

— **THE EASTERN** —
TRANSPORTATION
COALITION

CONNECTING FOR SOLUTIONS



CAPABLE II: Count All Pedestrian And Bicycle Locations Efficiently (Phase II)

Development of the Active Transportation Count Specification (ATCS)

The ATCS is a conceptual model and data exchange format to standardize bicycle, pedestrian, and other non-motorized traffic counts in The Eastern Transportation Coalition.

February 26, 2026

Executive Summary

This report presents a data specification developed as part of the CAPABLE II (Count All Pedestrian and Bicycle Locations Efficiently, Phase II) SCOOP initiative – a cooperative effort funded by states in The Eastern Transportation Coalition (TETC) that focused on bike and pedestrian (“bike/ped”) data. This specification, called the Active Transportation Count Specification (ATCS), is a data exchange format for standardizing bicycle, pedestrian, and other non-motorized traffic counts within the Coalition.

The Coalition’s CAPABLE Phase I SCOOP initiative ([link](#)), completed in 2025, characterized the state of bike/ped data across the Coalition and developed recommendations about how to improve its availability, quality, and cost. Based on committee feedback and additional stakeholder conversations, the report recommended the following steps for future Coalition involvement:

- **Establish an ongoing Coalition bike/ped committee** to ensure that data formats, specifications, and future procurement of bike/ped data in the TDM align with Coalition member needs.
- **Create a consistent data schema for bike/ped count data**, to be developed in coordination with practitioners and enforced through the TETC Transportation Data Marketplace procurement.
- **Establish Coalition resources to support agencies with bike/ped data collection and analysis.** This recommendation addresses the fact that bike/ped data collection is often (though not always) undertaken at the local level, and many small agencies/municipalities do not have internal resources or prior experience “wrangling” the data to common formats.

The CAPABLE Phase II initiative, which commenced in December 2024 and concluded in December 2025, focused primarily on developing the bike/ped data schema. With the guidance of an advisory committee consisting of both bike/ped data experts from within Coalition member states and subject matter experts across the United States, the project team developed the Active Transportation Count Specification (ATCS). The ATCS is intended to unambiguously describe bike and pedestrian counts and provide a standard data exchange format to support exchange of bicycle and pedestrian count data within the Coalition. Any vendors supplying bike/ped counts will be required to support the ATCS in the next iteration of the Transportation Data Marketplace that is anticipated to begin in 2027.

The remainder of this report contains Version 1.0 of the ATCS. It provides background and motivation for developing the specification, introduces the conceptual model and concepts used throughout, and finally describes the file formats and data fields required to implement it. The final section includes a set of contextual examples that thoroughly exhibit how the ATCS can be implemented in real-world counting scenarios. These examples are memorialized in a set of downloadable ATCS formatted files.

Acknowledgements

The project team would like to thank the following members of the CAPABLE II advisory committee for providing guidance and feedback throughout the project.

TETC Member State Representatives

- Bob Frey (Massachusetts DOT)
- Natalie Levine (Massachusetts DOT / MBTA)
- Sam Harris (Georgia DOT)
- Ron Knezevich (Georgia DOT)
- Will Rogers (Greater Nashville Regional Council)
- Amanda Holland (Vermont Agency of Transportation)
- Todd Sears (Vermont Agency of Transportation)
- David Cook (Virginia DOT)
- Mason Perrone (Rhode Island Department of Administration)
- Michael Stafford (North Carolina DOT)
- Christopher Roberts (Connecticut DOT)
- Eric Courchesne (Alabama Transportation Institute)

External Representatives

- Liz Stolz (SRF Consulting)
- Josh Roll (Oregon DOT)
- Shawn Turner (Texas A&M Transportation Institute)
- Phil Lasley (Texas A&M Transportation Institute)
- Andy Duvall (National Laboratory of the Rockies)

— THE EASTERN
TRANSPORTATION
COALITION

CONNECTING FOR SOLUTIONS



Active Transportation Count Specification (ATCS)

*Conceptual model and data exchange format to standardize bicycle,
pedestrian, and other non-motorized traffic counts in TETC.*

Version 1.0

Version History

Version 0.1: Initial Draft for internal review

Date: 2025-12-23

Changes: N/A

Version 0.2: for CAPABLE committee review

Date: 2026-01-21

Changes:

- Reformatted document and section headings
- Clarified explanations and examples
- Added Metadata file to ATCS package file structure

Version 1.0: Initial release at completion of CAPABLE II

Date: 2026-02-28

Changes:

- Incorporated minor edits from CAPABLE advisory committee

Table of Contents

1. Introduction and Background	1
2. Conceptual Framework	1
2.1 Description of Core Entities	1
2.2 Relationships between Entities	2
3. Data Representation	1
3.1 General Conventions	1
3.2 Site Entity	2
3.3 Flow Entity	6
3.4 Counter Entity	11
3.5 Deployment Entity	12
3.6 Count Record Entity	13
4. Package Specification	15
4.1 File Packaging	15
4.2 Use Cases and Scope	17
4.3 Adaptable Workflows	17
4.4 Database and Analytics Compatibility	17
5. Schema Reference	18
5.1 metadata.json	18
5.2 sites.geojson	19
5.3 flows.geojson	21
5.4 counters.csv	25
5.5 deployments.geojson	26
5.6 count_records.csv	27
6. Examples	28
6.1 Example Path Segment	29
6.2 Example Road Segment	31
6.3 Example Path Intersection	32
6.4 Example Road Intersection	34
6.5 Example Hybrid Intersection	35
6.6 Example Complex	37
Appendix A: Enumerated Values	38

1. Introduction and Background

Transportation agencies across The Eastern Transportation Coalition (or *Coalition*) have shown interest in quantifying bicycling and pedestrian activity to support diverse use cases, including determining where to plan and design bike/pedestrian infrastructure, assessing project impacts, contextualizing crash statistics, and identifying economic opportunities for local investment. To support these and other applications, agencies typically collect bike/ped volumes using a combination of permanently installed continuous counters and short-term counters that can be deployed to different locations.

An Eastern Transportation Coalition initiative called [CAPABLE](#) (**C**ount **A**ll **P**edestrian **A**nd **B**icycle **L**ocations **E**fficiently) characterized the state of bike/ped data across the Coalition and developed recommendations about how to improve its availability, quality, and cost. Initially, the project focused on crowdsourced bike/ped estimates, but conversations with data vendors quickly revealed that a key limiting factor is the lack of widespread ‘ground truth’ count datasets to which models can be calibrated. It became clear that agency count programs serve multiple purposes; they provide critical data for local planning and safety applications, and they additionally serve as a source of ‘ground truth’ for bike/ped activity to which crowdsourced volume estimates can be calibrated and independently validated. Thus, supporting Coalition agencies with bike/ped counting is expected to provide significant value to the entire bike/ped data ecosystem.

A key challenge, however, is that, although count programs do exist across Coalition states, data tends to be collected at the local level and siloed across organizations without consistent data formats or QA/QC processes. Unlike vehicle data, which has federal mandates for reporting vehicle volume counts as part of the Highway Performance Monitoring System (HPMS), bike/ped data have no similar federal reporting requirements, but rather is at the discretion of the state, city, or metropolitan planning organization to collect and use such data in their planning and operations. Without any federal reporting requirement, no pressure or expectation to evolve to a common reporting format has emerged. Consistent data schemas for bike/ped count data collection and aggregation would provide significant value to the entire bike/ped ecosystem, enabling efficient data collection and sharing as well as enabling more effective crowdsourced products.

As the Coalition’s Transportation Data Marketplace (TDM) looks forward to its next procurement for transportation datasets, there is an opportunity to establish a consistent and robust data format for non-motorized traffic count data into the TDM request for proposal (RFP) specifications. The intent of this bike/ped data standard, the aspects of which will be described below, is to provide a common language within the Coalition TDM to describe bike and pedestrian counts. This is critical for ensuring that all procured bike/ped counts result in consistently formatted data. This also enables agencies to more easily switch between vendors without worrying about proprietary vendor data formats, which creates interoperability. It also allows Coalition members to more easily combine data collected by different agencies. This latter capability is expected to form the basis for a Coalition-level database used for calibration (and validation of) third party, crowd-sourced data vendors for non-motorized count, and be easily importable to different commercial data management systems.

The Active Transportation Count Specification (ATCS) is inspired by several other data schemas that predate this effort, including FHWA’s Traffic Monitoring Guide (TMG), BikePed Portal (managed by Portland State University), the Texas Bike and Pedestrian Count Exchange (BP|CX, developed and managed by Texas A&M Transportation Institute), and MS2 (Midwestern Software Solutions). The ATCS builds upon – and seeks to retain compatibility with, where possible – many of these other schemas, with a particular focus on (i) data exchange, rather than long-term

storage, and (ii) the ability to capture a variety of counting movements that are expected to become more common at intersections, including turning movements and crossing counts at crosswalks.

The remainder of this document is organized as follows:

- **Section 2** presents the conceptual framework, introducing the five core entities that form the foundation of the schema.
- **Section 3** explains the data representation, including defining each entity, its attributes, and the validation logic that governs the relationships between entities.
- **Section 4** is the packaging specification, explaining how the ATCS files are organized into a practical file structure.
- **Section 5** is the official schema reference, showing the exact field names and data types required for implementation.
- **Section 6** walks through a series of examples for common and complex real-world bike/ped counting scenarios.
- **Appendix A** provides a glossary of key terms for reference.

2. Conceptual Framework

The Active Transportation Count Specification (ATCS) is a relational data model designed to standardize the collection, storage, and exchange of bike, pedestrian, and other micromobility count data. The model is built on five core entities that separate the *where*, *what*, and *how* of a counting event, creating a flexible and scalable system that is well-suited for storage in a relational database.

At its core, ATCS describes *locations* where counts occur, *movements* that are counted, *devices or observers* that collect data, *time periods* when they are active, and the resulting *count values*. These concepts are intentionally separated so that the same location or movement definition can be reused across multiple data collection efforts.

While this document's primary focus is on defining the functional rules and the file-based data exchange format, it is important to note that the conceptual model itself is intentionally designed as a **relational model**. This structure, with its distinct entities and defined relationships, is well-suited for implementation in a relational database. Such an implementation allows an agency to leverage powerful database tools for data validation, efficient querying, and long-term management, even if their data is received and shared using the standard file-based exchange format as described later in the document.

2.1 Description of Core Entities

Figure 2-1 shows a simplified version of an Entity-Relationship Diagram, which captures the hierarchical relationship between entities.

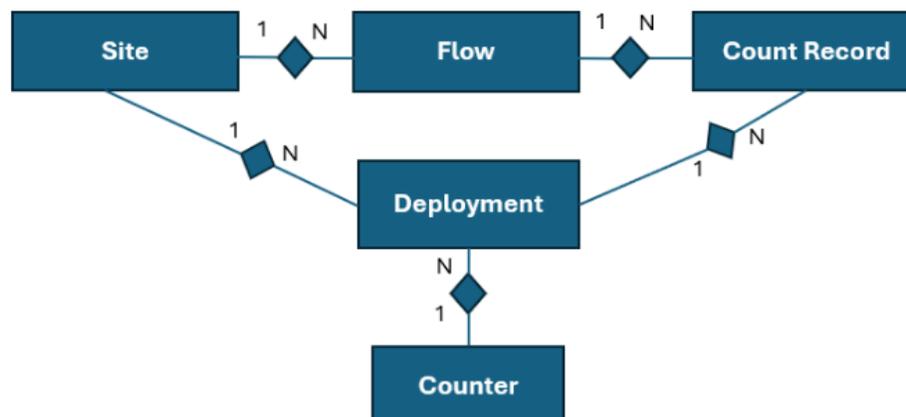


Figure 2-1: Simplified diagram showing core entities and how they relate to one another.

- **Site:** A Site is the top-level geographic location where counts are collected (typically a Segment or Intersection). It defines a counting location's fundamental geometry, its real-world infrastructure environment, and its specific layout.
- **Flow:** A Flow defines a specific movement to be counted within a Site; it does not contain the count data itself. It defines the category of movement (e.g., Screenline, Turning,

Crossing), the travel mode, facility type, and other details specific to the type of movement.

- **Counter:** A Counter describes the hardware or method used to collect the data. This could be a specific model of video camera, an infrared sensor, or a human observer.
- **Deployment:** A Deployment describes "when and where" a specific Counter was active. It links one Counter to a Site for a defined period and captures the specific sensor location, which may differ from the facility's location. This creates a record of a specific data collection effort.
- The **Count Record** contains aggregated counts by date and time. Each record, which captures the counts for a single Flow and time period, links back to the Deployment (which defines "when and where") and the Flow (which defines the specific movement).

2.2 Relationships between Entities

The five entities in the conceptual model are linked by a series of clear, one-to-many relationships that define how count data is organized.

- A **Site** is the parent for both **Flows** and **Deployments**. Each Site can have one or more Flows defined within it, and it can host one or more Deployments over time. This establishes a one-to-many relationship from Site to both Flow and Deployment.
- A **Counter** is linked to a **Deployment**. Each Counter can be used for many Deployments, but each Deployment uses only one Counter, creating a one-to-many relationship.
- The **Count Record** entity is the logical table that stores time series counts and serves as the "many" side of two key relationships. Each individual **Count** record is linked to exactly one **Deployment** and exactly one **Flow** (which specifies the specific counting movement). This creates a **many-to-one** relationship from Count to both Deployment and Flow.

3. Data Representation

This section provides the logical rules for the ATCS, focusing on the concepts rather than specific data formats. Subsection 3.1 starts with general conventions, Subsections 3.2 - 3.6 describe the key concepts associated with each of the five core entities outlined previously.

3.1 General Conventions

The following conventions are used throughout the specification:

- **Measuring Directionality: “Bearing” vs. “Heading”** – Although *bearing* and *heading* are both used in the schema to refer to a measured angle (0-359 degrees), each has slightly different intended purpose. The term *bearing* is used to refer to the orientation of a facility, while *heading* is used to capture travel direction. In both cases, 0 degrees is North, and positive angles increase in the clockwise direction (0 = N, 90 = E, 180 = S, 270 = W).
- **Spatial Reference (WGS-84)**: All spatial locations are represented as latitude and longitude coordinates in WGS84, the global reference system used for the Global Positioning System (GPS). When discussing locations informally, latitude is listed before longitude (e.g., “lat, lon”), but when encoding values in the data exchange format, note that GeoJSON specification uses a longitude, latitude (x,y) order for coordinates.
- **Temporal Reference (ISO 8601)**: Any temporal data (found in deployments and count records) must adhere to ISO 8601 standards for reporting dates and timestamps in a consistent manner (e.g., 2025-08-01 for date, and 2025-08-01T08:15:00 for timestamp). All timestamps are assumed to be in the local time zone unless a UTC offset is explicitly provided.
- **Drawing Conventions**: When annotating figures to describe concepts in this document, Site-level characteristics are colored **red**, while Flows (counting movements) are drawn in **black**.

3.2 Site Entity

A **Site** is the top-level location where counts are collected, and defines a counting location's fundamental geometry, its real-world infrastructure environment, and its specific layout. A properly defined site will provide the necessary context for defining and interpreting **Flows** (counting movements), which are defined in Section 3.3.

3.2.1 Defining Characteristics

Geometric Definition

A Site is defined spatially by a geographic bounding polygon. This polygon describes the extent of the physical area being monitored.

Base Type

The **Base Type** classifies a counting site into one of three categories, based on how it can be represented spatially using a simple geometric model. This geometric model will be defined in more detail in the **Site Diagram** section below.

Table 3-1: Base Type

Classification	Value	Definition	Geometric Model
Segment	segment	Represents a continuous, linear section of a road or path.	Line
Intersection	intersection	Represents a distinct junction where multiple roads or paths meet.	Node with radiating lines
Complex	complex	When the counting location is highly complex or represents an area that cannot be represented as a Segment or Intersection, it is classified as Complex and cannot be easily reduced to a simple model.	N/A

Figure 3-1 shows examples of each Base Type, with bounding polygons shown in red to show the spatial extent of the site. The left image shows a Segment along a shared use path, the middle shows an Intersection between two roads, and the right shows a Complex type for a complicated location that cannot be reduced to a simple geometric model. In these examples, the bounding polygon is a simple rectangle that encompasses the Site. However, the bounding polygon may be any type of closed polygon.

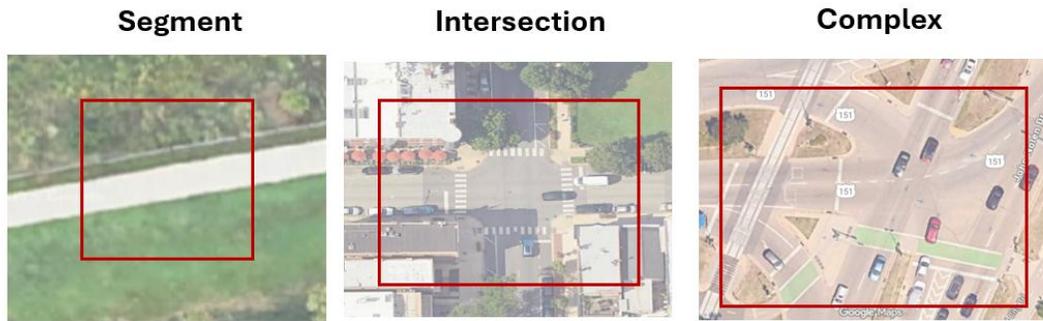


Figure 3-1: Example of Segment, Intersection, and Complex Base Types

Facility Classification

The **Facility Classification** is used to categorize the infrastructure at a Segment or Intersection site. This classification will be used to help communicate what type of facilities and travel movements can be defined in the Flows.

Segment sites may have either Road or Path facility classification. Intersections may have Road or Path classifications (if all legs are uniformly Road or Path, respectively) or be classified as Hybrid if legs are mixed Road/Path. Complex sites do not utilize this classification.

Table 3-2: Facility Classification

Category	Value	Definition	Applicable Base Types
Road	road	A Site that contains motor vehicle infrastructure, though it may also have pedestrian/bicycle facilities	- Segment - Intersection (if all legs are Road)
Path	path	A Site dedicated to non-motorized users, physically separated from the roadway network	- Segment - Intersection (if all legs are Path)
Hybrid	hybrid	A Site where Road and Path infrastructures intersect. This is typically used for Intersections where a trail crosses a street.	- Intersection (if legs include both Road and Path facilities)

Referring back to Figure 3-1, the leftmost image is an example of a Path Site (Segment Base Type), the middle image is a Road Site (Intersection Base Type) and the rightmost image does not have a facility classification because it is a Complex Base Type.

Base Type vs. Facility Classification: Note that **Base Type** describes the geometric structure of the location, while **Facility Classification** describes the transportation context. Both are needed because the same geometric structure (e.g., an intersection) can exist in different infrastructure environments.

Site Diagram

The **Site Diagram (SD)** is a simplified reference model used to unambiguously describe directions, legs, and relative orientation. It serves as a *logical* reference for the site (not a literal drawing on a map) and is intended to communicate how a single segment (for Segment Base Type) or multiple legs of an intersection (for Intersection Base Type) are oriented. Site Diagrams are defined as follows for the following Base Types:

- **Segments:** SD is defined by a Point (lon,lat) and bearing (0-360 deg).
 - *Interpretation:* The Point should represent a representative location on the linear segment (near the counting location), with the bearing communicating the primary orientation of the road/path. To visualize the segment, draw a short line centered on the Point with an angle defined according to the bearing. Note that the bearing focuses on how the segment is oriented, *not* the travel direction, which will be covered later in the Flows section.
- **Intersections:** SD is defined by a Point (lon,lat) and each leg is defined by a label and bearing (0-360 deg)
 - *Interpretation:* The Point represents center of the intersection and each leg's bearing represents the angle measured from the center point. To visualize the intersection, draw the center point and a line for each leg at the bearing angle.
 - Note that if the Facility Classification is "Hybrid" (an intersection of a Road/Path), the Site diagram should also indicate the facility class of each leg separately.
- **Complex:** Not applicable.

Figure 3-2 shows visual representations of Sites Diagrams for the same images shown previously in Figure 3-1. The leftmost image represents a Segment on a Path (Base Type = Segment, Facility Classification = Path), so the Site Diagram is defined by a point geometry and bearing. Visually, this is shown by the red dot placed in the center of the facility, with a line that follows the angle of the path. The middle image shows a visual representation of a SD for a Road Intersection (Base Type = Intersection, Facility Classification = Road). The red dot is located at the center of the intersection, and four legs (labeled N,E,S,W) are defined based on angles relative to the center point. Finally, the rightmost image does not have a SD because it is classified as Complex and cannot be reduced to a simple geometric model.

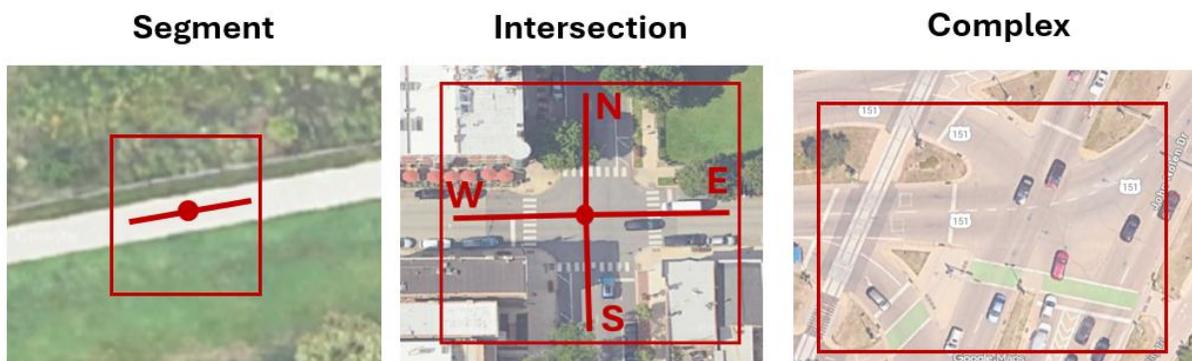


Figure 3-2: Examples of Site Diagrams for Segment, Intersection, and Complex Base Types

Intersection Control

The **Intersection Control** field communicates the control devices used at intersections, providing important context for the counting environment. This field is only relevant for Intersection Sites (not Segment or Complex), and categorizes values based on Highway Capacity Manual (HCM) and Manual on Uniform Traffic Control Devices (MUTCD) classifications.

Table 3-3: Intersection Control Classification

Category	Value	Definition
Signalized	signalized	Traffic is controlled by a full color traffic control signal (Red/Yellow/Green). This includes actuated, pre-timed, and adaptive signals, but excludes flashing red/yellow beacons (which are usually reinforced Stop or Yield controls).
All-Way Stop	all_way_stop	Every approaching leg must stop. This is commonly known as a 4-way stop (or 3-way stop at a T-Intersection).
Two-Way Stop	two_way_stop	Stop signs are present on the minor street(s) only; the major street flows freely.
Roundabout	roundabout	A circular intersection where entering traffic yields to circulating traffic.
Yield	yield	The intersection is controlled by Yield signs rather than Stop signs.
Uncontrolled	uncontrolled	No regulatory signs (Stop/Yield) or signals exist.
Other	other	Complex or non-standard controls (e.g., Police Officer directing traffic).

3.2.2 Site Validation Logic

To ensure the digital record accurately reflects the physical environment, a Site must satisfy the following logical constraints:

- **Mandatory Spatial Extent:** Every Site must be defined by a valid Bounding Polygon that encompasses the entire study area. Linear strings or single points are insufficient to define the spatial extent of a site.
- **Geometric Consistency:** The requirements for the Site Diagram are determined by the site's Base Class:
 - **Segment & Intersection:** These sites *must* possess a valid SD (containing defined Legs and Bearings) to support topological reference.
 - **Complex:** These sites *must not* contain a logical Site Diagram. The geometry is assumed to be unstructured.
- **Facility Class Inheritance (Intersections):** For Intersection Sites, when Facility Class is Road or Path, it is assumed that all legs of the intersection “inherit” that classification without needing to be explicitly labeled. However, if Facility Class is Hybrid (i.e., mixture of Roads and Paths), each leg must be labeled as either Road or Path in the SD.

3.3 Flow Entity

A **Flow** describes a specific counting movement within a Site. These flows – defined to capture specific travel modes, facilities, and travel directions, plus other details -- represent the movements that are measured by a counter.

3.3.1 Defining Characteristics

Geometric Definition

Regardless of the count type or the complexity of the site, every Flow is defined by two absolute geometric properties:

- **Lat/Long Coordinate (geometry):** A specific WGS84 Latitude/Longitude coordinate representing the primary location of the count.
- **Heading (heading):** An integer bearing (0–360°) representing the direction of travel or the orientation of the movement relative to True North.

A heading is required so that the flow can be oriented consistently relative to the Site Diagram, even when counts are bidirectional.

When the Flow represents a Turning Movement, a coordinate and heading *are needed at both start and end locations*. However, the start location and heading is considered the primary reference.

Types of Counting Movements

Flows can be categorized into three types of counting movements, which are summarized visually in Figure 3-3 and enumerated below: **Screenline**, **Turning Movement**, and **Crossing**. Specifying the count type will impact what other attributes need to be populated in the Flow description.

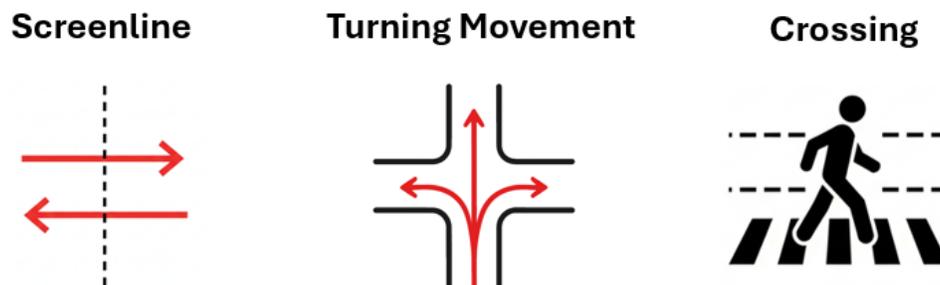


Figure 3-3: Visual Classification of Counting Movements

Table 3-4: Count Type

Category	Value	Definition	Applicable Base Types
Screenline	screenline	Users passing a single point along a facility (trip wire). Measures facility “throughput” or usage.	<ul style="list-style-type: none"> • Segment: along a segment • Intersection: along <i>one leg</i> of an intersection • Complex: along a facility within Site (measures facility usage)
Turning Movement	turning_movement	Users traveling between defined start and end points. Measures movement between locations.	<ul style="list-style-type: none"> • Intersection: movement between start and end legs of an intersection • Complex: movement between two points
Crossing	crossing	Users crossing a facility at a particular point. Measures “exposure”/conflict.	<ul style="list-style-type: none"> • Segment: crossing (perpendicular to) a segment • Intersection: crossing (perpendicular to) <i>one leg</i> of an intersection • Complex: crossing any facility within the Site (measures facility exposure/conflict)

When these Count Types are applied to a Complex site (in cases where the site cannot be defined as a Segment or Intersection), standard topological references (e.g., start_leg) are omitted because there is no Site Diagram to reference. The movement is defined solely by the absolute Geometry (Lat/Lon) and Heading of the Flow, supplemented by the description field.

Figure 3-4 below provides visual examples of each type of count. In each image, the Site’s Base Type and Facility Class fields are specified, with Site Diagram visualized in red for context, and the Flow shown in black. The black dot corresponds to the latitude/longitude coordinate, and the arrow is drawn in the direction of the Heading.

Travel Mode

Travel Mode classifies the counting movement by *how* the user travels. The intent of this field is to capture the broad category of mode (e.g., pedestrian versus bike) rather than detailed differences between users. However, note that these categories can later be broken down into sub-categories in the Count Record section (e.g., to categorize traditional versus e-assist bikes).

Table 3-5: Travel Mode

Category	Value	Definition	Notes
Non-Motorized	non_motorized	All-inclusive category for users that are not in a motor vehicle.	Used when a sensor does not distinguish modes.
Pedestrian	pedestrian	People on foot, in wheelchairs, or using mobility-assisted devices.	
Bicycle	bicycle	Traditional (“analog”) or electric-assist bicycles	
Scooter	scooter	Electric or kick scooters.	
Other	other	Skateboards, roller blades, or other non-motorized devices not fitting the previous categories	

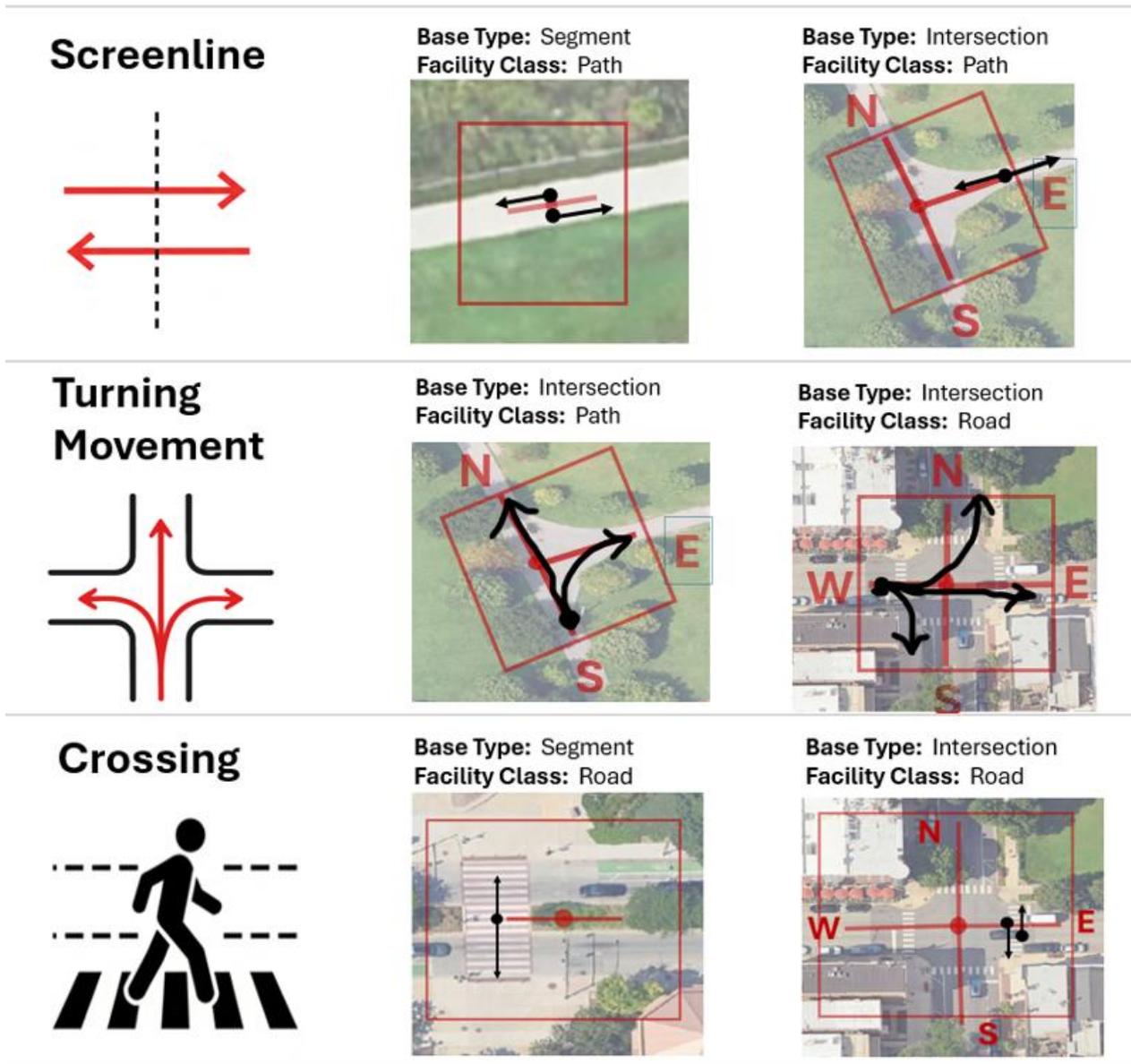


Figure 3-4: Examples of Screenline, Turning Movement, and Crossing Flows

Facility Type

The facility type describes the infrastructure where the counting movement is collected. Note that the allowable options for this field are constrained depending on the Site's Facility Classification (see 3.7 for validation rules). Additionally, for turning movements, this attribute can be specified at both the start and end location (in case a trip starts and ends on different types of facilities).

Table 3-6: Facility Type

Category	Value	Definition	Notes
Right-of-Way	right_of_way	The entire cross-section of a road segment, including all parallel facilities (e.g., road, sidewalks, shoulders).	Used on Road sites with parallel facilities.
General Lane	general_lane	General purpose lane(s) in a roadway.	
Bike Lane	bike_lane	Marked bike lane without physical separation.	
Separated Bike Lane	separated_bike_lane	Bike lane that is physically separated from motorized traffic (sometimes called “cycle track”).	
Shoulder	shoulder	Paved shoulder on the edge of a roadway, not designated as a travel lane, but usable by active transportation modes.	
Shared Use Path	shared_use_path	A trail/path that is physically separated from motorized vehicular traffic with an independent right-of-way	
Crosswalk	crosswalk	Designated point for crossing a roadway, typically marked with paint or signage.	

Facility Side

Facility Side describes the position of the facility infrastructure relative to the centerline of the segment (or intersection leg). It is used to disambiguate cases where there are multiple facilities of the same type (e.g., sidewalks on both sides of a roadway – both of which would be associated with the same Segment Site). The Facility Side field should be specified for paired facilities (e.g., sidewalks, bike lanes) where two facilities may exist someday– even if only one exists currently.

Table 3-7: Facility Side

Category	Value	Notes
North Side	N	
North-East Side	NE	
East Side	E	
South-East Side	SE	
South Side	S	
South-West Side	SW	
West Side	W	
North-West Side	NW	
Center	C	Located in the median or center of the right-of-way.

3.3.2 Flow Validation Logic

To ensure data integrity and analytical consistency, the following validation logic applies to Flows. In general, the validation of a Flow depends on the context provided by the parent Site.

- **“Vector” Representation:** Regardless of the Site type, every Flow must be defined by a primary coordinate location (latitude, longitude) and heading (direction). This ensures that any flow can be visualized independently of the site diagram.
- **Topological Consistency:** If the parent Site is structured (i.e., Base Type is Segment or Intersection), the Flow must be consistent with the Site Diagram.
 - **Segments:** For Screenline counts, the Flow’s heading (0-359 deg) should be aligned with the Site Diagram bearing (0-359 deg). This means the angular difference is very close to 0 or 180 degrees. For Crossing counts, which represent a perpendicular movement to the segment, the angular difference should generally be close to 90 or 270 degrees.
 - Note: There may be edge cases for which crossing counts are not perpendicular. This test should highlight the discrepancy for manual review.
 - **Intersections:** Screenline and Crossing counts must explicitly reference a valid leg label that is defined in the Intersection SD. As described previously for Segments, Flow headings must be aligned with or perpendicular to the SD bearing from the appropriate leg for Screenline and Crossing counts, respectively.

Turning Movement Counts must explicitly reference two valid leg labels defined in the Intersection SD – one representing the beginning, and the other representing the end of the movement. The start and end headings should align with the SD bearings associated with each leg.
 - **Complex:** If the parent Site is Complex, the Flow must *not* reference specific Legs. Instead, the validity of the flow relies solely on the coordinates and headings.
- **Uniqueness Constraint** To prevent ambiguity in the dataset, a Site cannot contain duplicate definitions for the same physical movement. The combination of *Mode*, *Facility Type*, and *Facility Side* must be unique for a given *Count Type* within a single Site. This prevents defining two indistinguishable Pedestrian Sidewalk Flows on the same side of a segment.
- **Site-Facility Compatibility** A Flow’s defined Facility Type must be physically compatible with the parent Site’s Facility Class. This prevents logical errors, such as defining a Sidewalk on a trail network. These rules are summarized below in Table 3-8.

Table 3-8: Site-Facility Compatibility Logic

If Site Facility Class is...	Then Allowed Flow Facility Types are...
Road	Right-of-Way, General Lane, Bike Lane, Separated Bike Lane, Shoulder, Sidewalk, Crosswalk
Path	Shared Use Path
Hybrid	<i>Determined by Leg Inheritance (each leg may be Road or Path facility class)</i>
N/A (Complex Site Type)	<i>All Types Permitted (Context defined by Description)</i>

3.4 Counter Entity

The **Counter** entity represents the specific instrument, device, or human observer responsible for collecting data. This information enables agencies to understand *how* the data was collected.

The Active Transportation Count Specification utilizes a method-agnostic approach to hardware. Whether the source is a pneumatic tube, a camera, or a person manually tallying, it is treated as a Counter with a unique identifier.

3.4.1 Defining Characteristics

Counter Type

To ensure data quality and allow for appropriate error modeling (e.g., occlusion biases or weather sensitivity), every counter must be classified by its type of data collection. The following methods align with FHWA Traffic Monitoring Guide (TMG) categories and NCHRP Report 797 (*Guidebook on Methods for Estimating Non-Motorized Travel Volume*).

Table 3-8: Counter Type

Category	Value	Notes
Inductive Loop	inductive_loop	
Passive Infrared	passive_infrared	
Active Infrared	active_infrared	
Pneumatic Tube	pneumatic_tube	
Piezoelectric	piezoelectric	
Radar	radar	
Magnetometer	magnetometer	
Camera	camera	
LiDAR	lidar	
Human	human	In-field observation
Other	other	Emerging/unclassified

Additional Counter Characteristics

Other optional attributes include information about equipment manufacturers (make, model), software versions, and other descriptive information.



Figure 3-5: Examples of Inductive Loop (left) and Pneumatic Tube (right) Count Technologies

3.5 Deployment Entity

A **Deployment** describes *when and where* a specific Counter was active. In contrast to Site and Flow locations, which focus on the facilities along which counts are collected, the Deployment location reflects where the sensor itself is deployed (which may be some distance from the facility). This entity links a particular Counter to a specific location for a time period.

3.5.1 Defining Characteristics

Consistent Site and Flow Definitions

Deployments are assumed to have a static physical configuration. If the Site infrastructure changes (e.g., painted Bike Lane is turned into a Separated Bike Lane) or the Flow logic changes, *the current deployment must end*. This will ensure that historical data is accurately linked to the Site infrastructure at the time of collection.

- **Note:** While the **Site** represents the permanent infrastructure being studied (e.g., the intersection of Main St & 1st Ave), the **Deployment** represents the specific physical placement of the counter.

For example, if you move a camera 50 feet to get a better viewing angle of the same crosswalk, you create a new **Deployment** record because the sensor location has changed. However, the **Site** (the crosswalk itself) and the **Flow** (the crossing movement) remain unchanged. This separation allows agencies to track exactly where equipment was placed without needing to redefine the facility being counted.

Deployment Location

The deployment location defines where the Counter is deployed. This is not critical for describing the intended movement (as this is handled via Site and Flow definitions) but is needed to be able to know where equipment was mounted (or a human observer was standing). If the Counter location changes (e.g., a sensor was moved), *the current deployment must end*.

Processing Method

Additional context is sometimes needed to indicate how data collected by a Counter is processed to obtain the actual counts. The most common example is a camera, which collects video footage at a location, which can either be reduced to counts by a person manually reviewing the video, or through automated algorithms/services. In fact, the same camera, which may be moved from location to location across deployments, may sometimes be used for manual counts, and other times fed to a video analytics service. The processing method should be specified as either Automated or Manual to provide context for the counts.

3.6 Count Record Entity

The **Count Record** contains aggregated counts by date and time. Each record, which captures the counts for a single Flow and time period, links back to the Deployment (which defines where and when a Counter was deployed) and the Flow (which defines the specific movement).

3.6.1 Defining Characteristics

Temporal Aggregation

Counts are associated with a timestamp and duration that captures the aggregated time period (e.g., 15 or 60 minutes). The following best practices are recommended:

- Within a particular Deployment, Count Records should use a consistent time aggregation.
- Timestamps should align with the beginning of a period. For example, 60-minute time bins should start at the beginning of an hour (e.g., 08:00, 09:00), while 15-minute bins should start at 0, 15, 30, and 45 minutes (e.g., 08:00, 08:15, 08:30, 08:45).

Mode Sub-Categories

Although the Flow entity defines the primary Travel Mode, the Count Record entity allows for optional, granular subcategorization via a Mode Subtype attribute (free-text input, not constrained by a set of predefined values). Whereas *the main Travel Mode (defined by the Flow) must be stable and constrained to the defined allowable values*, the subcategories may be user-defined.

Mode SubType can be useful when a specific sensor technology can distinguish classes beyond the primary mode (e.g., distinguishing different sub-classes of “Bicycle” such as “Cargo Bike”). It is expected that this will become more prevalent with vision-based solutions, as the same video footage may be re-processed with a new algorithm to extract different detailed classifications. Importantly, though, even if the subtype is specified, this does not change the primary category, which is part of the Flow definition.

Data Quality Flag

Quality Assurance and Quality Control (QA/QC) tests are often conducted after count records are imported to a database management system (e.g., within the Texas BP|CX or BikePed Portal), and may include statistical tests (e.g., as outlined in FHWA’s Traffic Monitoring Guide) and manual outlier detection determined through visual inspection of plots. Such tests are important to make sure that data used for reporting and analysis differentiates between valid and invalid data.

Although the Coalition ATCS does not provide guidance on exactly how to determine whether recorded count records are valid, the Data Quality Flag field is included to support translation between databases where QA/QC testing has already been executed, enabling proper interpretation of count records. In cases where the ATCS package reflects raw counts that have not yet been evaluated, this field can be left blank.

The four flags listed in Table 3-9 combine ideas from the BP|CX and BikePed Portal and are mutually exclusive. The first flag is “Valid,” which is used when the data has passed all tests and is considered to be high-confidence and reflective of typical conditions. Records that meet this

criterion can be used for any purpose. The next flag is “Valid, but Atypical,” which reflects high-confidence counts taken during atypical conditions. This flag may be used if high-confident counts were collected during a race/festival/special event, which is not reflective of typical conditions and may skew average values. This warns the user and lets them decide how to interpret the values (e.g., whether to include them in monthly averages), while simultaneously indicating that the data is not wrong. The third category is “Suspect,” which can be used to flag low-confidence data that appears to be incorrect, but has not been fully vetted and deemed invalid. The “Suspect” flag value might be generated automatically via statistical testing, and until further determination, indicates that the count records should not be used for analysis or reporting. Finally, the “Invalid” flag indicates that the count record is known to be incorrect (e.g., due to sensor failure) and should be excluded from all analysis.

Table 3-9: Data Quality Values

Classification	Value	Description	How to Use
Valid	valid	High confidence, normal traffic conditions.	Use for everything (totals, averages, baselines).
Valid, but Atypical	valid_atypical	High confidence (the count is correct), but the conditions were abnormal.	Include in totals (sum) but possibly exclude from averages.
Suspect	suspect	Low confidence. The data looks wrong (statistical outlier) or failed an automated check, but hasn't been manually verified yet.	Typically exclude from analysis and flag for human review.
Invalid	invalid	Known bad data (e.g., sensor failure, occlusion).	Always exclude from analysis.

4. Package Specification

While Section 3 described the entities, Sections 4 and 5 define how those entities are implemented as files and packaged as part of the ATCS. At a high level, this requires creating a set of text/GIS files that align with the data model, resulting in 5 files – one for Site, Flow, Counter, Deployment, and Count Record entities. This section focuses on file formats used and how they are packaged together to create a lightweight, portable, and database-ready exchange format.

4.1 File Packaging

The ATCS standard consists of five core files that function as a relational unit, plus one metadata file, typically packaged together in a Zipfile. While they are separate files, the core files are logically linked (via unique identifiers) and are intended to be processed together. Collectively, these files are used to unambiguously describe the details of bike/ped data collection.

- **Location Files (GeoJSON):** `sites.geojson`, `flows.geojson`, `deployments.geojson`. These files define the physical infrastructure, logic of the movements, and where counters are deployed – each of which has a spatial component. These are the most complex to define, and benefit from using GeoJSON's nested structure to model site geometries and counting movements, while also enabling non-technical users to visualize locations in standard GIS software. These files may take some work to initially define but tend to be static once created.
- **Data Files (CSV):** `counters.csv`, `counts.csv`. These files contain the records of counters and resulting time-series data for the counting movements (Flows) and deployment locations defined in the GeoJSON files. They are dynamic and can grow over time as additional sensors are deployed and counts are collected. Note that the decision to split the standard into GeoJSON and CSV file formats is intentional. It allows Site and Flow definitions to be appropriately specified using a spatially enabled format that can handle complexity well, while ensuring that the data collection files are compact and easy to prepare/process in standard spreadsheet software.
- **Metadata File (JSON):** `metadata.json`. The metadata file is used to specify the schema version, identify the names of the core files (e.g., Site file is `sites.geojson`), and other optional information to quickly identify the contents of the files.

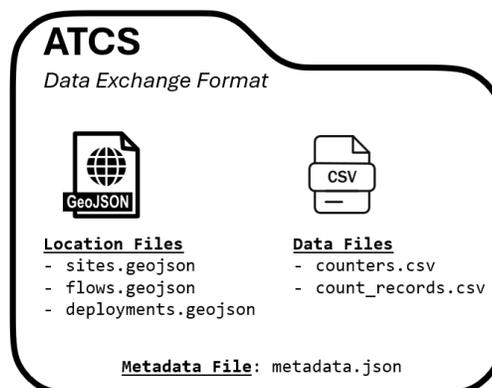


Figure 4-1: Visual Example of Files in ATCS Package

GeoJSON Format

Three files, Sites, Flows, and Deployments, are GeoJSON files that contain spatial information. The Sites file captures the overall study area (defined by Polygon boundaries, plus a Site Diagram) while the Flows file describes specific counting movements (Point locations). GeoJSON was selected as the format for representing these spatial files because (I) the file format is simple (human readable/editable), (II) it is fully supported by common GIS software and web viewers, and (III) can handle nested data structures that are well suited to represent the conceptual model.

Following the specified GeoJSON structure, each record in a GeoJSON file consists of “geometry” and “properties” JSON objects. The geometry object stores the spatial representation for each feature (Point and Polygon types are the relevant types for this implementation), while all other attributes are stored in the “properties” object.

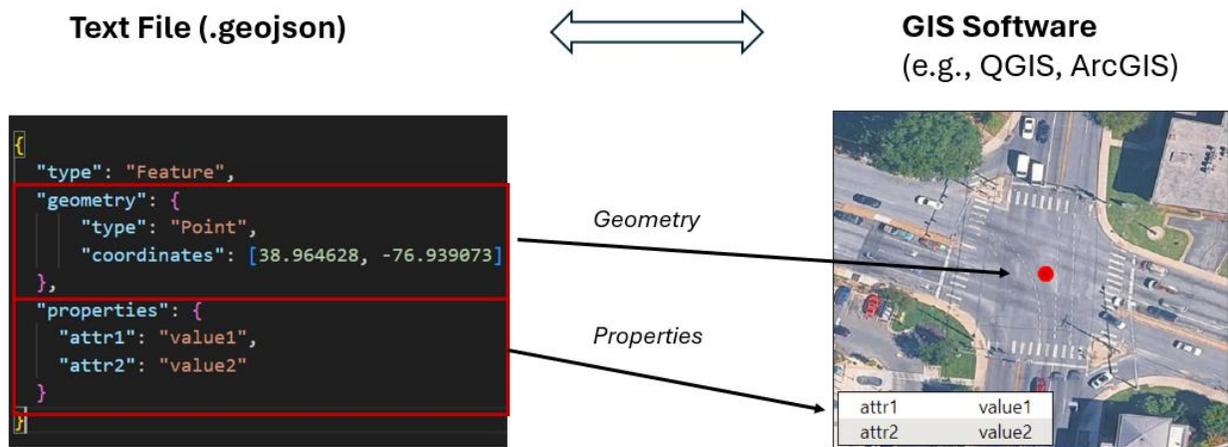


Figure 5-1: Example GeoJSON Record and Corresponding Visualization in GIS.

CSV Format

The two other tables, Counters and Count Records, contain tabular and time series data that is most naturally represented in a comma-separated value (CSV) format. CSV is a simple text format where each line represents a table row, and column values are separated by a character that serves as a delimiter (usually a comma).

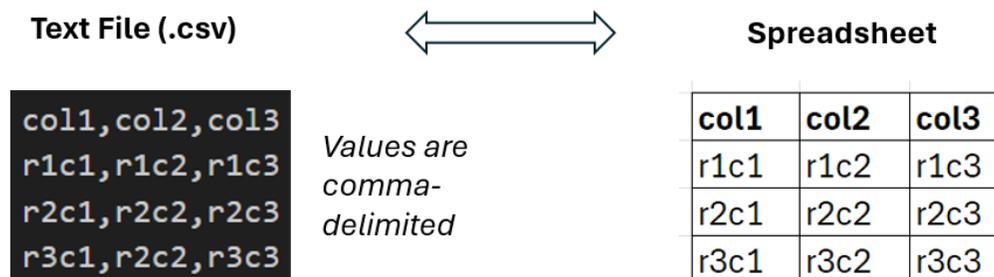


Figure 5-2: Example CSV Record and Corresponding Spreadsheet Representation.

4.2 Use Cases and Scope

The format is designed to work for a range of use cases with different sizes and scopes. For example, the file format is equally valid for the following examples:

- **Project Level Scope:** A single data collection event (e.g., a short-term count collected at 2 sites). In this example, the ATCS files might contain two Site definitions, several Flow definitions (depending on how many counting movements need to be captured), two Counters and Deployments (assuming one per site), and a small file of 15-minute Count Records.
- **Repository / Archive Level Scope:** A regional repository containing bike/ped count data across a city, state, or even Coalition-wide. The files might contain hundreds of Sites, thousands of Flows, and millions of Count Records.

4.3 Adaptable Workflows

A key feature of the ATCS is that the five files do not need to be created simultaneously. The schema supports various workflows, making it ideal for the contracting process.

For example, consider the following use case, where an agency hires a contractor to collect data at agency-defined sites.

- *Step 1 (Agency Side):* An agency prepares the **sites.geojson** and **flows.geojson** files. These files define the requirements: *"Here are the locations and the specific movements we want counted"* and are provided to the contractor.
- *Step 2 (Contractor Side):* The contractor deploys their equipment, collects counts for the specified period of time, and creates **counters.csv**, **deployments.geojson**, and **counts.csv**. These files define the results.
- *Step 3: Integration:* The contractor returns their three files. The agency joins them with the original GeoJSON files to create the complete ATCS dataset.

4.4 Database and Analytics Compatibility

The ATCS is an exchange format, designed for moving data between systems. However, because it is based on a relational model, it is well-suited for long-term storage in a relational database (e.g., PostgreSQL, SQL Server) and can be integrated with new or existing analytics systems. A stable data exchange format will allow data to be imported and analyzed in a consistent manner without needing ad-hoc solutions for many different formats.

Additionally, even without importing to a database, the GeoJSON files in the ATCS package can be directly loaded into GIS software (e.g., ArcGIS Pro, QGIS) to immediately visualize counting locations and movements.

5. Schema Reference

As described in Section 4, an ATCS package defined by a set of text/GIS files that align with the data model, resulting in 5 core files – one for Site, Flow, Counter, Deployment, and Count Record entities (plus a metadata file). Three files, Sites, Flows, and Deployments are GeoJSON files containing spatial information, while Counters and Count Records are CSV files containing tabular data. Additionally, there is a simple metadata file that is used to capture the schema version and other contextual information to help interpret the files.

5.1 metadata.json

The Metadata JSON file is a lightweight configuration file stored alongside the other core files. Its main purposes are to record the schema version, clarify how to interpret each of the files, and provide a unique namespace for the identifiers contained in the files (to make them uniquely identifiable if later imported to a database). Table 5-1 shows the properties of the metadata JSON file.

Table 5-1: Metadata Attributes

Field Name	Type	Required?	Description & Allowed Values
atcs_version	String	Yes	Indicates the ATCS version (e.g., "v0.1")
dataset_version	String	Yes	Indicates the release or revision of the specific data contained within the package (e.g., "2025-11-21" or "1.1"). This will allow systems to identify and overwrite older data with newer corrections or updates.
provider_id	String	Yes	Indicates the entity providing the data (e.g., "tetc", "district_dot") This will be used to prefix IDs provided in any of the core files to ensure they are globally unique.
package_version	String	Yes	Indicates the version of the dataset. Recommended to use a date format (e.g., YYYY-mm-dd) to indicate when the data was released.
name	String	No	Simple, human readable description of the data collection activity.
resources	Array	Yes	Array of objects that identifies the specific files included in the package and maps each one to its corresponding logical entity within the ATCS standard. Each resource object contains the following: <ul style="list-style-type: none"> • entity [String]: Identifies the entity (e.g., "site") • path [String]: Path to file/resource ("sites.geojson") • description [String]: Optional description of file

5.2 sites.geojson

Each record in the Sites GeoJSON file represents a single Site feature, which is defined by the following characteristics:

- **Geometry:** GeoJSON Polygon geometry object that defines the boundary of the Site area.
- **Properties:** A set of JSON fields that describe the Site characteristics, including the Base Type, Facility Classification, and Site Diagram.

Table 5-2 summarizes the information stored in each Site feature, defined in GeoJSON format. Each value in the Field Name column of the table corresponds to a JSON object within the “geometry” or “properties” object. In the case of nested JSON objects (e.g., `site_diagram`), indentation is used to communicate child attribute descriptions.

Note that fields marked as “required” must always be present as keys in the JSON object. If a field is marked “conditional”, see the “Description and Allowed Values” column to determine whether the field is applicable for the specific context. In cases where a field is not applicable to a specific site type (e.g., `site_diagram` for a Complex site), the field should be omitted.

Table 5-2: Site Feature Definitions (GeoJSON)

Field Name	Type	Required?	Description & Allowed Values
Geometry			
<code>geometry</code>	GeoJSON Object	Yes	A GeoJSON object containing the following objects: <ul style="list-style-type: none"> • type [String]: Polygon • coordinates [Array]: Array of Longitude, Latitude coordinates that describe the Polygon boundary.
Properties			
<code>site_id</code>	String	Yes	A unique identifier for the site. Note that this must be unique within the ATCS package.
<code>base_type</code>	String	Yes	Categorizes a site into a fundamental base type, based on how it can be modeled geometrically. <u>Allowed values:</u> Enumerated in Base Type Table .
<code>facility_class</code>	String	Conditional	High level categorization of infrastructure. <u>Allowed values:</u> Enumerated in Facility Class Table . <i>Omit if Base Type is Complex.</i>
<code>site_diagram</code>	Object	Conditional.	A JSON object that describes the site's simplified geometric layout, with structure depending on Base Type (<code>base_type</code>). <ul style="list-style-type: none"> • Segment: <ul style="list-style-type: none"> ○ reference_point [Array]: Centered on segment location ○ bearing [Int]: Orientation of the road/path segment • Intersection <ul style="list-style-type: none"> ○ reference_point [Array]: Defines center of intersection ○ legs [Array of Leg objects]: Defines intersection legs <i>Omit if Base Type is Complex.</i>

site_diagram └ reference_point	Array	Cond.	[Longitude, Latitude] Array representing the reference point. <i>Required if Site Diagram exists (i.e., not Complex).</i>
site_diagram └ legs	Array	Conditional	Array of Leg objects containing: <ul style="list-style-type: none"> • label [String]: Unique leg identifier • bearing [Integer]: Orientation of road/path segment (0-359) • facility_class [String]: <i>Conditional</i>. If facility_class is Hybrid, then specific facility classification per leg. <i>Only used for intersections.</i>
intersection_control	String	No	Describes type of intersection control (only relevant for intersections). Allowed values: Enumerated in Intersection Control Table . <i>Omit if Base Type is Segment or Complex.</i>
state	String	No	State where Site is located Allowed values: Two letter state abbreviation (e.g., "MD").
county	String	No	County where Site is located (e.g., "Prince George's").
municipality	String	No	Municipality (city, town, village, etc.) where Site is located. <i>This focuses on the location of the Site, not who owns and manages the facility.</i>
jurisdiction	String	No	The owner/authority responsible for the facility (e.g., "MDOT"). <i>This focuses on who owns and manages the facility, not where it is located.</i>
tags	Object	No	A flexible JSON object for additional key-value properties that describe the site.

5.3 flows.geojson

Each record in the Flows GeoJSON file represents a single Flow feature and is defined by the following characteristics:

- **Geometry:** GeoJSON Point geometry that defines the core counting location.
 - For *Screenline/Crossing Counts*, this is the specific location where the counting movement is observed.
 - For *Turning movements* that involve multiple intersection legs, it is the location where the counting movement starts (i.e., start leg).
 For Turning movements, the properties object also requires a mandatory GeoJSON Point object to capture the ending location. Only the start point will be included in the official geometry object (used for visualization GIS software), but the end Point information will be encoded if needed for more advanced use cases.
- **Properties:** A set of JSON fields that describe the Flow characteristics. There are a set of common fields (required as part of any Flow definition) followed by conditional fields that must be present (or omitted) based on the Count Type. Table 5-2 shows the common Flow properties, while Tables 5-3, 5-4, and 5-5 show the specific fields needed to describe Screenline Counts, Turning Movement Counts, and Crosswalk Counts.

Feature Definitions: All Flows

The following Flow feature definitions are relevant for all flows, regardless of what type of counting movement a flow represents.

Table 5-3: Flow Feature Definitions Common to All Scenarios (GeoJSON)

Field Name	Type	Required?	Description & Allowed Values
Geometry			
geometry	GeoJSON Object	Yes	A GeoJSON object containing the following objects: <ul style="list-style-type: none"> • type [String]: Point • coordinates [Array]: [Longitude, Latitude]
Properties			
flow_id	String	Yes	A unique identifier for the counting movement.
site_id	String	Yes	Must reference a valid site in sites.geojson .
count_type	String	Yes	The type of counting movement. <i>This selection impacts what other fields are required.</i> Allowed values: Enumerated in Count Type Table , although “crossing” is invalid when the Site’s Facility Classification is Path.
travel_mode	String	Yes	The primary classification of travel mode. <i>This can be further subcategorized in the count records.</i> Allowed values: Enumerated in Travel Mode Table .
description	String	Conditional	Human-readable description of the flow.

			<i>Required for Complex Sites where there is no Site Diagram, Optional for other cases.</i>
map_references	Object	No	<p>A JSON object where each key is a string representing the maps source (e.g., "osm", "overture", "gatis") and the corresponding value is an object containing the following keys:</p> <ul style="list-style-type: none"> • id [String]: External identifier • version [String] <i>Optional</i>: The dataset version or date • uri [String]: <i>Optional</i>. Link to the feature (e.g., OSM URL) <p><i>Multiple map references may be used – either to link to multiple maps (e.g., keys "osm" and "tmc") or when a turning movement starts and ends at different location (e.g, keys "start_osm", "end_osm").</i></p>
tags	Object	No	A JSON object containing non-standard, user-defined, or other custom key-value pairs.

Screenline Counts

In addition to the required featured definitions defined above (which are relevant for all Flows), the following additional properties are used for Screenline counts.

Screenline counts are typically collected on segments, either belonging to a Segment site or a single leg of an Intersection, although they can also be defined for Complex base types, if needed (see Screenline row in [Count Type Table](#)).

Table 5-4: Additional Flow Properties for Screenline Counts (GeoJSON)

Field Name	Type	Required?	Description & Allowed Values
Properties			
leg	String	Conditional	<p>The label of the leg where the count is located. The leg label must be defined in the Site Diagram.</p> <p><i>Required if Site Type is Intersection. Omit if Site Type is Segment or Complex.</i></p>
heading	Integer	Yes	<p>Direction of traveler movement (0-359 degrees, 0 = N, 90=E, ...).</p> <p><i>If flow represents bi-directional travel, the heading can be defined in either direction.</i></p>
is_bidirectional	Boolean	Yes	<p>Indicates whether the count is bi-directional (i.e., includes counts in both directions of travel).</p> <p>Allowed values: True, False</p>
facility_type	String	Yes	<p>The specific infrastructure where the count is collected.</p> <p>Allowed values: Enumerated in Facility Type Table, subject to the Site-Facility Compatibility Logic Table, which limits values by Site Facility Class.</p>

facility_side	String	Conditional	Location relative to the corridor (as defined by Site diagram bearing). Used to disambiguate multiple facilities with the same type on the same corridor. <u>Allowed values:</u> Enumerated in Facility Side Table . <i>Required if there are multiple flows on the same segment (or intersection leg) with the same facility type. Omit if not relevant.</i>
----------------------	--------	-------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Turning Movement Counts

In addition to the required featured definitions defined in the previous section (which are relevant for all Flows), the following additional properties are used for Turning Movement counts.

Turning movement counts are collected at Intersection sites, with movements reflecting trips between two legs, or Complex sites, with movements reflecting trips between arbitrary start and end points. However, as noted in the [Count Type Table](#), turning movements cannot be defined for Segment sites.

Table 5-5 Additional Flow Properties for Turning Movement Counts (GeoJSON)

Field Name	Type	Required?	Description & Allowed Values
Properties			
start_leg	String	Conditional	The label of the leg where the trip starts . The leg label must be defined in the Site Diagram. <i>Required if Site Type is Intersection. Omit if Site is Complex.</i>
start_heading	Integer	Yes	Direction of travel at the start leg location (0-359 degrees, 0 = N, 90=E, ...).
start_facility_type	String	Yes	The specific infrastructure where the trip starts . <u>Allowed values:</u> Enumerated in Facility Type Table , subject to the Site-Facility Compatibility Logic Table , which limits values by Site Facility Class.
start_facility_side	String	Conditional	Side of segment where the trip starts . <u>Allowed values:</u> Enumerated in Facility Side Table . <i>Required if there are multiple flows on the same segment (or intersection leg) with the same facility type. Omit if not relevant.</i>
end_leg	String	Conditional	The label of the leg where the trip ends . The leg label must be defined in the Site Diagram. <i>Required if Site Type is Intersection. Omit if Site is Complex.</i>
end_heading	Integer	Yes	Direction of travel at the ends leg location (0-359 degrees, 0 = N, 90=E, ...).
end_facility_type	String	Yes	The specific infrastructure where the trip ends .

			Allowed values: Enumerated in Facility Type Table , subject to the Site-Facility Compatibility Logic Table , which limits values by Site Facility Class.
end_facility_side	String	Cond.	Side of segment where the trip ends . Allowed values: Enumerated in Facility Side Table . <i>Required if there are multiple flows on the same segment (or intersection leg) with the same facility type. Omit if not relevant.</i>
end_latitude	Float	Yes	Latitude where trip ends .
end_longitude	Float	Yes	Longitude where trip ends .

Crossing Counts

In addition to the required featured definitions defined previously (which are relevant for all Flows), the following additional properties are used for Crossing counts.

Crossing counts are typically collected on segments, either belonging to a Segment site or a single leg of an Intersection, although they can also be defined for Complex base types, if needed (see Crossing row in [Count Type Table](#)). In addition to the required common properties, the following additional properties are required when Count Type is Crossing.

Table 5-6: Additional Flow Properties for Crossing Counts (GeoJSON)

Field Name	Type	Required?	Description & Allowed Values
Properties			
crossing_leg	String	Cond.	The label of the leg that the flow crosses. The leg label must be defined in the Site Diagram. <i>Required if Site Type is Intersection. Omit if Site Type is Segment or Complex.</i>
heading	Integer	Yes	Direction of travel (0-359 degrees, 0 = N, 90=E, ...). <i>If flow represents bi-directional travel, the heading can be defined in either direction.</i>
is_bidirectional	Boolean	Yes	Indicates whether the count is bi-directional (i.e., includes counts in both directions of travel). Allowed values: True, False
facility_type	String	Yes	The specific infrastructure where the crossing count is collected. Allowed values: “crosswalk” or “general_lanes”

5.4 counters.csv

Each record in the Counters CSV file represents a single Counter, which is defined by the following columns (attributes). Columns that are specified as required must be included, but additional columns may be optionally provided.

Table 5-7: Counter Columns (CSV)

Field Name	Type	Req'd?	Description & Allowed Values
counter_id	String	Yes	Unique identifier for counting device or method.
counter_type	String	Yes	Primary count technology used. <u>Allowed values:</u> Enumerated in Counter Type Table .
make	String	No	Hardware manufacturer (e.g., Eco-Counter, MetroCount).
model	String	No	Specific model name or number.
serial_number	String	No	Serial number for hardware.
...	...	No	Any additional user-defined fields may be optionally provided.

5.5 deployments.geojson

Each record in the Deployments GeoJSON file represents a single Deployment feature, which reflects where a Counter is deployed for a period of time. It is defined by the following characteristics:

- **Geometry:** GeoJSON Point geometry that defines the deployment location (where equipment is positioned).
- **Properties:** A set of JSON fields that describe the Deployment attributes.

Table 5-7 shows the properties associated with each Deployment. Note that the deployment location (latitude and longitude coordinate) is stored in the “geometry” object rather than the “properties”, so it is not included in the table below.

Table 5-8: Deployment Feature Definitions (GeoJSON)

Field Name	Type	Req'd?	Description & Allowed Values
Geometry			
geometry	GeoJSON Object	Yes	A GeoJSON object containing the following objects: <ul style="list-style-type: none"> • type [String]: Point • coordinates [Array]: [Longitude, Latitude] This Point object reflects the location where the counter is deployed.
Properties			
deployment_id	String	Yes	Unique identifier for specific data collection event.
site_id	String	Yes	Must reference a valid site in sites.geojson .
counter_id	String	Yes	Must reference a valid counter in counters.geojson .
processing_method	String	Yes	Indicates whether counts are machine-derived or manual. <u>Allowed values:</u> automated, manual, unknown <i>This is particularly useful for camera Counters, where the data collected by the same device (video) may be processed by human review or automated algorithms.</i>
start_datetime	String	Yes	Date/time when counter was deployed (ISO 8601 datetime format).
end_datetime	String	No	Date/time the counter was removed (ISO 8601 datetime format). <i>Should not be included if counter is currently deployed.</i>
tags	Object	No	Custom metadata that is associated with the deployment.

5.6 count_records.csv

Each record in the Count Records CSV file represents a single Count Record, which is defined by the following columns (attributes).

Table 5-8: Count Record Columns (CSV)

Field Name	Type	Req'd?	Description & Allowed Values
deployment_id	String	Yes	Must reference a valid site in <code>deployments.geojson</code> .
flow_id	String	Yes	Must reference a valid counter in <code>flows.geojson</code> .
start_time	String	Yes	The start timestamp of this specific interval (ISO 8601). <i>Must fall within the Deployment start_time and end_time.</i>
interval_minutes	Integer	Yes	Duration of count interval in minutes.
count	Integer	Yes	The observed volume. Must be ≥ 0 . <i>Note: Use 0 to indicate "System active, no traffic observed." Do not omit records unless the system was down.</i>
sub_mode	String	No	A more granular categorization of the travel mode. (E.g., if Flow's travel mode is "bicycle", this could specify "e-bike")
quality_flag	String	No	Flag indicating data quality. <u>Allowed Values:</u> Enumerated in Data Quality Table . <i>May be omitted if data has not gone through QA/QC.</i>

6. Examples

This section demonstrates how the ATCS standard is applied to common data collection environments, ranging from simple trail segments to complex intersections. The goal is to bridge the gap between the attribute definitions in Section 5 and real-world implementation.

To assist with clarity, each example is presented using the following components:

1. **Scenario Description:** A brief description of the physical location and the specific traffic counting logic being applied.
2. **Visual & Logical Summary:** A diagram illustrating the physical setup (**Red** elements represent Site Logic; **Black** elements represent Flow Logic) paired with a table of selected attributes.

Note: These tables are conceptual summaries designed to highlight critical logic (e.g., how a specific Heading defines a Flow). They do not reflect the actual file structure (particularly for GeoJSON files like Sites and Flows) and exclude many attributes for readability. Please refer to the actual reference files to see how the ATCS specification is implemented for each example.

3. **Reference Files:** A single, comprehensive ATCS data package (ATCS.zip) containing all the examples is available for download (see link below). By including multiple scenarios in one package, the standard is able to manage heterogeneous data sources simultaneously.

Note: These examples prioritize the definition of **Sites** and **Flows**, as these represent the most complex logical relationships in the standard. Files for **Deployments**, **Counters**, and **Count Records** are included in the package to demonstrate correct formatting but strictly utilize synthetic data.

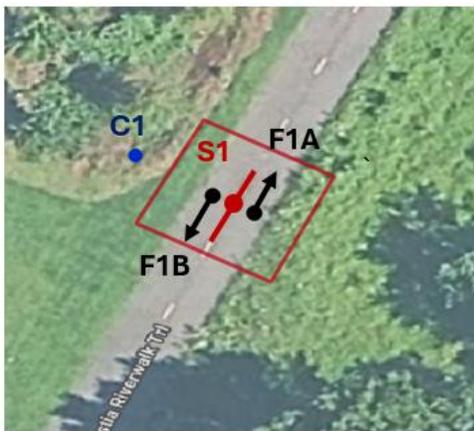
Download the example ATCS package (schema v1.0) [HERE](#)

Naming Convention: Because all the example Sites and Flows are stored together in the same ATCS package, Site, Flow, and Counter IDs were created to include the Example number to make it easy to sort by example. For instance, the Site in Example 1 is called S1 (“Site for example 1”), all Flows start with F1 (e.g., F1A, F1B, ...), the Counter is called C1, the Deployment is D1.

6.1 Example Path Segment

This example, shown visually in Figure 6-1, represents the most basic counting scenario: a **Screenline** count along a **Path Segment**. This scenario consists of a single Site (**S1**), two directional Flows (**F1A** and **F1B**), and one Counter (**C1**) deployed to the site to collect data.

- **Site Definition:** The Site is defined with a `base_type` of `segment` and a `facility_class` of `path`. This tells the system that the count occurs on a linear stretch of trail that is separate from the roadway network. It is defined spatially by a bounding polygon and its Site Diagram. The Site Diagram consists of a point plus a bearing, communicating the orientation of the path.
- **Flow Definitions:**
 - **F1A:** Represents bicycle traffic traveling at a heading of 15 degrees (NE).
 - **F1B:** Represents bicycle traffic traveling at a heading of 195 degrees (SW).



Site Properties (selected)

site_id	S1
base_type	segment
facility_class	path

Flow Properties (selected)

flow_id	F1A	F1B
site_id	S1	S1
count_type	screenline	screenline
travel_mode	bicycle	bicycle
heading	15	195
is_bidirectional	false	false
facility_type	shared_use_path	shared_use_path

Fig 6-1: Path Segment (Screenline Count)

Note the following information when reviewing this scenario in the accompanying ATCS files. These comments focus in particular on Sites and Flows – the two critical files for describing the site context and counting movements.

- Site S1's geometry (a geometry object within `sites.geojson`) defines the spatial extent of the study area. This corresponds to the red polygon on Figure 6-1.

```

"geometry": {
  "type": "Polygon",
  "coordinates": [[
    [ -76.969163401755665, 38.893384844760995 ],
    [ -76.969099978458729, 38.893356413627885 ],
    [ -76.969070296742416, 38.893407472463075 ],
    [ -76.969131533029113, 38.893434810091058 ],
    [ -76.969163401755665, 38.893384844760995 ]
  ]]
}

```

- Site S1's `site_diagram` (a property object within `sites.geojson`) defines the overall orientation of the roadway through a reference point and bearing. This corresponds to the red point and line on Figure 6-1.

```
"site_diagram": {
  "reference_point": [-76.96911591, 38.89339509],
  "bearing": 15
}
```

- Flows F1 and F2 each have a point geometry (see `geometry` object within `flows.geojson`) representing the screenline counting location. These points correspond to the black points on Figure 6-1. Note that GIS software will not automatically produce arrows like are shown in Figure 6-1, but the Flow's point location combined with the heading and bi-directionality flag (`heading` and `is_bidirectional` fields in the `properties` object) communicate this information.

```
"geometry": {
  "type": "Point",
  "coordinates": [ -76.969107953949077, 38.893390064939517 ]
}
```

- As described in Section 5, the definitions for F1 and F2 include attributes common to all Flows (Table 5-2) and other attributes specific to screenline counts (Table 5-3). All these flow attributes are stored in the property object of `flows.geojson`. Note that only Segment-specific flow attributes are included in this example (e.g., `leg` is omitted here because the Site is a segment, not an intersection).
 - *Validation Check:* Note that Flow F1 and F2 represent *directional* movements with headings 15 and 195, respectively, which align with the orientation of the segment (Site Diagram has bearing of 15 degrees).
- An optional attribute included in the F1 and F2 definitions is `map_ref`, which can be used to link the Flow to an external map reference. This example demonstrates how to link to an OpenStreetMap way (segment).

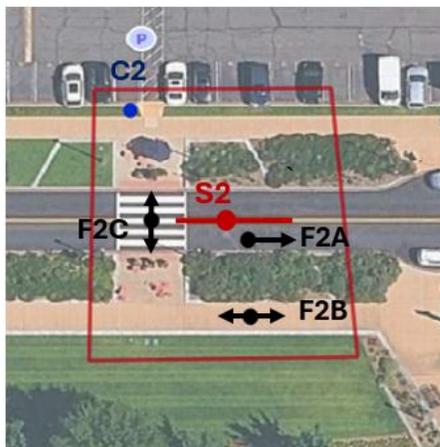
```
"map_refs": {
  "osm": {
    "id": "102419446",
    "version": "2025-12-03",
    "uri": "https://www.openstreetmap.org/way/102419446"
  }
}
```

- Counter C1 (a tube counter defined in `counters.csv`) was deployed to a specific lat/lon at Site S1 for a specified time period, defined in `deployments.geojson`. The C1 deployment location corresponds to the blue point in Figure 6-1. Note that this location is adjacent to the segment, representing where the control device was placed.

6.2 Example Road Segment

This example, shown visually in Figure 6-2, represents another common counting scenario: **Screenline** and **Crossing** counts on a **Road Segment**. This represents a step up in complexity relative to the first example, as there are multiple parallel facilities on the segment (general purpose lanes and sidewalks), and flows include both screenline counts (in the direction of the segment) and crossing counts (perpendicular, or crossing the segment). This scenario consists of a single Site (**S2**), three Flows (**F2A**, **F2B**, **F2C**), and one Counter (**C2**) deployed to the site to collect data.

- **Site Definition:** The Site is defined with a `base_type` of segment and a `facility_class` of road. This tells the system that the count occurs on a linear stretch of road, which may have parallel facilities and crossings. It is defined spatially by a bounding polygon and its Site Diagram consisting of a point plus a bearing to communicate the orientation of the road.
- **Flow Definitions:**
 - **F2A:** Represents bicycle traffic on `general_lanes` traveling at a heading of 90 degrees (EB). This is a screenline count along the segment.
 - **F2B:** Represents pedestrian traffic traveling on sidewalk at a heading of 90 or 270 degrees (combined EB/WB, because `is_bidirectional`). This is a screenline count along the segment.
 - **F2C:** Represents pedestrian traffic traveling on a crosswalk at a heading of 0 or 180 degrees (combined NB/SB, because `is_bidirectional`). This is a crossing count that reflects a movement perpendicular to the segment.
 - **Facility Type and Facility Side:** Each flow's `facility_type` is specified separately because Roads can have multiple types of facilities. Facility Side is needed in **F2B** to clarify that the sidewalk is on the south (S) side of the road.



Site Properties (selected)

<code>site_id</code>	S2
<code>base_type</code>	segment
<code>facility_class</code>	road

Flow Properties (selected)

<code>flow_id</code>	F2A	F2B	F2C
<code>site_id</code>	S2	S2	S2
<code>count_type</code>	screenline	screenline	crossing
<code>travel_mode</code>	bicycle	pedestrian	pedestrian
<code>heading</code>	90	90	0
<code>is_bidirectional</code>	false	true	true
<code>facility_type</code>	general_lanes	sidewalk	crosswalk
<code>facility_side</code>		S	

Fig 6-2: Road Segment (Screenline and Crossing Counts)

Note that the counting movements chosen for this example represent only a subset of the possible flows; they are illustrative of the types of flows that may be measured and not intended to enumerate every possibility.

As in 6.1, the geojson files will have additional attributes reflecting geometry, the site diagram and other optional attributes such as a map reference, and appropriate entries in the counters and deployments files. The details are omitted here but are included in the linked ATCS example package.

6.3 Example Path Intersection

This example, shown visually in Figure 6-3, represents another step up in complexity: a **Path Intersection** with both **Turning Movement** and **Screenline** counts. It consists of a Site (**S3**), three Flows (**F3A**, **F3B**, **F3C**), and one Counter (**C3**) deployed to the site to collect data.

- **Site Definition:** The Site is defined with a `base_type` of `intersection` and a `facility_class` of `path`. This tells the system that the count occurs at the intersection of multiple path segments. It is defined spatially by a bounding polygon and has a Site Diagram that describes the orientation of each leg relative to the center of the intersection.
- **Flow Definitions:**
 - **F3A:** Represents bicycle traffic traveling from leg L1 to L2 (turning_movement count).
 - **F3B:** Represents bicycle traffic traveling from leg L1 to L3 (turning_movement count).
 - **F3C:** Represents bicycle traffic traveling along leg L3 at a heading of 245 degrees (SW, screenline count).

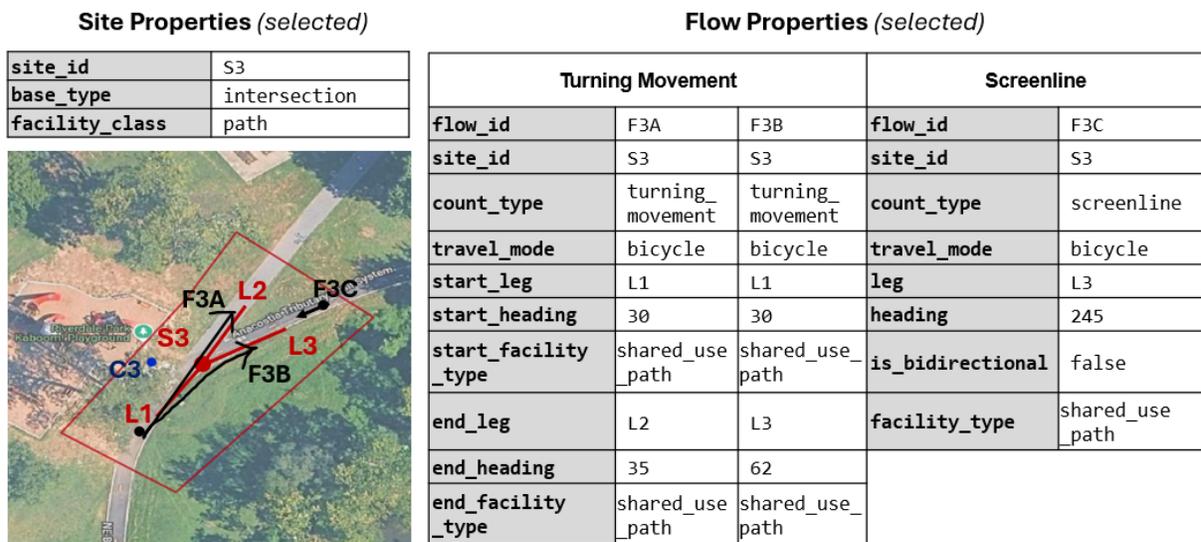


Fig 6-3: Path Intersection (Turning Movement and Screenline Counts)

Note the following information when reviewing this scenario in the accompanying ATCS files.

- Site S3's `site_diagram` (a property object within `sites.geojson`) defines the overall orientation of the intersection. It defines a reference point in the center of the intersection and labels plus corresponding bearings (measuring the angle relative to the center point) for each leg. This data structure corresponds to the red point and labeled legs in Figure 6-3.

```
"site_diagram": {
  "reference_point": [-76.92435510, 38.96314056],
  "legs": [
    {"id": "L1", "bearing": 220},
    {"id": "L2", "bearing": 35},
    {"id": "L3", "bearing": 62}
  ]
}
```

- Flows F3A, F3B, and F3C (defined in `flows.geojson`) have several common attributes (`flow_id`, `site_id`, `count_type`, `travel_mode`) and others that depend on the `count_type` (turning_movement vs screenline)
 - Turning movement counts require information about the start and end location, including the start and end leg labels (which must be defined in the Site Diagram).
 - Screenline counts require specifying a specific intersection leg (which must be defined in the Site diagram)
- Each Flow's geometry (defined in the geometry object of `flows.geojson`) is either the count location (for screenline counts such as F3C) or the location where the count starts (for turning movement counts such as F3A and F3B). For turning movement counts, the end location is stored as additional latitude/longitude coordinates within the `flows.geojson` properties object.

```
"properties": {
  "flow_id": "F3A",
  ...
  ...
  "end_leg": "L2",
  "end_heading": 36,
  "end_facility_type": "shared_use_path",
  "end_longitude": -76.9242946,
  "end_latitude": 38.9632254
}
```

6.4 Example Road Intersection

This example, shown visually in Figure 6-4, is a **Road Intersection** with both **Turning Movement** and **Crossing** counts. It consists of a Site (**S4**), three Flows (**F4A**, **F4B**, **F4C**), and one Counter (**C4**) deployed to the site to collect data. This is similar to the prior example, but because the Site's Facility Class is Road (i.e., all legs of the intersections are Roads), additional attributes are required to describe the Flows.

- **Site Definition:** The Site is defined with a `base_type` of `intersection` and a `facility_class` of `road`. This tells the system that the count occurs at the intersection of multiple road segments. It is defined spatially by a bounding polygon and has a Site Diagram that describes the orientation of each leg relative to the center of the intersection.
- **Flow Definitions:**
 - **F4A:** Represents bicycle traffic traveling from leg L1 to L2 (turning_movement count).
 - **F4B:** Represents bicycle traffic traveling from leg L1 to L3 (turning_movement count).
 - **F3C:** Represents pedestrian traffic crossing leg L4 at a heading of 0 and 180 degrees (crossing count).

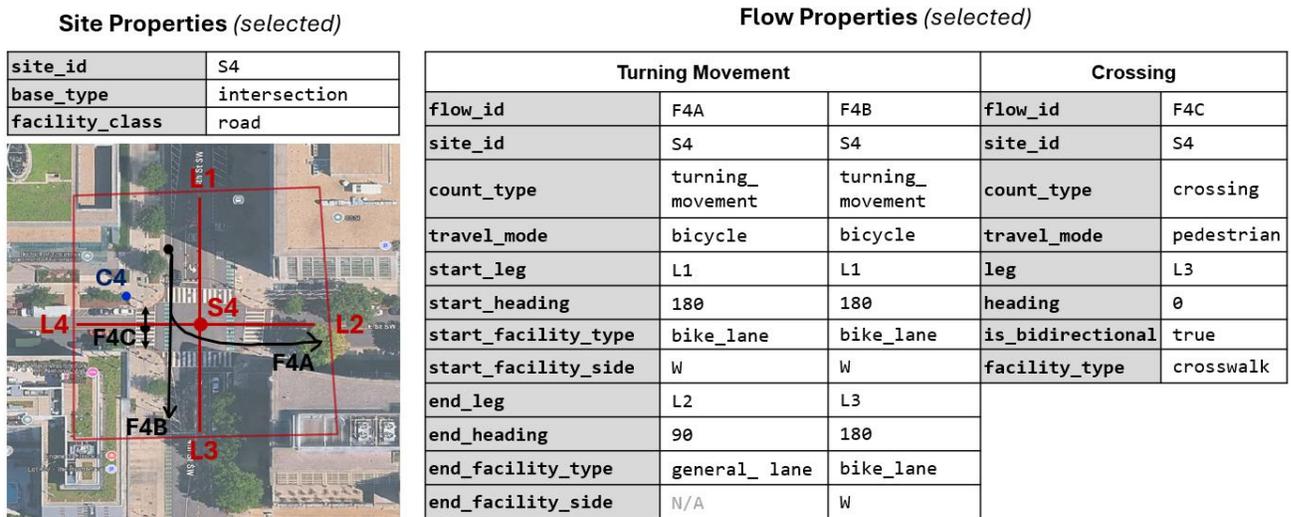


Fig 6-4: Road Intersection (Turning Movement and Crossing Counts)

Note that because the `facility_class` is `road`, each leg of the intersection can have multiple parallel facilities. For example, the North leg of the intersection, labeled L1 in Fig 6-4, has two sidewalks (W and E sides), two bike lanes (W and E sides), and general-purpose lanes. When defining flows on (or between pairs of) road segments, the facility type and facility side need to be specified (in cases where paired facilities exist). For example, Flow F4A captures the turning movement from a `bike_lane` (W side) on leg L1 to a `general_lane` on leg L2.

Please also note that the Flows here are not meant to be exhaustive; while these illustrate common counting movements, many more could theoretically be defined.

6.5 Example Hybrid Intersection

This example, shown visually in Figure 6-5, is a **Hybrid Intersection with Turning Movement** counts. It consists of a Site (**S5**), three Flows (**F5A**, **F5B**, **F5C**), and one Counter (**C5**) deployed to the site to collect data. The main point of this example is to show how to define a site diagram for a Hybrid site, where a road intersects with a path.

- **Site Definition:** The Site is defined with a `base_type` of `intersection` and a `facility_class` of `hybrid`. This tells the system that the count occurs at the intersection of both road and path segments. It is defined spatially by a bounding polygon and has a Site Diagram that describes the orientation of each leg relative to the center of the intersection. Additionally, the Site Diagram reports the `facility_class` for each leg.
- **Flow Definitions:**
 - **F5A:** Represents pedestrian traffic from leg L1 to L2 (turning_movement count).
 - **F5B:** Represents pedestrian traffic from leg L1 to L3 (turning_movement count).
 - **F5C:** Represents pedestrian traffic from leg L1 to L4 (turning_movement count).

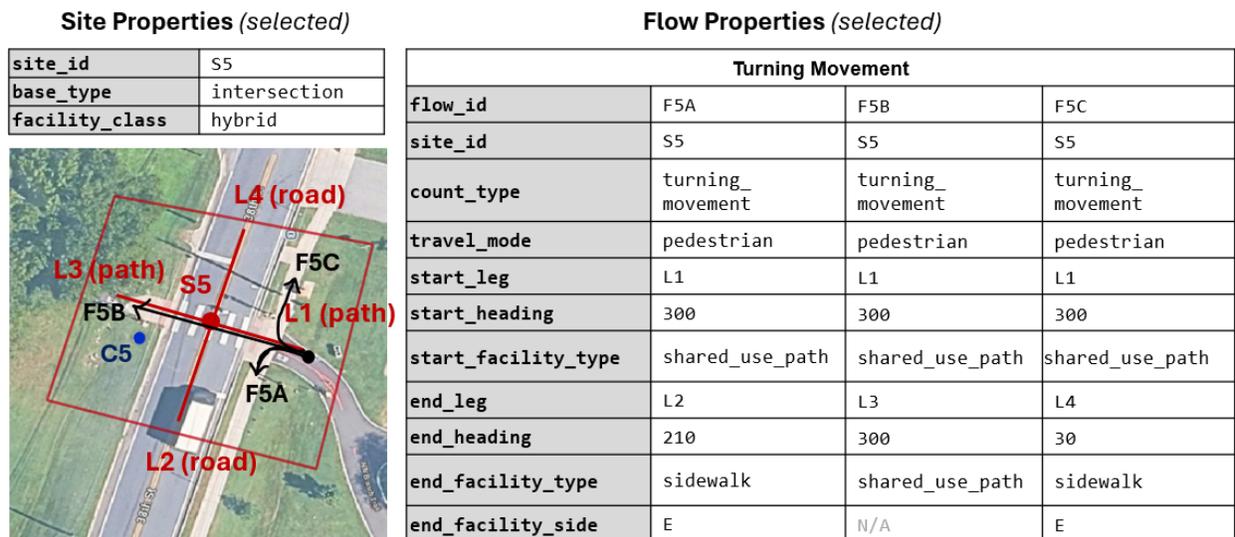


Fig 6-5: Hybrid Intersection (Turning Movement Counts)

Note that each of these flows starts on a Path segment (L1), so the `facility_type` is automatically `shared_use_path` and the `facility_side` does not need to be populated. Two of the flows (F5A and F5C) end on road segments (i.e., legs L2 and L4, where `facility_class` has a value of `road`). In these cases, the `facility_type` needs to be specified (`sidewalk`) to distinguish between general purpose lanes on the roadway. To be extra clear, the `facility_side` is also specified to indicate that the sidewalk is on the East side of the road, although this is not strictly necessary because this road only has only one sidewalk.

The main thing to note about the ATCS files for this example is how the Site Diagram (in the properties object of `sites.geojson`) is defined. Because the Site has a `facility_class` of `hybrid` (i.e., it represents a junction between path and road segments), each leg's `facility_class` must be explicitly defined in the SD.

```
"site_diagram": {
  "reference_point": [-76.9561328,38.9498203],
  "legs": [
    {"id": "L1", "bearing": 120, "facility_class": "path"},
    {"id": "L2", "bearing": 210, "facility_class": "road"},
    {"id": "L3", "bearing": 300, "facility_class": "path"},
    {"id": "L4", "bearing": 30, "facility_class": "road"}
  ]
}
```

6.6 Example Complex

This example, shown visually in Figure 6-6, is a **Complex** Site capturing a **Screenline** count. It consists of a Site (**S6**), one Flow (**F6**), and one Counter (**C6**) deployed to the site to collect data. The main point of this example is to show how to define a site diagram for a site that does not fit neatly into the simple geometric models for segments or intersections. In this case, the Site S6 is an area encompassing the staircase leading to the Lincoln Memorial in Washington, D.C., and the Flow is a screenline counting movement counting pedestrians climbing the stairs.

Site Definition: The Site is defined with a `base_type` of `complex`, meaning that it can't be easily classified as either a segment or intersection. This also means that the `facility_class` is undefined. Because it is Complex, there will be no Site Diagram to model the location, and any counting movement (Flow) will need to be explicitly defined to communicate the intent of the count. It is defined spatially by a bounding polygon.

In this example, the Site is located at the staircase outside the Lincoln memorial in Washington, DC. This location is on the edge of a plaza and is not obviously a road/path segment or the intersection of segments.

- **Flow Definitions:**
 - **F6A:** Represents pedestrian traffic ascending the staircase to the Lincoln Memorial at 270 degrees (screenline count).

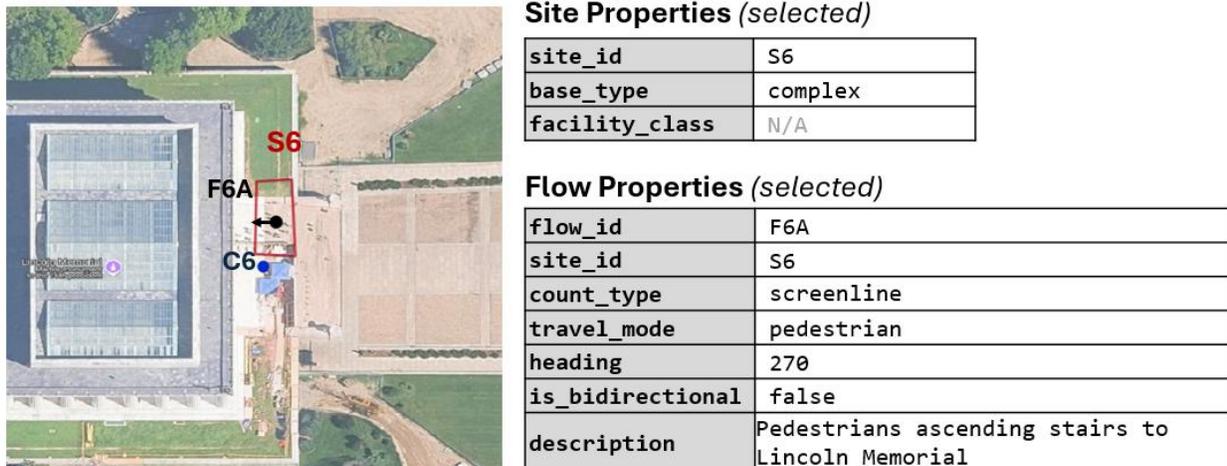


Fig 6-6: Complex Site (Screenline Counts)

The Main thing to note in the ATCS files for this example is that `sites.geojson` has no `site_diagram` or `facility_class` for Site S6 because it is Complex. Likewise, the `flows.geojson` is free to use whatever fields are relevant to describe the intent of the counting movement and is not constrained by the Site logic. In addition to required location information (stored in the geometry object), a mandatory `description` field is used to describe the counting scenario.

Appendix A: Enumerated Values

Site – Base Type

- segment
- intersection
- complex

Site – Facility Classification

- road
- path
- hybrid

Site – Intersection Control

- signalized
- all_way_stop
- two_way_stop
- roundabout
- yield
- uncontrolled
- other

Flow – Count Type

- screenline
- turning_movement
- crossing

Flow – Travel Mode

- non_motorized
- pedestrian
- bicycle
- scooter
- other

Flow – Facility Type

- right_of_way
- general_lane
- bike_lane
- separated_bike_lane
- shoulder
- shared_use_path
- crosswalk

Flow – [Facility Side](#)

- N
- NE
- E
- SE
- S
- SW
- W
- NW
- C

Counter – [Technology Type](#)

- inductive_loop
- passive_infrared
- active_infrared
- pneumatic_tube
- piezoelectric
- radar
- magnetometer
- video_analytics
- manual
- lidar
- other

Count Record – [Quality Flag](#)

- valid
- valid_atypical
- suspect
- invalid