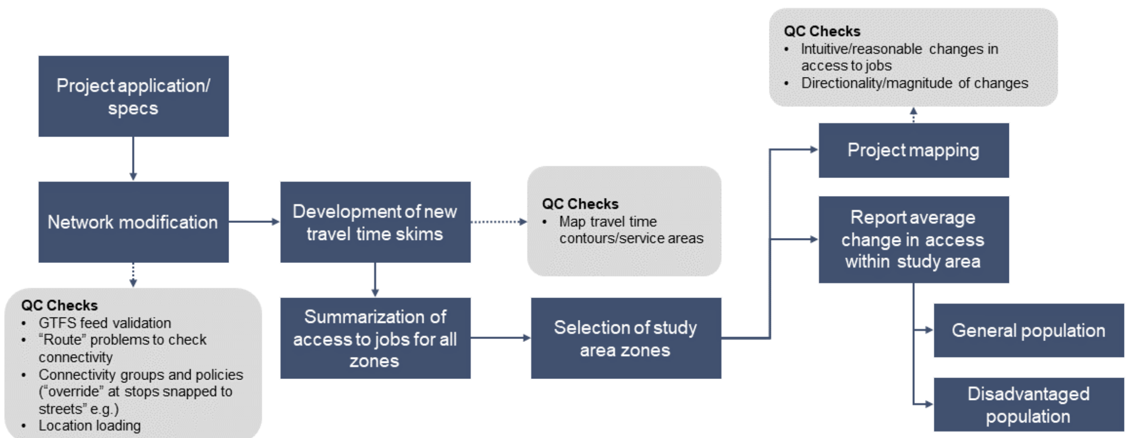
3. **Map Study Area** - Based on a table of study area zones, create a feature layer selecting all zones in the study area or a feature class showing the dissolved boundary of the study area.

4. **Stops to Streets Connectors** - If working in the ArcGIS Basic license level when creating transit networks using the "Add GTFS to a Network Dataset" toolbox, use this alternative tool to generate required network features (this is a substitute for completing step 2 of that toolbox's workflow, which requires the Standard or Advanced license levels).

3.3 Chapter 30 Transit Scoring Workflow Overview

The Chapter 30 project scoring process follows this general workflow:

1. Prepare data
2. Network setup
3. Run accessibility tools
4. Quality assurance
5. Prepare Chapter 30 Transit Project Report



3.3.1 Data Preparation

Maryland Statewide Transportation Model

The Maryland State Highway Administration (SHA) developed and maintains the Maryland Statewide Transportation Model (MSTM). to support a variety of transportation planning and system operation and performance applications. The MSTM is a multiresolution travel demand modeling platform providing consistent data on land uses and travel networks across the state at multiple scales. Level 1 is the coarsest scale and is primarily utilized for statewide analyses; Level 2 is an intermediate scale suitable for regional-level analyses; and Level 3 is a fine-grained scale supporting local area analyses.

The following data from the MSTM are utilized for Chapter 30 transit project accessibility scoring:

- Point (centroid) and polygon feature classes representing MSTM Level 2 zones.
- Socio-economic and demographic data summarized to MSTM Level 2 zones for the scoring horizon year.
- Polyline feature class representing MSTM Level 3 network features. Although the transit analysis is carried out at the Level 2 scale, the Level 3 network is utilized to model the access and egress to/from transit stops and stations at a sufficiently fine level of detail.

- Trip table estimating number of person trips between MSTM level 2 zones.

Existing transit network GTFS feeds

GTFS is a standard format for storing and sharing open transit data, including route and schedule information. GTFS feeds are collections of comma-delimited text (csv) files that provide sufficient information to model transit routing options by time of day for a selected service day (specific date or typical day of the week). The details of the tables included in a typical feed and the data recorded in each table are outlined here.

For Chapter 30 transit scoring, General Transit Feed Specification (GTFS) feeds for all transit properties in Maryland and neighboring jurisdictions (Washington, DC and northern Virginia, e.g.) were obtained from the Transportation Resource Information Point website. These feeds offer the best available representation of currently available fixed-route transit services across the state and serve as the basis of the "base" transit network. All feeds were utilized "as is," assuming the feed developers adequately validated the information contained in each feed.

For 2019 Chapter 30 scoring, the latest available feeds were downloaded on April 30, 2018 (originally downloade for the 2018 inaugural round of Chapter 30 project scoring).

EXISTING GTFS FEEDS

Agency	Name	Start	Date	Date
Allegany County Transit	MD	2016/05/19	2018/01/01	Summer/Fall 2016
Annapolis Transit	MD	2010/01/01	2019/12/31	2016/12/07
BWI Thurgood Marshall Airport	MD	2016/01/01	2017/12/31	2016/12/06
Calvert County Public Transportation	MD	2015/09/23	2017/12/31	2016/12/08
Carroll Transit System	MD	2016/04/13	2018/01/01	2016/12/27
Cecil Transit	MD	201001/01	2019/12/31	2016/06/14
Charles County VanGo	MD	2015/10/06	2017/12/31	2016/12/19
Charm City Circulator	MD	2016/05/06	2018/12/31	5/17/2016
Delmarva Community Transit	MD	2015/09/23	2017/12/31	2016/10/20
Harford Transit LINK	MD	2015/10/01	2017/12/31	2016/11/28
Maryland Transit Administration	MD	2010/01/01	2019/12/31	[no data]
Montgomery County MD Ride On	MD	2010/01/01	2019/12/31	[no data]
Ocean City Transportation	MD	2016/11/24	2017/12/31	2017/07/07
Queen Anne's County Ride	MD	2016/03/08	2017/12/31	2016/12/21
Regional Transportation Agency of Central Maryland	MD	2016/06/01	2018/03/01	2016/11/15
Shore Transit	MD	2016/03/18	2020/01/01	Summer 2016
St. Mary's Transit System	MD	2015/09/23	2017/12/31	2016/12/21
The Bus of Prince George's County	MD	2016/09/25	2017/12/31	2016/12/21
TransIT Services of Frederick County	MD	2010/01/01	2019/12/31	2013/01/28
Washington County Transit	MD	2016/05/12	2018/01/01	2016/12/21
DC Circulator	DC	2010/01/01	2019/12/31	[no data]
WMATA	DC	2010/01/01	2019/12/31	[no data]
Alexandria Transit Company (DASH)	VA	2010/01/01	2019/12/31	[no data]
Arlington Transit	VA	2010/01/01	2019/12/31	[no data]
Fairfax Connector	VA	2010/01/01	2019/12/31	[no data]
Fairfax CUE	VA	2008/07/01	2020/01/01	[no data]
Loudon County Transit	VA	2010/01/01	2019/12/31	[no data]
Virginia Railway Express	VA	2010/01/01	2019/12/31	[no data]
Winchester Transit	VA	2010/01/01	2019/12/31	[no data]

Project Transit Network

Descriptive information on the proposed service changes are needed for all projects. The most recent and authoritative planning documents should be identified for the development of these GTFS feeds. Attributes needed to code new or changed service include: mode (including dedicated right of way), transit stop locations, estimated departure times, service hours, frequency of service, and travel times. If one or more pieces of information is unavailable, reasonable assumptions should be made and documented.

After proposed service documentation is identified, GTFS files need to be developed to represent the proposed service. The matrix below outlines possible proposed infrastructure projects and the associated actions required for a GTFS feed to represent this project. Recommended best practice is to create a copy o f the base GTFS feed to edit separately, or if new routes are proposed a new GTFS feed containing only the new route should be developed.

SCENARIOS FOR UPDATING GTFS

Improvement	GTFS attributes affected	Action
Alignment change	Stops, Stop Times	Update
Service change	Calendar	Update
Headway improvement	Stop Times, Frequencies	Update
New route	All	Build new
Route deviation	All	Build new

The 2019 Chapter 30 scoring included two transit projects. These projects are described below and links to the final project reports submitted to MDOT are included for reference. These project reports include detailed information on how the projects were coded into GTFS to create the project transit network GTFS feed and will provide examples of how to create these project transit feeds.

Project Name	Description	Report Link
South Side Transit	A proposed light rail line between Alexandria, Va., and the Washington, D.C. suburb of Oxon Hill in Prince George's County that would connect two existing Washington Metro stations serving the Green, Blue, and Yellow lines.	Report
US 29 Bus Rapid Transit	A potential BRT line with an alignment between Burtonsville and Mount Hebron in Maryland with four intermediate stops. The project would connect with the planned US 29 Flash BRT corridor in Burtonsville	Report

Standard GTFS Tables

The example tables above focus on the GTFS files that provide the richest details in modeling the specific service characteristics of new and/or improved transit services. A standards-compliant GTFS feed rquires a series of files that meet validation requirements. In developing a new GTFS feed, it can be helpful to begin with standard tables provided by Google Developer Resources website. All required files not shown in this user guide must be included in the project GTFS feed using simplisitc coding (assuming a new route will operate on all days of the week in the calendar.txt file, e.g.).

These tables can form the basis for coding in a new build scenario network. Variables in the standard feed can be modified as needed to represent the proposed service. The standard tables are availabe at this link.

Note: Before proceeding it is necessary to validate GTFS feeds. See "GTFS Feed Validation" in "Chapter 30 Quality Assurance" for information on performing this step.

3.3.2 Network setup

To score projects, Network Datasets must be created to represent both the existing and project transportation networks. These networks are stored in their own geodatabases which are created in the Networks folder. Several steps in this process require the use of the Add GTFS to a Network Dataset toolbox.

See also:

- Basic instructions on creating a network dataset
- Add GTFS to a Network Dataset User Guide.

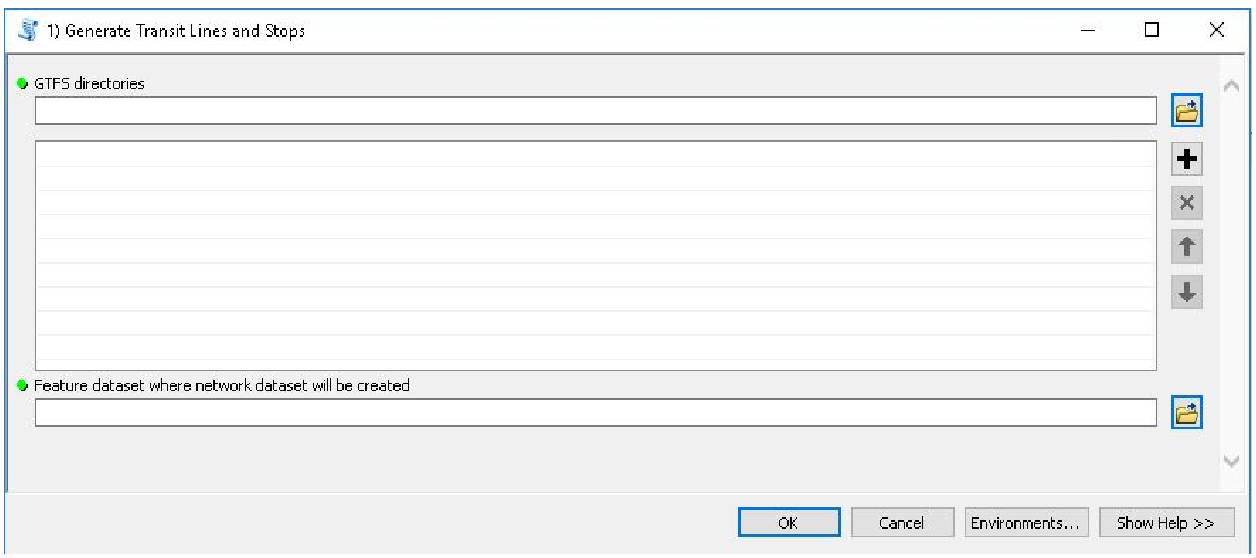
Step 1: Create feature dataset

Create a feature dataset within the appropriate geodatabase for a given project. This feature dataset will house the required feature classes to build a network dataset. Import the MSTM Level 3 highway network into the feature dataset.

Step 2: Generate transit lines and stops

Feature classes are created from the GTFS feeds to represent both the base network and project scenarios and should be added to the feature dataset created during Step 1. This step uses the "1) Generate Transit Lines and Stops" from the "Add GTFS to a Network Dataset" tool.

GTFS data are used to create line and point features classes and build a SQL database of the transit schedule described in the GTFS feed. These will be stored in the working geodatabase containing the network dataset built in Step 1. All GTFS feeds should be loaded onto the network at one time, including feeds created to model new projects. As described below, routes and trips available under proposed projects can be excluded to reflect existing service by creating network parameters.

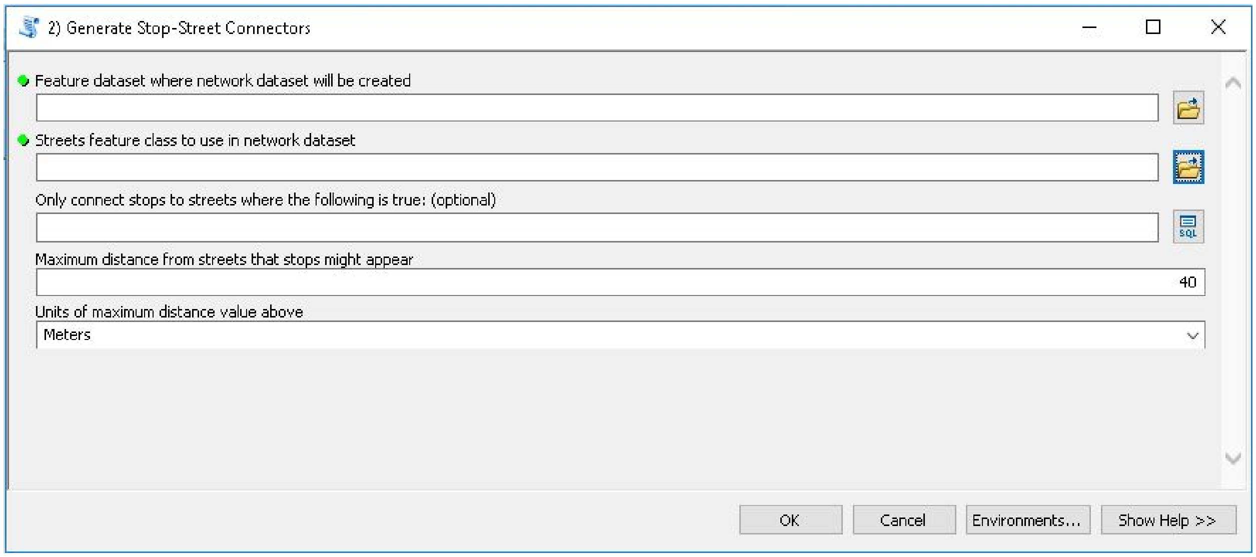


Field	Description
GTFS directories	GTFS feeds covering the state + project feed(s)
Feature dataset where network dataset will be created	Feature dataset (sits within GDB) where the network will be created.

Note: Each scenario should be in a separate geodatabase.

Step 3: Generate Stop-Street Connectors

Run the "2) Generate Stop Street Connectors" tool in the "Add GTFS to a Network Dataset" tool. This step creates connections between transit stop locations and the roadway network to allow interaction between transit lines and the access/egress network. The output files should be stored in the feature dataset created in Step 1. In the case of Chapter 30 scoring, the streets feature class is the MSTM Level 3 network which was already imported.

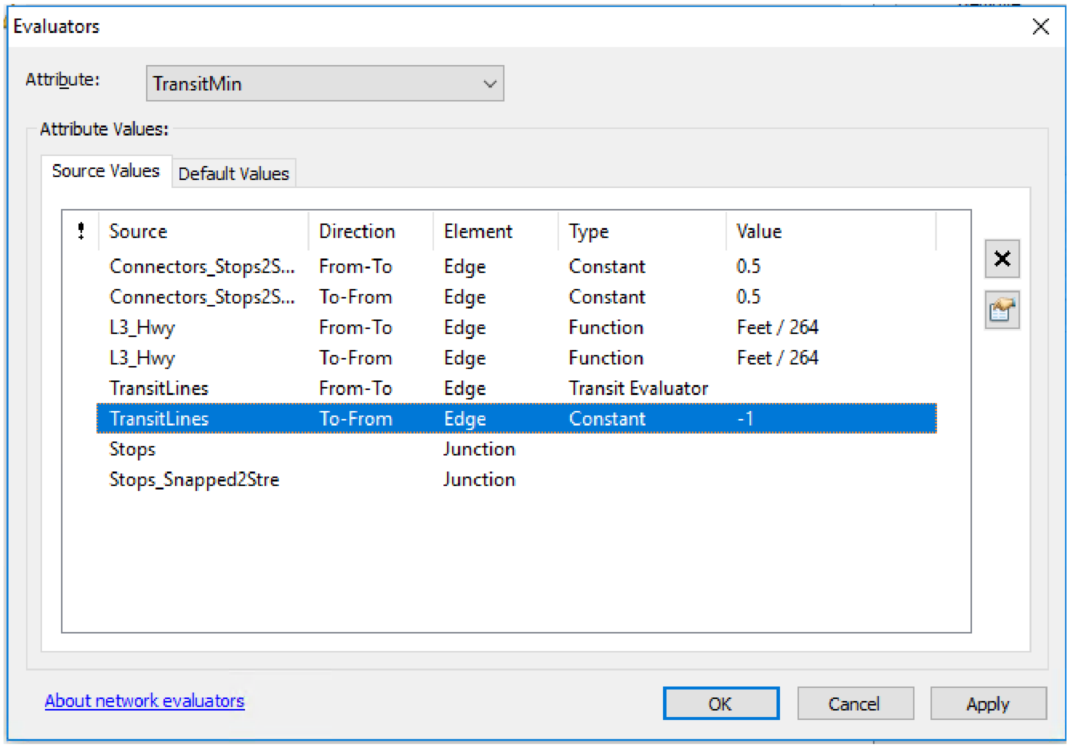


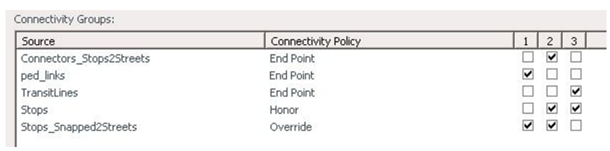
Field	Description	Recommended Value
Feature dataset where network dataset will be created	Feature dataset where the network will be created.	Geodatabase for project
Streets feature class to use in network dataset	Streets features: Level 3 MSTM network imported into feature dataset	Output of 1)Generate Transit Lines and Stops
Only connect stops to streets where the following is true: (optional)	Optional connection criteria.	L3_Hwy
Maximum distance from streets that stop might appear	Maximum distance value.	100
Unit of maximum distance value above	Maximum distance unit.	Feet

Note: In order to run the Generate Stop-Street Connectors tool, you must have the Desktop Standard or Advanced license. If using the Basic license, use the alternative tool provided in the Chapter 30 Project Scoring Toolbox

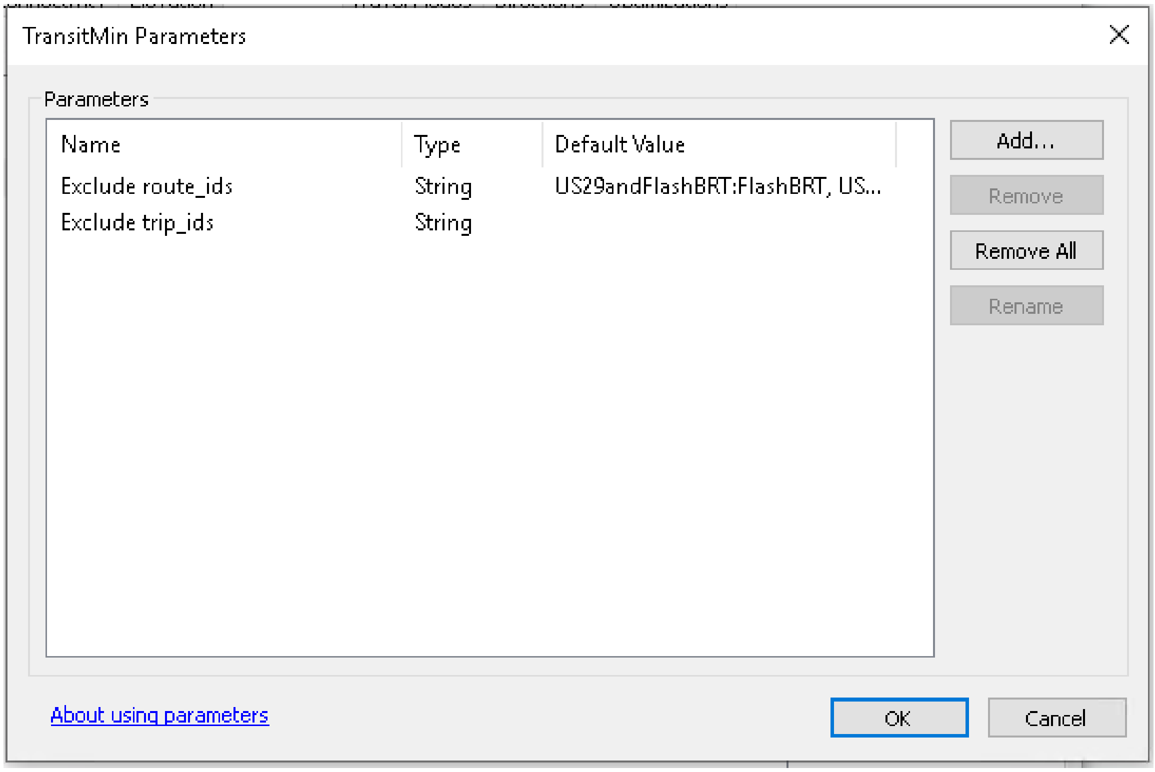
Step 4: Build Network Dataset

Once all feature classes are created, right click the feature dataset in the catalog view to create a new network dataset. During network dataset creation, evaluators for calculating travel times on the network and connectivity rules must be established. The two screens below show the recommended evaluator and connectivity group settings for Chapter 30 transit analysis. Pay close attention to the Connectivity Policy in the connectivity group settings step.



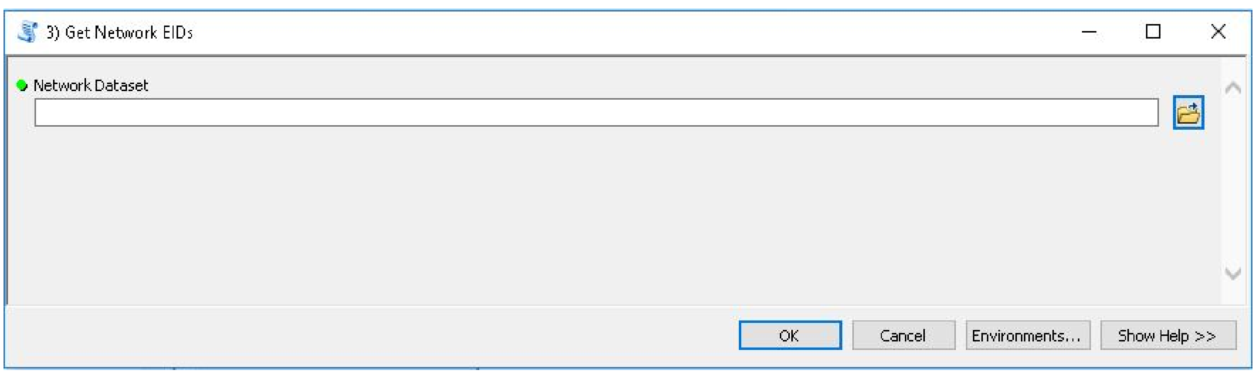


Parameters should be added to your transit evaluators that will allow you to modify later analyses by excluding specific routes or trips. These parameters must be a “String” data type and named exactly “Exclude route_ids” and “Exclude trip_ids” to work properly. The values entered can be a single or list of route or trip IDs prefixed with the name of the GTFS dataset and a colon. The pictures below show an example of these parameters with a default exclusion included. Please see the Add GTFS to a Network Dataset documentation on excluding routes and trips for more information.



Step 5: Get Network EIDs

Run the "3) Get Network EIDs" tool in the "Add GTFS to a Network Dataset" tool. After creating and building the network dataset, this script prepares the network for use in Network Analyst.



Field	Description
Network Dataset	The built transit network dataset

Note: This tool retrieves the network dataset's edge IDs (EIDs) for the transit lines features and adds the EIDs to a SQL table that will be referenced by the GTFS transit evaluator. The network dataset must be built prior to running this tool, and the tool must be re-run every time the network dataset is rebuilt in order to update the EID values.

Warning: Quality checks are required once networks are built. Run the Routing Problems and Service Area Problems quality checks before continuing.

3.3.3 MMA Processing Steps

Following the preparation of socio-economic and network data as described in previous sections of this guide, accessibility scores can be calculated, summarized, and compared to a base condition to identify changes in access to jobs. The following workflow describes the overall approach to this analysis using the MMA and Chapter 30 toolboxes.

1. Define Decay Curves

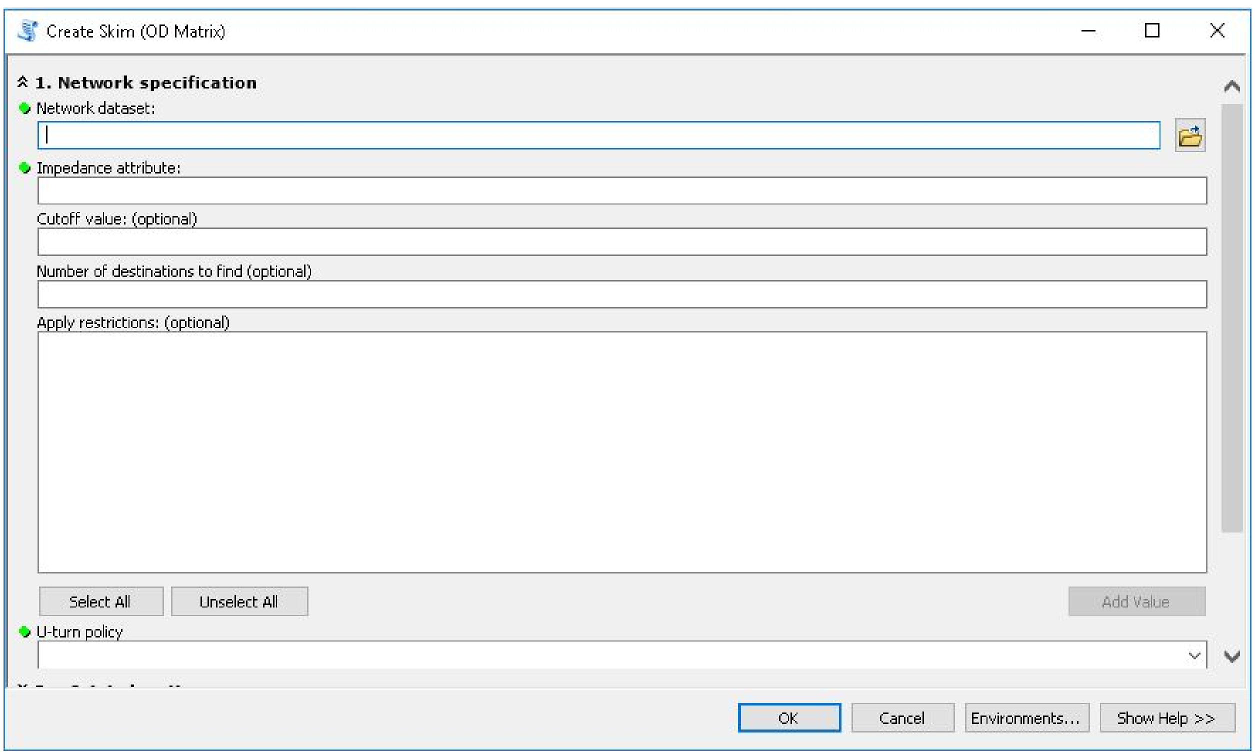
Decay rates define how travel impedance alters the value of a given activity. In general, activities that are far away are less attractive than activities that are nearby. The Manage Decay Rates tool located in the MMA toolbox is used for creating and modifying these values.

2. Create Skims (CS)

Use the Create Skims tool in the MMA toolbox to create a transit skim. This tool interface consists of five sections, which are described in detail below.

CS.1. Network specification

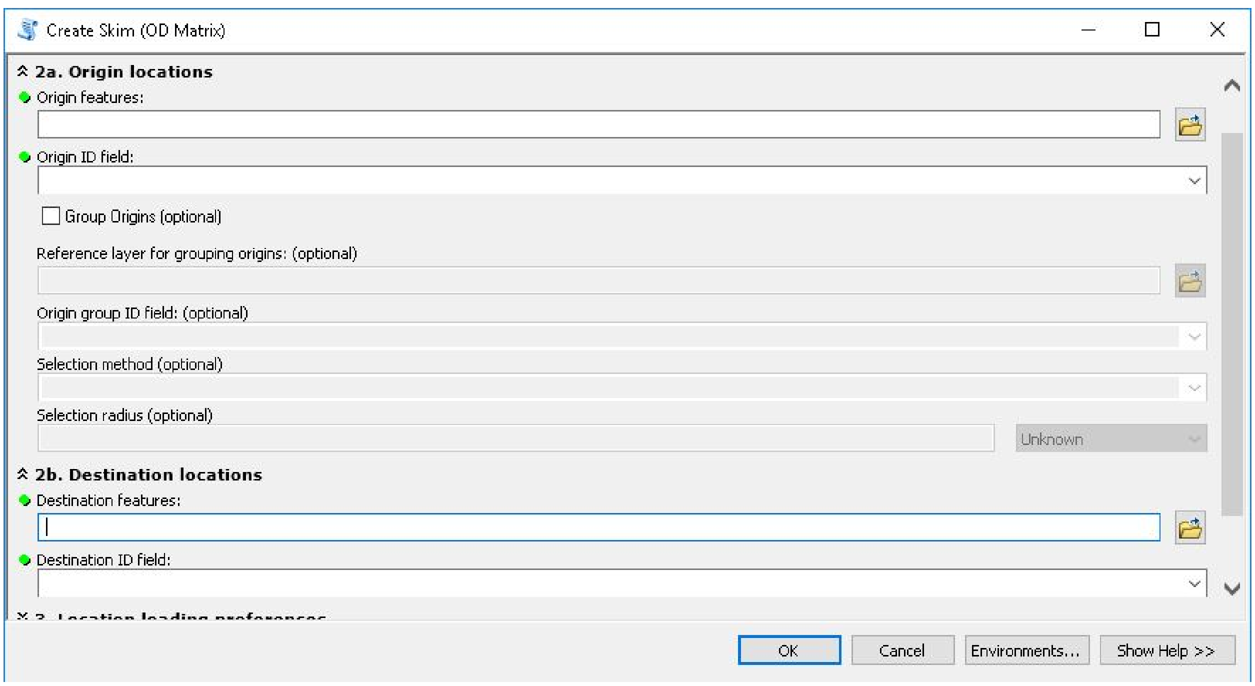
This section of the tool provides details of the network dataset supporting the analysis.



Field	Description	Recommended Value
Network dataset	The built transit network dataset	
Impedance attribute	Travel time impedance attribute in the network dataset	Transit-Min
Cutoff value (optional)	Cutoff value: 60 minutes	60
Number of destinations to find (optional)	Leaving this blank will find all destinations within the cutoff	
Apply restrictions (optional)	This network was constructed such that no restriction attributes were required, so this list appears blank. If restrictions have been set up, choose the appropriate restrictions to apply when running the analysis (do not walk on limited access highways, e.g.)	
U-turn policy	In most applications for walking or transit, there is no reason to adopt a strict U-turn policy, so “ALLOW_UTURNS” is generally recommended	Allow U turns

CS.2. OD locations

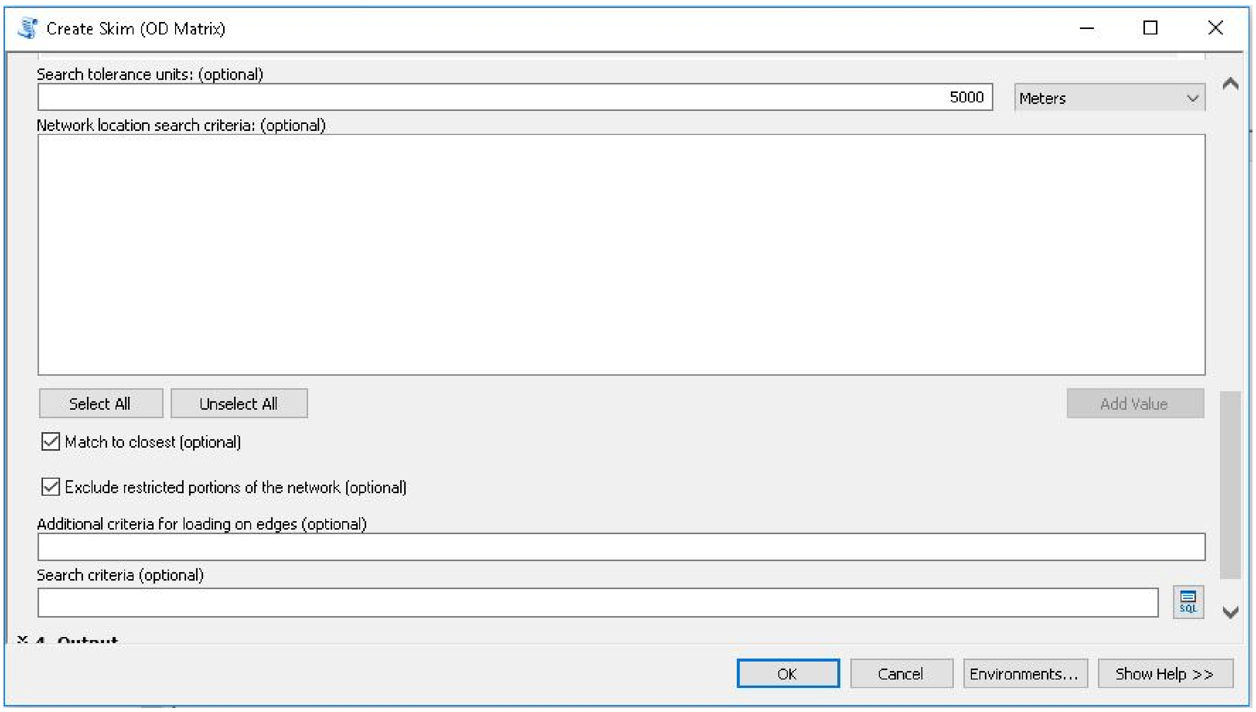
This section specifies the features that will represent origin and destination locations and how to group these features (if desired or needed for memory management).



Field	Description	Recommended Value
Origin features	Origins for statewide transit analysis are represented by Level 2 zonal centroids from the MSTM.	L2_Centroids.shp
Origin ID field	A field identifying each zone will be available in the centroids feature class to use as the "ID field" for each location type.	N
Destination features	Origins for statewide transit analysis are represented by Level 2 zonal centroids from the MSTM.	L2_Centroids.shp
Destination ID field	A field identifying each zone will be available in the centroids feature class to use as the "ID field" for each location type.	N

CS.3. Location loading preferences

This section defines how OD features will load onto the network using pre-calculated network location fields or spatial analysis.



Field	Description	Recommended Value
Search tolerance (optional)	Search for loadable features within this tolerance (most locations will still load on features much nearer than the tolerance if it is set generously)	5000 Meters
Network location search criteria (optional)	Origins and destinations (zone centroids) should only be able to load onto pedestrian links – not directly at bus stops or on transit lines.	Only check L3_Hwy End
Match to closest (optional)	Match to closest should be checked in almost all applications	Check
Exclude restricted portions of the network (optional)	Excluding restricted portions of the network is usually a good idea so that locations load on traversable links.	Check
Additional criteria fr loading on edges (optional)	Additional search criteria may be applied to links in the network.	L3_HWY
Search criteria (optional)		SWFT=11

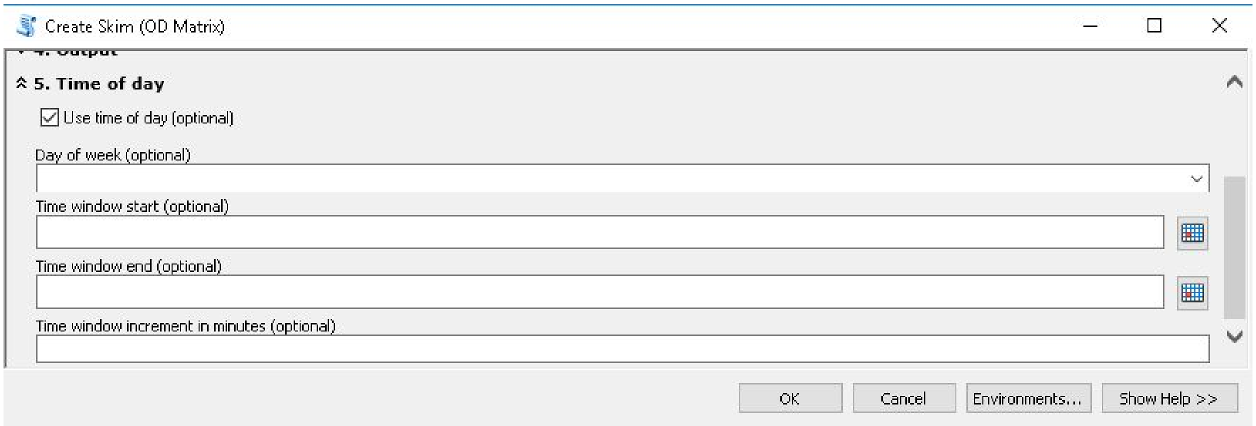
CS.4. Output

This section specifies where the output table(s) will be stored and how to name them.

Field	Description	Recommended Value
Output workspace	The create skims tool will produce at least one skim table and possibly multiple tables, depending on how the form has been completed. The tool needs to know where to save these outputs and a generalized analysis name.	Skims[Create a new folder for the base and for each project]
Analysis name	The output tables will include the analysis name as well as suffixes to describe their geography or time of day attributes, as needed.	Tran_Prj#

CS.5. Time of day

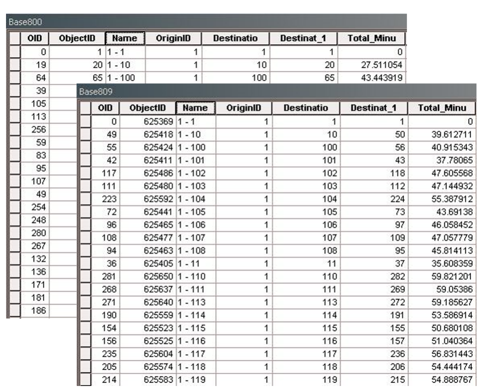
Specify the time of day for the analysis and setup iterative runs covering multiple times throughout the day. Chapter 30 parameters for these variables are indicated in the screenshot below.



Field	Description	Recommended Value
Use time of day (optional)	The create skims tool will produce at least one skim table and possibly multiple tables, depending on how the form has been completed. The tool needs to know where to save these outputs and a generalized analysis name.	Checked
Day of week (optional)	To generalize analysis to a typical day of the week, select a value from the "Day of Week" menu. If using a generalized day of the week, select "Time Only" in the "Time window start" dialog, and specify the initial departure time to analyze.	Wednesday
Time window start (optional)	If analyzing a specific day, select "Date and Time" in the "Time window start" dialog, and specify the date on the calendar as well as the initial departure time to analyze (the "day of week" field will be ignored if a specific date is selected)	7:00 AM
Time window end (optional)	If analyzing multiple departure times, specify the closing time for the window as "Time only" in the "Time window end" dialog	8:30 AM
Time window increment in minutes (optional)	If analyzing multiple departure times, specify the interval of minutes.	9

CS.Output Resulting Skim File(s)

The Create Skims tool produces multiple tables with estimated travel times between zones for each specified departure time. Travel times and accessibility vary by time of day.



Base800

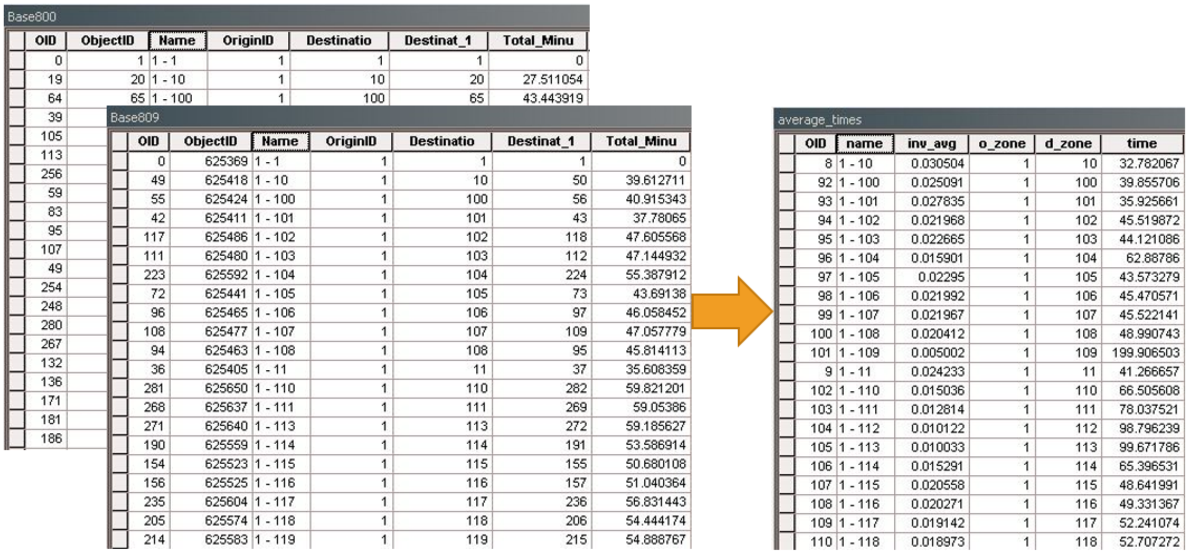
OID	ObjectID	Name	OriginID	Destinatio	Destinat_1	Total_Minu
0	1	1 - 1	1	1	1	0
19	20	1 - 10	1	10	20	27.511054
64	65	1 - 100	1	100	65	43.443919
39						
105						
113						
256						
59						
83						
95						
107						
49						
254						
248						
280						
267						
132						
136						
171						
181						
186						

Base809

OID	ObjectID	Name	OriginID	Destinatio	Destinat_1	Total_Minu
0	625369	1 - 1	1	1	1	0
49	625418	1 - 10	1	10	50	39.612711
55	625424	1 - 100	1	100	56	40.915343
42	625411	1 - 101	1	101	43	37.78065
117	625486	1 - 102	1	102	118	47.605568
111	625480	1 - 103	1	103	112	47.144932
223	625592	1 - 104	1	104	224	55.387912
72	625441	1 - 105	1	105	73	43.69138
96	625465	1 - 106	1	106	97	46.058452
108	625477	1 - 107	1	107	109	47.057779
94	625463	1 - 108	1	108	95	45.814113
36	625405	1 - 11	1	11	37	35.608359
281	625650	1 - 110	1	110	282	59.821201
268	625637	1 - 111	1	111	269	59.05386
271	625640	1 - 113	1	113	272	59.185627
190	625559	1 - 114	1	114	191	53.586914
154	625523	1 - 115	1	115	155	50.680108
156	625525	1 - 116	1	116	157	51.040364
235	625604	1 - 117	1	117	236	56.831443
205	625574	1 - 118	1	118	206	54.444174
214	625583	1 - 119	1	119	215	54.888767

3. Produce average travel times skim

Separate transit skims are created for specific departure times as specified in the "CS.5. Time of Day" above. All of the resulting skims will be stored in a single folder for calculating average transit travel times during the travel period specified by the Time Window settings. Run the Average travel times tool to summarize the multiple skims created during the "Create skims" step into a single skim reflecting typical conditions for the travel period.



Base800

OID	ObjectID	Name	OriginID	Destinatio	Destinat_1	Total_Minu
0	1	1 - 1	1	1	1	0
19	20	1 - 10	1	10	20	27.511054
64	65	1 - 100	1	100	65	43.443919
39						
105						
113						
256						
59						
83						
95						
107						
49						
254						
248						
280						
267						
132						
136						
171						
181						
186						

Base809

OID	ObjectID	Name	OriginID	Destinatio	Destinat_1	Total_Minu
0	625369	1 - 1	1	1	1	0
49	625418	1 - 10	1	10	50	39.612711
55	625424	1 - 100	1	100	56	40.915343
42	625411	1 - 101	1	101	43	37.78065
117	625486	1 - 102	1	102	118	47.605568
111	625480	1 - 103	1	103	112	47.144932
223	625592	1 - 104	1	104	224	55.387912
72	625441	1 - 105	1	105	73	43.69138
96	625465	1 - 106	1	106	97	46.058452
108	625477	1 - 107	1	107	109	47.057779
94	625463	1 - 108	1	108	95	45.814113
36	625405	1 - 11	1	11	37	35.608359
281	625650	1 - 110	1	110	282	59.821201
268	625637	1 - 111	1	111	269	59.05386
271	625640	1 - 113	1	113	272	59.185627
190	625559	1 - 114	1	114	191	53.586914
154	625523	1 - 115	1	115	155	50.680108
156	625525	1 - 116	1	116	157	51.040364
235	625604	1 - 117	1	117	236	56.831443
205	625574	1 - 118	1	118	206	54.444174
214	625583	1 - 119	1	119	215	54.888767

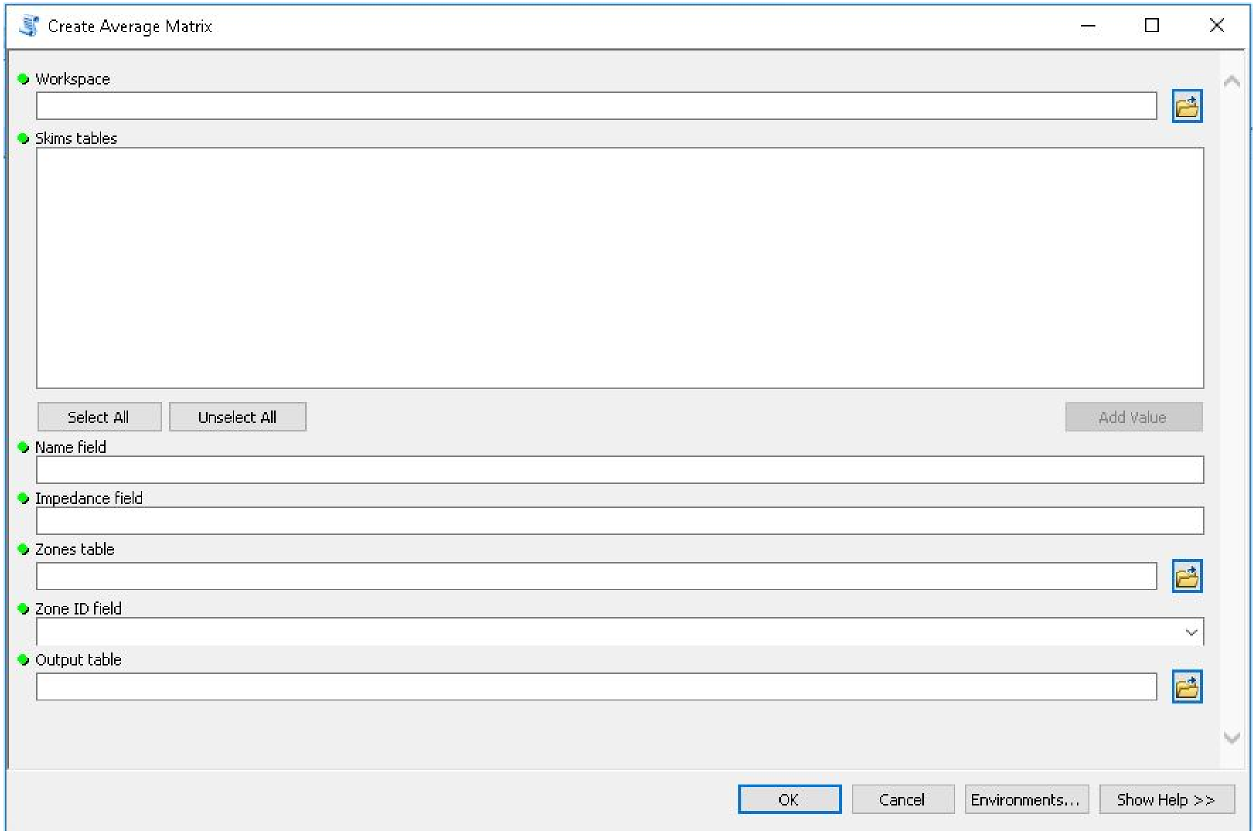
average_times

OID	name	inv_avg	o_zone	d_zone	time
8	1 - 10	0.030504	1	10	32.782067
92	1 - 100	0.025091	1	100	39.855706
93	1 - 101	0.027835	1	101	35.925661
94	1 - 102	0.021968	1	102	45.519872
95	1 - 103	0.022665	1	103	44.121086
96	1 - 104	0.015901	1	104	62.88786
97	1 - 105	0.02295	1	105	43.573279
98	1 - 106	0.021992	1	106	45.470571
99	1 - 107	0.021967	1	107	45.522141
100	1 - 108	0.020412	1	108	48.990743
101	1 - 109	0.005002	1	109	199.906503
9	1 - 11	0.024233	1	11	41.266657
102	1 - 110	0.015036	1	110	66.505608
103	1 - 111	0.012814	1	111	78.037521
104	1 - 112	0.010122	1	112	98.796239
105	1 - 113	0.010033	1	113	99.671786
106	1 - 114	0.015291	1	114	65.396531
107	1 - 115	0.020558	1	115	48.641991
108	1 - 116	0.020271	1	116	49.331367
109	1 - 117	0.019142	1	117	52.241074
110	1 - 118	0.018973	1	118	52.707272

The "Name" field should be "Name" if the skims were developed using the "Create Skims" tool. The name field anticipates values structured as "{origin name} – {destination name}" (note the " – " delimiter).

The "impedance" field stores travel time estimates between each zone pair for specific departure times, as specified in the Time Window settings of the "Create Skims" tool.

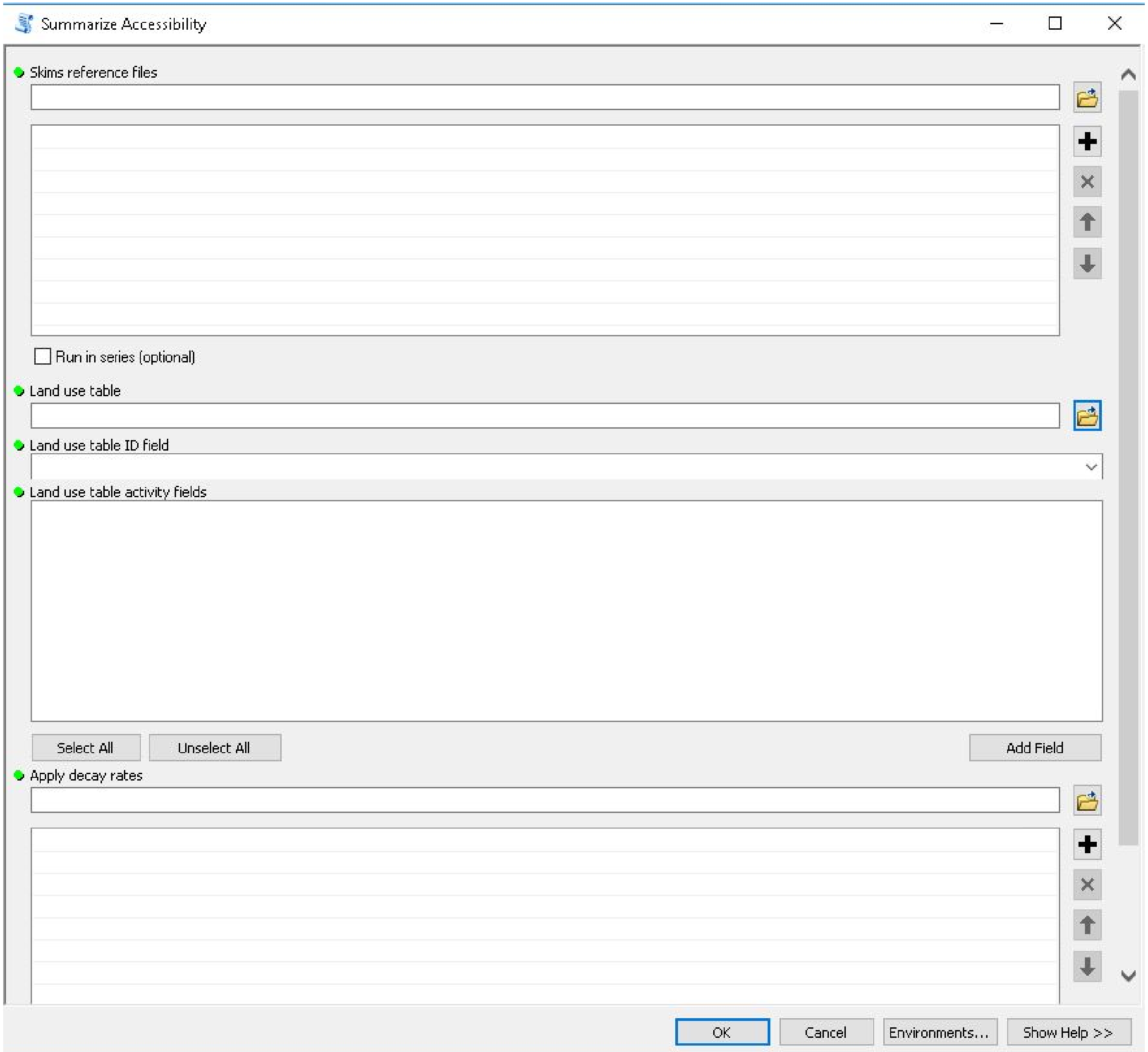
Reference zones and ID information are provided so that origin and destination zones can be properly indexed during the averaging process. These zones should reflect those used during the skim generation process.



Field	Description	Recommended Value
Workspace with skims tables (OD matricies)	Separate transit skims are created for specific departure times as specified in section 5 of the "Create OD Matrix" tool. All of the resulting skims should be stored in a single folder for calculating average transit travel times during the travel period. The folder with the time-of-day results is provided here, and the specific tables to summarize are selected from the list below.	
Skim tables (OD matrices) to average	Attributes of the time-of-day skims are listed to ensure the correct columns are referenced when developing the period-wide average skim. The "Name" field should be "Name" if the skims were developed using the "Create OD Matrix" tool. The name field anticipates values structured as "[origin name] – [destination name]"	
Skim tables (OD matrices) "Name" field	Attributes of the time-of-day skims are listed to ensure the correct columns are referenced when developing the period-wide average skim. The "Name" field should be "Name" if the skims were developed using the "Create OD Matrix" tool. The name field anticipates values structured as "[origin name] – [destination name]"	
Skim table (OD matrix) impedance weight field	The "impedance weight" field is the field storing time-of-day travel time estimates between each zone pair.	
Reference zones	Reference zones and ID information are provided so that origin and destination zones can be properly indexed during the averaging process. These zones should reflect those used during the skim generation process.	
Zone ID field	Reference zones and ID information are provided so that origin and destination zones can be properly indexed during the averaging process. These zones should reflect those used during the skim generation process.	
Output table	Location to create database.	

4. Summarize accessibility

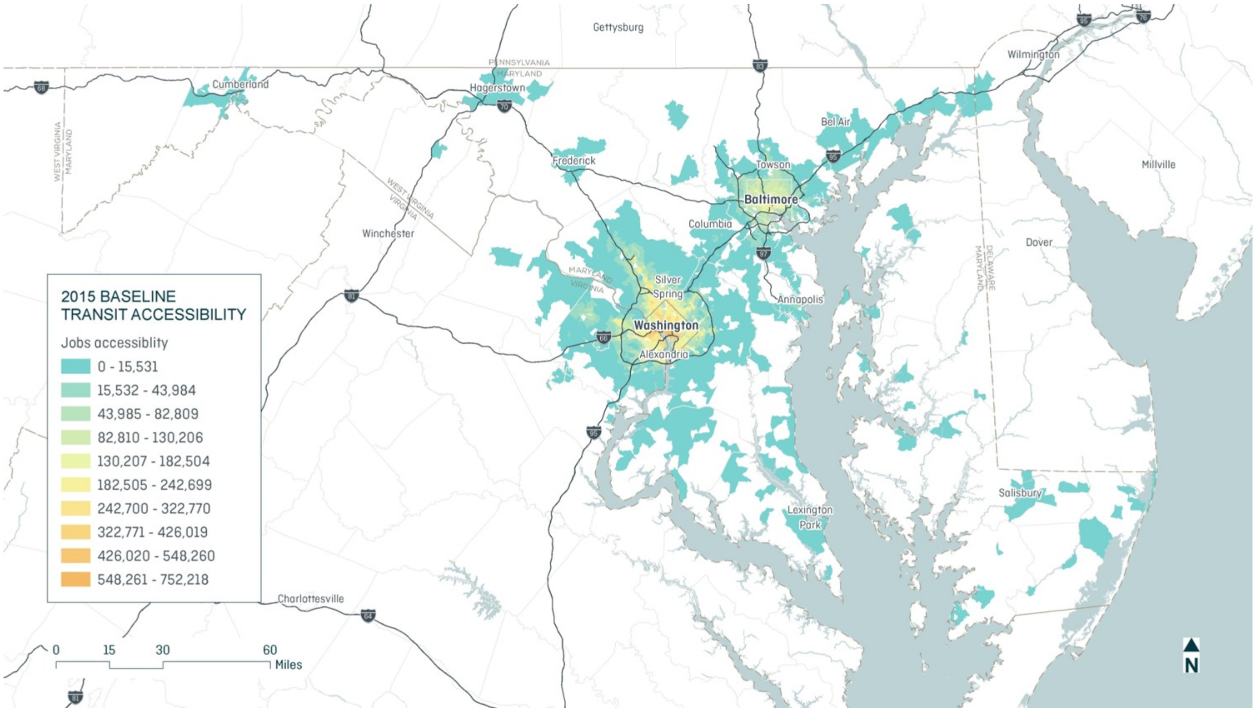
This is the final step in developing MMA scores for each scenario. Use the Summarize Accessibility tool in the MMA toolbox to process skims and zonal data to produce accessibility scores by zone.



Field	Description	Recommended Value
Skim reference files	Skim reference files (JSON format) store metadata about skim tables. These files are parsed by the "Summarize Accessibility" tool and define which fields to use for identifying origin/destination zones and travel time information. The tool can handle multiple skim reference files. Multiple files should be used when: • Analyzing multiple modes at the same geographic scale (run in series) A single study area requires multiple skim tables due to its size (i.e., if origins were "grouped" in the "Create OD Matrix (skim)" tool) (do not run in series) • Accessibility is being summarized for multiple departure times (run in series)	
Run in series (optional)	When the Summarize Accessibility tool is "run in series," a distinct output table will be created for each skim reference file provided in the list at top. Otherwise, all results will be "collapsed" into a single table.	Not checked
Land use table	Accessibility is summarized based on land use data that define the number of activities in each zone. For each origin zone, the number of activities at reachable destinations is summarized. The table of land uses providing these zonal data must be specified here.	
Land use table ID field	In the skims files, the origin and destination zones are stored using zone ID values. The corresponding ID values for the land use table are specified here. The data type for the land use table ID field should match the data type for OD data stored in the skims tables (if OD values are stored as text, the land use table ID field should be a text field also, e.g.)	
Land use table activity fields	Activities to summarize are listed here. Multiple fields may be selected. In the example, access to "education" and "health care" jobs will be summarized (as well as all jobs, which is not pictured due to the length of the fields list)	
Apply decay rates	Decay rates define the value of time, weighting destinations nearby as being more valuable than destinations farther away. They can also be used to create "time slices" of accessibil-	

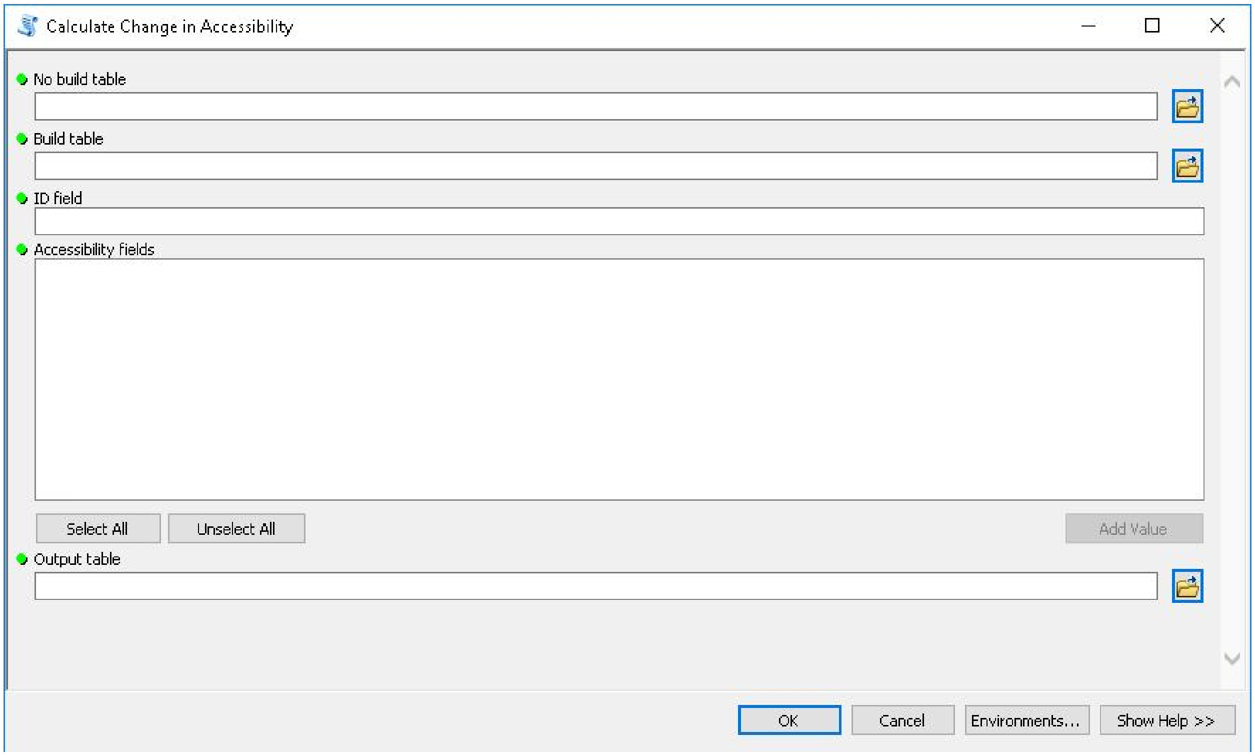
Summarize Accessibility Results

The tool yields one or more tables of accessibility results that can be joined to a zone feature class for mapping.



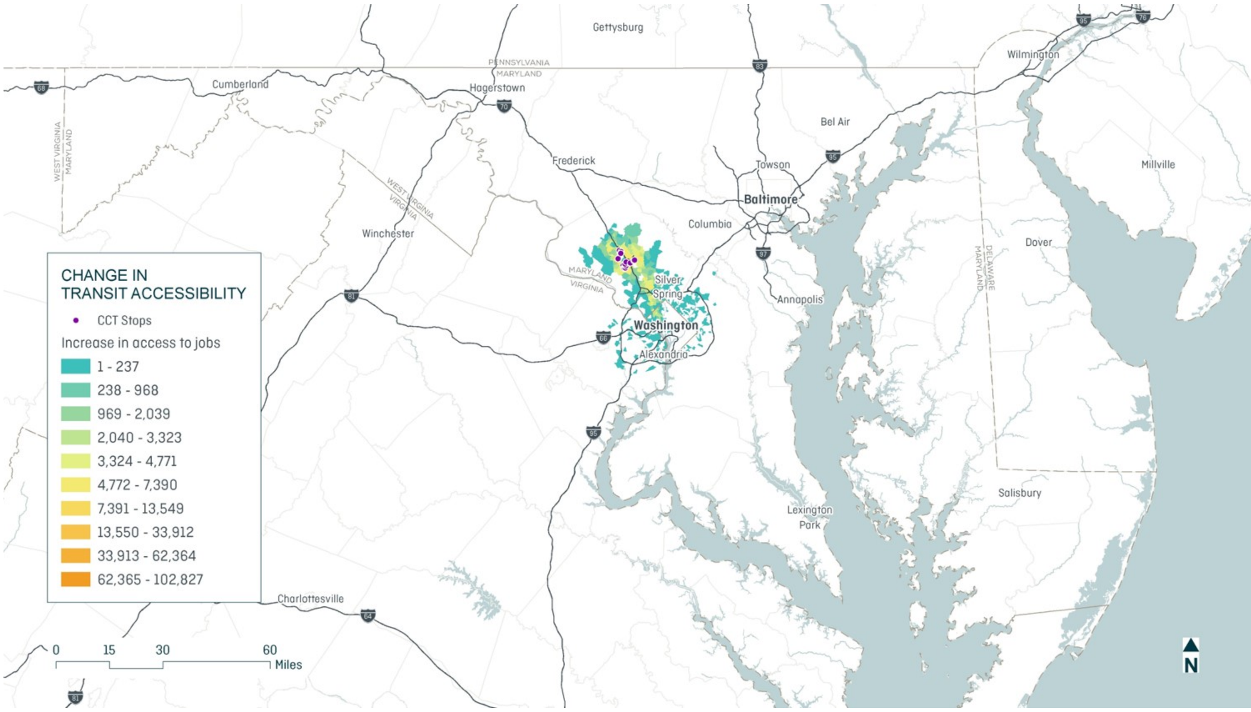
5. Calculate change in accessibility

Use the Calculate Change in Accessibility tool to understand how accessibility is modified by a project altering the transportation system and/or land uses. Provide a "no build" table reflecting baseline accessibility scores and a build table reflecting new accessibility scores assuming the project is implemented.



Field	Description	Recommended Value
No build table	The "no build" table is the table containing the baseline accessibility scores.	
Build table	The "build" table is the table containing the project accessibility scores. The tool assumes each table has the same structure (column names and data types).	
ID field	The "ID field" is the zone ID for zones in both tables.	
Accessibility fields	The "Accessibility fields" are the accessibility scores for which the tool will produce the "change" values for. In the example, "HBWA_TOTAL_40" in the no build table will be subtracted from "HBWA_TOTAL_40" in the build table to produce a field called "HBWA_TOTAL_40" in the output table representing the change in accessibility for that activity (gravity-weighted jobs in 2040 in this case)	
Output table	The "output table" is the table that will be produced by the tool defining changes in accessibility for the selected "accessibility fields."	

Change in accessibility result

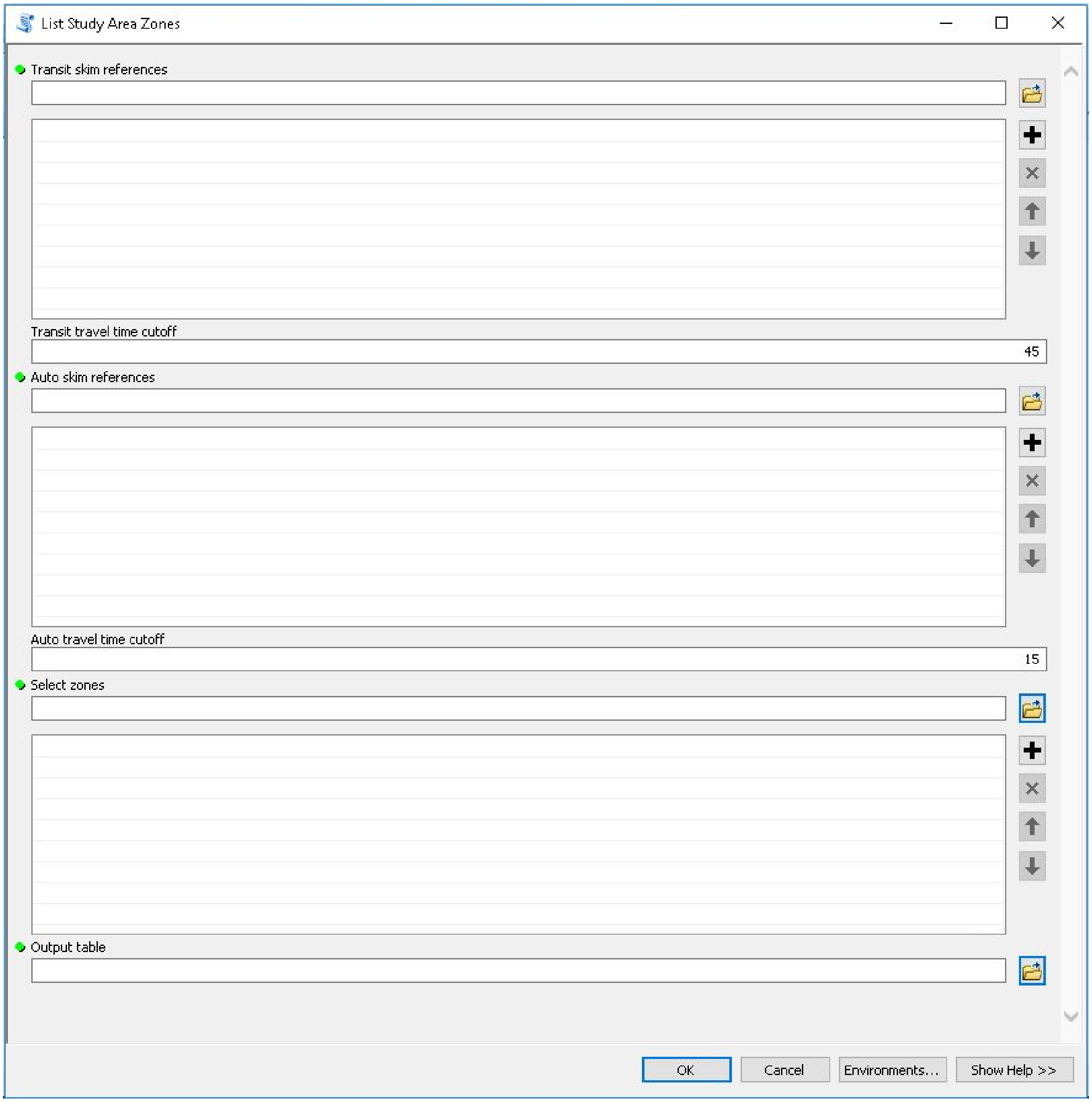


3.3.4 Chapter 30 Score Development

1. Define project study area

The List Study Area Zones tool is used to create a table defining all zones within the project study area.

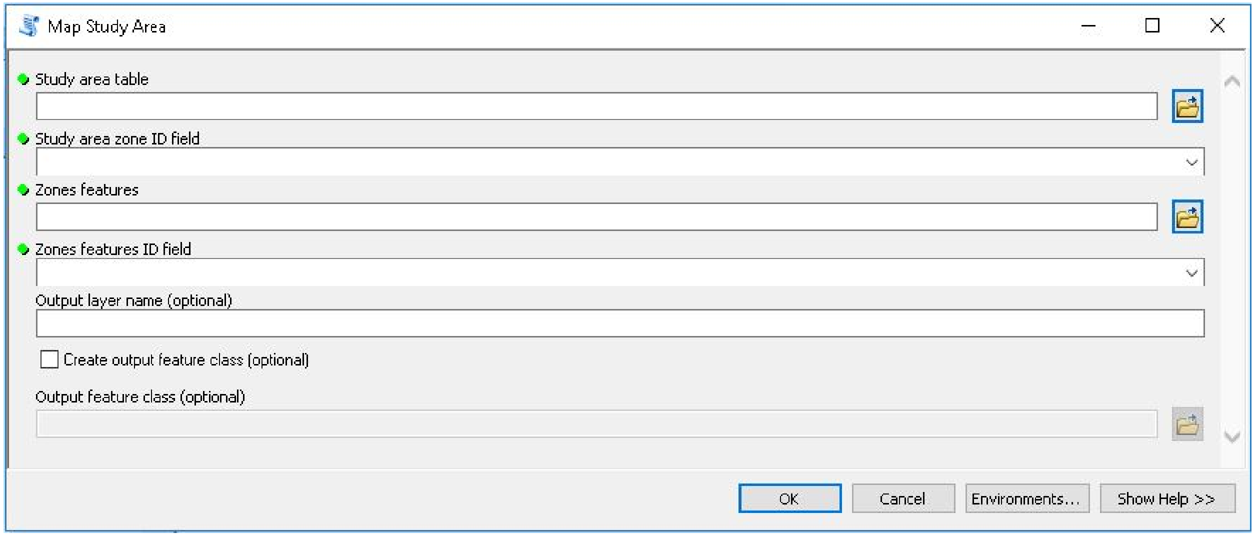
The project study area is based on travel time to the zones in which the project is implemented. Project zones are zones intersecting project features (new stops, stops affected by frequency enhancements, e.g.). Any zones within 45 minutes by transit (in the project no-build scenario) are part of the study area as are any zones within 15 minutes by driving (based on MSTM highway skims, which are provided as an a priori data source to inform this step of score development).


List Study Area Zones
Transit skim references
Transit travel time cutoff
45
Auto skim references
Auto travel time cutoff
15
Select zones
Output table
OK
Cancel
Environments...
Show Help >>

Field	Description	Recommended Value
Transit skim reference	The transit skim reference files point to the transit skim(s) that	
Transit travel time cutoff	define average OD travel times for the project “no-build” condition.	15 minutes
Auto skim reference	The auto skim reference files point to the auto skim(s) that define average OD travel times for the base condition.	
Auto travel time cutoff	Zones that can reach the “project zones” within the auto travel time cutoff will be included in the study are definition.	15 minutes
Select zones	To look up which zones meet the above travel time criteria (consistent with the study area definition), the tool must know the “project zones” (zones that overlap affected project features). Each project zone ID is listed here, either as separate entries or as a semicolon-delineated list of values.	L2 zones
Output table	The user specifies the study area definition table to be produced by the tool.	

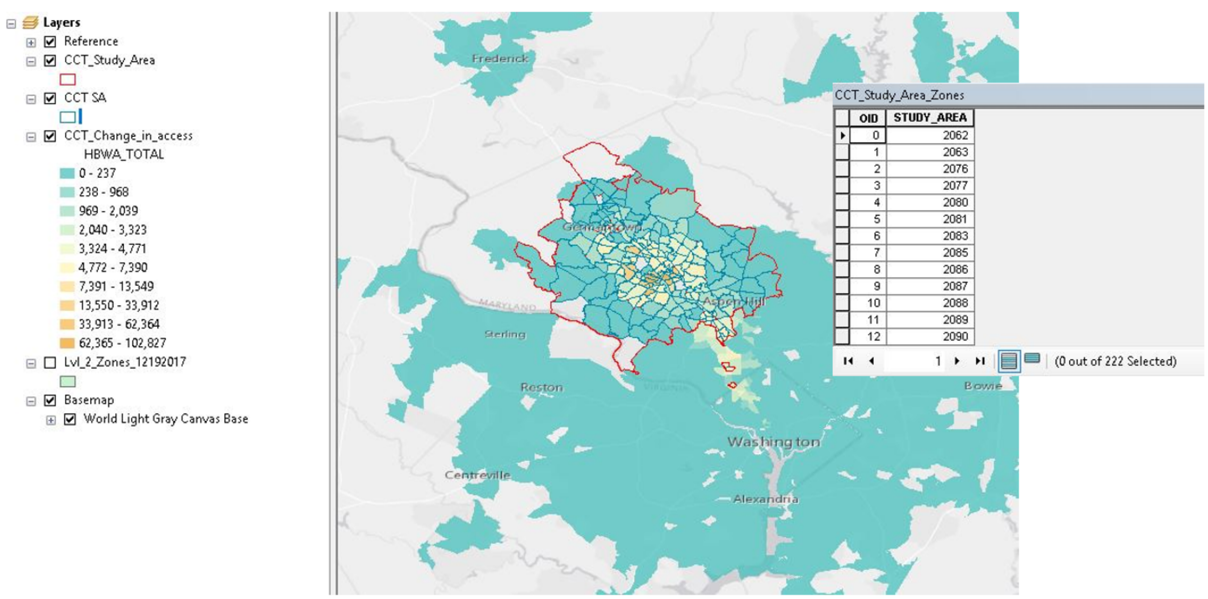
2. Map project study area

The Map Study Area tool is used to plot a table of project study area zones (created using the List Study Area Zones tool) on a map.



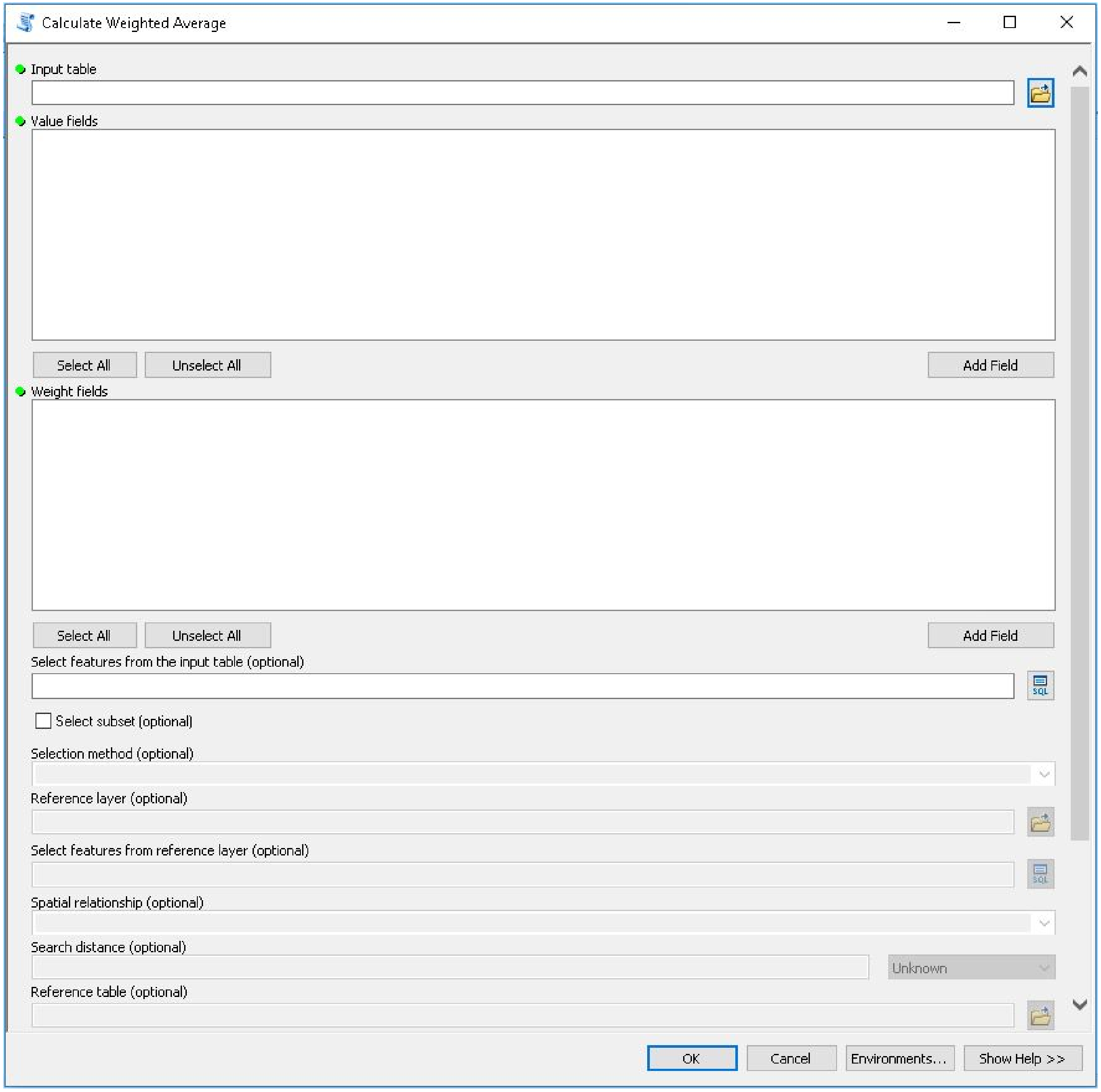
Field	Description	Recommended Value
Study area table	The study area zones table is a table containing a list of zones included in the project study area (this can be generated using the "List study area zones" tool).	
Study area zone ID field	The study area zoneID field is the field in the study area zones table to reference when looking up which zones are in the study area.	"STUDY_AREA" is the default
Zones features	The zones feature class is a polygon feature class that will be used to map the study are based on the zone values in the zone features ID field	
Zone features ID field	The zone IDs listed in the study area zones table.	
Output layer name (optional)	The tool produces a feature layer, querying the zones feature class based on values in the study area zones table; the resulting layer name can be specified here. If blank, the layer name will be given a random unique name.	
Create output feature class (optional)	Optionally, the results can be dissolved into a single feature representing the project study area. If this option is selected, provide an output feature class.	
Output feature class (outional)	Location to save dataset.	

Map project study area output



3. Calculate weighted average

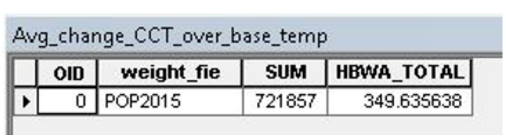
The Calculate Weighted Average tool summarizes the average accessibility score for a collection of zones based on the distribution of popuation groups in those zones. For Chapter 30 scoring, this tool needs to be run twice - once to generate average change in access to jobs scores for the general population and once to generate average change in access to jobs scores for disadvantaged populations. The provided MSTM Level 2 zones contain the appropriate population group data. In both cases, the table of project study area zones should be used to select a subset of MSTM level 2 zones, allowing the average change in access to be assessed only for zones in the project study area.



Field	Description	Recommended Value
Input table	The input table is the table of zonal accessibility scores from which the weighted average will be calculated	
Value fields	The value fields are the fields for which a weighted average score will be calculated. The weighted average of "HBWA_TOTAL" will be produced in this example (not visible due to length of field list).	
Weight fields	The weight fields are the fields that inform the weighting in the weighted average calculation. For a population-weighted average, for example, choose the field representing zonal populations in this list (POP_2015 field not visible due to length of list).	
Select features from the input table (optional)	A SQL query can be specified to allow the user to run the analysis for specific zones in the table (this will generally not be necessary for Chapter 30 scoring purposes).	
Select subset (optional)	The "Select subset" option allows the user to utilize features or table records from another data source (such as the study area definition tables, e.g.) to limit the records that will be utilized in the development of weighted average results. If the "Selection method" is "SPATIAL": The user may define a reference feature layer or feature class for selecting features to analyze (the input table must also be a feature layer for this option to work) A SQL expression can be defined to focus on specific features in the reference layer The spatial relationship and selection tolerance (search distance) can be set to define which features in the input table will be selected based on their spatial relationship to features in the reference layer. If the "Selection method" is "TABULAR": The user may define a reference table for selecting feature to analyze. The features in the input table to select will be those with common values in the reference table key field and the input table lookup field A SQL expression can be defined to focus on specific records in the reference table	
Selection method (optional)		
Reference layer (optional)		
Select features from reference layer (optional)		
Spatial relationship (optional)		
Search distance (optional)		
Reference table (optional)		
Reference table key field (optional)		
Output table	The output table is the summary table of weighted average results that will be produced by the tool. The table has columns for each value field and rows for each weight field.	

Calculate weighted average results

The output of this tool is is used to generate final figures for the report.



Avg_change_CCT_over_base_temp

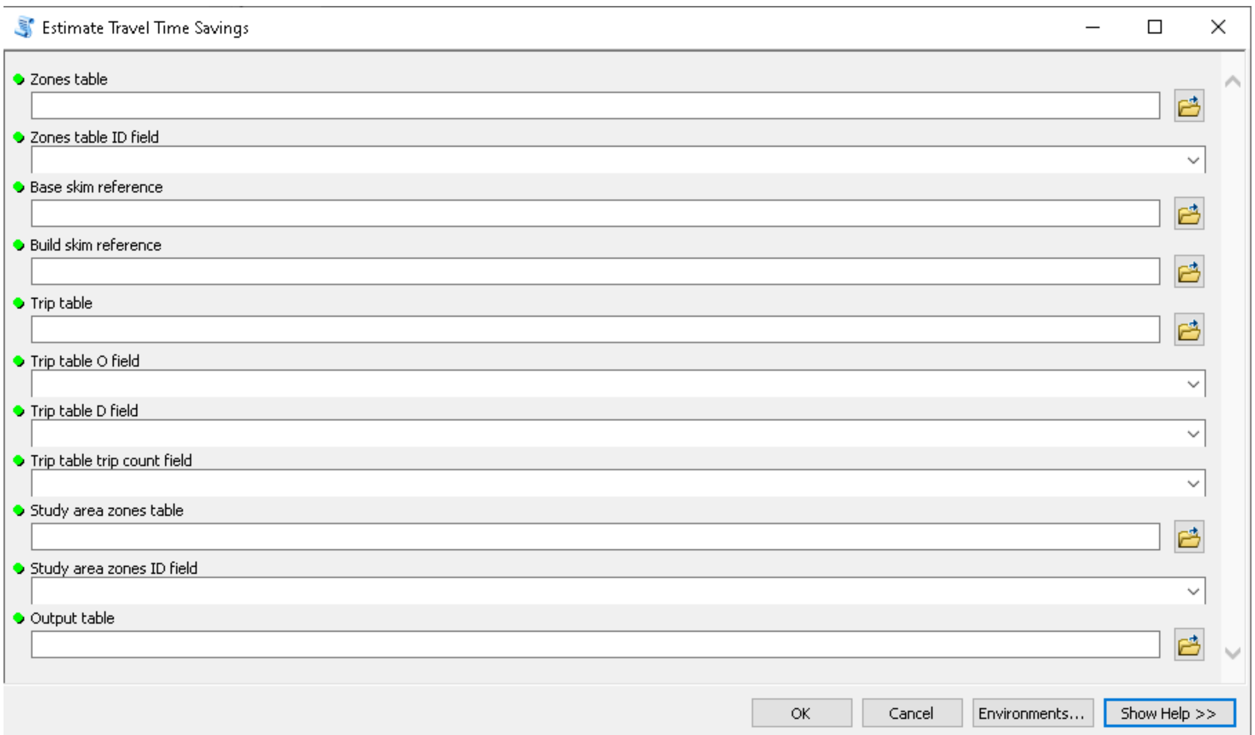
OID	weight_fie	SUM	HBWA_TOTAL
0	POP2015	721857	349.635638

4. Calculate travel time savings

The Estimate Travel Time Savings tool analyzes the differences in travel times between a "build" skim and a "base" skim. The travel time savings estimates are weighted by a trip table such that projects that improve travel times between heavily-traveled O-D pairs will generate greater savings relative to projects the improve travel times between sparsely-traveled O-D pairs. For Chapter 30 scoring purposes, the travel time savings calculation generated by the geoprocessing tool assess how transit travel times improve for existing transit riders. For new transit riders, travel time savings are expected to accrue to highway users. The travel time savings calculation consists of four steps:

- Calculate transit travel time savings from all origins in the study area to all destinations for all MSTM level 2 zones.
- Multiply this matrix against the person trip table, zeroing out all cells where transit travel is not possible.
- Compute a weighted average transit travel time savings (a single number) for transit users.
- Multiply this travel time savings by estimated total daily existing ridership on the project (supplied by the project applicant). Applicants supply total ridership. For 2019 Chapter 30 scoring, it was assumed that 80 percent of the total ridership estimated provided by the applicant represents existing ridership, with the remaining 20 percent representing new riders.

The geoprocessing tool generates travel time savings by origin zone in the study area for mapping purposes. The output table can be summarized to generate a single estimate of average travel time savings throughout the study area. To do this, take column sums for the "SumTTChg" and "SumTrips" columns for the whole table. This yields the total minutes saved by travelers from each zone and the total trips from each zone across the study area. Then divide the total "SumTTChg" value by the "SumTrips" value for the average change in travel time on a per trip basis.



3.4 Chapter 30 Transit Project Report

Transit project scoring results are assembled into a document that contains key information about the project, data development and assumptions made for accessibility scoring, quality assurance mapping, and summaries of the accessibility impact findings. The report is organized into 5 sections, as outlined below. A brief description of the content expected in each section is provided here.

Links to the 2019 Transit Project Reports are also provided below for reference.

- US 29 BRT (Project 20-19)
- South Side Transit (Project 20-34)

Chapter I. Introduction

1. Overview of Project - *Provide a brief overview of the project include a project map.*

Chapter II. Coding Assumptions

1. Alignment – *Describe alignment and right of way needs.*
2. Attributes – *Describe project details and include relevant tables of the GTFS files to describe details of project. Include maps of the transit networks.*
3. Modifications of Existing GTFS Feeds – *Describe modifications required to code project in GTFS.*

Chapter III. Network Review Results

1. Network Dataset Configuration Review – *Provide network dataset configurations as part of a quality check.*
2. Connectivity Tests – *Provide service area analysis quality assurance tests.*
3. Shortest Path – *Provide route analysis quality assurance tests.*

Chapter IV. Reasonableness of results

1. Extent of Study Area – *Describe study area and development.*
2. Travel Time Contours to Project – *Map travel time contours used to develop the study area.*
3. MMA Results – *Map MMA results.*

Chapter V. Mapping and Final Results

1. Network (Project Links and Stops) – *Map of project.*
2. Build vs No Build – *Map average travel time changes by zone.*
3. Project Study Area – *Map study area boundary.*
4. Positive and Negative Accessibility Changes – *Map accessibility changes.*

CHAPTER 4

Chapter 30 Quality Assurance

4.1 GTFS feed validation

Google FeedValidator

GTFS feeds should be validated using the Google FeedValidator utility. Tool documentation is available on the tool's wiki page. This tool will check for a variety of GTFS feed issues, including:

- Missing files
- Missing table columns, rows, and values
- Coding errors
- Basic GTFS geometry

A full list of errors identified by this tool is included in the FeedValidator utility documentation.

Additional Validation

When the FeedValidator utility detects errors in a GTFS feed, appropriate modifications should be made to the GTFS feed to correct the issue. In some cases, errors produced by the validator will require futher investigation to identify the root cause.

As part of the 2019 socring process, a script was written in R to provide additional validation information beyond what the FeedValidator utility provides. Specifically, the script identifies stop-to-stop pairs in a given GTFS feed for which the feed's schedule information suggests abnormally fast or abnormally slow transit vehicle travel speeds. The tool, provided in the form of a R Markdown file, computes transit vehicle speed based on the linear distance between stops and the transit schedules present in the feed. The tool produces tables and maps of station pairs with servie speeds less than 4.3 miles per hour (7 kmph) or greater than 89 miles per hour (110 kmph).

The script was used to identify and map all stop-to-stop pairs with very high or very low speeds in the GTFS feeds used for Chapter 30 scoring. While resolving these schedule anomalies was beyond the scope of the transit project scoring effort, the maps help provide insight into the quality of the supporting GTFS data for future reference. These represent the base (no-build) network feeds, which are common to all scored projects. As such, each project's relative score remains a reliable estimate of the accessibility and travel time savings benefits of the project.

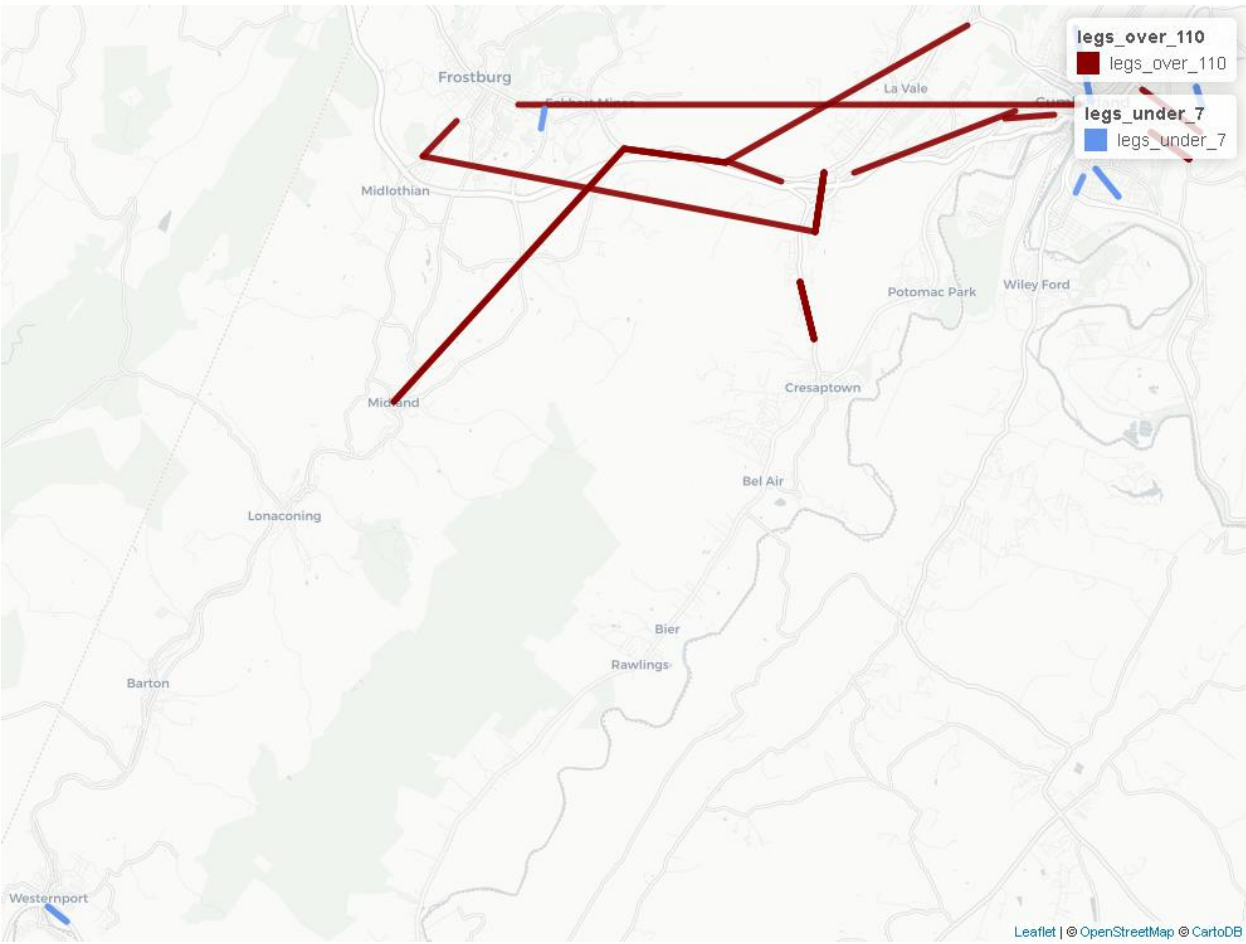
Maps and tables of stop-to-stop speeds are provided via the links listed below. Only feeds having speed issues are shown. Maps display speeds in kmph.

The findings of the GTFS validation procedures identifying very fast or very slow travel speeds between transit stops are provided on this page. Results are listed by agency name.

4.1.1 Validation Outputs

The following maps and tables are the outputs produced by the speed validator tool. This tool was run to validate base transit network GTFS feeds in cases where the FeedValidator utility indicated further validation steps were necessary.

Allegany Transit



from_stop_name	to_stop_name	route_id	route_long_name	max_kmph	max_distance_miles	number_of_trips
Main St & Sleeman St	Frederick St & Mechanic St	Gold	Country Club Mall / Frostburg	281.3	8.16	1
Walmart Grocery Entrance	Active Network, Inc	Silver	Morning Service	168.31	5.81	1
Red Hill Plaza (Cumberland Treatment Center)	LaVale Sheetz	Green 1	LaVale / Cresaptown / Bedford Rd	164.35	0.85	1
Golden Living Center	Urology Associates	Red 1	Willowbrook Rd / South Cumberland	162.16	0.67	8
Midland MD 36 & Big Lane Ave	Clarysville Motel	Purple	Westernport / LaVale / Bedford Rd	153.77	4.96	2
Opposite Lions Ball Field	Pyzano's	Yellow	Evening Service	145.41	2.51	1
Greene St & Lee St	Gold Dingle	Yellow	Evening Service	139.16	0.72	1
MVA	Tradewinds	Green 1	LaVale / Cresaptown / Bedford Rd	132.12	0.84	5
Red Hill Plaza (Cumberland Treatment Center)	Valley Plaza	Gold	Country Club Mall / Frostburg	129.29	4.02	1
Willowbrook Rd & Pine Ave	Brook Building	Yellow	Evening Service	126.09	1.04	1
Walmart Grocery Entrance	Lowe's & Burton's Plaza	FSU 3	FSU Saturday Shuttle	125.18	0.86	7
MVA	Tradewinds	Purple	Westernport / LaVale / Bedford Rd	122.21	0.84	1
MVA	Tradewinds	Silver	Morning Service	122.21	0.84	1
Active Network, Inc	Cordts PE Center Shelter	Yellow	Evening Service	116.53	0.72	1
Clarysville Motel	Opposite Red Hill Plaza (Cumberland Treatment Center)	Gold	Country Club Mall / Frostburg	113.74	1.51	3
Westernport Bus Stop	Westernport McDonalds	Purple	Westernport / LaVale / Bedford Rd	6.61	0.34	2
Allegany College of Maryland	Cumberland Meadows Apartments	Blue 1	White Oaks / Willowbrook Rd	6.04	0.31	4
Weis Market	Frostburg Plaza (ACS)	Silver	Morning Service	5.93	0.31	1
Virginia Ave & Industrial Blvd (Rite Aid)	Opposite HRDC	Yellow	Evening Service	5.89	0.27	1
Mountain View Apartments	Oldtown Rd & Race St	Silver	Morning Service	4.53	0.52	2
Allegany Nursing Home & Adult Day Care	Rose's	Silver	Morning Service	3.5	1.09	2

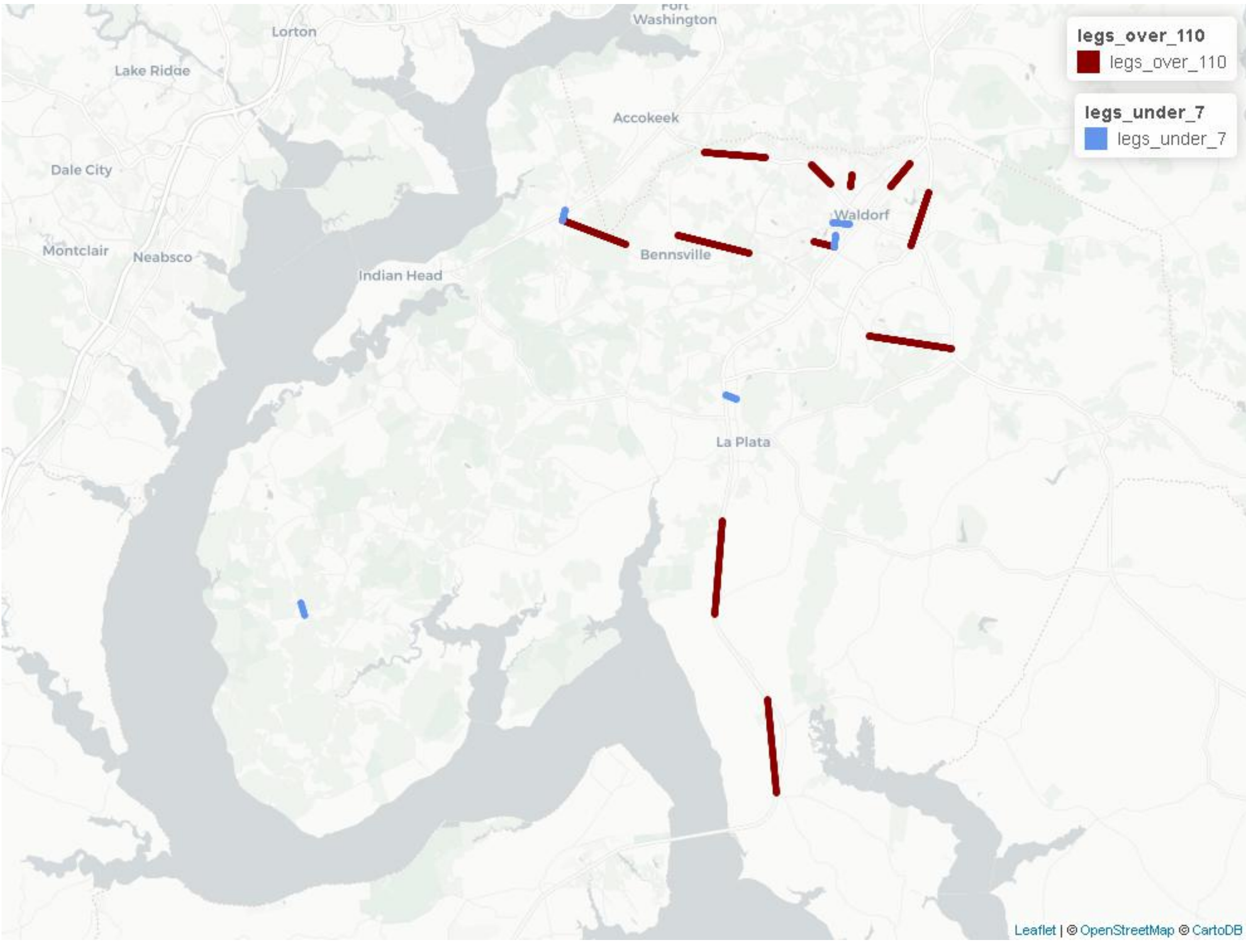
Calvert County Public Transit

from_stop_name	to_stop_name	route_id	route_long_name	max_kmph	max_distance_miles	number_of_trips
Williams Wharf Rd. and Kings Rd.	Broomes Island Rd. and Oyster House	6	Mid-County	867.84	3	2
Broomes Island Rd. and Oyster House	Ross Rd.	6	Mid-County	840.32	2.9	1
Dowell Rd.	Giant Food	7	Lusby Shuttle	633.33	3.28	3
Broomes Island Rd. and Oyster House	Broomes Island Rd. and Grays Rd.	6	Mid-County	288.75	4.49	1
Broomes Island Rd. and Williams Wharf Rd.	Williams Wharf Rd. and Kings Rd.	6	Mid-County	267.88	0.92	1
Saint Leonard Rd. and Calvert Beach Rd.	Courthouse	5	South Route	196.74	6.11	1
Sixes Rd. and T Hance Turn Around	Sixes Rd. and Adelina Rd.	6	Mid-County	185.69	2.88	2
Patuxent Plaza Shopping Center at Bus Stop	Solomons Info Center and Island	7	Lusby Shuttle	183.54	0.95	3
Detention and Substance Abuse Center	Yardley Hills at Community Building	1	Prince Frederick Shuttle I	179.98	0.93	1
Ross Rd. and Broomes Island Rd.	Broomes Island Rd. and Oyster House	6	Mid-County	170.19	2.64	1
Dowell Rd.	Southern Pines Community Center	7	Lusby Shuttle	135	3.5	6
Route 4 and Route 2	Route 2 and Mount Harmony Rd.	4	North Route	127.77	2.65	8
Route 2 and Mount Harmony Rd.	Route 260 and Route 261	4	North Route	125.85	3.91	3
Walmart	Detention and Substance Abuse Center	1	Prince Frederick Shuttle I	125.45	1.3	5
Giant Food	Patuxent Plaza Shopping Center at Bus Stop	7	Lusby Shuttle	112.59	2.92	3
Southern Pines Community Center	Giant Food	7	Lusby Shuttle	6.55	0.27	6
Mill Bridge Rd.	Coster Rd. and Bafford Rd.	5	South Route	6.36	0.33	1
Southern Pines Senior Apartments	Southern Pines Community Center	7	Lusby Shuttle	5.74	0.3	3
Giant Food	Southern Pines Community Center	7	Lusby Shuttle	5.24	0.27	9
Yardley Hills at Community Building	North Prince Frederick Blvd and Hallowing Point Rd	8	Charlotte Hall	4.69	0.58	1

Cecil Transit

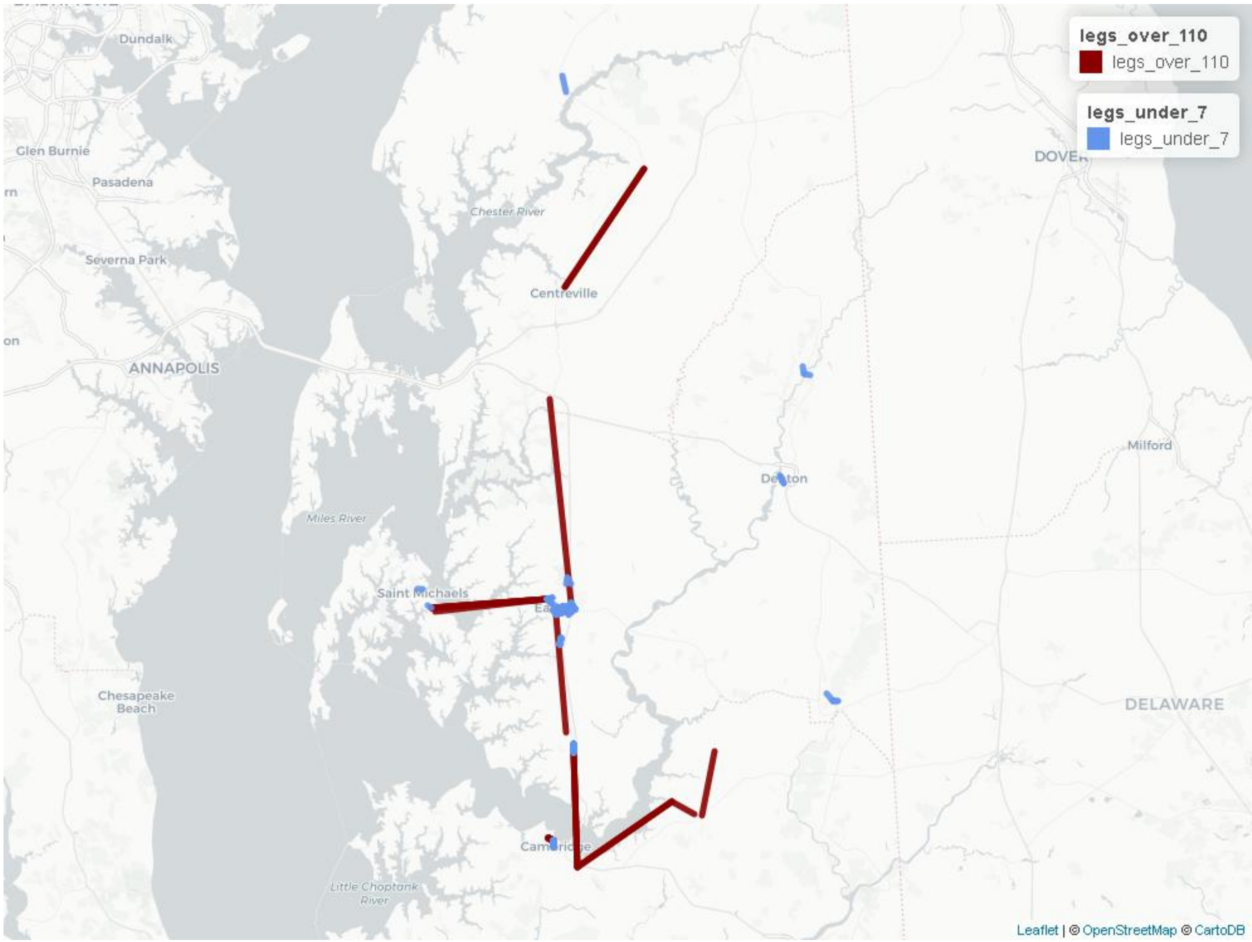
from_stop_name	to_stop_name	route_id	route_long_name	max_kmph	max_distance_miles	number_of_trips
RT 40 at Belvidere Rd. (EB)	RT 40 - Charlestown Crossing	24047	Perryville Connection	230.19	2.38	7
RT 40 - After Montgomery Dr.	RT 40 - Activity Center Ln.	24047	Perryville Connection	165.09	1.71	7
RT 40 - Across from Stony Run	RT 40 - Ritchie Bros.	24047	Perryville Connection	148.76	1.54	8
RT 40 - Charlestown Crossing	RT 40 - Stony Run Apts.	24047	Perryville Connection	147.54	1.53	5
Super Wal-Mart	RT 40 - Kohls	24046	Glasgow Connection	140.61	1.46	4
RT 40 - Ritchie Bros.	RT 40 - Belvidere Rd. (WB)	24047	Perryville Connection	114.49	2.37	8
RT 40 - Before Old Elk Neck Rd.	Landing Ln. & W. Main St.	24047	Perryville Connection	112.43	1.16	7
200 Chesapeake Blvd	RT 40 - Kohls	24046	Glasgow Connection	111.76	1.16	1
RT 40 - John Deere	Safeway - Food Court Entrance	24046	Glasgow Connection	6.55	0.37	1
4th St. and Ave. 'D'	Perry Pt. Bldg. 361	24047	Perryville Connection	6.25	0.52	1
200 Chesapeake Blvd	Acme - Big Elk Mall	24048	Mid -County Connection	6.04	2.19	2
Bridge St. Plaza - American Cash Traders	150 E. Main St.	24046	Glasgow Connection	5.06	0.31	1
RT 40 - Happy Harry's	RT 40 - La Grange	24046	Glasgow Connection	4.98	0.26	1
Davita Dialysis (Call in stop)	Acme - Big Elk Mall	24047	Perryville Connection	4.62	0.33	3
NE Walmart	NE - Walgreens / Food Lion	24048	Mid -County Connection	3.96	0.29	4
W. Main St. and Maffit St.	Acme - Big Elk Mall	24047	Perryville Connection	3.81	0.59	1
W. Main St. and Maffit St.	Davita Dialysis (Call in stop)	24047	Perryville Connection	3.55	0.26	3

Charles County VanGo



from_stop_name	to_stop_name	route_id	route_long_name	max_kmph	max_distance_miles	number_of_trips
Billingsley Rd & Piney Church Rd (Charlotte Hall)	Rt 5 S & Zachia Manor Ct	28	Charlotte Hall	177.23	2.45	13
Salvation Army Super Store	Washington Square	31	Pinefield	170.62	0.88	16
Mattawoman-Beantown Rd & Idlewood Park Rd	St Charles Pkwy & Northgate Pl	25	Business A	159.02	1.65	13
Berry Rd & Buttonbush Dr	Berry Rd & Greenwood Dr	24	Berry Road	150.31	0.78	14
Billingsley Rd & Middletown Rd	Bensville Rd & Bancroft Dr	29	Indian Head	138.57	2.15	15
Rt 301 S & Crossover Rd	Rt 301 S & Edge Hill Rd	30	Newburg	133.92	2.77	6
Relax Inn	Rt 301 N & Fairgrounds Rd	30	Newburg	132.72	2.75	7
Billingsley Rd & Prince Edward Dr	Billingsley Rd & Livingston Rd	29	Indian Head	126.46	1.96	15
Hamilton Rd & Stoney Cover Dr	Hamilton Rd & Firethorne St	31	Pinefield	125.18	0.32	15
St Patricks Dr & Highgate Pl	Waldorf Transfer Point	34	St. Charles C	118.71	0.61	13
Berry Rd & Sharperville Pl	Berry Rd & Bensville Rd	24	Berry Road	117.22	1.82	14
Billingsley Rd & Livingston Rd	Bryans Rd McDonalds (Indian Head)	29	Indian Head	6.91	0.29	15
Victoria Park Apartments	Charles County Plaza	26	Business B	6.5	0.47	12
Nanjemoy Community Center	Rt 6 & Liverpool Pt Rd	38	Nanjemoy	6.18	0.38	2
Rt 6 & Liverpool Pt Rd	Nanjemoy Community Center	38	Nanjemoy	6.18	0.38	2
LaPlata Transfer Point (Walmart Lot)	Heritage Green Pkwy & Lelia Ct	40	301 Connector	5.65	0.29	21
Waldorf Transfer Point	St Charles Towne Center Mall	33	St. Charles B	2.99	0.34	16

Delmarva Community Transit

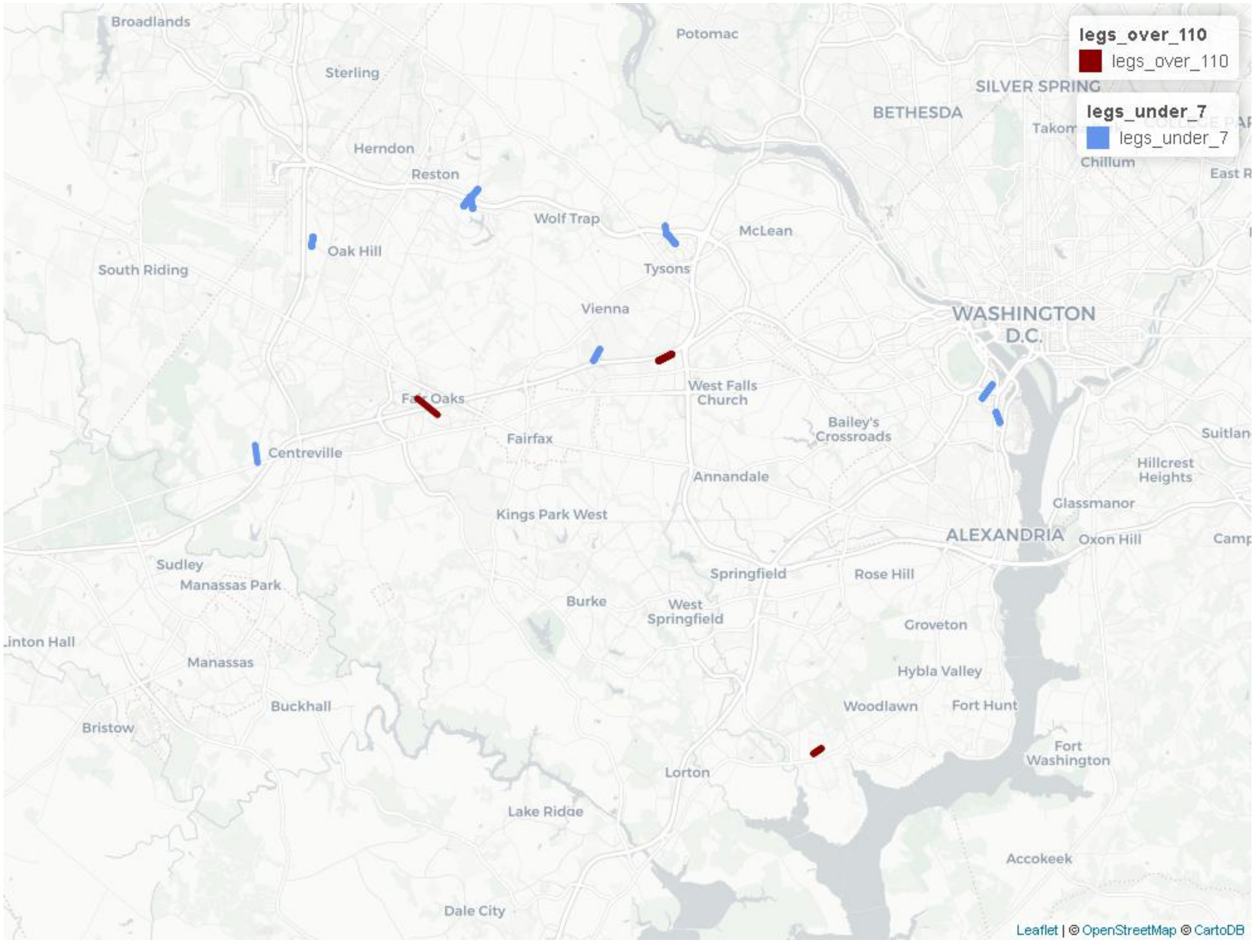


from_stop_name	to_stop_name	route_id	route_long_name
Talbot & Tubman (YMCA)	Travers & West End	18	Cambridge North
Target	St. Michaels Village	13	St. Michaels Shuttle
Waddells Corner SB	Dockins Market WB	11	Route 11 - Cambridge - Secretary - East N
Secretary Main St.	Main St. East New Market EB	11	Route 11 - Cambridge - Secretary - East N
Memorial Hospital (Easton)	Lover's Lane SB	8	Route 9 - Cambridge - Trappe - Easton
Walmart EB	Shore Stop	8	Route 9 - Cambridge - Trappe - Easton
Secretary Main St. WB	Walmart	7	Route 8 - Cambridge - Hurlock - Secretar
Shore Stop SB	Walmart	8	Route 9 - Cambridge - Trappe - Easton
Walmart EB	Secretary Main St.	7	Route 8 - Cambridge - Hurlock - Secretar
Grauls Market	Target	13	St. Michaels Shuttle
612/Greyhound	Chesapeake College	1	Route 4 - Rock Hall - Chestertown - Cent
Kramer Center	Rhode's Store NB	1	Route 4 - Rock Hall - Chestertown - Cent
Federalsburg Town Parking WB	Laurel Grove Apartments	4	Route 6 - Denton - Federalsburg - Preston
St. Michaels Village	Choptank Community	13	St. Michaels Shuttle
Kinnamon's Gas Station NB	Greensboro Parking Lot NB	5	Route 7 - Greensboro - Denton - Easton
Food Lion - Easton	Target	15	Saturday Denton - Easton Shuttle
Magnolia Meadows	The Greens	17	Route D Easton
Target	301 Bay St. (Social Services) Easton	13	St. Michaels Shuttle

Table 1 – continued from previous p

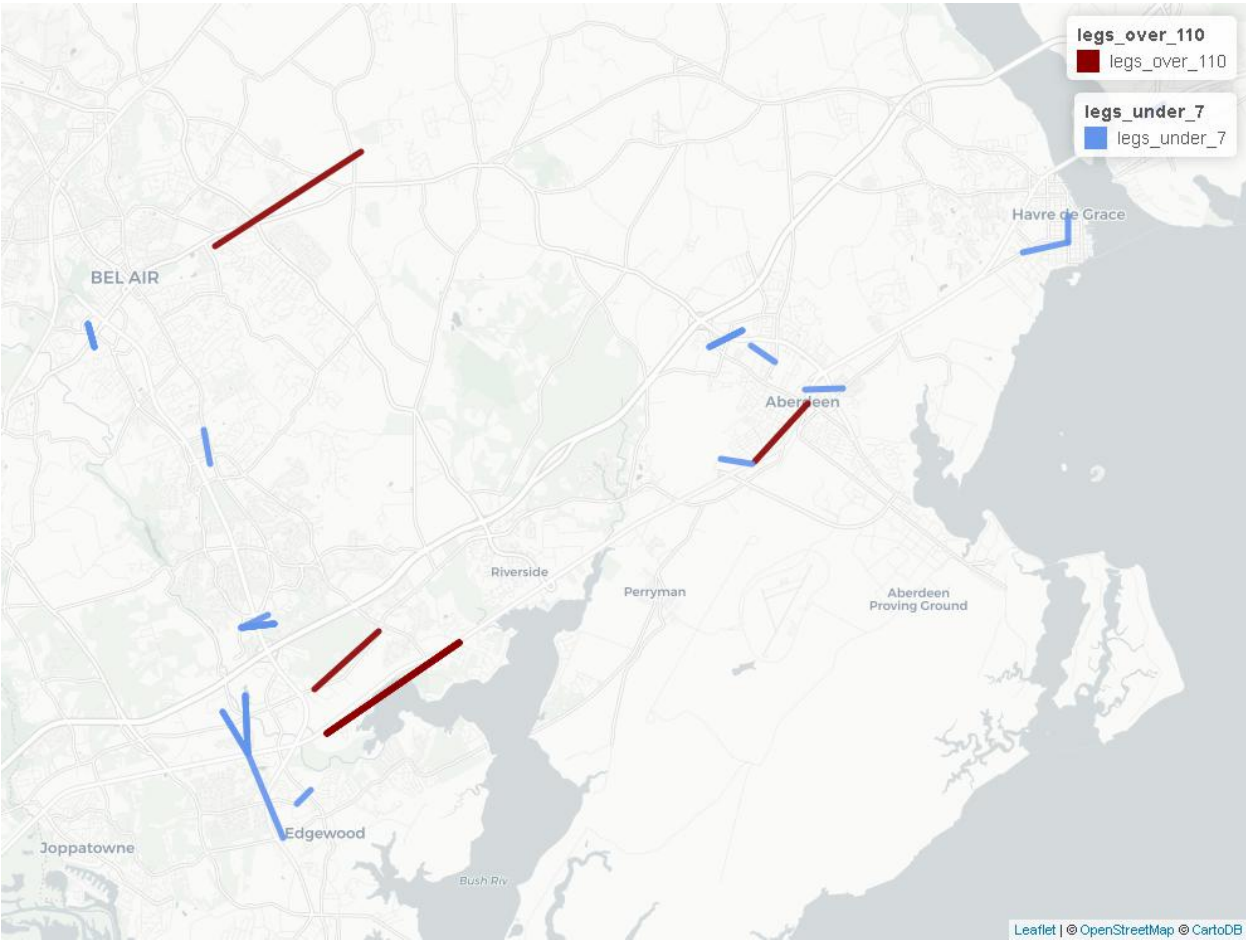
Inn at Perry Cabin	113 Mitchells St.	13	St. Michaels Shuttle
Memorial Hospital (Easton)	Walmart	3	Route 5 - Denton - Easton
Walmart	Memorial Hospital (Easton)	3	Route 5 - Denton - Easton
Senior Center	Doverbrook	17	Route D Easton
301 Bay St. (Social Services) Easton	Post Office EB	13	St. Michaels Shuttle
Easton Post Office	Easton Neighborhood Service Center	14	Denton - Easton AM Shuttle
Goldsborough & Calvert	Memorial Hospital (Easton)	3	Route 5 - Denton - Easton
Rose's Store	Dollar General	1	Route 4 - Rock Hall - Chestertown - Cent
Easton Neighborhood Service Center	Post Office EB	16	Route C Easton
Shore Stop	Trappe Post Office	8	Route 9 - Cambridge - Trappe - Easton
Post Office EB	East Ave. at Goldsborough	16	Route C Easton
Post Office SB	Shore Stop SB	8	Route 9 - Cambridge - Trappe - Easton
Vaughn St. & School St.	Greensboro Parking Lot	5	Route 7 - Greensboro - Denton - Easton
Memorial Hospital (Easton)	Walmart	8	Route 9 - Cambridge - Trappe - Easton
Bob Evans/Panera Bread	612/Greyhound/MVA	14	Denton - Easton AM Shuttle
Goldsborough & Calvert	High & Choptank	1	Route 4 - Rock Hall - Chestertown - Cent
Goldsborough & Calvert	High & Choptank	9	Route 9A - Cambridge - Trappe - Easton
High & Choptank	Goldsborough & Calvert	3	Route 5 - Denton - Easton
301 Bay St. (Social Services) Easton	Target	13	St. Michaels Shuttle
Goldsborough and Calvert EB	High & Choptank WB	13	St. Michaels Shuttle
High & Choptank WB	Goldsborough and Calvert EB	9	Route 9A - Cambridge - Trappe - Easton
HAPS Building	Town Parking Lot	15	Saturday Denton - Easton Shuttle
HAPS Building	Town Parking Lot	3	Route 5 - Denton - Easton
Walmart	612/Greyhound	1	Route 4 - Rock Hall - Chestertown - Cent
High & Choptank	Walmart	1	Route 4 - Rock Hall - Chestertown - Cent
High & Choptank WB	Walmart	13	St. Michaels Shuttle
Federal Manor & East Main	Federalsburg Town Parking WB	4	Route 6 - Denton - Federalsburg - Preston
612/Greyhound/MVA	Airport Industrial Park	14	Denton - Easton AM Shuttle
High & Choptank	Goldsborough & Calvert	13	St. Michaels Shuttle
Travers & West End	Zip Mart	18	Cambridge North
Goldsborough & Calvert	Walmart	13	St. Michaels Shuttle
Goldsborough & Calvert	Walmart	3	Route 5 - Denton - Easton
Walmart	High & Choptank	11	Route 11 - Cambridge - Secretary - East N
Walmart	High & Choptank WB	9	Route 9A - Cambridge - Trappe - Easton
Walmart	High & Choptank	3	Route 5 - Denton - Easton

Fairfax County Connector (VA)



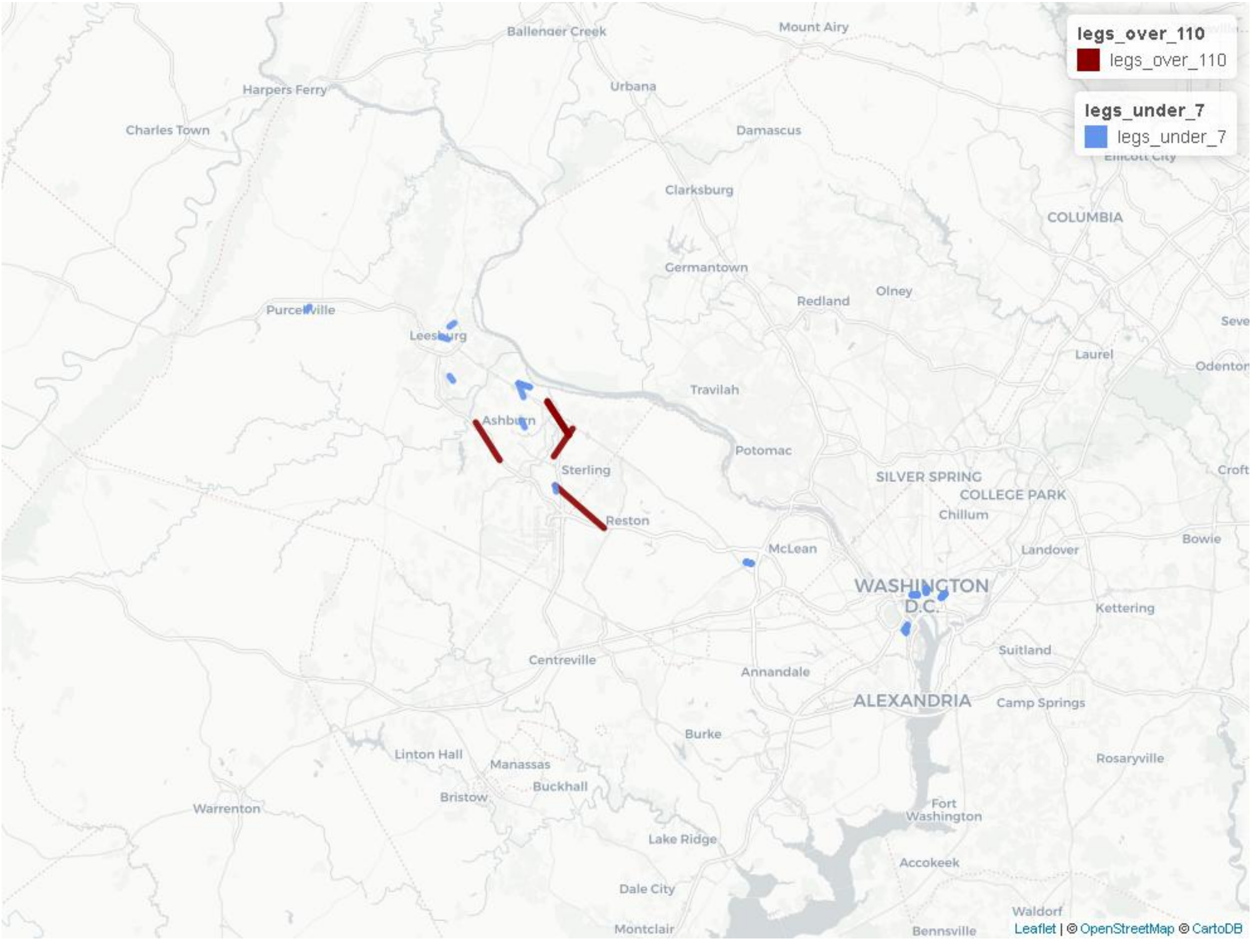
from_stop_name	to_stop_name	route_id	route_long_name	max_kmph	max_distance_miles	number_of_trips
GOETHALS RD @ CONSTITUTION RD	GUNSTON RD @ GOETHALS RD	6651	Fort Belvoir Express	1686.81	0.29	6
2711 PROSPERITY AVE	DUNN LORING METRO	6667	Dunn Loring - Navy Federal - Tysons	1212.95	0.42	30
MONUMENT DR @ FAIRFAX CORNER AVE	MONUMENT DR @ WEST OX	6689	Fairfax Govt Center	114.99	0.73	2
SUNRISE VLY @ CAMPUS COMMONS DR W	WIEHLE METRO BAY J	6675	Reston South - Glade - South Lakes	6.98	0.33	26
WIEHLE METRO BAY C	WIEHLE AVE @ ROGER BACON DR	6718	Hunters Woods - Lake Anne	6.95	0.3	16
PARK CENTER RD @ TOWERVIEW RD	EDS DR @ MCLEAREN RD	6707	Centreville Rd	6.72	0.27	34
SUNRISE VALLEY DR @ WETHERSFIELD CT	WIEHLE METRO	6680	Reston South - Soapstone	6.24	0.31	22
WYCOMBE ST @ REGENTS PARK RD	CENTREVILLE P+R	6697	Sully Station	6.19	0.52	2
PENTAGON CITY METRO	METRO PENTAGON Bay L7	6685	PENTAGON EXPRESS	6.02	0.5	7
SUNRISE VALLEY DR @ WETHERSFIELD CT	WIEHLE METRO BAY G	6710	Sunrise Valley	5.97	0.32	14
SPRING HILL RD @ TURNING LEAF LA	TYSONS WESTPARK	6703	Lewinsville Road	5.96	0.26	5
18TH ST @ CRYSTAL CITY METRO	23RD ST @ CRYSTAL DRIVE	6685	PENTAGON EXPRESS	5.81	0.3	13
NUTLEY ST @ KINGSLEY RD	VIENNA METRO NORTH BAY F	6668	Maple Avenue - Tysons	5.64	0.37	8
WESTPARK DR @ PARK RUN DR	JONES BRANCH DR @ PARK RUN DR	6663	Central Tysons - Tysons Corner Metr	4.49	0.35	26

Hartford TransitLink



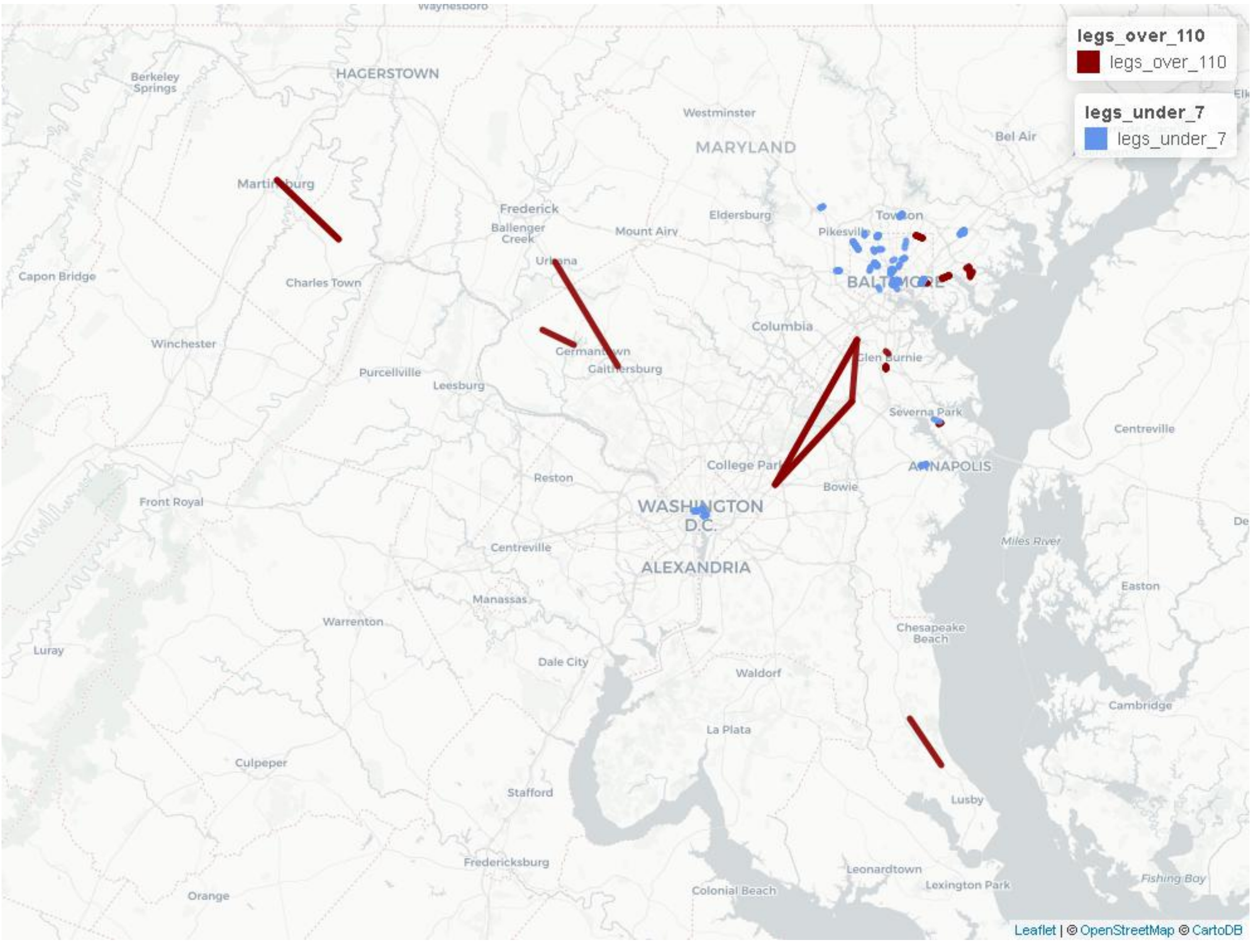
from_stop_name	to_stop_name	route_id	route_long_name	max_kmph	max_distance_miles	number_of_trips
Aberdeen Train Station	McDonald's (across from Walmart on US 40)	4	Yellow	132.48	1.14	1
Harford Community College - Fallston Hall Westbound	Greenbrier Shopping Plaza/Safeway Westbound	1	Green	122.51	2.54	1
William Paca Industrial Park - Eastbound	Cokesbury Rd at Rte. 7-McComus Funeral Home-Eastbound	6	Purple	121.08	1.25	1
Home Depot - Edgewood - Eastbound	Long Bar Harbor Road at US 40 - Eastbound	6A	Light Purple	112.45	2.33	7
Winters Run Industrial Park	Rte. 40 & Paul Martin Dr - Walgreens	2	Blue	6.85	0.71	1
Target - Beards Hill	Aberdeen High School	4	Yellow	6.78	0.42	1
Woodsdale Apartments	Walmart - Abingdon	2	Blue	6.47	0.47	3
Woodsdale Apartments	Walmart - Abingdon	2A	Light Blue	6.47	0.47	3
Aberdeen Shopping Plaza/Ollie's	Wage Connection/Klein's Beards Hill Shopping Ctr	4	Yellow	6.46	0.54	3
Saks & Frito Lay	Walmart - Aberdeen - Eastbound	6A	Light Purple	6.4	0.46	1
Kohl's Distribution Center - Northbound	Rte. 40 & Paul Martin Dr - Walgreens	5	Red	6.39	1.32	1
Fire Station No. 2 - Southbound	Harford Memorial Hospital	7	Teal	6.15	0.38	1
Walmart - Abingdon	WaWa Food Market - Abingdon	2A	Light Blue	5.83	0.42	1
Food Lion - Perryville	White Horse Apartments - Perryville	7	Teal	5.45	0.34	1
Aberdeen Senior Activity Center - Westbound	Affinity Post Apts - Eastbound	4	Yellow	5.38	0.56	1
Hanson Road at Fern Street	Edgewood Shopping Plaza - Southbound	5	Red	5.24	0.27	1
Revolution at Seneca Ave.	Harford Memorial Hospital	7	Teal	5.09	0.69	1
Lakeside Business Park	Rte. 40 & Paul Martin Dr - Walgreens	2	Blue	4.8	0.84	2
Lakeside Business Park	Rte. 40 & Paul Martin Dr - Walgreens	2A	Light Blue	4.8	0.84	1
Harford Mall (Sears)	Harford Senior Housing	3	Orange	4.45	0.32	5
Winters Run Industrial Park	Rte. 40 & Paul Martin Dr - Walgreens	5	Red	4.28	0.71	1
Lorien at Bel Air	Festival at Bel Air - Shoprite	2A	Light Blue	3.5	0.51	1

Loudon Transit (VA)



from_stop_name	to_stop_name	route_id	route_long_name	max_kmh	max_distance_miles	number_of_trips
George Washington UniversityExploration Hall	Dulles Town Center Macy's and Lord & Taylor upper level outside	70E	7 to 7 on 7 Eastbound	203.41	2.35	1
Pacific Blvd @ Auto World Circle	Lerner Building @ Dulles Center Blvd	84X	Atlantic Pacific Connector	189.03	1.96	1
Dulles Town Center Macy's and Lord & Taylor upper level outside	George Washington Blvd @ George Washington University Exploration Hall	70W	7 to 7 on 7 Westbound	127.49	2.31	2
Broadlands 772 Lot	Goose Creek Village 20785 Century Corner Dr, Ashburn	SMC	Silver Metro Connection	126.71	2.62	1
Pacific Blvd @ Prologis Plaza	Herndon/Monroe Park & Ride lot	84X	Atlantic Pacific Connector	124.2	3.86	1
Army/Navy Drive & Fern Street	Pentagon Transit Station Bus Bay L5	DCC	DC Commuter- Dulles South, Dulles North, CFC, Ashburn North & Brambleton to The Rosslyn, Pentagon, Crystal City & DC	6.9	0.36	1
State Dept (21st & Virginia Ave Metro Bus Stop)	8th & E Streets, NW (Metro Bus Stop)	DCC	DC Commuter- Dulles South, Dulles North, CFC, Ashburn North & Brambleton to The Rosslyn, Pentagon, Crystal City & DC	6.85	0.28	26
State Dept (21st & Virginia Ave Metro Bus Stop)	8th & E Streets, NW (Metro Bus Stop)	PDCC	DC Commuter - Purcellville, Harmony & Leesburg to Rosslyn, Pentagon, Crystal City & Washington DC	6.85	0.28	15
Maple Ave Apartments	16th St Bus Shelter	40	Pucellville Connector	6.82	0.28	1
Leesburg Loudoun County Government, 1 Harrison St SE, Leesburg	Leesburg Plaza (NOVA Urgent Care)	55	Route 55	6.71	0.42	1
INOVA Loudoun Hospital (Main Entrance)	HHMI Janelia Farm Research Campus	WE_PM	Weihle Express Afternoon Schedule	6.59	0.75	1
14th & F Streets, NW (National Press Bldg) PM	I & 15th Sts., NW (The McPherson Bldg)	DCC	DC Commuter- Dulles South, Dulles North, CFC, Ashburn North & Brambleton to The Rosslyn, Pentagon, Crystal City & DC	6.54	0.27	20
14th & F Streets, NW (National Press Bldg) PM	I & 15th Sts., NW (The McPherson Bldg)	PDCC	DC Commuter - Purcellville, Harmony & Leesburg to Rosslyn, Pentagon, Crystal City & Washington DC	6.54	0.27	11
County Complex/County Garage	Loudoun Interfaith Relief/Miller Dr	56	Route 56	6.52	0.34	1
Eads Street & 12th Street PM	Pentagon Transit Center Bus Bay L5 PM	DCC	DC Commuter- Dulles South, Dulles North, CFC, Ashburn North & Brambleton to The Rosslyn, Pentagon, Crystal City & DC	6.41	0.46	2
Eads Street & 12th Street PM	Pentagon Transit Center Bus Bay L5 PM	PDCC	DC Commuter - Purcellville, Harmony & Leesburg to Rosslyn, Pentagon, Crystal City & Washington DC	6.41	0.46	2
Pacific Blvd @ Asurion	Pacific Blvd @ Prologis Plaza	84X	Atlantic Pacific Connector	5.84	0.38	1

MTA

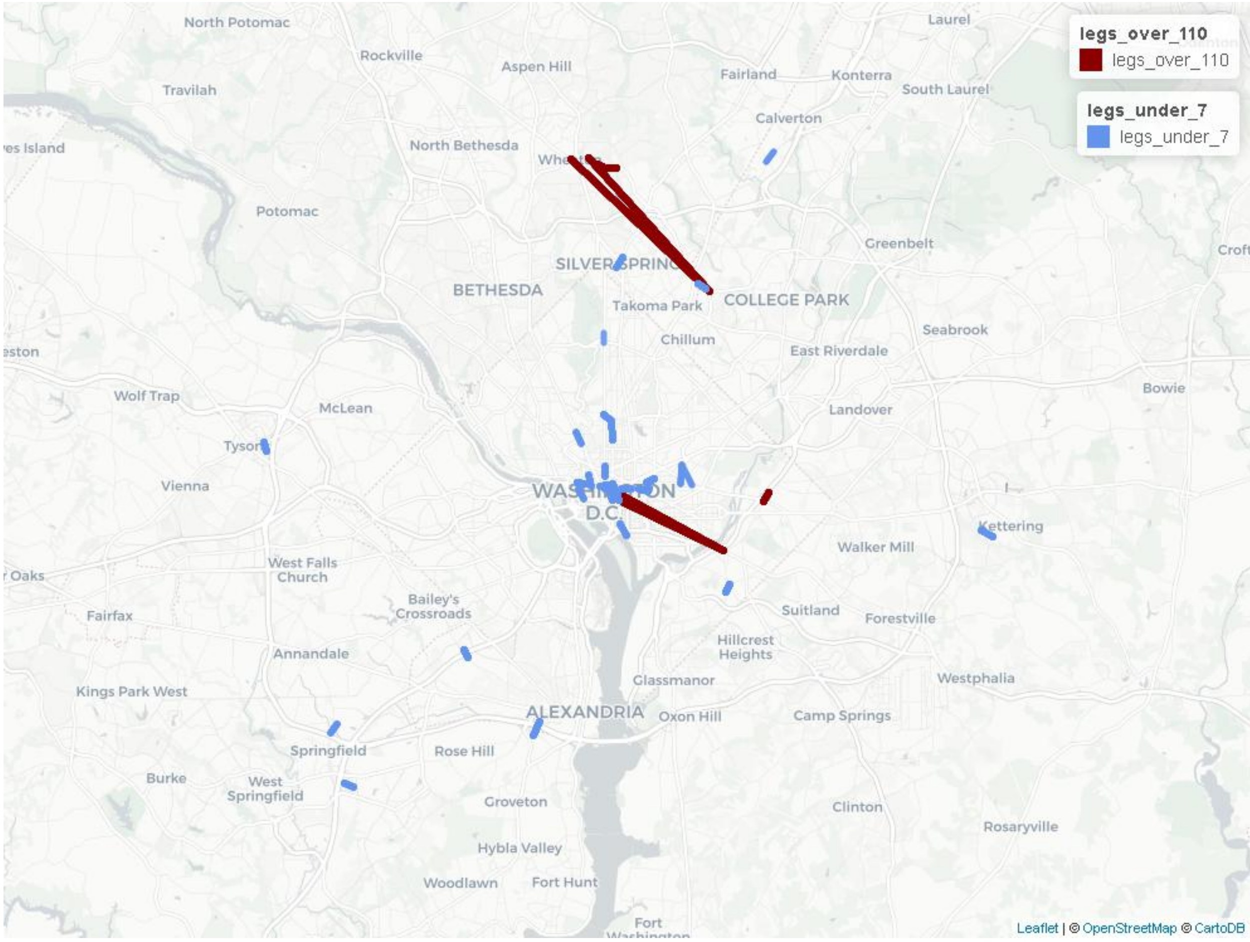


from_stop_name	to_stop_name
HARFORD RD & NORTHERN PKY sb	NORTHERN PKWY & MCCLEAN BLVD
NORTHERN PKWY & MCCLEAN BLVD fs eb	NORTHERN PKWY & HAMPNETT AVE
WEST CAMPUS DR & FINE ARTS BUILDING eb	COLLEGE DR & RING RD EAST nb fs
BACK RIVER NECK RD & LANFLAIR RD sb	MIDDLEBOROUGH RD & BRIGHTON M
BAYVIEW BLVD & HOPKINS-BAYVIEW DR sb	EASTERN AVE & ELRINO ST eb
HOSPITAL DR & CRAIN HWY wb	CRAIN HWY & OAK MANOR DR nb
MIDDLEBOROUGH RD & EDDYSTONE PL fs sb	STEMMERS RUN RD & OLD EASTERN
OLD EASTERN AVE & ESSEX AVE eb	BACK RIVER NECK RD & EVERGREEN
DUFFIELFS W VA MARC wb	MARTINSBURG W VA MARC
EASTERN AVE & DIAMOND POINT RD wb	EASTERN AVE & ROLLING MILL RD wb
GAITHERSBURG PARK & RIDE	URBANA & PARK & RIDE LOT
BOYDS MARC wb	BARNESVILLE MARC wb
PRINCE FREDERICK P&R (FAIRGROUND & ARMORY RD)	ST LEONARD CHURCH MD 2 & BALL R
BWI RAIL STATION MARC sb	ODENTON MARC sb
ODENTON MARC sb	NEW CARROLLTON MARC sb
CROMWELL STATION LOOP	BALTIMORE ANNAPOLIS BLVD & MAP
BWI RAIL STATION MARC sb	NEW CARROLLTON MARC sb
COVINGTON ST & CROSS ST mid sb	SARATOGA ST & HOWARD ST wb

Table 2 –

KIRK AVE & 25TH ST sb	KIRK AVE & BARTLETT ST
EASTERN AVE & PONCA ST eb	LOMBARD ST & BIOSCIENCE DR fs wb
TOWANDA AVE & ANOKA AVE opp sb	MONDAWMIN STATION
CHARLES ST & 31ST ST nb	33RD ST & ABELL AVE eb
DOLPHIN ST & HOWARD ST fs wb	HOWARD ST & NORTH AVE nb
JONES STATION RD & SEVERNA PARK PARK & RIDE eb	COLLEGE PKY & ANNE ARUNDEL COM
PENNSYLVANIA AVE & 10TH ST NW wb	12TH ST & G ST NW nb
PENNSYLVANIA AVE & 10TH ST NW wb	12TH ST & G ST NW nb
PENNSYLVANIA AVE & 10TH ST NW wb	12TH ST & G ST NW nb
PENNSYLVANIA AVE & 10TH ST NW wb	12TH ST & G ST NW nb
K ST & 14TH ST NW wb	15TH ST & M & MASSACHUSETTS AVE
K ST & 14TH ST NW wb	15TH ST & M & MASSACHUSETTS AVE
REISTERSTOWN RD & REISTERSTOWN PLAZA ENTRANCE	ROGERS AVE METRO STATION BAY 6
RIVA RD & FOREST DR opp wb	HARRY S TRUMAN P & R
RIVA RD & FOREST DR opp wb	HARRY S TRUMAN P & R
PRATT ST & LIGHT ST eb	GAY ST & FAYETTE ST nb
FAYETTE ST & FRONT ST wb	SARATOGA ST & GUILFORD AVE wb
YORK RD & CHESAPEAKE AVE nb	TOWSON TOWN CENTER BAY 1
GUILFORD AVE & SARATOGA ST fs sb	BALTIMORE ST & PRESIDENT ST fs eb
FAYETTE ST & SAINT PAUL ST wb	HOPKINS PL & BALTIMORE ARENA mb
SANDPIPER CIR & WHITE MARSH HEALTH CENTER wb	WHITE MARSH PARK & RIDE
PARALLEL DR & SECURITY WEST BUILDING wb	WOODLAWN DR & PARALLEL DR fs nb
LOMBARD ST & CHARLES ST mb wb	HOWARD ST & CAMDEN ST sb
LOMBARD ST & CHARLES ST mb wb	HOWARD ST & CAMDEN ST sb
11TH ST & E ST NW sb	14TH ST & CONSTITUTION AVE NW sb
GREENSPRING AVE & COLD SPRING LN nb	DUPONT LOOP SCHOOL
23RD ST & I ST NW nb	K ST & 20TH ST NW eb
23RD ST & I ST NW nb	K ST & 20TH ST NW eb
23RD ST & I ST NW nb	K ST & 20TH ST NW eb
FAYETTE ST & GAY ST wb	SAINT PAUL ST & FAYETTE ST fs sb
LOCH RAVEN BLVD & ARLINGTON AVE fs nb	LOCH RAVEN BLVD & KITMORE RD nb
DUKELAND AVE & CONNEXIONS ACADEMY AND BARD HIGH SCHOOL nb	NORTH AVE & LONGWOOD ST eb
LOCH RAVEN BLVD & WOODBOURNE AVE nb	LOCH RAVEN BLVD & BELVEDERE AV
SAINT PAUL PL & SARATOGA ST sb	CHARLES ST & PRATT ST nb
WILKENS AVE & PAYSON ST eb	MONROE ST & MONTGOMERY PARK fs
35TH ST & HILLEN RD	33RD ST & LOCH RAVEN BLVD wb
KELLY AVE & SOUTH RD eb	FALLS RD & BELLEMORE RD nb
PENNSYLVANIA AVE & 10TH ST NW wb	12TH ST & G ST NW nb
PENNSYLVANIA AVE & 10TH ST NW wb	12TH ST & G ST NW nb
NORTH AVE LT RAIL sb	PENN STATION LIGHT RAIL
35TH ST & HILLEN RD	THE ALAMEDA & 33RD ST sb
8221 TOWN CENTER DR opp wb	WHITE MARSH PARK AND RIDE BAY 1
TORAH INSTITUTE	REISTERSTOWN RD & PAINTERS MILL
COLD SPRING LN & LIGHT RAIL eb	POLY WESTERN HIGH SCHOOL
WILLOW GLEN DR & PIMLICO RD fs wb	GREENSPRING AVE & TANEY RD eb

WMATA (MD, DC, VA)



from_stop_name	to_stop_name
PENNSYLVANIA AVE SE & LENFANT	F ST & 11TH ST
PENNSYLVANIA AVE & 10TH ST NW	PENNSYLVANIA AVE SE & LENFANT
UNIVERSITY BLVD & INWOOD AVE	UNIVERSITY BLVD & SLIGO CREEK PKWY
UNIVERSITY BLVD & INWOOD AVE	UNIVERSITY BLVD & SLIGO CREEK PKWY
MINNESOTA AVE STATION & BUS BAY B	MINNESOTA AVE NE & BENNING RD NE
UNIVERSITY BLVD & EASECREST DR	UNIVERSITY BLVD & INWOOD AVE
UNIVERSITY BLVD & EASECREST DR	UNIVERSITY BLVD & INWOOD AVE
UNIVERSITY BLVD & RIGGS RD	UNIVERSITY BLVD W & WESTCHESTER DR
UNIVERSITY BLVD & RIGGS RD	UNIVERSITY BLVD W & WESTCHESTER DR
VEIRS MILL RD & UNIVERSITY BLVD	UNIVERSITY BLVD & RIGGS RD
VEIRS MILL RD & UNIVERSITY BLVD	UNIVERSITY BLVD & RIGGS RD
PENNSYLVANIA AVE NW & 12TH ST NW	I ST & 15TH ST NW
ALABAMA AVE SE & AINGER PL SE	NAYLOR RD SE & GOOD HOPE RD SE
ALABAMA AVE SE & AINGER PL SE	NAYLOR RD SE & GOOD HOPE RD SE
ALABAMA AVE SE & AINGER PL SE	NAYLOR RD SE & GOOD HOPE RD SE
H ST NW & 4TH ST NW	H ST & 1ST ST NW
15TH ST NW & NEW YORK AVE NW	PENNSYLVANIA AVE NW & 14TH ST NW
15TH ST NW & NEW YORK AVE NW	PENNSYLVANIA AVE NW & 14TH ST NW

Table 3 – continued from

15TH ST NW & NEW YORK AVE NW	PENNSYLVANIA AVE NW & 14TH ST NW
15TH ST NW & NEW YORK AVE NW	PENNSYLVANIA AVE NW & 14TH ST NW
15TH ST NW & NEW YORK AVE NW	PENNSYLVANIA AVE NW & 14TH ST NW
FRANKLIN SQUARE & BUS BAY B	16TH ST & I ST
H ST NW & 7TH ST NW	H ST NW & 11TH ST NW
EISENHOWER AVE & SWAMP FOX RD	HUNTINGTON STATION (N) & BUS BAY B
PENNSYLVANIA AVE NW & 24TH ST NW	K ST NW & 21ST ST NW
SEMINARY RD & KENMORE AVE	SOUTHERN TOWERS RD & STRATFORD BLDG
LOISDALE CT & #6564	SPRINGFIELD MALL ROADWAY & BETWEEN FOODFEST & PA
16TH ST & M ST	16TH ST & P ST
16TH ST & M ST	16TH ST & P ST
H ST & MADISON PL NW	PENNSYLVANIA AVE NW & 13TH ST NW
MASSACHUSETTS AVE NW & NEW JERSEY AVE NW	H ST NW & 5TH ST NW
MT LUBENTIA WAY & MT LUBENTIA CT	HARRY S TRUMAN DR & TRUMAN MANOR #600-70
14TH ST NW & IRVING ST NW	PARK RD & 16TH ST
13TH ST NW & G ST NW	FRANKLIN SQUARE & BUS BAY D
PENNSYLVANIA AVE NW & 12TH ST NW	I ST & 15TH ST NW
16TH ST & SHERIDAN ST	16TH ST & MISSOURI AVE
23RD ST NW & I ST NW	VIRGINIA AVE & E ST
I ST & 15TH ST NW	I ST & 17TH ST NW
NORTH CAPITOL ST NW & PIERCE ST NW	K ST NW & NEW JERSEY AVE NW
UNIVERSITY BLVD & 14TH AVE	UNIVERSITY BLVD & 15TH AVE
UNIVERSITY BLVD & 14TH AVE	UNIVERSITY BLVD & 15TH AVE
H ST NW & 16TH ST X	H ST NW & 14TH ST NW
CONNECTICUT AVE & BELMONT RD	CONNECTICUT AVE & CALVERT ST
14TH ST & BELMONT ST	14TH ST NW & IRVING ST NW
TYSONS CORNER STATION & BUS BAY G	TYSONS CORNER SHOPPING CENTER & PARKING TERRACE C
MAINE AVE + 9TH STREET	12TH ST SW & D ST SW
I ST NW & 19TH ST NW	20TH ST NW & M ST NW
COLESVILLE RD & SPRING ST	SILVER SPRING STATION & BUS BAY 114
COLESVILLE RD & SPRING ST	SILVER SPRING STATION & BUS BAY 113
BRENTWOOD PKWY NE & PENN ST NE	FLORIDA AVE NE & 13TH ST NE
BACKLICK RD & HECHINGER DR	COMMERCIAL DR & INDUSTRIAL RD
POWDER MILL RD & HIGH POINT HIGH SCHOOL	POWDER MILL RD & CHERRY HILL RD
I ST & 15TH ST NW	I ST & 17TH ST NW
K ST NW & 14TH ST NW	13TH ST NW & G ST NW
BRENTWOOD PKWY NE & PENN ST NE	8TH ST & K ST

Note: The stop-to-stop speed tool discussed in this section can be accessed here. R and R Studio are required to run the script.

4.2 Routing problems

To confirm the project networks are properly configured, a network routing test should be conducted on both the base and project networks to identify the shortest path between two points. For these tests, beginning and ending points are identified near the termini of the project. The purpose of this test is to confirm that the network is utilizing the new path created by the project. If the build and no-build networks are configured properly, the build scenario route should traverse the proposed network segment. This test may also show a travel time reduction. Additional detail on Route analysis in Arcmap and Network analyst is available here. Recommended settings include:

- A 8:00 AM departure time
- Use the travel time evaluator create during network development
- Route endpoints should be near the project termini

The below example from the 2018 scoring illustrates the results of this test. The top map shows a base case route between Montgomery College and Wheaton Metro Station and the bottom map shows the route with BRT project in place. Additional detail on this project can be found here.

The below example from the 2018 scoring illustrates the results of this test. The top map shows a base case route between Montgomery College and Wheaton Metro Station and the bottom map shows the route with BRT project in place. Additional detail on this project can be found here [ADD LINK TO PROJECT 32 PDF HERE].

No-Build Route Montgomery College to Wheaton Metro Station at 8:00 am

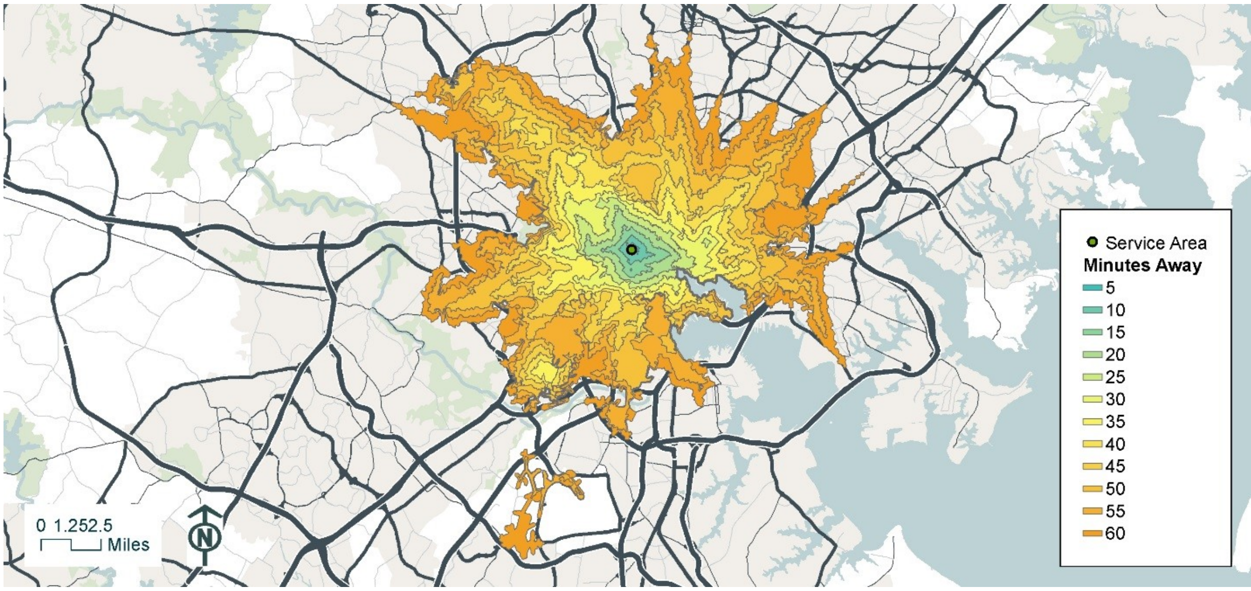
Build Route Montgomery College to Wheaton Metro Station at 8:00 am

4.3 Service area problems (travel time contours)

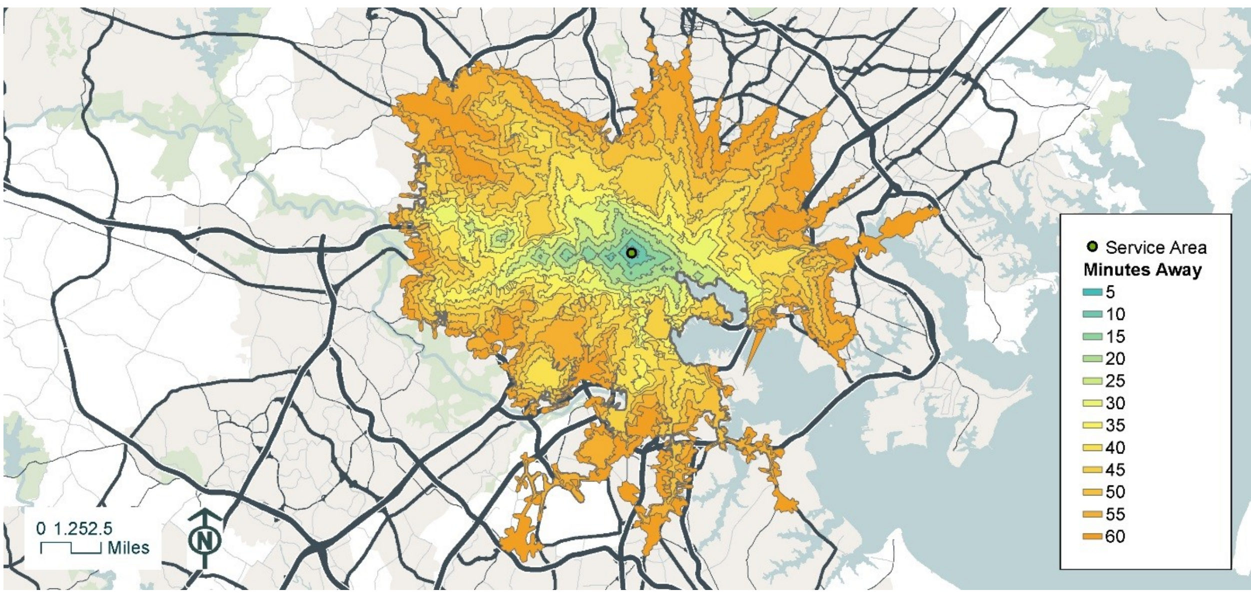
A service area analysis should be conducted under build and no-build scenarios for each project. The result of this analysis is a map of travel time isochrones for a single location within the project study area. A simple check that the isochrones expand as the project provides additional service and that the expansion is intuitive given the nature of the project is sufficient to confirm that the project is appropriately integrated into the base multimodal network. A Service Area test in Network Analyst, which is documented here, is used to conduct this test. The following settings are recommended:

- Breaks in 5 minute increments between 5 and 60 minutes
- Departure time on typical Wednesday at 8:00 AM
- Direction should be away from the facility
- Use the travel time evaluator create during network development
- Start points should be along the project

The below example from the 2018 scoring illustrates the results of this test. The top map shows a base case service area from N. Carey St and W. Franklin St. and the bottom map shows the same service area with project in place. Additional detail on this project can be found here.



No-Build Service Area from North Carey St @ West Franklin St. at 8:00 am



Build Service Area from North Carey St @ West Franklin St. at 8:00 am

Python Module Index

Index