Transportation Systems Management & Operations

Interface Control Document for:

ITSIQA Turning Movement Counts Application Programming Interface (TMC API)

Version: ***1.2***

Approval date: ***12/29/2018***

DOCUMENT CONTROL PANEL		
File Name:	ITSIQA - TMC API - Interface Control Document - 2018-12-29.docx	
File Location:		
Version Number:	1.2	
	Name	**Date**
Created By:	Natesh Kirpalani, AECOM	10/08/2018
Reviewed By:	John Hope, AECOM	10/14/2018
	Natesh Kirpalani, AECOM	11/09/2018
	Clay Packard, VHB	11/12/2018
	Katie King, Metric Engineering	11/12/2018
	Natesh Kirpalani, AECOM	12/29/2018
Modified By:	Natesh Kirpalani, AECOM	10/08/2018
	John Hope, AECOM	11/08/2018
	John Hope, AECOM	11/12/2018
	John Hope, AECOM	12/29/2018
Approved By:		

Table of Contents

1. Overview 1
   - *1.1 Document Overview* *1*
   - *1.2 System Overview* *1*
2. Connecting to API 2
   - *2.1 Success Response* *2*
   - *2.2 Error Response* *2*
   - *2.3 Output Request Methods* *3*
     - 2.3.1 GetIntersectionNetwork 4
     - 2.3.2 GetIntersectionNetworkFilteredByRadius 6
     - 2.3.3 GetIntersectionNetworkFilteredByCounty 6
     - 2.3.4 GetCountyList 7
     - 2.3.5 GetVehicleClassList 7
   - *2.4 Input Request Methods* *8*
     - 2.4.1 AllDataByApproach 9
     - 2.4.2 AllDataByLane 10
     - 2.4.3 BicycleCounts 11
     - 2.4.4 ClassificationAll 11
     - 2.4.5 ClassificationLeftTurn 12
     - 2.4.6 ClassificationRightTurn 12
     - 2.4.7 ClassificationThroughTurn 12
     - 2.4.8 ClassificationUTurn 13
     - 2.4.9 CountArrival 13
     - 2.4.10 FlowRate 14
     - 2.4.11 LightTimingPerApproach 14
     - 2.4.12 LightTimingPerLane 14
     - 2.4.13 Occupancy 15
     - 2.4.14 PeakHourFactor 15
     - 2.4.15 PedCounts 15
     - 2.4.16 PercentArrival 16
     - 2.4.17 RightTurnOnRed 16
     - 2.4.18 TurnMovements 16
     - 2.4.19 TurnMovementsOnLight 17
     - 2.4.20 TurnMovementsOnLightWithSpeed 17
     - 2.4.21 TurnMovementsWithSpeed 18

List of Tables

Table 1: Valid County Names 3
Table 2: Valid Vehicle Classifications 4
Table 3: Intersection Configuration Fields 4
Table 4: County Configuration Fields 7
Table 5: Vehicle Classification Configuration Fields 7
Table 6: Valid Light States 8
Table 7: Valid Bicycle and Pedestrian Count Heading Values 9

List of Figures

Figure 1: High Level Description of ITSIQA TMC API Software 1

List of Acronyms and Abbreviations

API ..... Application Programming Interface
FDOT ..... Florida Department of Transportation
ICD..... Interface Control Document
ITSIQA .....Intelligent Transportation Systems Integration Quality and Analysis
TMC.....Turning Movement Count
WSDL .....Web Service Definition Language
XML…….. Extensible Markup Language

1. Overview

1.1 Document Overview

The purpose of this document is to provide an Interface Control Document (ICD) for Intelligent Transportation Systems Integration Quality and Analysis (ITSIQA) Application Programming Interface (API) for receiving Turning Movement Count (TMC) data.

1.2 System Overview

The ITSIQA TMC API utilizes Web Service Definition Language (WSDL) software to receive incoming TMC data and/or respond to requests with ITSIQA intersection configuration information. Per standard WSDL protocol, data is formatted in XML with parameters specified in this document.

The following diagram describes the data flow at a high level.

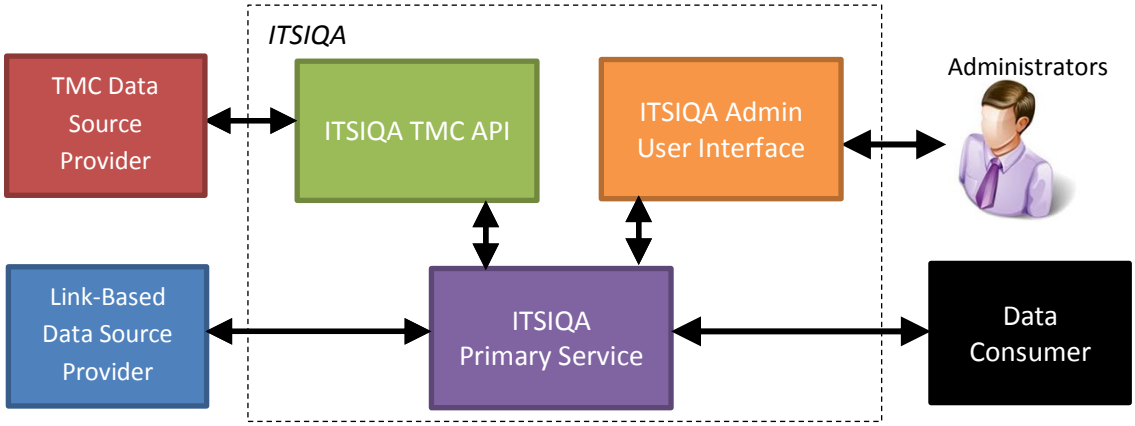

Figure 1: High Level Description of ITSIQA TMC API Software

TMC Data Source Providers are the primary users of the ITSIQA TMC API. TMC Data Source Providers calls WSDL methods to request intersection configuration information housed in the ITSIQA configuration database. The ITSIQA TMC API retrieves requested data from the ITSIQA database and the ITSIQA Primary Service and responds with this data to the TMC Data Source Provider. Other WSDL methods are available to push turning movement count information from TMC Data Source Providers to the ITSIQA TMC API and then onto the ITSIQA Primary Service. ITSIQA consolidates all receives data and produces both link-based and TMC output to Data Consumers.

2. Connecting to API

The ITSIQA TMC API is accessible as a standard WSDL web service via a Universal Resource Location (URL) on port 80. TMC Data Source Providers can access methods using the following path:

http://{ITSIQA IP}/TMCAPI

The ITSIQA TMC API does not maintain persistent sessions with TMC Data Source Providers. TMC Data Source Providers initiate all requests. Once the ITSIQA TMC API responds to a request, the transaction is logged, data received is passed onto the ITSIQA Primary Service, and then the session is closed and forgotten.

A TMC API tester application is available with the source code to assist developers in connecting to and calling methods through the API.

2.1 Success Response

Success responses are returned for input request methods that are successfully received and processed. The format of the response should include the following:

```
<NameOfMethod>
        <TimeStamp>01/01/1900 3:39:41 PM</TimeStamp>
        <Status>Success</Status>
        <Detail>Success</Detail >
</NameOfMethod>
```

Note that the actual responses will use the name of the method rather than "NameOfMethod". The given timestamp is the date and time in local time that the method was processed.

When output request methods are successfully received and processed, the response will be the requested data. Section 2.3 of this document describes the output XML from the output request methods.

2.2 Error Response

Error responses are returned for input or output request methods that are not successfully received or processed. The format of the response should include the following:

```
<NameOfMethod>
        <TimeStamp>01/01/1900 3:39:41 PM</TimeStamp>
        <Status>Error</Status>
        <Detail>Some sort of error details</Detail >
</NameOfMethod>
```

Note that the actual responses will use the name of the method rather than "NameOfMethod". The given timestamp is the date and time in local time that the method was processed. The value of Detail will include a message in English that describes the nature of the error. Specific data fields and values in question will be provided in Detail if applicable.

2.3 Output Request Methods

The following output request methods are available from the ITSIQA TMC API.

- **GetIntersectionNetwork()** – Returns configuration information for all intersections contained with ITSIQA.

- **GetIntersectionNetwork(Latitude, Longitude, Radius)** – Returns configuration information for intersections within the circle defined by the given Latitude and Longitude coordinates in degrees and radius in miles.

- **GetIntersectionNetwork(CountyList)** – Returns configuration information for intersections within the given comma-delimited list of counties. Table 1 defines all valid county names.

- **GetCountyList()** – Returns complete list of valid county names, as noted in Table 1.

- **GetVehicleClassList()** – Returns complete list of valid vehicle classification bins per current FDOT standards, as noted in Table 2.

Table 1: Valid County Names

Alachua	Franklin	Lee	Pinellas
Baker	Gadsden	Leon	Polk
Bay	Gilchrist	Levy	Putnam
Bradford	Glades	Liberty	Santa Rosa
Brevard	Gulf	Madison	Sarasota
Broward	Hamilton	Manatee	Seminole
Calhoun	Hardee	Marion	St. Johns
Charlotte	Hendry	Martin	St. Lucie
Citrus	Hernando	Miami-Dade	Sumter
Clay	Highlands	Monroe	Suwannee
Collier	Hillsborough	Nassau	Taylor
Columbia	Holmes	Okaloosa	Union
DeSoto	Indian River	Okeechobee	Volusia
Dixie	Jackson	Orange	Wakulla
Duval	Jefferson	Osceola	Walton
Escambia	Lafayette	Palm Beach	Washington
Flagler	Lake	Pasco	

Table 2: Valid Vehicle Classifications

Vehicle Classification Bin	Classification Definition
Bin1	Vehicles of lengths up to 10 feet.
Bin2	Vehicles of lengths between 10 and 24 feet.
Bin3	Vehicles of lengths between 10 and 54 feet.
Bin4	Vehicles of lengths greater than 54 feet.
Bin5	*Not used.*
Bin6	*Not used.*
Bin7	*Not used.*
Bin8	*Not used.*

The following sections describe each output request method in more detail.

2.3.1 GetIntersectionNetwork

This method provides all intersection configuration that exists within ITSIQA. This information does not change frequently. It is recommended that this function not be called more than once per day.

Intersection configuration data is provided in XML format. Table 3 describes the fields reported.

Table 3: Intersection Configuration Fields

Data Field	Data Description	Example Value
Intersections / TimeStamp (attribute)	Date and time when configuration was last updated, in local time.	6/18/2018 8:52:39 AM
Intersections / Intersection	XML element containing all configuration information related to each intersection. The number of intersections varies depending on the list of counties configured for the Data Output Stream and the enabled intersections within each county.	<Intersection>… </Intersection >
Intersections / Intersection / ID	Unique alpha-numeric identifier for the reported intersection. External systems should not make any assumptions of the format of this identifier other than it is unique to the reported intersection.	ORL-00254
Intersections / Intersection / Location	XML element containing Latitude and Longitude values, indicating the geographic center of the intersection.	<Latitude> 29.285916 </Latitude> <Longitude> -81.083468 </Longitude>

Data Field	Data Description	Example Value
Intersections / Intersection / County	Name of the county where the intersection is located. An intersection has exactly one county associated with it. The value for this field is one of the county names listed in Table 4.	Orange
Intersections / Intersection / Approaches / Approach	XML element containing all configuration information related to each approach within an intersection. The number of approaches varies with each intersection, although a common two-road intersection typically has four approaches.	<Approach>… </Approach >
Intersections / Intersection / Approaches / Approach / ApproachID	Unique alpha-numeric identifier for the reported approach. External systems should not make any assumptions of the format of this identifier other than it is unique to the reported approach.	ORL-00254-00001-N
Intersections / Intersection / Approaches / Approach / Road	Name of the reported roadway on which the approach is located. Roadway names are named using standard formatting for interstates, US roads, state roads, and county roads, using the format I-XXX, US-XXX, SR-XXX, and CR-XXX, respectively, where XXX is the integer associated with road. If a road has multiple names, only one is used. All roads are consistently reported. All approaches on I-4, for example, will have a Road value of I-4.	I-4 SR-408 US-441
Intersections / Intersection / Approaches / Approach / RoadID	Unique identifier for the roadway on which the approach is located.	1517
Intersections / Intersection / Approaches / Approach / MileMarker	Optional field that provides a mile marker closest to the intersection. If populated, this value may range from 0 to 65,000 and may have any number of significant digits.	10 23.1 1.5542
Intersections / Intersection / Approaches / Approach / Direction	Direction of travel for the approach. Each approach has exactly one reported direction of travel. The only values for this field include one of the following: Northbound, Southbound, Eastbound, or Westbound	Northbound
Intersections / Intersection / Approaches / Approach / Lanes / Lane	XML element containing all configuration information related to each lane within an approach. The number of lanes varies with each approach, although there must be at least one.	<Lane>… </Lane>

Data Field	Data Description	Example Value
Intersections / Intersection / Approaches / Approach / Lanes / Lane / LaneID	Unique alpha-numeric identifier for the reported lane. External systems should not make any assumptions of the format of this identifier other than it is unique to the reported lane.	ORL-00254-00001-N-1
Intersections / Intersection / Approaches / Approach / Lanes / Lane / LaneTypeIDs	Comma-delimited list of identifiers that describe the allowable movement through the lane. There may be a variable number of identifiers reported in this field, although there must be at least one. All valid identifiers include the following: Left, Through, Right, UTurn	Through,Right
Intersections / Intersection / Links	XML element containing all links associated with the intersection. There can be any number of links in this list, however, there are typically four, six, or eight links associated with each intersection.	<Links>… </Links>
Intersections / Intersection / Links / LinkID	A unique Link ID of a link associated with the intersection. This identifier should match a link reported from the LinkConfig file exactly. This field may be blank if a link is not associated.	102+10817

2.3.2 GetIntersectionNetworkFilteredByRadius

This method provides intersection configuration that exists within the circle defined by the given latitude and longitude in degrees and radius in miles. All three parameters should be provided as type Double with 64-bit precision. For example, the following call will retrieve all configured intersections within a circle with the center located at 28.531122 latitude and -81.397117 longitude and radius of 5.25 miles:

GetIntersectionNetworkFilteredByRadius(28.531122, -81.397117, 5.25)

This information does not change frequently. It is recommended that this function not be called more than once per day.

Intersection configuration data is provided in XML format. Table 2 describes the fields reported.

2.3.3 GetIntersectionNetworkFilteredByCounty

This method provides intersection configuration that exists within the given comma-delimited list of counties. Parameter should be provided as a String list of county names, each separated by

a county name. All valid county names are defined in Table 1. For example, the following call will retrieve all configured intersections within Orange, Seminole, and Lake Counties.

GetIntersectionNetworkFilteredByCounty("Orange,Seminole,Lake")

The following call will retrieve all configured intersections within Orange County only.

GetIntersectionNetworkFilteredByCounty("Orange")

This information does not change frequently. It is recommended that this function not be called more than once per day.

Intersection configuration data is provided in XML format. Table 3 describes the fields reported.

2.3.4 GetCountyList

This method provides a valid list of counties. Table 4 defines the parameters returned from this method.

Table 4: County Configuration Fields

Data Field	Data Description	Example Value
Counties / TimeStamp (attribute)	Date and time when configuration was last updated, in local time.	6/18/2018 8:52:39 AM
Counties / County	XML element defining county information.	<County>… </County>
Counties / County / CountyName	Valid county name.	Orange
Counties / County / FDOTDistrict	Name of FDOT District.	District 5

2.3.5 GetVehicleClassList

This method provides a valid list of vehicle classifications, per FDOT standards. Table 5 defines the parameters returned from this method.

Table 5: Vehicle Classification Configuration Fields

Data Field	Data Description	Example Value
VehicleClassifications / TimeStamp (attribute)	Date and time when configuration was last updated, in local time.	6/18/2018 8:52:39 AM
VehicleClassifications / VehicleClass	XML element defining vehicle classification information.	<VehicleClass>… </VehicleClass>
VehicleClassifications / VehicleClass / BinIndex	Index of vehicle classification bin, starting with one (1).	1

Data Field	Data Description	Example Value
VehicleClassifications / VehicleClass / Used	True/False flag indicating if vehicle classification bin is used.	True
VehicleClassifications / VehicleClass / MinLength	Minimum length of vehicles (in feet) within vehicle classification bin. A value of -1 means there is no minimum length.	0
VehicleClassifications / VehicleClass / MaxLength	Maximum length of vehicles (in feet) within vehicle classification bin. A value of -1 means there is no maximum length.	10

2.4 Input Request Methods

Input request methods require TMC Data Source Providers to provide data using unique identifiers. TMC Data Source Providers should first request configuration information via one or more of the output request methods. Using the unique identifiers provided from these methods, TMC Data Source Providers can provide data using the input request methods.

It is recommended for TMC Data Source Providers to call input request methods exactly once per minute. For timing accuracy, providers should be synchronized to the same timing source as the server(s) hosting the ITSIQA software.

For all input data provided, reported data should be less than zero (such as -1) if no data is available for a specified field. For example, if TurnMovements is called but only LeftTurnCount, ThroughCount, and RightTurnCount values are known and UTurnCount is unknown, all counts should be zero or greater except for UTurnCount should be less than zero. It is valid to provide no data for all fields. In this case, all fields should have values less than zero. A reported value of zero for any data field is considered valid.

For all input request methods, the Provider must give a unique source identifier (SourceID) as an alpha-numeric string that identifies the provider and a date/time (TimeStamp) in local time for when the data applies. The Provider should always use the same SourceID for all data provided. ITSIQA does not regulate the SourceIDs so the Provider must ensure that their SourceID is unique. Close coordination with the ITSIQA administrator will help to ensure that a unique SourceID is selected for the Provider.

Tables 6 and 7 below are used for the input request methods described in the following sections.

Table 6: Valid Light States

Red
Green
Yellow
FlashingYellow

Table 7: Valid Bicycle and Pedestrian Count Heading Values

Forward
Backward

The following sections describe all input request methods from the ITSIQA TMC API.

2.4.1 AllDataByApproach

This method allows providing all approach-level TMC data. ApproachID must match the unique lane identifier reported from the ITSIQA TMS API via the GetIntersectionNetwork output request methods. Counts should be reported as integers by turning type. If no data exists for a turning type, Provider should provide the count value less than zero, such as -1. Input fields include the following:

- SourceID: Unique identifier for the provider.
- TimeStamp: Date/time (format MM/DD/YYYY HH:mm:ss) when data was last updated.
- ApproachID: Intersection Network identifier for the approach.
- TotalVolume: Total number of vehicles reported for approach.
- GreenOccupancy: Value reported in seconds of the time which vehicles occupied the lanes during the green light.
- RedOccupancy: Value reported in seconds of the time which vehicles occupied the lanes during the red light.
- Speed: Average report speed of vehicles in miles per hour.
- FlowRate: Numeric value that represents the calculation for flow rate. The calculation consists of the total volume divided by the addition of green and yellow time, then multiplied by 3600. This calculation is not performed on approaches that have a lower value of green and yellow time that is less than a configured value.
- SaturationFlowRate: Maximum flow rate in the past hour.
- PercentArrivalOnGreen: Percent of vehicles arriving during the green light.
- PercentArrivalOnRed: Percent of vehicles arriving during the red light.
- GreenTime: Number of seconds of the green light during the reporting period.
- YellowTime: Number of seconds of the yellow light during the reporting period.
- RedTime: Number of seconds of the red light during the reporting period.
- UnderutilizedGreen: Numeric value that represents the calculation for underutilized green. The calculation consists of the green time subtracted from the outcome of that turn type's volume multiplied by a configured value of 3600 to which the value is divided by the saturation flow.
- PedCountForward: Reported number of pedestrians that moved in the direction of the approach through the intersection.
- PedCountBackward: Reported number of pedestrians that moved in the reverse direction of the approach through the intersection.
- BicycleCountForward: Reported number of bicycles that moved in the direction of the approach through the intersection.
- BicycleCountBackward: Reported number of bicycles that moved in the reverse direction of the approach through the intersection.

- CountArrivalOnGreen: Count of vehicles arriving during the green light.
- CountArrivalOnRed: Count of vehicles arriving during the red light.
- PeakHourFactor: Calculated value that uses the volume of vehicles during the peak hour over the past 24 hours.
- RightTurnOnRed: Count of vehicles that turned right during a red light.

2.4.2 AllDataByLane

This method allows providing all lane-level TMC data. LaneID must match the unique lane identifier reported from the ITSIQA TMS API via the GetIntersectionNetwork output request methods. Counts should be reported as integers by turning type. If no data exists for a turning type, Provider should provide the count value less than zero, such as -1. Input fields include the following:

- SourceID: Unique identifier for the provider.
- TimeStamp: Date/time (format MM/DD/YYYY HH:mm:ss) when data was last updated.
- LaneID: Intersection Network identifier for the lane.
- TotalVolume: Total number of vehicles reported for lane.
- GreenOccupancy: Value reported in seconds of the time which vehicles occupied the lanes during the green light.
- RedOccupancy: Value reported in seconds of the time which vehicles occupied the lanes during the red light.
- Speed: Average report speed of vehicles in miles per hour.
- FlowRate: Numeric value that represents the calculation for flow rate. The calculation consists of the total volume divided by the addition of green and yellow time, then multiplied by 3600. This calculation is not performed on lanes that have a lower value of green and yellow time that is less than a configured value.
- SaturationFlowRate: Maximum flow rate in the past hour.
- PercentArrivalOnGreen: Percent of vehicles arriving during the green light.
- PercentArrivalOnRed: Percent of vehicles arriving during the red light.
- GreenTime: Number of seconds of the green light during the reporting period.
- YellowTime: Number of seconds of the yellow light during the reporting period.
- RedTime: Number of seconds of the red light during the reporting period.
- UnderutilizedGreen: Numeric value that represents the calculation for underutilized green. The calculation consists of the green time subtracted from the outcome of that turn type's volume multiplied by a configured value of 3600 to which the value is divided by the saturation flow.
- CountArrivalOnGreen: Count of vehicles arriving during the green light.
- CountArrivalOnRed: Count of vehicles arriving during the red light.
- PeakHourFactor: Calculated value that uses the volume of vehicles during the peak hour over the past 24 hours.
- RightTurnOnRed: Count of vehicles that turned right during a red light.

2.4.3 BicycleCounts

This method allows providing bicycle counts per intersection approach and heading. ApproachID must match the unique lane identifier reported from the ITSIQA TMS API via the GetIntersectionNetwork output request methods. Count values should be given as integers. If no data exists for a bicycle count, a value less than zero should be provided, such as -1. Input fields include the following:

- SourceID: Unique identifier for the provider.
- TimeStamp: Date/time (format MM/DD/YYYY HH:mm:ss) when data was last updated.
- ApproachID: Intersection Network identifier for the approach.
- Heading: Valid heading compared to the direction of the approach. Values are listed in Table 7. Forward is the same direction of the approach and Backward is the opposite direction of the approach.
- Count: Reported number of bicycles that moved through the intersection in the direction of the heading.

2.4.4 ClassificationAll

This method allows providing total vehicle counts by vehicle classification. LaneID must match the unique lane identifier reported from the ITSIQA TMS API via the GetIntersectionNetwork output request methods. Vehicle classifications must match district-defined classification. These classifications are provided from the output method GetVehicleClassList(). Vehicle classification counts should be given as integers. If no data exists for a vehicle classification bin, a value less than zero should be provided, such as -1. Input fields include the following:

- SourceID: Unique identifier for the provider.
- TimeStamp: Date/time (format MM/DD/YYYY HH:mm:ss) when data was last updated.
- LaneID: Intersection Network identifier for the lane.
- Bin1: Count of vehicles within classification bin 1.
- Bin2: Count of vehicles within classification bin 2.
- Bin3: Count of vehicles within classification bin 3.
- Bin4: Count of vehicles within classification bin 4.
- Bin5: Count of vehicles within classification bin 5.
- Bin6: Count of vehicles within classification bin 6.
- Bin7: Count of vehicles within classification bin 7.
- Bin8: Count of vehicles within classification bin 8.

2.4.5 ClassificationLeftTurn

This method allows providing Left Turn only vehicle counts by vehicle classification. LaneID and vehicle classification requirements are the same as the ClassificationAll method. Input fields include the following:

- SourceID: Unique identifier for the provider.
- TimeStamp: Date/time (format MM/DD/YYYY HH:mm:ss) when data was last updated.
- LaneID: Intersection Network identifier for the lane.
- Bin1: Count of vehicles within classification bin 1.
- Bin2: Count of vehicles within classification bin 2.
- Bin3: Count of vehicles within classification bin 3.
- Bin4: Count of vehicles within classification bin 4.
- Bin5: Count of vehicles within classification bin 5.
- Bin6: Count of vehicles within classification bin 6.
- Bin7: Count of vehicles within classification bin 7.
- Bin8: Count of vehicles within classification bin 8.

2.4.6 ClassificationRightTurn

This method allows providing Right Turn only vehicle counts by vehicle classification. LaneID and vehicle classification requirements are the same as the ClassificationAll method. Input fields include the following:

- SourceID: Unique identifier for the provider.
- TimeStamp: Date/time (format MM/DD/YYYY HH:mm:ss) when data was last updated.
- LaneID: Intersection Network identifier for the lane.
- Bin1: Count of vehicles within classification bin 1.
- Bin2: Count of vehicles within classification bin 2.
- Bin3: Count of vehicles within classification bin 3.
- Bin4: Count of vehicles within classification bin 4.
- Bin5: Count of vehicles within classification bin 5.
- Bin6: Count of vehicles within classification bin 6.
- Bin7: Count of vehicles within classification bin 7.
- Bin8: Count of vehicles within classification bin 8.

2.4.7 ClassificationThroughTurn

This method allows providing Through traffic only vehicle counts by vehicle classification. LaneID and vehicle classification requirements are the same as the ClassificationAll method. Input fields include the following:

- SourceID: Unique identifier for the provider.

- TimeStamp: Date/time (format MM/DD/YYYY HH:mm:ss) when data was last updated.
- LaneID: Intersection Network identifier for the lane.
- Bin1: Count of vehicles within classification bin 1.
- Bin2: Count of vehicles within classification bin 2.
- Bin3: Count of vehicles within classification bin 3.
- Bin4: Count of vehicles within classification bin 4.
- Bin5: Count of vehicles within classification bin 5.
- Bin6: Count of vehicles within classification bin 6.
- Bin7: Count of vehicles within classification bin 7.
- Bin8: Count of vehicles within classification bin 8.

2.4.8 ClassificationUTurn

This method allows providing UTurn only vehicle counts by vehicle classification. LaneID and vehicle classification requirements are the same as the ClassificationAll method. Input fields include the following:

- SourceID: Unique identifier for the provider.
- TimeStamp: Date/time (format MM/DD/YYYY HH:mm:ss) when data was last updated.
- LaneID: Intersection Network identifier for the lane.
- Bin1: Count of vehicles within classification bin 1.
- Bin2: Count of vehicles within classification bin 2.
- Bin3: Count of vehicles within classification bin 3.
- Bin4: Count of vehicles within classification bin 4.
- Bin5: Count of vehicles within classification bin 5.
- Bin6: Count of vehicles within classification bin 6.
- Bin7: Count of vehicles within classification bin 7.
- Bin8: Count of vehicles within classification bin 8.

2.4.9 CountArrival

This method allows providing count arrival lane-level TMC data. LaneID must match the unique lane identifier reported from the ITSIQA TMS API via the GetIntersectionNetwork output request methods. If no data exists for a turning type, Provider should provide the count value less than zero, such as -1. Input fields include the following:

- SourceID: Unique identifier for the provider.
- TimeStamp: Date/time (format MM/DD/YYYY HH:mm:ss) when data was last updated.
- LaneID: Intersection Network identifier for the lane.
- CountArrivalOnGreen: Count of vehicles arriving during the green light.
- CountArrivalOnRed: Count of vehicles arriving during the red light.

2.4.10 FlowRate

This method allows providing flow rate approach-level TMC data. ApproachID must match the unique lane identifier reported from the ITSIQA TMS API via the GetIntersectionNetwork output request methods. If no data exists for a turning type, Provider should provide the count value less than zero, such as -1. Input fields include the following:

- SourceID: Unique identifier for the provider.
- TimeStamp: Date/time (format MM/DD/YYYY HH:mm:ss) when data was last updated.
- ApproachID: Intersection Network identifier for the approach.
- FlowRate: Numeric value that represents the calculation for flow rate. The calculation consists of the total volume divided by the addition of green and yellow time, then multiplied by 3600. This calculation is not performed on approaches that have a lower value of green and yellow time that is less than a configured value.
- SaturationFlowRate: Maximum flow rate in the past hour.

2.4.11 LightTimingPerApproach

This method allows providing the length of time in seconds each light state was active over the previous 60 seconds on a per approach basis. All timing values should be reported as non-zero integers and must total 60 seconds or less for all light states. This method should not be used if approach contains multiple traffic lights with different light timings per light. Input fields include the following:

- SourceID: Unique identifier for the provider.
- TimeStamp: Date/time (format MM/DD/YYYY HH:mm:ss) when data was last updated.
- ApproachID: Intersection Network identifier for the approach.
- GreenTime: Number of seconds of the green light during the reporting period.
- YellowTime: Number of seconds of the yellow light during the reporting period.
- RedTime: Number of seconds of the red light during the reporting period.

2.4.12 LightTimingPerLane

This method allows providing the length of time in seconds each light state was active over the previous 60 seconds on a per lane basis. All timing values should be reported as non-zero integers and must total 60 seconds or less for all light states. Input fields include the following:

- SourceID: Unique identifier for the provider.
- TimeStamp: Date/time (format MM/DD/YYYY HH:mm:ss) when data was last updated.
- LaneID: Intersection Network identifier for the lane.
- GreenTime: Number of seconds of the green light during the reporting period.
- YellowTime: Number of seconds of the yellow light during the reporting period.

- RedTime: Number of seconds of the red light during the reporting period.

2.4.13 Occupancy

This method allows providing occupancy approach-level TMC data. ApproachID must match the unique lane identifier reported from the ITSIQA TMS API via the GetIntersectionNetwork output request methods. If no data exists for a turning type, Provider should provide the count value less than zero, such as -1. Input fields include the following:

- SourceID: Unique identifier for the provider.
- TimeStamp: Date/time (format MM/DD/YYYY HH:mm:ss) when data was last updated.
- ApproachID: Intersection Network identifier for the approach.
- RedOccupancy: Value reported in seconds of the time which vehicles occupied the lanes during the red light.
- GreenOccupancy: Value reported in seconds of the time which vehicles occupied the lanes during the green light.

2.4.14 PeakHourFactor

This method allows providing peak hour factor lane-level TMC data. LaneID must match the unique lane identifier reported from the ITSIQA TMS API via the GetIntersectionNetwork output request methods. If no data exists for a turning type, Provider should provide the count value less than zero, such as -1. Input fields include the following:

- SourceID: Unique identifier for the provider.
- TimeStamp: Date/time (format MM/DD/YYYY HH:mm:ss) when data was last updated.
- LaneID: Intersection Network identifier for the lane.
- PeakHourFactor: Calculated value that uses the volume of vehicles during the peak hour over the past 24 hours.

2.4.15 PedCounts

This method allows providing pedestrian counts per intersection approach and heading. ApproachID must match the unique lane identifier reported from the ITSIQA TMS API via the GetIntersectionNetwork output request methods. Count values should be given as integers. If no data exists for a bicycle count, a value less than zero should be provided, such as -1. Input fields include the following:

- SourceID: Unique identifier for the provider.
- TimeStamp: Date/time (format MM/DD/YYYY HH:mm:ss) when data was last updated.
- ApproachID: Intersection Network identifier for the approach.

- Heading: Valid heading compared to the direction of the approach. Values are listed in Table 7. Forward is the same direction of the approach and Backward is the opposite direction of the approach.
- Count: Reported number of pedestrians that moved through the intersection in the direction of the heading.

2.4.16 PercentArrival

This method allows providing percent arrival approach-level TMC data. ApproachID must match the unique lane identifier reported from the ITSIQA TMS API via the GetIntersectionNetwork output request methods. If no data exists for a turning type, Provider should provide the count value less than zero, such as -1. Input fields include the following:

- SourceID: Unique identifier for the provider.
- TimeStamp: Date/time (format MM/DD/YYYY HH:mm:ss) when data was last updated.
- ApproachID: Intersection Network identifier for the approach.
- PercentArrivalOnRed: Percent of vehicles arriving during the red light.
- PercentArrivalOnGreen: Percent of vehicles arriving during the green light.

2.4.17 RightTurnOnRed

This method allows providing right turn on red count lane-level TMC data. LaneID must match the unique lane identifier reported from the ITSIQA TMS API via the GetIntersectionNetwork output request methods. If no data exists for a turning type, Provider should provide the count value less than zero, such as -1. Input fields include the following:

- SourceID: Unique identifier for the provider.
- TimeStamp: Date/time (format MM/DD/YYYY HH:mm:ss) when data was last updated.
- LaneID: Intersection Network identifier for the lane.
- RightTurnOnRed: Count of vehicles that turned right during a red light.

2.4.18 TurnMovements

This method allows providing turning movement counts for a given lane. LaneID must match the unique lane identifier reported from the ITSIQA TMS API via the GetIntersectionNetwork output request methods. Counts should be reported as integers by turning type. If no data exists for a turning type, Provider should provide the count value less than zero, such as -1. Input fields include the following:

- SourceID: Unique identifier for the provider.
- TimeStamp: Date/time (format MM/DD/YYYY HH:mm:ss) when data was last updated.

- LaneID: Intersection Network identifier for the lane.
- UTurnCount: Count of vehicles that made a UTurn during reporting period.
- LeftTurnCount: Count of vehicles that made a Left Turn during reporting period.
- ThroughCount: Count of vehicles that traveled through the intersection during reporting period.
- RightTurnCount: Count of vehicles that made a Right Turn during reporting period.

2.4.19 TurnMovementsOnLight

This method allows providing turning movement counts for a given lane and Light State. All valid Light States are listed in Table 6. Requirements for LaneID and counts are the same as the TurnMovements method. Input fields include the following:

- SourceID: Unique identifier for the provider.
- TimeStamp: Date/time (format MM/DD/YYYY HH:mm:ss) when data was last updated.
- LaneID: Intersection Network identifier for the lane.
- UTurnCount: Count of vehicles that made a UTurn during reporting period.
- LeftTurnCount: Count of vehicles that made a Left Turn during reporting period.
- ThroughCount: Count of vehicles that traveled through the intersection during reporting period.
- RightTurnCount: Count of vehicles that made a Right Turn during reporting period.
- LightState: Light state associated with turning movement counts.

2.4.20 TurnMovementsOnLightWithSpeed

This method allows providing turning movement counts for a given lane, Light State, and includes reported speed. All valid Light States are listed in Table 6. Requirements for LaneID and counts are the same as the TurnMovements method. Input fields include the following:

- SourceID: Unique identifier for the provider.
- TimeStamp: Date/time (format MM/DD/YYYY HH:mm:ss) when data was last updated.
- LaneID: Intersection Network identifier for the lane.
- UTurnCount: Count of vehicles that made a UTurn during reporting period.
- LeftTurnCount: Count of vehicles that made a Left Turn during reporting period.
- ThroughCount: Count of vehicles that traveled through the intersection during reporting period.
- RightTurnCount: Count of vehicles that made a Right Turn during reporting period.
- LightState: Light state associated with turning movement counts.
- Speed: Average speed of vehicles in miles per hour.

2.4.21 TurnMovementsWithSpeed

This method allows providing turning movement counts for a given lane and includes reported speed. Requirements for LaneID and counts are the same as the TurnMovements method. Input fields include the following:

- SourceID: Unique identifier for the provider.
- TimeStamp: Date/time (format MM/DD/YYYY HH:mm:ss) when data was last updated.
- LaneID: Intersection Network identifier for the lane.
- UTurnCount: Count of vehicles that made a UTurn during reporting period.
- LeftTurnCount: Count of vehicles that made a Left Turn during reporting period.
- ThroughCount: Count of vehicles that traveled through the intersection during reporting period.
- RightTurnCount: Count of vehicles that made a Right Turn during reporting period.
- Speed: Average speed of vehicles in miles per hour.