



I-95 Vehicle Probe Project II



Interface Guide
April 2018



TABLE OF CONTENTS

Understanding & Using INRIX Data	1
INRIX Connected Services	1
The INRIX Road Network Architecture.....	1
TMC Segments.....	1
XD Segments.....	2
Sub-Segments.....	2
Segment Sets	3
Accessing INRIX Data.....	3
Establishing Credentials.....	3
Using Security Tokens.....	3
Connecting to the INRIX Gateway Server	3
Tips and Programming Suggestions.....	3
Traffic Data.....	4
Requests	4
Responses.....	4
Tips and Programming Suggestions.....	6
Graphical Traffic Data	7
Requests	7
Responses.....	7
Tips and Programming Suggestions.....	8
Incident Information	8
Requests	8
Responses.....	8
API Calls.....	10
GetSecurityToken.....	10
Description	10
Example CALL Request	10
Example CALL Response	10
Syntax	10
Parameters	10
Return Values	11
Response Elements.....	11
GetRoadSpeedInTMCs	13
Description	13
Example CALL Request	13
Example CALL Response	13

Syntax	13
Parameters	13
Return Values	14
Response Elements.....	15
GetRoadSpeedInSet	17
Description	17
Example CALL Request	17
Example CALL Response	17
Syntax	17
Parameters	17
Return Values	18
Response Elements.....	19
GetSegmentSpeed	21
Description	21
Example CALL Request	21
Example CALL Response	21
Syntax	22
Parameters	22
Return Values	23
Response Elements.....	23
Notes	26
GetSegmentSpeedInSet	27
Description	27
Example CALL Request	27
Example CALL Response	27
Syntax	27
Parameters	28
Return Values	29
Response Elements.....	29
Notes	31
GetRoadSpeedAtPoints.....	32
Description	32
Example CALL Request	32
Example CALL Response	32
Syntax	32
Parameters	32
Return Values	33
Response Elements.....	34
GetTrafficTileForSets.....	37
Description	37
Example CALL Request	37

Example CALL Response	37
Syntax	37
Parameters	38
Return Values	39
Notes	39
GetTrafficTileForSegmentSets	40
Description	40
Example CALL Request	40
Example CALL Response	40
Syntax	40
Parameters	41
Return Values	42
Notes	42
GetXDIncidentsInSegments	43
Description	43
Example CALL Request	43
Example CALL Response	43
Syntax	48
Parameters	48
Return Values	49
Response Elements.....	50
GetXDIncidentsInSet	61
Description	61
Example CALL Request	61
Example CALL Response	61
Syntax	65
Parameters	66
Return Values	67
Response Elements.....	68
GetSpeedBucket.....	79
Description	79
Example CALL Request	79
Example CALL Response	79
Syntax	79
Parameters	79
Return Values	80
Response Elements.....	80
Notes	84
CreateSpeedBucket.....	85
Description	85
Example CALL Request	85

Example CALL Response	85
Syntax	86
Parameters	86
Return Values	86
Response Elements.....	87
Notes	91
Appendix A – TMCs	92
Appendix B – INRIX XD Segments	97
Appendix C – Sub-Segments	99
Appendix D – INRIX-Managed Set Files	100
Appendix E – Integrating Graphical Traffic Data in User Applications	101
Appendix F – Speed Buckets	102
Appendix G – Score	104
Appendix H – Confidence (C-Value)	105
Appendix I – QuadKeys	106
Appendix J – RESTful Tile API.....	107
appToken API	107
Description	107
Example CALL Request	107
Example CALL Response	107
Syntax	107
Parameters	108
Return Values	108
Tile API	108
Description	108
Example CALL Request	108
Example CALL Response	109
Syntax	109
Parameters	109
Return Values	110
Appendix K – Common Status IDs	111
Appendix L – Contacts and Resources	112

This document contains links to help the user navigate.

Clicking on section titles in the Table of Contents will take the user directly to those sections.

Other links throughout the document will either jump within the document to the appropriate place or open the referenced item in a browser window.

To navigate BACK to the previously viewed page in the document, use <Alt><LeftArrow>.

INRIX Connected Services

INRIX Connected Services is a suite of online applications that provide traffic information and location-based services to connected devices over the Internet. These services can be used to build a wide variety of compelling client applications that provide users with useful, timely, and accurate information.

This document discusses a subset of the total INRIX suite that is available under the I-95 VPPII contract, providing access to traffic information in the following areas:

- Traffic data detailing road speed parameters (reference speed, current speed, historical average speed, time required to travel across a segment, etc.) for:
 - Road segments, called TMC Segments or XD Segments (specified by TMCIds or XDIds)
 - Groups of road segments, listed in Segment Sets (specified by SetIds)
 - Road segments closest to a point, defined by latitude/longitude and heading
- Graphical representations of road speed parameters, called Traffic Tiles, to use as overlays on maps
- Incident Information, providing alerts about slow speed areas or scheduled construction

The INRIX Road Network Architecture

The INRIX architecture utilizes TMC Segments and/or XD Segments as the basis for defining road sections on which to report speed and incident data.

TMC SEGMENTS

The Traffic Message Channel (TMC) is a specific application of the FM Radio Data System (RDS) used for broadcasting real-time traffic and weather information.¹ The development of RDS-TMC standards started in Europe and included the development of TMC location codes. TMC codes were originally conceived of as points on the road network, typically assigned at significant decision points, interchanges or intersections for the purpose of describing locations of traffic incidents (accidents, construction, traffic slowdowns, etc.) in an unambiguous format, independent of map vendor.

The North American Location Code Alliance (a consortium of TomTom and HERE (previously NAVTEQ), the nation's leading suppliers of commercial map databases) created, owns, maintains, and expands a US/Canada TMC location code table that adheres to the international standard on location referencing (ISO 14819-3:2004²). INRIX has the rights to utilize the location table provided by TomTom.

INRIX reports traffic flow data by considering the road segments implied by the distance between consecutive TMC location codes, referred to as TMC Segments. As updates to the TMC location code table are published, INRIX updates and expands available road coverage.

¹ For more information on RDS-TMC, go to: <http://www.tisa.org/technologies/tmc/>

² See: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=35984

[Appendix A](#) provides more information on interpreting TMC Codes and illustrates the elements that are contained in the TMC Segments database.

XD SEGMENTS

INRIX XD Segments, first introduced in 2013, are similar to TMC Segments in that they delineate a specific section of roadway, but they also address some of the limitations of TMC Segments, such as road coverage, the ability to cover new roads more quickly, segment overlap and gapping, and segment resolution.

XD Segments are defined and maintained solely by INRIX. Therefore, it is possible for INRIX to create XD Segments for sections of roads or new roads that do not yet have defined TMC location codes or TMC Segments. XD Segments also differ in that they provide a more consistent, unambiguous, and granular definition of road segments when compared to TMC Segments, which vary greatly in length depending on the distance between neighboring TMC location codes.

Through feedback and collaboration with customers who have successfully implemented and are benefitting from the improvements offered by XD, INRIX identified new functionality that was desired by the XD user base. XD v2 improves the utility of data on arterial networks with two key enhancements: intentional termination of segments at junctions/intersections/interchanges and increased granularity of segments, particularly in urban signalized corridors.

XD v2 segments terminate (or “break”) at junctions/intersections/interchanges or at their maximum length of 1,600 meters. This is a significant change from XD v1, where breaks were frequently at arbitrary positions between junctions. XD v2 breakpoints are at junctions between roads of equivalent hierarchical rank, determined mostly by Functional Road Classification (FRC) similarity — that is, they do not break at junctions with roads of lower rank. (Note: The FRC classifications are determined by INRIX map data using a classification scheme consistent worldwide. The FRC reference here should not be confused with a customer’s own FRC classification.) In addition, if an XD segment has a corresponding TMC location code for the same stretch of road, the XD segment will not extend past the break point of the TMC location code.

As a result of this upgrade, XD v2 segments are generally shorter (providing more granularity) and cover the underlying road network in a more logical and useful fashion. Users benefit from better sequencing of segments — the next/previous relationships of XD segments are improved, and the segments form more logical chains especially through junctions. There are also no overlaps of segments or gaps in coverage when using XD segments. Collectively, this enhances the accuracy of travel times between junctions/intersections/interchanges.

In addition, the new segment break points provide a more stable segmentation system with far fewer changes to segment locations and IDs over time, increasing consistency and decreasing the work required of XD users for map updates.

[Appendix B](#) provides more information about INRIX XD Segments.

SUB-SEGMENTS

Sub-segment options for INRIX XD Segments and TMC Segments further improve granularity, which enables reporting on greater resolution on any road covered by INRIX XD Segments or TMC Segments.

[Appendix C](#) provides more information about sub-segments.

SEGMENT SETS

In order to reduce the complexity of requests and responses, individual TMC Segments and/or XD Segments can be grouped into a Segment Set identified by a SetId. INRIX will create and maintain distinct set files for each state that could contain, for example, contracted road mileage separated into freeway coverage and arterial coverage. Additional details about the INRIX-managed set files are included in [Appendix D](#).

INRIX allows for the creation of custom Segment Sets that would not be created, managed or provided by INRIX, but rather created by a user for analyzing custom coverage. There is specific documentation for creating custom Segment Sets that is not contained within this interface guide. Please [contact INRIX support](#) to obtain initial permission and credentials that will allow for Segment Set creation.

Accessing INRIX Data

INRIX delivers data to customers via a Web Services Application Programming Interface (API). All API requests are made via Hypertext Transport Protocol (HTTP). Depending on the type of data returned, responses are either delivered as XML files (for numerical or text data) or as PNG or GIF files (for graphical data).

ESTABLISHING CREDENTIALS

Agencies must initially establish their credentials. They must sign a [Data Use Agreement \(DUA\)](#) and submit it to the [contracting agency](#). Then they must request a Vendor ID and Consumer ID for themselves and/or their contractors. A Vendor ID and Consumer ID are the required credentials for accessing all data discussed within this document.

USING SECURITY TOKENS

Renewable security tokens, requested via API, are used to control access to INRIX data.

To obtain a security token, a user must make a call to the API, [GetSecurityToken](#), supplying their credentials for authentication. The response to this call is an XML file that returns a security token and the URLs specifying the server paths to use for all subsequent API calls.

When a security token is issued by the INRIX system, it is valid for a preset number of minutes. Presently the default token expiry is 60 minutes. Once that time has expired, all API calls will return with failures until a new security token has been requested.

CONNECTING TO THE INRIX GATEWAY SERVER

GetSecurityToken will return the server paths to specify in subsequent API calls. The URL with type="API" and region="NA" will be used for most calls; however, calls that return graphical data (Traffic Tiles) will use the URL with type="TTS" and region="NA".

TIPS AND PROGRAMMING SUGGESTIONS

- Any request using an expired token will return with the attributes statusId="43" statusText="TokenExpired" in the <Inrix> root element of the XML response. Therefore, the calling code should be designed either to obtain a new

token from GetSecurityToken at a set interval less than the expiry duration or to request a new token if a call using the previous token is rejected and then re-issue the request.

- The server path for subsequent calls should NOT be hardcoded, since it may change in the future. The correct path for the type of call (“API” or “TTS” URL) should be chosen from the paths returned in the GetSecurityToken response.
- To increase flexibility of the implementation, it is best practice to store the URL for the GetToken request in a configuration file so that it can be altered if needed during the lifecycle of the project.

Traffic Data

The APIs in the following list return data specifying road speed parameters (reference speed, measured speed, historical average speed, time required to travel across a segment, etc.) to help analyze traffic conditions in real time. Details for the API requests and responses can be found in the next section of this document. (Click on the API names to jump directly to the referenced locations.)

REQUESTS

- [GetRoadSpeedInTMCs](#)** – For use with an individual TMC Segment or a list of TMC Segments, identified by TMCIds.
- [GetRoadSpeedInSet](#)** – For use with INRIX-managed or user-created Segment Sets containing only TMC Segments, identified by TMCSetIds.
- [GetSegmentSpeed](#)– For use with an individual Segment or a list of Segments, identified by XDIds and/or TMCIds.
- [GetSegmentSpeedInSet](#)– For use with INRIX-managed or user-created Segment Sets containing XD and/or TMC Segments, identified by SetIds.
- [GetRoadSpeedAtPoints](#) – To access traffic conditions at the road segment nearest to a specified point (latitude/longitude) and heading.

****The TMC-only APIs, GetRoadSpeedInTMCs and GetRoadSpeedInSet, will be deprecated in the near future.** Users requiring TMC data should transition to the GetSegmentSpeed and GetSegmentSpeedInSet APIs, which perform the same functions and have been designed to accept both XDIds and/or TMCIds.

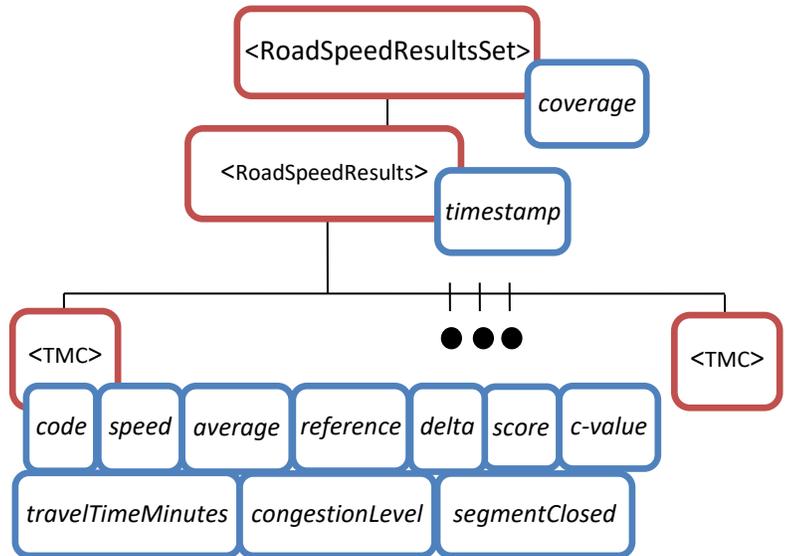
RESPONSES

API responses for standard traffic data are delivered in XML format. In general, each XML response contains a common root element <Inrix>, which has several attributes (*docType, copyright, versionNumber, createdDate, statusId, statusText, responseId*). It also contains a child element that serves as a container for the traffic data requested in the API call.

TMC Segments

Responses to the API calls requesting traffic data about TMC Segments (GetRoadSpeedInTMCs, GetRoadSpeedInSet) include a <RoadSpeedResultSet> element, which has an attribute called coverage specifying that all types of coverage (real-time core and/or extended, reference, historical) are being returned, and serves as a container for the returned data with the following structure:

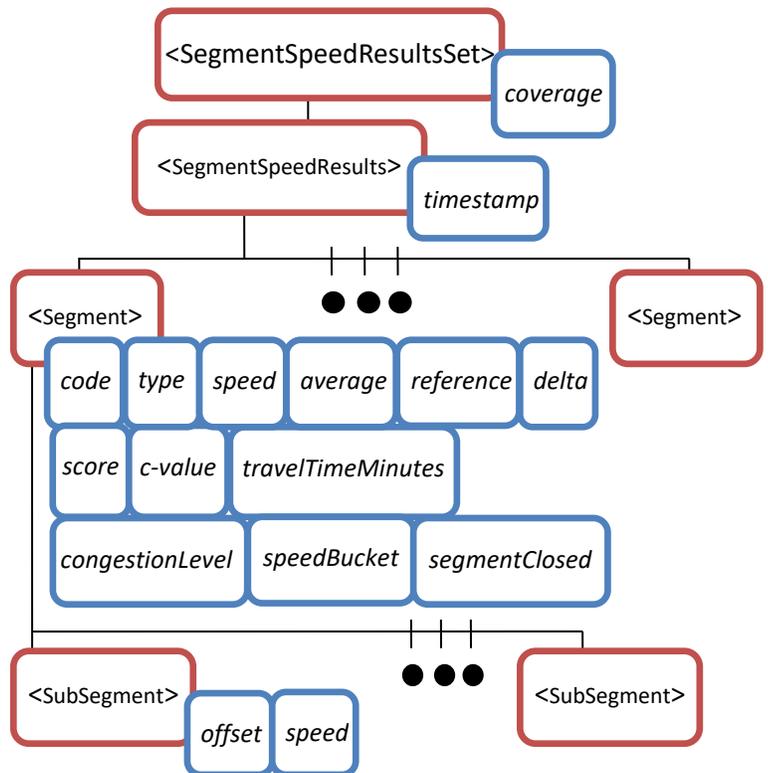
- The requested data itself is contained in a child element called `<RoadSpeedResults>`, which has a *timestamp* attribute.
- The `<RoadSpeedResults>` element also contains one or more child elements called `<TMC>`, one for each TMC Segment in the API call.
- The `<TMC>` element includes the traffic data attributes requested in the API call. The specific attributes will vary as specified in the call parameters but will include some or all of the following: *code, speed, average, reference, delta, score, c-value, travelTimeMinutes, congestionLevel, and segmentClosed*.



XD and/or TMC Segments

Similarly, responses to the API calls requesting traffic data about XD and/or TMC Segments (`GetSegmentSpeed`, `GetSegmentSpeedInSet`) include a `<SegmentSpeedResultSet>` element, which has an attribute called *coverage* specifying that all types of coverage (real-time core and/or extended, reference, historical) are being returned, and serves as a container for the returned data.

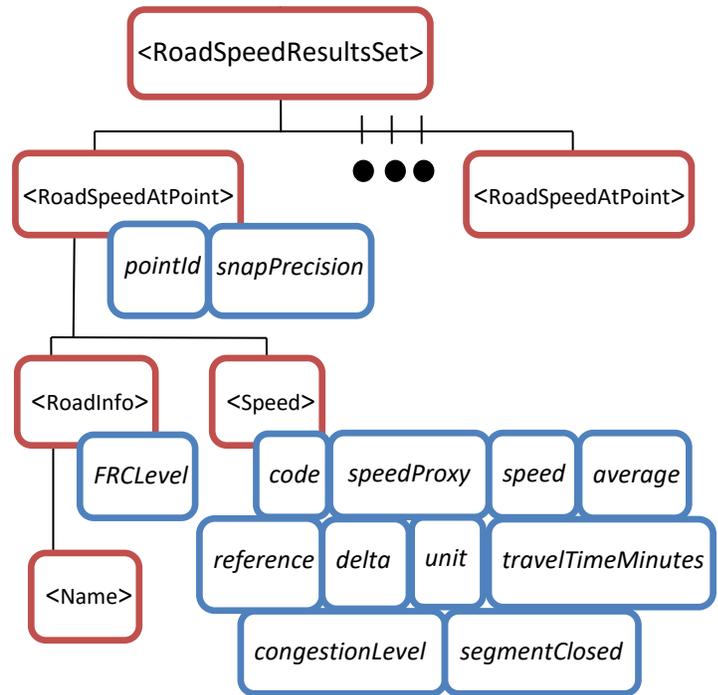
- The requested data itself is contained in a child element called `<SegmentSpeedResults>`, which has a *timestamp* attribute.
- The `<SegmentSpeedResults>` element also contains one or more child elements called `<Segment>`, one for each XD and/or TMC Segment in the API call.
- The `<Segment>` element includes the traffic data attributes requested in the API call. The specific attributes will vary as specified in the call parameters but will include some or all of the following: *code, type, speed, average, reference, delta, score, c-value, travelTimeMinutes, congestionLevel, speedBucket, and segmentClosed*.
- If sub-segments are requested, the `<Segment>` element will contain a child element called `<SubSegment>` for each sub-segment that meets the parameter requirements.



Segments Closest to a Point

Using the same general structure, responses to the API call `GetRoadSpeedAtPoints` include a `<RoadSpeedResultSet>` element that serves as a container for the returned data.

- The requested data itself is contained in one or more child elements called `<RoadSpeedAtPoint>`. Each of these elements contains `pointId` and `snapPrecision` attributes, and there is a `<RoadSpeedAtPoint >` element for each point requested in the API call.
- Each `< RoadSpeedAtPoint >` element contains two child elements called `<RoadInfo>` and `<Speed>`.
- The `<RoadInfo>` element may include the `FRCLLevel` attribute and/or the child element `<Name>`
- The `<Speed>` element includes the traffic data attributes requested in the API call. The specific attributes will vary as specified in the call parameters but will include some or all of the following: `code`, `speedProxy`, `speed`, `average`, `reference`, `delta`, `unit`, `travelTimeMinutes`, `congestionLevel` and `segmentClosed`.



TIPS AND PROGRAMMING SUGGESTIONS

Managing Sub-Segments

To manage the return file size when requesting sub-segments, several approaches that allow balance between payload size and usefulness of returned data are available.

- As a default, if sub-segment data is requested in a call, the speeds of all sub-segments are returned. However, a parameter that specifies the minimum speed delta between the parent segment and sub-segment can be utilized to decrease the file size and highlight areas with slower and/or faster speeds.
 - If this parameter is set to zero, all sub-segments are returned. This is the default.
 - If it is set to one, only sub-segments with a speed that is not the same as the parent segment are returned. This can greatly reduce the potential size of the response and highlight areas where traffic is slower and/or faster than the average on the parent segment.
 - The parameter can also be set to a higher value, allowing the user to highlight only those areas where traffic is much slower or faster and to decrease the file size further.
- INRIX suggests that users request a minimum resolution of 250 meters for sub-segments to help limit the file size.
- If a user employs set files containing a large number of TMCIDs and/or XDIDs, but only requires sub-segment data for a small subset of those segments, it would be beneficial to take one of these approaches:
 - Make an API call specifying the large set file, but no sub-segments. Then make a separate call requesting sub-segments and specifying the segment IDs of interest in a comma-delimited list.

- Create a smaller custom set file containing the segments of interest. Make a call with the large set file with no sub-segments, followed by a call with the smaller set file with sub-segments.

Graphical Traffic Data

The INRIX Traffic Tile Service (TTS) generates graphic image representations of Traffic Tiles that allow users to overlay traffic data onto a map for visualization. The service works with most Mercator projected maps (Google, Virtual Earth, Yahoo, MapQuest) and custom configurations are possible. See [Appendix E](#) for more information about integrating graphical traffic data into user applications.

Details for the API requests and responses can be found in the next section of this document. (Click on the API names to jump directly to the referenced locations.)

REQUESTS

****[GetTrafficTileForSets](#)**— For use with INRIX-managed or user-created Segment Sets containing only TMC Segments, identified by TMCSetIds.

[GetTrafficTileForSegmentSets](#)— For use with INRIX-managed or user-created Segment Sets containing XD and/or TMC Segments, identified by SetIds.

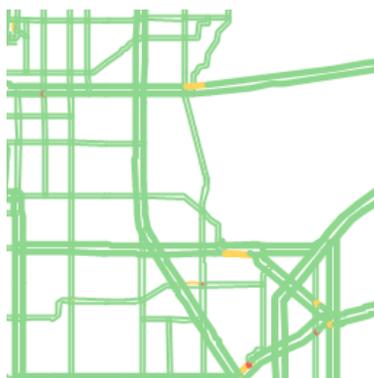
(Remember that these “GetTile” calls use the URL with type=“TTS”, as returned in the [GetSecurityToken](#) API.)

****The TMC-only API, [GetTrafficTileForSets](#), will be deprecated in the near future.** Users requiring TMC data should transition to the [GetTrafficTileForSegmentSets](#) API, which performs the same function and has been designed to accept both XDIds and/or TMCIds.

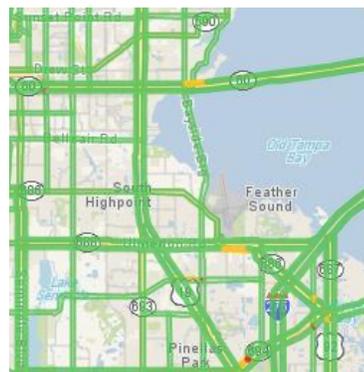
Note: A new Tile API has been created with RESTful guidelines in a cloud-based environment, making use of the API easier for the developer and more reliable and scalable for the end user. Details about how to use the new Tile API call can be found in [Appendix J](#).

RESPONSES

The TTS returns Traffic Tiles, consisting of graphical data with a transparent background that can be superimposed over maps, in either PNG or GIF format. The overlay depicts all segments and sub-segments specified in the call. For example:



Traffic Tile Overlay



Tile Superimposed Over User's Map

TIPS AND PROGRAMMING SUGGESTIONS

Traffic APIs versus Tile APIs?

To display traffic flow on a map, users can employ the Tile APIs. They provide transparent overlays that can be used with user-generated maps. The other option is for users to utilize the Traffic APIs to receive data from which they can build their own images.

Note that Tiles are two-dimensional and cannot be overlaid on a 3D surface.

Managing Sub-Segments

Unlike with the Traffic APIs, the file size for a Tile does not change significantly if sub-segments are requested. However, due to the underlying segment architecture and available granularity of the data, INRIX suggests that users request a minimum resolution of 100 meters for sub-segments when making Tile calls.

Varying the Look of a Traffic Tile

INRIX uses the concept of a Speed Bucket to separate data into groups (or “buckets”), based on speed or congestion level, and to specify how road segments that fall into these groups will be depicted. The pen style, color, hatching, and width of the lines associated with each bucket, as well as with a road closure, can be defined. Users can also specify different line styles for different road classifications, allowing skinnier lines for smaller roads, for example.

More information about how to use predefined or to create custom Speed Buckets is included in [Appendix F](#).

Incident Information

The INRIX Incident Service provides information on incidents that can impact traffic, including accidents, events, construction, and congestion

The main usage scenarios are displaying incidents on a map, creating routes that avoid incidents, and creating alerts for current or upcoming incidents. Details for the API and response can be found in the next section of this document. (Click on the API name to jump directly to the referenced location.)

REQUESTS

- [GetXDIncidentsInSegments](#)– For use with an individual Segment or a list of Segments, identified by XDIds and/or TMCIds.
- [GetXDIncidentsInSet](#)– For use with INRIX-managed or user-created Segment Sets containing XD and/or TMC Segments, identified by SetIds.

RESPONSES

Incident responses are delivered in XML format, similar to other basic traffic data. In general, each XML response contains a common root element <Inrix>, which has several attributes (*docType, copyright, versionNumber, createdDate,*

API CALLS

GetSecurityToken

DESCRIPTION

Returns a server path and a security token to use for authentication in subsequent API calls.

EXAMPLE CALL REQUEST

```
http://api.inrix.com/Traffic/Inrix.ashx?Action=GetSecurityToken&Vendorid=<InsertYourVendorID>&Consumerid=<InsertYourConsumerID>
```

EXAMPLE CALL RESPONSE

```
<Inrix docType="GetSecurityToken" copyright="Copyright INRIX Inc." versionNumber="10.4" createDate="2017-07-26T18:58:03Z" statusId="0" statusText="" responseId="329a9e76-92ec-40bb-be44-eadbda59597c">
  <AuthResponse>
    <AuthToken expiry="2017-07-26T19:57:00Z">toMt*t7nnkAl*MDklsw4lyCbChfru4uh3*hcx9WcRcQ|</AuthToken>
    <ServerPath>devzone.inrix.com/traffic/inrix.ashx</ServerPath>
    <ServerPaths>
      <ServerPath type="API" region="NA">http://na.api.inrix.com/Traffic/Inrix.ashx</ServerPath>
      <ServerPath type="TTS" region="NA">http://na-rseg-tts.inrix.com/RsegTiles/tile.ashx</ServerPath>
    </ServerPaths>
  </AuthResponse>
</Inrix>
```

SYNTAX

```
ServerPath
?Action=GetSecurityToken
&vendorId=string
&consumerId=string
```

PARAMETERS

- ServerPath*** (Type: string, Required)
The designated URL for the GetSecurityToken request (http://api.inrix.com/Traffic/Inrix.ashx?).
- vendorId*** (Type: string, Required)
A unique ID assigned by INRIX to each vendor.
- consumerId*** (Type: string, Required)
The user's consumer ID assigned by INRIX.

RETURN VALUES

Each XML response to this API call will contain two basic elements, and each may have attributes and/or child elements.

- **Inrix** – The root element in the response for all APIs is always <Inrix>. It contains response data in child elements and information about the request in the following attributes:
 - *docType* identifies the API called and the schema used for the response.
 - *copyright* identifies the INRIX, Inc. copyright.
 - *versionNumber* refers to the version number of the INRIX traffic services platform that produced the response.
 - *createdDate* is the timestamp indicating when the response was created. The timestamp specifies a DateTime value, formatted according to the ISO 8601 standard and expressed in UTC time. The format of the attribute is YYYY-MM-DDTHH:MM:SSZ. The “T” character separates the date from the time, and the “Z” character specifies that UTC time is used.
 - *statusId* indicates whether an API request succeeded or failed. Each *statusId* value is associated with a unique *statusText* value. A *statusId* is intended to be convenient to handle in code. *statusId=0* is a successful call. [Appendix K](#) contains a list of common statusID responses.
 - *statusText* corresponds to the *statusId*. A *statusText* is intended to be convenient for humans to interpret.
 - *responseID* is logged by INRIX for future use in assisting customer inquiries. Service Requestors can use *responseID* as a way for INRIX to identify the specific API call that was made. It is recommended that Service Requestors log this information to expedite technical support incidents.
- **AuthResponse** – This child element is a container for the requested data.

RESPONSE ELEMENTS

AuthResponse (Parent element: Inrix)

A container for authentication and server path information required to make requests.

```
Syntax: <AuthResponse>
         <AuthToken ... />
         <ServerPaths ... />
       </ AuthResponse >
```

Element	Required	Occurs	Description
AuthToken	yes	1	Contains a security token.
ServerPaths	yes	1	A container for ServerPath elements.

AuthToken (Parent element: AuthResponse)

Contains a security token.

```
Syntax: <AuthToken
         expiry = "string">
         SECURITY TOKEN
       </ AuthToken >
```

Attribute	Required	Type	Usage
expiry	yes	string	The date and time of expiration of the security token.

ServerPaths (Parent element: AuthResponse)

A container for ServerPath elements.

Syntax: <ServerPaths
 <ServerPath ... />
 </ ServerPaths>

Element	Required	Occurs	Description
ServerPath	yes	1...n	Specifies the URLs of the servers to which requests should be sent.

ServerPath (Parent element: ServerPaths)

Specifies the URLs of the servers to which requests should be sent.

Syntax: <ServerPath
 type = "string"
 region = "string">
 SERVER PATH
 </ ServerPath>

Attribute	Required	Type	Usage
type	yes	string	Indicates the type of request that should be sent to this server: API = non-Traffic Tile calls TTS = Traffic Tile calls
region	yes	string	Indicates the server's region. NA = North America

GetRoadSpeedInTMCs

This API will be deprecated in the near future. Use the [GetSegmentSpeed](#) API instead.

DESCRIPTION

Retrieves road speed data for a specified list of TMC Segments.

EXAMPLE CALL REQUEST

```
http://na.api.inrix.com/Traffic/Inrix.ashx?Action=GetRoadSpeedinTMCs&Token=2lkHQ79BL5mA*mrjawHZQOw6dD4d892nghh8HFJUNgo|&tmcs=110%2B05548,110-05548,110P05548,110N05548
```

EXAMPLE CALL RESPONSE

```
<Inrix docType="GetRoadSpeedInTmcs" copyright="Copyright INRIX Inc." versionNumber="10.4" createdDate="2017-07-26T19:14:26Z" statusId="0" statusText="" responseId="2945417b-4d91-49fc-882f-2bdad1926ddf">
  <RoadSpeedResultSet coverage="255">
    <RoadSpeedResults timestamp="2017-07-26T19:14:26Z">
      <TMC code="110+05548" speed="73" average="66" reference="66" delta="7" score="30"
        c-value="73" travelTimeMinutes="1.323" congestionLevel="3"/>
      <TMC code="110-05548" speed="74" average="67" reference="67" delta="7" score="30"
        c-value="100" travelTimeMinutes="3.038" congestionLevel="3"/>
      <TMC code="110P05548" speed="72" average="67" reference="67" delta="5" score="30"
        c-value="93" travelTimeMinutes="0.375" congestionLevel="3"/>
      <TMC code="110N05548" speed="75" average="67" reference="67" delta="8" score="30"
        c-value="85" travelTimeMinutes="0.355" congestionLevel="3"/>
    </RoadSpeedResults>
  </RoadSpeedResultSet>
</Inrix>
```

SYNTAX

ServerPath

?Action=GetRoadSpeedInTMCs

&Token=*string*

&Tmcs=*string*

&SpeedOutputFields=*string*

&Units=*integer*

PARAMETERS

ServerPath (Type: string, Required)

The URL of the server to which requests should be addressed as returned in the ServerPath element of the response to a GetSecurityToken request. (Should use the API path.)

Token (Type: string, Required)

A valid security token as returned in the AuthToken element of the response to a GetSecurityToken request.

Tmcs (Type: string, Required)

TMCIDs for which to get data. Can be one or more TMCIDs specified in a comma-delimited list. Any duplicate TMCIDs are ignored. The TMCIDs must be in 9-digit format. If the TMCID has a "+" for the direction component then it must be escaped as a %2B in the requesting URL.

SpeedOutputFields (Type: string, Optional)

The traffic speed fields to be returned in the response. Can specify a single field, or multiple fields in a comma-separated list such as "Reference,Speed". If this parameter is not specified, all fields are returned. (Further definition of the output fields is included in the Response Element section below.)

All (default)	Returns all fields.
Reference	Reference speed field.
Speed	Measured speed field.
Average	Historical average speed field.
Delta	Difference between average and current speed.
TTM	Time required to travel across the segment.
Congestion	Represents the level of congestion.
Score	Represents the source of the speed data returned.
Confidence	Represents the confidence in the real-time data.

Units (Type: integer, Optional)

Used for SpeedOutputFields data.

0 (default)	US (English units)
1	Metric units

RETURN VALUES

Each XML response to this API call will contain two basic elements, each having several attributes and/or child elements.

- *Inrix* – The root element in the response for all APIs is always <Inrix>. It contains response data in child elements and information about the request in the following attributes:
 - *docType* identifies the API called and the schema used for the response.
 - *copyright* identifies the INRIX, Inc. copyright.
 - *versionNumber* refers to the version number of the INRIX traffic services platform that produced the response.
 - *createdDate* is the timestamp indicating when the response was created. The timestamp specifies a DateTime value, formatted according to the ISO 8601 standard and expressed in UTC time. The format of the attribute is YYYY-MM-DDTHH:MM:SSZ. The "T" character separates the date from the time, and the "Z" character specifies that UTC time is used.
 - *statusId* indicates whether an API request succeeded or failed. Each *statusId* value is associated with a unique *statusText* value. A *statusId* is intended to be convenient to handle in code. *statusId*=0 is a successful call. [Appendix K](#) contains a list of common statusID responses.
 - *statusText* corresponds to the *statusId*. A *statusText* is intended to be convenient for humans to interpret.

— *responseID* is logged by INRIX for future use in assisting customer inquiries. Service Requestors can use *responseID* as a way for INRIX to identify the specific API call that was made. It is recommended that Service Requestors log this information to expedite technical support incidents.

- *RoadSpeedResultSet* – This child element is a container for all of the requested traffic data.

RESPONSE ELEMENTS

RoadSpeedResultSet (Parent element: Inrix)

A container for RoadSpeedResults elements.

Syntax: <RoadSpeedResultSet coverage = "Integer" >
 <RoadSpeedResults ... />
 </RoadSpeedResultSet>

Attribute	Required	Type	Usage
coverage	yes	Integer	Indicates the type of coverage returned. 255=All types of coverage

Element	Required	Occurs	Description
RoadSpeedResults	yes	1	Represents a collection of traffic data for a specified point in time.

RoadSpeedResults (Parent element: RoadSpeedResultSet)

Represents a collection of traffic data for a specified point in time.

Syntax: <RoadSpeedResults timestamp = "DateTime">
 <TMC ... />
 </RoadSpeedResults>

Attribute	Required	Type	Usage
timestamp	yes	DateTime	The point in time expressed in ISO 8963 format.

Element	Required	Occurs	Description
TMC	no	0..n	Represents traffic data for one TMC Segment.

TMC (Parent element: RoadSpeedResults)

Represents traffic data for one TMC Segment.

Syntax: <TMC
 code = "string"
 speed = "integer"
 average = "integer"
 reference = "integer"
 delta = "integer"
 travelTimeMinutes = "integer"
 congestionLevel = "integer">

```

score = "integer"
c-value = "integer"
segmentClosed = "boolean">
</TMC>

```

Attribute	Required	Type	Usage
code	yes	string	The TMC code identifying the segment of road for which data is being reported.
speed	no	integer	The traffic speed on the segment. This value is 0 if the road segment is closed.
average	no	integer	The historical average speed on the segment at the time of day specified in the RoadSpeedResults element. If the average speed is not available, this value may be empty.
reference	no	integer	The typical traffic speed on this segment under free flow conditions.
delta	no	integer	The difference between the speed and average values for the segment.
travelTimeMinutes	no	integer	The time in minutes required to traverse the segment.
congestionLevel	no	integer	The amount of congestion on the segment. A lower number represents higher congestion. 0 = speed is 0-31% of reference speed 1 = speed is 32-62% of reference speed 2 = speed is 63-92% of reference speed 3 = speed is 93-100% of reference speed 255= road segment is closed
score	no	integer	The source of the speed data field returned. 30 = Real-time data 20 = Historical data 10 = Reference speed (For more information, see Appendix G.)
c-value	no	integer	A measure of the confidence in the real-time data. Ranges from 0 to 100. The higher the c-value, the more confident the data. (For more information, see Appendix H.)
segmentClosed	no	boolean	Indicates whether the segment is closed or not. Speed data is not returned for a closed segment.

GetRoadSpeedInSet

This API will be deprecated in the near future. Use the [GetSegmentSpeedInSet](#) API instead.

DESCRIPTION

Retrieves road speed data for one or more TMC Segment Sets.

EXAMPLE CALL REQUEST

```
http://na.api.inrix.com/Traffic/Inrix.ashx?Action=GetRoadSpeedInSet&Token=36jjLnERspd07We5O5Fz6ZubfeRe9wgDy*v7W545SXc|&TmcSetID=1000010
```

EXAMPLE CALL RESPONSE

```
<Inrix docType="GetRoadSpeedInSet" copyright="Copyright INRIX Inc." versionNumber="10.4" createdDate="2017-07-26T19:46:26Z" statusId="0" statusText="" responseId="ff8b6a71-45d0-4f78-ae4e-7a93e5603333">
  <RoadSpeedResultSet coverage="255">
    <RoadSpeedResults timestamp="2017-07-26T19:46:26Z">
      <TMC code="125+08122" speed="46" average="39" reference="44" delta="7" score="30"
        c-value="97" travelTimeMinutes="0.773" congestionLevel="3"/>
      <TMC code="125N05113" speed="75" average="69" reference="68" delta="6" score="30"
        c-value="90" travelTimeMinutes="0.455" congestionLevel="3"/>
      ...
      <TMC code="125N04920" speed="56" average="54" reference="56" delta="2" score="30"
        c-value="100" travelTimeMinutes="0.015" congestionLevel="3"/>
      <TMC code="125P17002" speed="72" average="68" reference="68" delta="4" score="30"
        c-value="99" travelTimeMinutes="0.512" congestionLevel="3"/>
    </RoadSpeedResults>
  </RoadSpeedResultSet>
</Inrix>
```

SYNTAX

ServerPath

?Action=GetRoadSpeedInSet

&Token=*string*

&TmcSetID=*integer*

&SpeedOutputFields=*string*

&Units=*integer*

PARAMETERS

ServerPath (Type: string, Required)

The URL of the server to which requests should be addressed as returned in the ServerPath element of the response to a GetSecurityToken request. (Should use the API path.)

Token (Type: string, Required)

A valid security token as returned in the AuthToken element of the response to a GetSecurityToken request.

TmcSetID (Type: integer, Required)

TMCSetIds for which to get data. Can be one or more TMCSetIds specified in a comma-delimited list. Any duplicate TMCIds are ignored.

SpeedOutputFields (Type: string, Optional)

The traffic speed fields to be returned in the response. Can specify a single field, or multiple fields in a comma-separated list such as "Reference,Speed". If this parameter is not specified, all fields are returned. (Further definition of the output fields is included in the Response Element section below.)

All (default)	Returns all fields.
Reference	Reference speed field.
Speed	Measured speed field.
Average	Historical average speed field.
Delta	Difference between average and current speed.
TTM	Time required to travel across the segment.
Congestion	Represents the level of congestion.
Score	Represents the source of the speed data returned.
Confidence	Represents the confidence in the real-time data.

Units (Type: integer, Optional)

Used for SpeedOutputFields data.

0 (default)	US (English units)
1	Metric units

RETURN VALUES

Each XML response to this API call will contain two basic elements, each having several attributes and/or child elements.

- **Inrix** – The root element in the response for all APIs is always <Inrix>. It contains response data in child elements and information about the request in the following attributes:
 - *docType* identifies the API called and the schema used for the response.
 - *copyright* identifies the INRIX, Inc. copyright.
 - *versionNumber* refers to the version number of the INRIX traffic services platform that produced the response.
 - *createdDate* is the timestamp indicating when the response was created. The timestamp specifies a DateTime value, formatted according to the ISO 8601 standard and expressed in UTC time. The format of the attribute is YYYY-MM-DDTHH:MM:SSZ. The "T" character separates the date from the time, and the "Z" character specifies that UTC time is used.
 - *statusId* indicates whether an API request succeeded or failed. Each *statusId* value is associated with a unique *statusText* value. A *statusId* is intended to be convenient to handle in code. *statusId*=0 is a successful call. [Appendix K](#) contains a list of common statusID responses.
 - *statusText* corresponds to the *statusId*. A *statusText* is intended to be convenient for humans to interpret.

— *responseID* is logged by INRIX for future use in assisting customer inquiries. Service Requestors can use *responseID* as a way for INRIX to identify the specific API call that was made. It is recommended that Service Requestors log this information to expedite technical support incidents.

- *RoadSpeedResultSet* – This child element is a container for all of the requested traffic data.

RESPONSE ELEMENTS

RoadSpeedResultSet (Parent element: Inrix)

A container for RoadSpeedResults elements.

Syntax: <RoadSpeedResultSet coverage = "Integer" >
 <RoadSpeedResults ... />
 </RoadSpeedResultSet>

Attribute	Required	Type	Usage
coverage	yes	Integer	Indicates the type of coverage returned. 255=All types of coverage

Element	Required	Occurs	Description
RoadSpeedResults	yes	1	Represents a collection of traffic data for a specified point in time.

RoadSpeedResults (Parent element: RoadSpeedResultSet)

Represents a collection of traffic for a specified point in time.

Syntax: <RoadSpeedResults timestamp = "DateTime">
 <TMC ... />
 </RoadSpeedResults>

Attribute	Required	Type	Usage
timestamp	yes	DateTime	The point in time expressed in ISO 8963 format.

Element	Required	Occurs	Description
TMC	no	0..n	Represents traffic data for one TMC Segment.

TMC (Parent element: RoadSpeedResults)

Represents traffic data for one TMC Segment.

Syntax: <TMC
 code = "string"
 speed = "integer"
 average = "integer"
 reference = "integer"
 delta = "integer"
 travelTimeMinutes = "integer">

```

congestionLevel = "integer"
score = "integer"
c-value = "integer"
segmentClosed = "boolean">
</TMC>

```

Attribute	Required	Type	Usage
code	yes	string	The TMC code identifying the segment of road for which data is being reported.
speed	no	integer	The traffic speed on the segment. This value is 0 if the road segment is closed.
average	no	integer	The historical average speed on the segment at the time of day specified in the RoadSpeedResults element. If the average speed is not available, this value may be empty.
reference	no	integer	The typical traffic speed on this segment under free flow conditions.
delta	no	integer	The difference between the speed and average values for the segment.
travelTimeMinutes	no	integer	The time in minutes required to traverse the segment.
congestionLevel	no	integer	The amount of congestion on the segment. A lower number represents higher congestion. 0 = speed is 0-31% of reference speed 1 = speed is 32-62% of reference speed 2 = speed is 63-92% of reference speed 3 = speed is 93-100% of reference speed 255= road segment is closed
score	no	integer	The source of the speed data field returned. 30 = Real-time data 20 = Historical data 10 = Reference speed (For more information, see Appendix G.)
c-value	no	integer	A measure of the confidence in the real-time data. Ranges from 0 to 100. The higher the c-value, the more confident the data. (For more information, see Appendix H.)
segmentClosed	no	boolean	Indicates whether the segment is closed or not. Speed data is not returned for a closed segment.

GetSegmentSpeed

DESCRIPTION

Retrieves road speed data for a specified list of XD and/or TMC Segments.

EXAMPLE CALL REQUEST

XD segments only:

```
http://na.api.inrix.com/Traffic/Inrix.ashx?Action=GetSegmentSpeed&Token=bn77EgdCuCtoLJpqElbClmHgonclAXWa0Jvn-wljzXw|&Segments=1386654365,1386654376,1386654385&resolution=250&speedvariation=1
```

XD and TMC segments:

```
http://na.api.inrix.com/Traffic/Inrix.ashx?Action=GetSegmentSpeed&Token=S8fPAn5vt6hIOHykd5ulEjlZwquWVEl03tZgo02qB8A|&Segments=1386654365|XDS,1386654376|XDS,1386654385|XDS,111%2B11479|TMC,113-04803|TMC&resolution=250&speedvariation=1
```

EXAMPLE CALL RESPONSE

```
<Inrix docType="GetSegmentSpeed" copyright="Copyright INRIX Inc." versionNumber="10.4" createdDate="2017-07-26T20:20:27Z" statusId="0" statusText="" responseId="57adbc25-06de-4f8b-ad57-4cb2842ab51c">
```

```
  <SegmentSpeedResultSet coverage="255">
```

```
    <SegmentSpeedResults timestamp="2017-07-26T20:20:27Z">
```

```
      <Segment code="1386654365" type="XDS" speed="65" average="64" reference="64" score="30" c-value="100" travelTimeMinutes="0.573" speedBucket="3">
```

```
        <SubSegment speed="69" offset="0,251"/>
```

```
        <SubSegment speed="66" offset="251,501"/>
```

```
        <SubSegment speed="63" offset="752,1002"/>
```

```
      </Segment>
```

```
      <Segment code="1386654376" type="XDS" speed="71" average="66" reference="66" score="30" c-value="100" travelTimeMinutes="0.389" speedBucket="3"/>
```

```
      <Segment code="1386654385" type="XDS" speed="60" average="61" reference="62" score="30" c-value="100" travelTimeMinutes="0.728" speedBucket="3">
```

```
        <SubSegment speed="61" offset="0,235"/>
```

```
        <SubSegment speed="59" offset="235,471"/>
```

```
      </Segment>
```

```
      <Segment code="111+11479" type="TMC" speed="37" average="32" reference="29" score="30" c-value="100" travelTimeMinutes="1.074" speedBucket="3">
```

```
        <SubSegment speed="38" offset="268,537"/>
```

```
        <SubSegment speed="39" offset="537,805"/>
```

```
      </Segment>
```

```
      <Segment code="113-04803" type="TMC" speed="67" average="62" reference="62" score="30" c-value="98" travelTimeMinutes="0.819" speedBucket="3"/>
```

```
    </SegmentSpeedResults>
```

```
  </SegmentSpeedResultSet>
```

```
</Inrix>
```

SYNTAX

ServerPath

?Action=GetSegmentSpeed

&Token=*string*&Segments=*string*&SpeedOutputFields=*string*&Units=*integer*&Resolution=*integer*&SpeedVariation=*integer*

PARAMETERS

ServerPath (Type: string, Required)

The URL of the server to which requests should be addressed as returned in the ServerPath element of the response to a GetSecurityToken request. (Should use the API path.)

Token (Type: string, Required)

A valid security token as returned in the AuthToken element of the response to a GetSecurityToken request.

Segments (Type: string, Required)

XIDs and/or TMCIDs for which to get data. Can be one or more XID and/or TMCID specified in a comma-delimited list. Any duplicates are ignored.

If the list contains only XD segments, just the XIDs may be specified.

If the list contains XD segments and TMCs, the XIDs must be followed by a pipe character ('|') and 'XDS'. (E.g., XID '123456' would be represented as '123456|XDS'.)

If the list contains TMC segments, the TMCIDs must be followed by '|TMC'. The TMC '+' symbol should be replaced with '%2B'. (E.g., TMCID '111+11479' would be represented as '111%2B11479|TMC'.)

SpeedOutputFields (Type: string, Optional)

The traffic speed fields to be returned in the response. Can specify a single field, or multiple fields in a comma-separated list such as "Reference,Speed". If this parameter is not specified, all fields are returned. (Further definition of the output fields is included in the Response Element section below.)

All (default)	Returns all fields.
Reference	Reference speed field.
Speed	Measured speed field.
Average	Historical average speed field.
TTM	Time required to travel across the segment.
Score	Represents the source of the speed data returned.
Confidence	Represents the confidence in the real-time data.
SpeedBucket	Represents the level of congestion. (This field is the same as <i>congestionLevel</i> in the TMC Traffic API calls.)

Units (Type: integer, Optional)

Used for SpeedOutputFields data.

0 (default)	US (English units)
1	Metric units

Resolution (Type: integer, Optional)

This parameter is used to request speeds on sub-segments. It specifies the desired length of sub-segments in meters. If specified, the response will include sub-segments of equal size that are as close as possible to the resolution specified. For instance, if a given a segment has length of 1100 meters and resolution is 250, 4 sub-segments of 275 meters will be returned. 4 sub-segments will be favored over 5 because 5 sub-segments would have resulted in sub-segments length of 220 meters. 220 meters is farther away from the 250 meters than 275 meters. If this parameter is not specified, or is set to 0, no sub-segments are returned.

SpeedVariation (Type: integer, Optional)

This parameter specifies how much the sub-segment speed has to differ from the parent segment speed to be returned. The sub-segment speed is returned if the difference between the parent segment speed and the sub-segment speed is greater than or equal to the speed variation. If this parameter is not specified, it defaults to 0, which means that all sub-segments will be returned without consideration for variance from segment speed.

RETURN VALUES

Each XML response to this API call will contain two basic elements, each having several attributes and/or child elements.

- *Inrix* – The root element in the response for all APIs is always <Inrix>. It contains response data in child elements and information about the request in the following attributes:
 - *docType* identifies the API called and the schema used for the response.
 - *copyright* identifies the INRIX, Inc. copyright.
 - *versionNumber* refers to the version number of the INRIX traffic services platform that produced the response.
 - *createdDate* is the timestamp indicating when the response was created. The timestamp specifies a DateTime value, formatted according to the ISO 8601 standard and expressed in UTC time. The format of the attribute is YYYY-MM-DDTHH:MM:SSZ. The “T” character separates the date from the time, and the “Z” character specifies that UTC time is used.
 - *statusId* indicates whether an API request succeeded or failed. Each *statusId* value is associated with a unique *statusText* value. A *statusId* is intended to be convenient to handle in code. *statusId*=0 is a successful call. [Appendix K](#) contains a list of common statusID responses.
 - *statusText* corresponds to the *statusId*. A *statusText* is intended to be convenient for humans to interpret.
 - *responseID* is logged by INRIX for future use in assisting customer inquiries. Service Requestors can use *responseID* as a way for INRIX to identify the specific API call that was made. It is recommended that Service Requestors log this information to expedite technical support incidents.
- *SegmentSpeedResultSet* – This child element is a container for all of the requested traffic data.

RESPONSE ELEMENTS

SegmentSpeedResultSet (Parent element: Inrix)

A container for SegmentSpeedResults elements.

Syntax: <SegmentSpeedResultSet coverage = “Integer” >
 <SegmentSpeedResults ... />
 </ SegmentSpeedResultSet >

Attribute	Required	Type	Usage
coverage	yes	Integer	Indicates the type of coverage returned. 255=All types of coverage

Element	Required	Occurs	Description
SegmentSpeedResults	yes	1	Represents a collection of traffic data for a specified point in time.

SegmentSpeedResults (Parent element: SegmentSpeedResultSet)

Represents a collection of traffic data for a specified point in time.

Syntax: <SegmentSpeedResults timestamp = "DateTime">
 <Segment... />
 </SegmentSpeedResults>

Attribute	Required	Type	Usage
timestamp	yes	DateTime	The point in time expressed in ISO 8963 format.

Element	Required	Occurs	Description
Segment	no	0..n	Represents traffic data for one XD Segment.

Segment (Parent element: SegmentSpeedResults)

Represents traffic data for one segment.

Syntax: <Segment
 code = "string"
 type = "string"
 speed = "integer"
 average = "integer"
 reference = "integer"
 travelTimeMinutes = "integer"
 score = "integer"
 c-value = "integer"
 speedBucket = "integer"
 segmentClosed = "boolean">
 <SubSegment ... />
 </Segment>

Attribute	Required	Type	Usage
code	yes	string	The XD Segment ID identifying the segment of road for which data is being reported.
type	yes	string	The type of segment. XDS (XD Segment) or TMC.
speed	no	integer	The traffic speed on the segment. This value is 0 if the road segment is closed.

average	no	integer	The historical average speed on the segment at the time of day specified in the SegmentSpeedResults element. If the average speed is not available, this value may be empty.
reference	no	integer	The typical traffic speed on this segment under free flow conditions.
travelTimeMinutes	no	integer	The time in minutes required to traverse the segment.
score	no	integer	The source of the speed data field returned. 30 = Real-time data 20 = Historical data 10 = Reference speed (For more information, see Appendix G.)
c-value	no	integer	A measure of the confidence in the real-time data. Ranges from 0 to 100. The higher the c-value, the more confident the data. (For more information, see Appendix H.)
speedBucket	no	integer	The amount of congestion on the segment. A lower number represents higher congestion. 0 = speed is 0-31% of reference speed 1 = speed is 32-62% of reference speed 2 = speed is 63-92% of reference speed 3 = speed is 93-100% of reference speed 255= road segment is closed (This field is the same as <i>congestionLevel</i> in other Traffic API calls.)
segmentClosed	no	boolean	Indicates whether the segment is closed or not. Speed data is not returned for a closed segment.

Element	Required	Occurs	Description
SubSegment	no	0..n	Represents speed data for one sub-segment.

SubSegment (Parent element: Segment)

Represents speed data for one sub-segment.

Syntax: <SubSegment
 offset = "integer,integer"
 speed = "integer">
 </SubSegment >

Attribute	Required	Type	Usage
offset	no	integer,integer	The start and end offsets of the sub-segment on the parent segment in meters.
speed	no	integer	Current speed on the sub-segment.

NOTES

- If the Resolution parameter is specified, the speeds of sub-segments are returned. The SpeedVariation parameter can be used to limit the sub-segments returned to those that vary from the speed of the parent segment by at least a specified value.
- To help limit the size of the response data file, INRIX suggests that users request a minimum resolution of 250 meters for sub-segments.

GetSegmentSpeedInSet

DESCRIPTION

Retrieves road speed data for one or more Segment Sets, consisting of XD and/or TMC Segments.

EXAMPLE CALL REQUEST

```
http://na.api.inrix.com/Traffic/Inrix.ashx?Action=GetSegmentSpeedInSet&Token=UT63caWPRV9ORerKYPZTk1-jdvOrN*UEQWPSZsdgkpo|&SegmentSetID=1512428504&resolution=250&speedvariation=1
```

EXAMPLE CALL RESPONSE

```
<Inrix docType="GetSegmentSpeedInSet" copyright="Copyright INRIX Inc." versionNumber="10.4" createdDate="2017-07-26T21:48:08Z" statusId="0" statusText="" responselId="4b030ea8-ae6e-417b-8ef2-dd85784b992a">
  <SegmentSpeedResultSet coverage="255">
    <SegmentSpeedResults timestamp="2017-07-26T21:48:08Z">
      <Segment code="1386654365" type="XDS" speed="64" average="64" reference="64" score="30"
        c-value="100" travelTimeMinutes="0.584" delta="0" congestionLevel="3">
        <SubSegment speed="65" offset="0,251"/>
        <SubSegment speed="65" offset="251,501"/>
        <SubSegment speed="63" offset="501,752"/>
        <SubSegment speed="62" offset="752,1002"/>
      </Segment>
      ...
      <Segment code="125N04958" type="TMC" speed="46" average="64" reference="68" score="30"
        c-value="100" travelTimeMinutes="0.831" delta="-18" congestionLevel="2">
        <SubSegment speed="53" offset="0,256"/>
        <SubSegment speed="53" offset="256,513"/>
        <SubSegment speed="44" offset="513,769"/>
        <SubSegment speed="37" offset="769,1025"/>
      </Segment>
    </SegmentSpeedResults>
  </SegmentSpeedResultSet>
</Inrix>
```

SYNTAX

ServerPath

?Action=GetSegmentSpeedInSet

&Token=*string*

&SegmentSetID=*string*

&SpeedOutputFields=*string*

&Units=*integer*

&Resolution=*integer*

&SpeedVariation=*integer*

PARAMETERS

ServerPath (Type: string, Required)

The URL of the server to which requests should be addressed as returned in the ServerPath element of the response to a GetSecurityToken request. (Should use the API path.)

Token (Type: string, Required)

A valid security token as returned in the AuthToken element of the response to a GetSecurityToken request.

SegmentSetID (Type: string, Required)

Segment Set Ids for which to get data. Can be one or more Segment Set Ids specified in a comma-delimited list. Segment Sets can include XDIds, TMCIds, or both. Any duplicate Ids are ignored.

SpeedOutputFields (Type: string, Optional)

The traffic speed fields to be returned in the response. Can specify a single field, or multiple fields in a comma-separated list such as "Reference,Speed". If this parameter is not specified, all fields are returned. (Further definition of the output fields is included in the Response Element section below.)

All (default)	Returns all fields.
Reference	Reference speed field.
Speed	Measured speed field.
Average	Historical average speed field.
Delta	Difference between average and current speed.
TTM	Time required to travel across the segment.
Congestion	Represents the level of congestion.
Score	Represents the source of the speed data returned.
Confidence	Represents the confidence in the real-time data.

Units (Type: integer, Optional)

Used for SpeedOutputFields data.

0 (default)	US (English units)
1	Metric units

Resolution (Type: integer, Optional)

This parameter is used to request speeds on sub-segments. It specifies the desired length of sub-segments in meters. If specified, the response will include sub-segments of equal size that are as close as possible to the resolution specified. For instance, if a given a segment has length of 1100 meters and resolution is 250, 4 sub-segments of 275 meters will be returned. 4 sub-segments will be favored over 5 because 5 sub-segments would have resulted in sub-segments length of 220 meters. 220 meters is farther away from the 250 meters than 275 meters. If this parameter is not specified, or is set to 0, no sub-segments are returned.

SpeedVariation (Type: integer, Optional)

This parameter specifies how much the sub-segment speed has to differ from the parent segment speed to be returned. The sub-segment speed is returned if the difference between the parent segment speed and the sub-segment speed is greater than or equal to the speed variation. If this parameter is not specified, it defaults to 0, which means that all sub-segments will be returned without consideration for variance from segment speed.

RETURN VALUES

Each XML response to this API call will contain two basic elements, each having several attributes and/or child elements.

- *Inrix* – The root element in the response for all APIs is always <Inrix>. It contains response data in child elements and information about the request in the following attributes:
 - *docType* identifies the API called and the schema used for the response.
 - *copyright* identifies the INRIX, Inc. copyright.
 - *versionNumber* refers to the version number of the INRIX traffic services platform that produced the response.
 - *createdDate* is the timestamp indicating when the response was created. The timestamp specifies a DateTime value, formatted according to the ISO 8601 standard and expressed in UTC time. The format of the attribute is YYYY-MM-DDTHH:MM:SSZ. The “T” character separates the date from the time, and the “Z” character specifies that UTC time is used.
 - *statusId* indicates whether an API request succeeded or failed. Each *statusId* value is associated with a unique *statusText* value. A *statusId* is intended to be convenient to handle in code. statusId=0 is a successful call. [Appendix K](#) contains a list of common statusID responses.
 - *statusText* corresponds to the statusId. A statusText is intended to be convenient for humans to interpret.
 - *responseID* is logged by INRIX for future use in assisting customer inquiries. Service Requestors can use *responseID* as a way for INRIX to identify the specific API call that was made. It is recommended that Service Requestors log this information to expedite technical support incidents.
- *SegmentSpeedResultSet* – This child element is a container for all of the requested traffic data.

RESPONSE ELEMENTS

SegmentSpeedResultSet (Parent element: Inrix)

Represents the collection of SegmentSpeedResults for the given coverage.

Syntax: <SegmentSpeedResultSet coverage = “Integer” >
 <SegmentSpeedResults ... />
 </SegmentSpeedResultSet>

Attribute	Required	Type	Usage
coverage	yes	Integer	Indicates the type of coverage returned. 255=All types of coverage

Element	Required	Occurs	Description
SegmentSpeedResults	yes	1	Represents a collection of traffic data for a specified point in time

SegmentSpeedResults (Parent element: SegmentSpeedResultSet)

Represents a collection of traffic data for a specified point in time.

Syntax: <SegmentSpeedResults timestamp = “DateTime”>
 <Segment... />
 </SegmentSpeedResults>

Attribute	Required	Type	Usage
timestamp	yes	DateTime	The point in time expressed in ISO 8963 format.

Element	Required	Occurs	Description
Segment	no	0..n	Represents traffic data for one XD Segment.

Segment (Parent element: SegmentSpeedResults)

Represents traffic data for one segment.

Syntax: <Segment

```

  code = "string"
  type = "string"
  speed = "integer"
  average = "integer"
  reference = "integer"
  travelTimeMinutes = "integer"
  score="integer"
  c-value="integer"
  delta="integer"
  congestionLevel="integer"
  segmentClosed = "boolean">
    <SubSegment ... />

```

</Segment >

Attribute	Required	Type	Usage
code	yes	string	The XD Segment ID identifying the segment of road for which data is being reported.
type	yes	string	The type of segment. XDS (XD Segment) or TMC.
speed	no	integer	The traffic speed on the segment. This value is 0 if the road segment is closed.
average	no	integer	The historical average speed on the segment at the time of day specified in the SegmentSpeedResults element. If the average speed is not available, this value may be empty.
reference	no	integer	The typical traffic speed on this segment under free flow conditions.
travelTimeMinutes	no	integer	The time in minutes required to traverse the segment.
score	no	integer	The source of the speed data field returned. 30 = Real-time data 20 = Historical data 10 = Reference speed (For more information, see Appendix G.)

c-value	no	integer	A measure of the confidence in the real-time data. Ranges from 0 to 100. The higher the c-value, the more confident the data. (For more information, see Appendix H.)
delta	no	integer	The difference between the speed and average values for the segment.
congestionLevel	no	integer	The amount of congestion on the segment. A lower number represents higher congestion. 0 = speed is 0-31% of reference speed 1 = speed is 32-62% of reference speed 2 = speed is 63-92% of reference speed 3 = speed is 93-100% of reference speed 255= road segment is closed
segmentClosed	no	boolean	Indicates whether the segment is closed or not. Speed data is not returned for a closed segment.

Element	Required	Occurs	Description
SubSegment	no	0..n	Represents speed data for one sub-segment.

SubSegment (Parent element: Segment)

Represents speed data for one sub-segment.

Syntax: <SubSegment
 offset = "integer,integer"
 speed = "integer">
 </SubSegment >

Attribute	Required	Type	Usage
offset	no	integer,integer	The start and end offsets of the sub-segment on the parent segment in meters
speed	no	integer	Current speed on the sub-segment.

NOTES

- If the Resolution parameter is specified, the speeds of sub-segments are returned. The SpeedVariation parameter can be used to limit the sub-segments returned to those that vary from the speed of the parent segment by at least a specified value.
- To help limit the size of the response data file, INRIX suggests that users request a minimum resolution of 250 meters for sub-segments.

GetRoadSpeedAtPoints

DESCRIPTION

Send vehicle speed data and location and receive relevant traffic information for vehicle location.

EXAMPLE CALL REQUEST

```
http://na.api.inrix.com/Traffic/Inrix.ashx?Action=GetRoadSpeedAtPoints&Token=LNMQ1TkR79A17G5*9gCkrjncTgzSW6oeMe9FtLRMTM|&pointId=9776943440&timeStamp=2011-07-27T22:03:01.173&latitude=39.130300&longitude=-94.507580&heading=92
```

EXAMPLE CALL RESPONSE

```
<Inrix docType="GetRoadSpeedAtPoints" copyright="Copyright INRIX Inc." versionNumber="10.4" createdDate="2017-07-26T22:13:33Z" statusId="0" statusText="OK" responseId="6f4cc0fb-a6bf-464e-9f47-e54f8b904431">
  <RoadSpeedResultSet>
    <RoadSpeedAtPoint snapPrecision="1" pointId="9776943440">
      <RoadInfo FRCLevel="3">
        <Name>E Front St</Name>
      </RoadInfo>
      <Speed code="119+14145" speed="26" reference="31" speedProxy="31" average="27"
        delta="-1" travelTimeMinutes="2.6521273295084637" congestionLevel="2" unit="MPH"/>
    </RoadSpeedAtPoint>
  </RoadSpeedResultSet>
</Inrix>
```

SYNTAX

ServerPath

?Action=GetRoadSpeedAtPoints

&Token=*string*

&RequestID=*string*

&PointID=*string*

&TimeStamp=*datetime*

&Latitude=*double*

&Longitude=*double*

&Heading=*short*

&SpeedOutputFields=*string*

PARAMETERS

ServerPath (Type: string, Required)

The URL of the server to which requests should be addressed as returned in the *ServerPath* element of the response to a *GetSecurityToken* request. (Should use the API path.)

Token (Type: string, Required)

A valid security token as returned in the AuthToken element of the response to a GetSecurityToken request.

PointID (Type: string, Required)

Unique number which is defined by the caller and returned in the response. Can be used to match up the response to the original request. Can be unique per requestID (batch) or unique across a system (like a GUID).

TimeStamp (Type: DateTime, Required)

UTC Time the point was collected in ISO 8601 format.

Latitude (Type: Double, Required)

Latitude in WGS98 format.

Longitude (Type: Double, Required)

Longitude in WGS98 format.

Heading (Type: Short, Required)

SpeedOutputFields (Type: string, Optional)

The traffic speed fields to be returned in the response. Can specify a single field, or multiple fields in a comma-separated list such as "Reference,Speed". If this parameter is not specified, all fields are returned. (Further definition of the output fields is included in the Response Element section below.)

All (default)	Returns all fields.
Reference	Reference speed field.
Speed	Measured speed field.
Average	Historical average speed field.
Delta	Difference between average and current speed.
TTM	Time required to travel across the segment.
Congestion	Represents the level of congestion.

RETURN VALUES

Each XML response to this API call will contain two basic elements, each having several attributes and/or child elements.

- *Inrix* – The root element in the response for all APIs is always <Inrix>. It contains response data in child elements and information about the request in the following attributes:
 - *docType* identifies the API called and the schema used for the response.
 - *copyright* identifies the INRIX, Inc. copyright.
 - *versionNumber* refers to the version number of the INRIX traffic services platform that produced the response.
 - *createdDate* is the timestamp indicating when the response was created. The timestamp specifies a DateTime value, formatted according to the ISO 8601 standard and expressed in UTC time. The format of the attribute is YYYY-MM-DDTHH:MM:SSZ. The "T" character separates the date from the time, and the "Z" character specifies that UTC time is used.
 - *statusId* indicates whether an API request succeeded or failed. Each *statusId* value is associated with a unique *statusText* value. A *statusId* is intended to be convenient to handle in code. *statusId*=0 is a successful call. [Appendix K](#) contains a list of common statusID responses.
 - *statusText* corresponds to the *statusId*. A *statusText* is intended to be convenient for humans to interpret.

— *responseID* is logged by INRIX for future use in assisting customer inquiries. Service Requestors can use *responseID* as a way for INRIX to identify the specific API call that was made. It is recommended that Service Requestors log this information to expedite technical support incidents.

- *RoadSpeedResultSet* – This child element is a container for all of the requested traffic data.

RESPONSE ELEMENTS

RoadSpeedResultSet (Parent element: Inrix)

A container for RoadSpeedResults elements.

```
Syntax: <RoadSpeedResultSet>
         <RoadSpeedAtPoint ... />
</RoadSpeedResultSet>
```

Element	Required	Occurs	Description
RoadSpeedAtPoint	yes	1..n	Information about the snapped road at the point provided.

RoadSpeedAtPoint (Parent element: RoadSpeedResultSet)

Information about the snapped road at the point provided.

```
Syntax: < RoadSpeedAtPoint
         pointid = "string"
         snapPrecision = "boolean">
         <RoadInfo ... />
         <Speed ... />
</RoadSpeedAtPoint >
```

Attribute	Required	Type	Usage
pointid	yes	String	ID of the point provided.
snapPrecision	yes	Integer	Value between 1 and 5 that indicates the accuracy of the TMC the road is snapped to. A value of 1 indicates a solid match.

Element	Required	Occurs	Description
RoadInfo	yes	1	Defines information about the road segment of the snapped TMC.
Speed	yes	1	Defines traffic information for the road segment of the snapped TMC.

RoadInfo (Parent elements: RoadSpeedAtPoint)

Defines information about the road segment of the snapped TMC.

```
Syntax: <RoadInfo
    FRCLevel= "integer">
    <Name ... />
</RoadInfo>
```

Attribute	Required	Type	Usage
FRCLevel	no	integer	Federal Road Classification (FRC) code. Indicates the size of the road: 1= First class roads, such as national highway network roads 2= Second class roads, such as state highway network roads 3= Third class roads, such as state interconnecting network roads 4= Fourth class roads, such as major connecting roads 5= Fifth class roads, such as minor roads connecting suburbs 6= Sixth class roads, such as destination and destination collector roads 7= Seventh class roads, such as destination dead-end roads

Element	Required	Occurs	Description
Name	no	1..n	Defines the name of the road.

Speed (Parent element: RoadSpeedAtPoint)

Defines traffic information for the road segment of the snapped TMC.

```
Syntax: <Speed
    code = "string"
    speedproxy="byte"
    speed = "byte"
    reference = "short"
    average = "byte"
    delta = "byte"
    travelTimeMinutes = "decimal"
    congestionLevel = "byte"
    unit = "string"
    segmentClosed = "boolean"
/>
```

Attribute	Required	Type	Usage
code	yes	String	TMCId associated with the road segment.
speedproxy	yes	String	Uncapped reference speed.
speed	no	byte	Traffic speed on the TMC Segment. This value is 0 if the road segment is closed.
reference	no	short	Typical traffic speed on the TMC Segment under free flow conditions.
average	no	byte	Historical average speed on the TMC Segment at the time of day specified in the RoadSpeedResults element.
delta	no	byte	Difference between the speed and average values for the TMC Segment.
travelTimeMinutes	no	decimal	Time in minutes required to traverse the TMC Segment.
congestionLevel	no	byte	The amount of congestion on the segment. A lower number represents higher congestion. 0 = speed is 0-31% of reference speed 1 = speed is 32-62% of reference speed 2 = speed is 63-92% of reference speed 3 = speed is 93-100% of reference speed 255= road segment is closed
unit	no	string	MPH. Included if a speed attribute is returned.
segmentClosed	no	boolean	Indicates whether the segment is closed or not. Speed data is not returned for a closed segment.

GetTrafficTileForSets

This API will be deprecated in the near future. Use the [GetTrafficTileForSegmentSets](#) API instead.

DESCRIPTION

Generates a Traffic Tile overlay with an area specified by a Bing Maps Tile System quadkey or by the corners of a bounding box for one or more specified TMC Segment Sets.

EXAMPLE CALL REQUEST

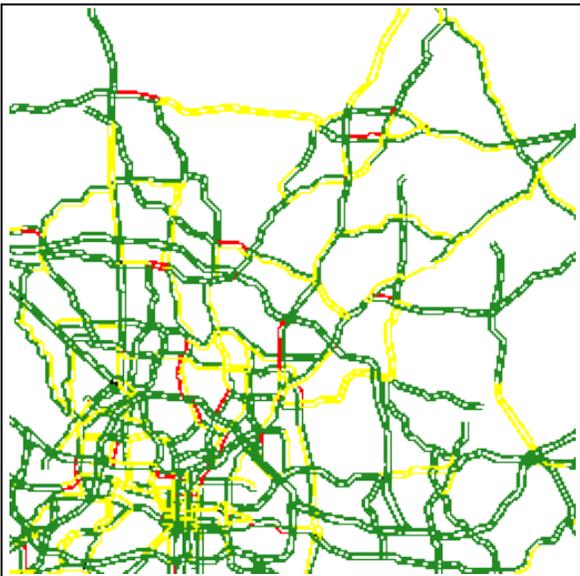
With quadkey:

```
http://na-rseg-tts.inrix.com/RsegTiles/tile.ashx?action=GetTrafficTileForSets&token=Az0UcqKgMXJOBlaADq2bRtbPhPb  
cicz-nyHiJMS2Vm4|&quadkey=0320120020&TMCSetIds=84632162
```

With bounding box:

```
http://na-rseg-tts.inrix.com/RsegTiles/tile.ashx?action=GetTrafficTileForSets&token=Az0UcqKgMXJOBlaADq2bRtbPhPb  
cicz-nyHiJMS2Vm4|&corner1=35.74651225991851|-78.75&corner2=36.03133177633187|-78.3984375  
&TMCSetIds=84632162
```

EXAMPLE CALL RESPONSE



SYNTAX

ServerPath

?Action=GetTrafficTileForSets

&Token=*string*

&Quadkey=*string*

&Corner1=*string*

&Corner2=*string*

&PenWidth=*integer*
 &Height=*string*
 &Width=*string*
 &Format=*string*
 &FRCLLevel=*string*
 &SpeedBucketID=*integer*
 &Resolution= *integer*
 &TMCSetsIds= *list*

PARAMETERS

ServerPath (Type: string, Required)

The URL of the server to which requests should be addressed as returned in the ServerPath element of the response to a GetSecurityToken request. (Should use the TTS path.)

Token (Type: string, Required)

A valid security token as returned in the AuthToken element of the response to a GetSecurityToken request.

Quadkey (Type: string, Required)

A Bing Maps Tile System quadkey that corresponds to the area for which to get data. The minimum quadkey length is 6 digits and the maximum is 17. A quadkey request always returns a 256x256 image. (For more information about using quadkeys, see Appendix I.)

OR

Corner1 (Type: string, Required)

The first corner of the region for which to get data. This parameter must be specified as a pair of latitude and longitude values separated by a pipe character (|), in the form Corner1="latitude|longitude". The latitude and longitude values are expressed using the WGS 84 datum. Northern latitudes are positive and southern latitudes are negative. Eastern hemisphere longitudes are positive and western hemisphere longitudes are negative. Longitudes in North America are negative. The corner specified by the Corner1 parameter can be any of the four corners of the bounding rectangle.

Corner2 (Type: string, Required)

The second corner of the region for which to get data. The Corner2 parameter is the geometric opposite of Corner1.

PenWidth (Type: float, Optional)

The pen width of the lines on the Traffic Tile overlay, in pixels. The default is 4.

Height (Type: integer, Optional)

The height in pixels of the generated Tile. The default is 256.

Width (Type: integer, Optional)

The width in pixels of the generated Tile. The default is 256.

Format (Type: string, Optional)

The format of the Traffic Tile to be returned. The default is .GIF.

GIF (default)	Return the tile in GIF format.
PNG	Return the tile in PNG format.

FRCLevel (Type: string, Optional)

The Federal Road Classification code of the roads to report. Multiple codes can be specified in a comma-delimited list.

All (default)	All road types.
1	First class roads, such as national highway network roads.
2	Second class roads, such as state highway network roads.
3	Third class roads, such as state interconnecting network roads.
4	Fourth class roads, such as major connecting roads.
5	Fifth class roads, such as minor roads connecting suburbs
6	Sixth class roads, such as destination and destination collector roads.
7	Seventh class roads, such as destination dead-end roads.

SpeedBucketID (Type: integer, Optional)

A Speed Bucket is a range of speeds or percentages that is used to categorize data and specify how it should be displayed on a Traffic Tile. All Speed Buckets are identified by a unique number called a SpeedBucketID.

If no Speed Bucket is specified, the default is the INRIX-created Speed Bucket with SpeedBucketID=1, which is defined as:

- Color=black, if current speed is 0-31% of reference speed
- Color=red, if current speed is 32-62% of reference speed
- Color=yellow, if current speed is 63-92% of reference speed
- Color=green, if current speed is 92-100% of reference speed

(For more information about creating custom Speed Buckets, see [Appendix F.](#))

Resolution (Type: integer, Optional)

This parameter is used to request speeds on sub-segments. It specifies the desired length of sub-segments in meters. If specified, the response will include sub-segments of equal size that are as close as possible to the resolution specified. For instance, if a given a Segment has length of 1100 meters and resolution is 250, 4 sub-segments of 275 meters will be returned. 4 sub-segments will be favored over 5 because 5 sub-segments would have resulted in sub-segments length of 220 meters. 220 meters is farther away from the 250 meters than 275 meters. If this parameter is not specified, or is set to 0, no sub-segments are returned.

TMCSetIds (Type: list, *Required*)

TMCSetIds for which to get data. Can be one or more TMCSetIds specified in a comma-delimited list. Any duplicate TMCIds are ignored.

RETURN VALUES

A GIF or PNG image whose size, bounds, and contents are determined by the parameters specified in API call.

NOTES

- There is a limit to how large of an area a tile can be requested for. This is currently set to .03 degrees per pixel. Any request for a larger tile will return an error.
- There is a limit to how small of an area a tile can be requested for. This is currently set to 0.00001063 degrees per pixel. Any request for a smaller tile will return an error.
- Due to the underlying segment architecture and available granularity of the data, INRIX suggests that users request a minimum resolution of 100 meters for sub-segments when making Tile calls.

GetTrafficTileForSegmentSets

DESCRIPTION

Generates a Traffic Tile overlay with an area specified by a Bing Maps Tile System quadkey or by the corners of a bounding box for one or more specified Segment Sets, consisting of XD and/or TMC Segments.

EXAMPLE CALL REQUEST

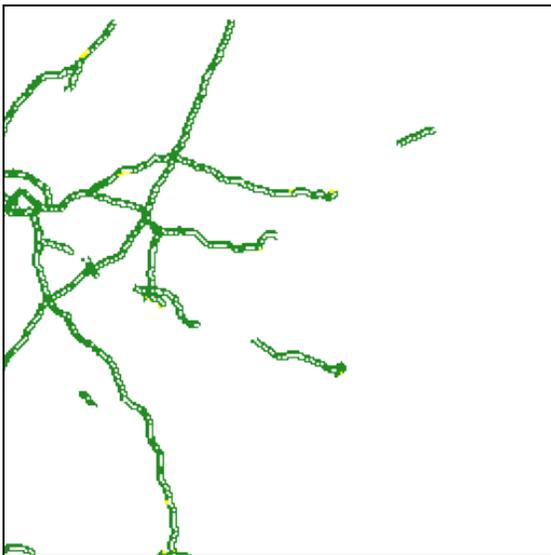
With quadkey:

```
http://na-rseg-tts.inrix.com/RsegTiles/tile.ashx?action=GetTrafficTileForSegmentSets&token=tFdTXBRjshhfEGtOxTTRRpVCch8Yah4FnjAdPgoTauY|&quadkey=0320120&SegmentSetIds=2061742903
```

With bounding box:

```
http://na-rseg-tts.inrix.com/RsegTiles/tile.ashx?action=GetTrafficTileForSegmentSets&token=tFdTXBRjshhfEGtOxTTRRpVCch8Yah4FnjAdPgoTauY|&corner1=34.30714385628804|-78.75&corner2=36.597889133070225|-75.9375 &SegmentSetIds=2061742903
```

EXAMPLE CALL RESPONSE



SYNTAX

ServerPath

?Action= GetTrafficTileForSegmentSets

&Token=*string*

&Quadkey=*string*

&Corner1=*string*

&Corner2=*string*

&PenWidth=*integer*

&Height=*string*

&Width=*string*
 &Format=*string*
 &FRCLLevel=*string*
 &SpeedBucketID=*integer*
 &Resolution= *integer*
 & SegmentSetIds = *list*

PARAMETERS

ServerPath (Type: string, Required)

The URL of the server to which requests should be addressed as returned in the ServerPath element of the response to a GetSecurityToken request. (Should use the TTS path.)

Token (Type: string, Required)

A valid security token as returned in the AuthToken element of the response to a GetSecurityToken request.

Quadkey (Type: string, Required)

A Bing Maps Tile System quadkey that corresponds to the area for which to get data. The minimum quadkey length is 6 digits and the maximum is 17. A quadkey request always returns a 256x256 image. (For more information about using quadkeys, see Appendix I.)

OR

Corner1 (Type: string, Required)

The first corner of the region for which to get data. This parameter must be specified as a pair of latitude and longitude values separated by a pipe character (|), in the form Corner1="latitude|longitude". The latitude and longitude values are expressed using the WGS 84 datum. Northern latitudes are positive and southern latitudes are negative. Eastern hemisphere longitudes are positive and western hemisphere longitudes are negative. Longitudes in North America are negative. The corner specified by the Corner1 parameter can be any of the four corners of the bounding rectangle.

Corner2 (Type: string, Required)

The second corner of the region for which to get data. The Corner2 parameter is the geometric opposite of Corner1.

PenWidth (Type: float, Optional)

The pen width of the lines on the Traffic Tile overlay, in pixels. The default is 4.

Height (Type: integer, Optional)

The height in pixels of the generated Tile. The default is 256.

Width (Type: integer, Optional)

The width in pixels of the generated Tile. The default is 256.

Format (Type: string, Optional)

The format of the Traffic Tile to be returned. The default is .GIF.

GIF (default)	Return the tile in GIF format.
PNG	Return the tile in PNG format.

FRCLevel (Type: string, Optional)

The Federal Road Classification code of the roads to report. Multiple codes can be specified in a comma-delimited list.

All (default)	All road types.
1	First class roads, such as national highway network roads.
2	Second class roads, such as state highway network roads.
3	Third class roads, such as state interconnecting network roads.
4	Fourth class roads, such as major connecting roads.
5	Fifth class roads, such as minor roads connecting suburbs
6	Sixth class roads, such as destination and destination collector roads.
7	Seventh class roads, such as destination dead-end roads.

SpeedBucketID (Type: integer, Optional)

A Speed Bucket is a range of speeds or percentages that is used to categorize data and specify how it should be displayed on a Traffic Tile. All Speed Buckets are identified by a unique number called a SpeedBucketID.

If no Speed Bucket is specified, the default is the INRIX-created Speed Bucket with SpeedBucketID=1, which is defined as:

- Color=black, if current speed is 0-31% of reference speed
- Color=red, if current speed is 32-62% of reference speed
- Color=yellow, if current speed is 63-92% of reference speed
- Color=green, if current speed is 92-100% of reference speed

(For more information about creating custom Speed Buckets, see [Appendix F.](#))

Resolution (Type: integer, Optional)

This parameter is used to request speeds on sub-segments. It specifies the desired length of sub-segments in meters. If specified, the response will include sub-segments of equal size that are as close as possible to the resolution specified. For instance, if a given a Segment has length of 1100 meters and resolution is 250, 4 sub-segments of 275 meters will be returned. 4 sub-segments will be favored over 5 because 5 sub-segments would have resulted in sub-segments length of 220 meters. 220 meters is farther away from the 250 meters than 275 meters. If this parameter is not specified, or is set to 0, no sub-segments are returned.

SegmentSetIds (Type: list, *Required*)

Segment Set Ids for which to get data. Can be one or more Segment Set Ids specified in a comma-delimited list. Segment Sets can include XDIds, TMCIds, or both. Any duplicate Ids are ignored.

RETURN VALUES

A GIF or PNG image whose size, bounds, and contents are determined by the parameters specified in the API call.

NOTES

- There is a limit to how large of an area a tile can be requested for. This is currently set to .03 degrees per pixel. Any request for a larger tile will return an error.
- There is a limit to how small of an area a tile can be requested for. This is currently set to 0.00001063 degrees per pixel. Any request for a smaller tile will return an error.
- Due to the underlying segment architecture and available granularity of the data, INRIX suggests that users request a minimum resolution of 100 meters for sub-segments when making Tile calls.

GetXDIncidentsInSegments

DESCRIPTION

Retrieves traffic-related incidents for a specified list of XD and/or TMC Segments.

EXAMPLE CALL REQUEST

```
http://na.api.inrix.com/Traffic/Inrix.ashx?Action=GetXDIncidentsInSegments&Token=InnZGDC0VuryDZy*OZA*HO69wq  
wq3xSvwdxhhLhXrp4|&Segments=125P05270|tmc,1386729936|xds,1386708057|xds&IncidentOutputFields=All  
&LocRefMethod=TMC,XD
```

EXAMPLE CALL RESPONSE

```
<Inrix docType="GetXDIncidentsInSegments" copyright="Copyright INRIX Inc." versionNumber="11.1" createdDate=  
"2017-12-04T20:46:36Z" statusId="0" statusText="" responseId="32e87f66-170b-4011-b4e4-fca275a86c9b">  
  <XDIncidents>  
    <XDIncident id="112422" version="2" type="1" severity="2" latitude="36.477728" longitude="-  
78.237807" impacting="N" status="active">  
      <Messages>  
        <AlertCMessageCodes>  
          <EventCode level="Primary" quantifierType="0">701</EventCode>  
          <EventCode level="Secondary" quantifierType="0">701</EventCode>  
        </AlertCMessageCodes>  
        <InrixMessage>  
          <InrixCode type="Effect">701</InrixCode>  
          <InrixCode type="Information">292</InrixCode>  
          <InrixCode type="Cause">1037</InrixCode>  
        </InrixMessage>  
      </Messages>  
      <Location countryCode="1" direction="both ways" biDirectional="true">  
        <Segment type="XDS" code="1386708057" offset="211,221"/>  
        <Segment type="XDS" code="1386877262" offset="593,603"/>  
        <Segment type="TMC" code="125+05471" offset="1794,1804"/>  
        <Segment type="TMC" code="125-05470" offset="2159,2169"/>  
      </Location>  
      <Schedule planned="true" advanceWarning="false">  
        <OccurrenceStartTime>2015-04-28T11:00:00Z</OccurrenceStartTime>  
        <OccurrenceEndTime>2019-07-01T21:00:00Z</OccurrenceEndTime>  
        <Description xml:lang="en-US">  
          Starts at 4/28/2015 7:00 AM, ends at 7/1/2019 5:00 PM.  
        </Description>  
      </Schedule>  
      <Descriptions>  
        <Desc xml:lang="en-US" type="short">  
          I-85: intermittent lane closures at I-85 Exit 229 / Oine Rd  
        </Desc>
```

```

<Desc xml:lang="en-US" type="long">
  Intermittent lane closures due to construction work on I-85 both ways at I-85 Exit 229 / Oine Rd.
  Expect delays.
</Desc>
<Desc xml:lang="en-US" type="Text-to-Speech">
  There are intermittent lane closures because of construction work on I-85 both ways at I-85 Exit 229.
  Expect delays.
</Desc>
<ParameterizedDescription eventCode="701">
  <EventText>
    Intermittent Lane Closures, expect Delays, construction work
  </EventText>
  <RoadName>I-85</RoadName>
  <Direction>both ways</Direction>
  <ToLocation>Oine</ToLocation>
  <Crossroad2 tmcCode="125P00000">I-85 Exit 229 / Oine Rd</Crossroad2>
  <Position1>at</Position1>
  <Position2/>
</ParameterizedDescription>
</Descriptions>
<Head latitude="36.477728" longitude="-78.237807"/>
<DLRs type="XDSegment">
  <DLR offset="211,221">1386708057</DLR>
  <DLR offset="593,603">1386877262</DLR>
</DLRs>
<RDS alertcMessage="00C2BD155F004957A00000000000000000000000000000000000000000000000" tmcLocation="5471"
tmcCountry="1" tmcRegion="25" direction="1" extent="0" duration="865611" diversion="false" directionality
yChanged="false">
  <EventCodes>
    <EventCode value="701" primary="true"/>
    <EventCode value="701" primary="false"/>
  </EventCodes>
</RDS>
<DelayImpact fromTypicalMinutes="0.00" fromFreeFlowMinutes="0.00" distance="0.00"/>
</XDIncident>
<XDIncident id="61103688" version="8" type="3" severity="1" latitude="35.170351" longitude="-
80.887715" impacting="Y" status="active">
  <Messages>
    <AlertCMessageCodes>
      <EventCode level="Primary">74</EventCode>
    </AlertCMessageCodes>
    <InrixMessage>
      <InrixCode type="Effect">986</InrixCode>
      <InrixCode type="Effect" quantifierData="2" quantifierType="minute(s),
minute(s)">989</InrixCode>
      <InrixCode type="Information" quantifierData="125" quantifierType="severity
value">1000</InrixCode>

```

```

    <InrixCode type="Information" quantifierData="-0.04" quantifierType="percentage,
    %">998</InrixCode>
    <InrixCode type="Information" quantifierData="0.05" quantifierType="percentage,
    %">997</InrixCode>
    <InrixCode type="Information" quantifierData="-3.55" quantifierType="speed,
    mph">996</InrixCode>
    <InrixCode type="Information" quantifierData="84" quantifierType="severity
    value">995</InrixCode>
    <InrixCode type="Information" quantifierData="48.75 / 14.82 / 1" quantifierType="speed,
    mph">991</InrixCode>
    <InrixCode type="Information" quantifierData="29.2" quantifierType="speed, mph">11</InrixCode>
  </InrixMessage>
</Messages>
<Location countryCode="1" direction="Northbound" biDirectional="false">
  <Segment type="XDS" code="1386729936" offset="1208,1367"/>
  <Segment type="XDS" code="1386831031" offset="0,1006"/>
  <Segment type="XDS" code="1386745798" offset="0,320"/>
  <Segment type="XDS" code="1386904189" offset="0,636"/>
  <Segment type="XDS" code="1386908421" offset="0,979"/>
  <Segment type="XDS" code="1386700530" offset="0,1012"/>
  <Segment type="XDS" code="1386747907" offset="0,397"/>
  <Segment type="TMC" code="125P04777" offset="508,1673"/>
  <Segment type="TMC" code="125P04776" offset="1978,2137"/>
  <Segment type="TMC" code="125+04778" offset="0,320"/>
  <Segment type="TMC" code="125P04778" offset="0,636"/>
  <Segment type="TMC" code="125+04779" offset="0,1018"/>
  <Segment type="TMC" code="125P04779" offset="0,897"/>
  <Segment type="TMC" code="125+04780" offset="0,474"/>
</Location>
<Schedule planned="false" advanceWarning="false">
  <OccurrenceStartTime>2017-12-04T20:01:00Z</OccurrenceStartTime>
  <OccurrenceEndTime>2017-12-04T21:16:40Z</OccurrenceEndTime>
  <Description xml:lang="en-US">
    Starts at 12/4/2017 3:01 PM, ends at 12/4/2017 4:16 PM.
  </Description>
</Schedule>
<Descriptions>
  <Desc xml:lang="en-US" type="short">
    I-77 N/B: delays increasing between Exits 1A,3 I-485 and Exits 6A,6B Woodlawn Rd
  </Desc>
  <Desc xml:lang="en-US" type="long">
    Delays increasing and delays of two minutes on I-77 Northbound between Exits 1A,3 I-485 and Exits
    6A,6B Woodlawn Rd. Average speed 30 mph.
  </Desc>
  <Desc xml:lang="en-US" type="Text-to-Speech">
    Delays are increasing and there are delays of two minutes on I-77 Northbound between Exits 1A,3 I-
    485 and Exits 6A,6B Woodlawn Road. Average speed 30 mph.
  </Desc>

```

```

<ParameterizedDescription eventCode="74">
  <EventText>
    XDFI Delays increasing, xDFI Delays, xDFI Average Speed
  </EventText>
  <RoadName>I-77</RoadName>
  <Direction>Northbound</Direction>
  <Crossroad1 tmcCode="125P04776">I-485 Exit 67 / I-77 Exits 1A,3</Crossroad1>
  <Crossroad2 tmcCode="125P04780">I-77 Exits 6A,6B / Woodlawn Rd / Holiday Inn Dr</Crossroad2>
  <Position1>between</Position1>
  <Position2>and</Position2>
</ParameterizedDescription>
</Descriptions>
<Head latitude="35.170351" longitude="-80.887715"/>
<Tails>
  <Tail latitude="35.135312" longitude="-80.910100"/>
</Tails>
<LastDetourPoints>
  <LastDetourPoint latitude="35.120029" longitude="-80.920508"/>
</LastDetourPoints>
<DLRs type="XDSegment">
  <DLR offset="1208,1367">1386729936</DLR>
  <DLR offset="0,1006">1386831031</DLR>
  <DLR offset="0,320">1386745798</DLR>
  <DLR offset="0,636">1386904189</DLR>
  <DLR offset="0,979">1386908421</DLR>
  <DLR offset="0,1012">1386700530</DLR>
  <DLR offset="0,397">1386747907</DLR>
</DLRs>
<RDS alertcMessage="08604A12AC" tmcLocation="4780" tmcCountry="1" tmcRegion="25" direction="1" extent="4" duration="35" diversion="false" directionalityChanged="false">
  <EventCodes>
    <EventCode value="74" primary="true"/>
  </EventCodes>
</RDS>
<DelayImpact fromTypicalMinutes="1.27" fromFreeFlowMinutes="3.05" fromNas="0" distance="2.80" abnormal="false"/>
</XDIncident>
<XDIncident id="61102661" version="3" type="4" severity="3" latitude="36.400329" longitude="-77.64397" impacting="Y" status="active">
  <Messages>
    <AlertCMessageCodes>
      <EventCode level="Primary" quantifierType="0">201</EventCode>
    </AlertCMessageCodes>
    <InrixMessage>
      <InrixCode type="Cause">2</InrixCode>
    </InrixMessage>
  </Messages>
  <Location countryCode="1" direction="Northbound" biDirectional="false">

```

```

    <Segment type="XDS" code="1386799752" offset="653,663"/>
    <Segment type="TMC" code="125P05270" offset="653,663"/>
  </Location>
  <Schedule planned="false" advanceWarning="false">
    <OccurrenceStartTime>2017-12-04T19:43:24Z</OccurrenceStartTime>
    <OccurrenceEndTime>2017-12-04T21:01:55Z</OccurrenceEndTime>
    <Description xml:lang="en-US">
      Starts at 12/4/2017 2:43 PM, ends at 12/4/2017 4:01 PM.
    </Description>
  </Schedule>
  <Descriptions>
    <Desc xml:lang="en-US" type="short">I-95 N/B: accident after Exit 171 NC-125</Desc>
    <Desc xml:lang="en-US" type="long">Accident on I-95 Northbound after Exit 171 NC-125.</Desc>
    <Desc xml:lang="en-US" type="Text-to-Speech">
      There has been an accident on I-95 Northbound after Exit 171 NC-125.
    </Desc>
    <ParameterizedDescription eventCode="201">
      <EventText>Accident</EventText>
      <RoadName>I-95</RoadName>
      <Direction>Northbound</Direction>
      <ToLocation>Weldon</ToLocation>
      <Crossroad2 tmcCode="125P00000">I-95 Exit 171 / NC-125</Crossroad2>
      <Position1>after</Position1>
      <Position2/>
    </ParameterizedDescription>
  </Descriptions>
  <Head latitude="36.400329" longitude="-77.643970"/>
  <Tails>
    <Tail latitude="36.395731" longitude="-77.645981"/>
  </Tails>
  <LastDetourPoints>
    <LastDetourPoint latitude="36.395731" longitude="-77.645981"/>
  </LastDetourPoints>
  <DLRs type="XDSegment">
    <DLR offset="653,663">1386799752</DLR>
  </DLRs>
  <RDS alertcMessage="0840C91496" tmcLocation="5270" tmcCountry="1" tmcRegion="25" direction="1" ext
ent="0" duration="26" diversion="false" directionalityChanged="false">
    <EventCodes>
      <EventCode value="201" primary="true"/>
    </EventCodes>
  </RDS>
  <DelayImpact fromTypicalMinutes="0.00" fromFreeFlowMinutes="0.00" distance="0.00"/>
</XDIncident>
</XDIncidents>
</Inrix>

```

SYNTAX

ServerPath

?Action=GetXDIncidentsInSegments
 &Token=*string*
 &Segments=*string*
 &IncidentType=*string*
 &Severity=*string*
 &IncidentOutputFields=*string*
 &LocRefMethod=*string*

PARAMETERS

ServerPath (Type: string, Required)

The URL of the server to which requests should be addressed as returned in the ServerPath element of the response to a GetSecurityToken request. (Should use the API path.)

Token (Type: string, Required)

A valid security token as returned in the AuthToken element of the response to a GetSecurityToken request.

Segments (Type: integer, Required)

Segments for which to get data. Can be one or more XDIDs and/or TMCIDs specified in a comma-delimited list. Any duplicate Ids are ignored. In the call, XDIDs must be followed by “|xds” and TMCIDs must be followed by “|tmc”.

IncidentType (Type: string, Optional)

The types of incidents to return. Multiple incident types can be specified in a comma-delimited list. If this parameter is not specified, all fields are returned.

Incidents	All unusual incidents that may slow down traffic such as a car accident.
Construction	Construction incidents.
Events	Unusual events slated for the area such as a major sporting event.
Flow	Reports about the slowing down of traffic on route.

Severity (Type: string, Optional)

Filters incident reports based on severity level. This value can be in the range of 0-4, with 4 indicating the highest severity. Multiple severity values can be specified in a comma-delimited list. If this parameter is not specified, all severity levels are returned.

IncidentOutputFields (Type: string, Optional)

The incident fields to be returned in the response. Multiple fields can be specified in a comma-delimited list. If this parameter is not specified, all fields except “RDS”, “DelayImpact” and “HeadTail” are returned. To return all fields, including new fields added in the future, specify “All”.

All	Returns all fields.
ID	Unique identifier of an incident.
Version	Version number of the incident report, incremented each time an incident report is updated.

Type	Type of the incident (Incidents, Construction, Events, Flow). Incidents can be determined from the Alert-C event code: Construction indicates the presence of road construction; Events can be weather-related or a scheduled sporting/public event; and Flow indicates a blocking incident.
Severity	Severity of the incident. This value can be in the range of 0-4, with 4 indicating the highest severity.
EventCode	Event code of the incident. These are standard Alert-C event codes.
LatLong	Latitude and longitude of the incident.
Impacting	Whether the incident impacts traffic flow. This field is set to “yes” if the appearance of the incident changes the traffic flow below a certain percentage from that which is normally expected for the given segment of road at that time, given the current conditions.
StartTime	Starting time of the incident.
EndTime	Ending time of the incident.
ShortDescription	Short textual description of the incident including language code.
FullDescription	Longer textual description of the incident including language code.
TMCs	TMC Segments located at the incident.
ParameterizedDescription	Detailed information about the event specified by the eventCode.
RDS	Radio Data System data.
DelayImpact	Information about the delay caused by the event.
HeadTail	Location of the furthest point on the road where the incident starts, location of the point behind the head where the incident ends, and location of the last detour point to avoid the incident.

LocRefMethod (Type: string, Optional)

The location reference method returned for the incidents. Multiple types can be specified in a comma-delimited list. If this parameter is not specified, it defaults to TMC.

TMC (default)	Specify TMCs located at the incident.
XD	Specify XD Segments located at the incident.

RETURN VALUES

Each XML response to this API call will contain two basic elements, each having several attributes and/or child elements.

- *Inrix* – The root element in the response for all APIs is always <Inrix>. It contains response data in child elements and information about the request in the following attributes:
 - *docType* identifies the API called and the schema used for the response.
 - *copyright* identifies the INRIX, Inc. copyright.
 - *versionNumber* refers to the version number of the INRIX traffic services platform that produced the response.
 - *createdDate* is the timestamp indicating when the response was created. The timestamp specifies a DateTime value, formatted according to the ISO 8601 standard and expressed in UTC time. The format of the attribute is YYYY-MM-DDTHH:MM:SSZ. The “T” character separates the date from the time, and the “Z” character specifies that UTC time is used.

- *statusId* indicates whether an API request succeeded or failed. Each *statusId* value is associated with a unique *statusText* value. A *statusId* is intended to be convenient to handle in code. *statusId*=0 is a successful call. [Appendix K](#) contains a list of common *statusId* responses.
- *statusText* corresponds to the *statusId*. A *statusText* is intended to be convenient for humans to interpret.
- *responseId* is logged by INRIX for future use in assisting customer inquiries. Service Requestors can use *responseId* as a way for INRIX to identify the specific API call that was made. It is recommended that Service Requestors log this information to expedite technical support incidents.
- *XDIncidents* – This child element is a container for all of the requested incident data.

RESPONSE ELEMENTS

XDIncidents (Parent element: Inrix)

A container for incident data.

Syntax: <XDIncidents>
 <XDIncident ... />
 </XDIncidents>

Element	Required	Occurs	Description
XDIncident	no	0..n	Represents a single traffic incident.

XDIncident (Parent element: XDIncidents)

Represents a single traffic incident.

Syntax: <Incident
 id = "string"
 version = "version"
 type = "integer"
 severity = "integer"
 latitude = "double"
 longitude = "double"
 impacting = "boolean"
 status = "string">
 <Messages ... />
 <Location ... />
 <Schedule ... />
 <Descriptions ... />
 <Head ... />
 <Tails ... />
 < LastDetourPoints ... />
 <DLRs ... />
 <RDS ... />
 < DelayImpact ... />
 </Incident>

Attribute	Required	Type	Usage
id	no	string	Unique identifier associated with the incident
version	no	version	Version of the incident report
type	no	integer	Type of incident: 1=Construction 2=Event 3=Flow 4=Incident 6 = Police
severity	no	integer	Severity of the incident: 0=Minimal impact 1=Low impact 2=Moderate impact 3=High impact 4=Severe impact
latitude	no	double	Latitude of the incident
longitude	no	double	Longitude of the incident
impacting	no	boolean	If the incident is impacting traffic
status	yes	string	Status of the incident: Active, Cleared, Inactive

Element	Required	Occurs	Description
Messages	yes	0..1	Container for message codes describing the incident.
Location	no	0..1	Container for the Segments located at the incident.
Schedule	yes	0..1	Schedule for the incident.
Descriptions	no	0..1	Container for textual descriptions of the incident.
Head	no	0..1	Latitude/longitude of the furthest point on the road where the incident starts.
Tails	no	0..1	Latitude/longitude of the point behind the head where the incident ends.
LastDetourPoints	no	0..1	Latitude/longitude of the last detour point to avoid the incident.
DLRs	yes	0..1	Container for the XD Segments located at the incident.

RDS	no	0..1	Radio Data System data.
DelayImpact	no	0..1	Information about the delay caused by the incident.

Messages (Parent element: XDIncident)

Container for message codes describing the incident.

Syntax: <Messages>
 <AlertCMessageCodes ... />
 <InrixMessage ... />
 </Messages>

Element	Required	Occurs	Description
AlertCMessageCodes	yes	0..1	Container for Alert C message codes associated with the incident.
InrixMessage	yes	0..1	Container for INRIX message codes describing the incident.

AlertCMessageCodes (Parent element: Messages)

Container for Alert C message codes associated with the incident.

Syntax: <AlertCMessageCodes>
 <EventCode ... />
 </AlertCMessageCodes>

Element	Required	Occurs	Description
EventCode	yes	1..n	Alert C event code associated with the incident.

EventCode (Parent element: AlertCMessageCodes)

Alert C event code associated with the incident.

Syntax: <EventCode
 level = "string"
 quantifierType = "integer">
 ALERTC EVENT CODE
 </EventCode>

Attribute	Required	Type	Usage
level	yes	string	Primary, secondary.
quantifierType	no	Integer	Additional information.

InrixMessage (Parent element: Messages)

Container for INRIX message codes describing the incident.

Syntax: <InrixMessage>
 <InrixCode ... />
 </InrixMessage>

Element	Required	Occurs	Description
InrixCode	yes	1..n	INRIX message code describing the incident.

InrixCode (Parent element: InrixMessage)

INRIX message code describing the incident.

Syntax: <InrixCode
 type = "string"
 quantifierData = "string">
 quantifierType = "string">
 INRIX CODE
 </ InrixCode >

Attribute	Required	Type	Usage
type	yes	string	Cause, Effect, Information.
quantifierData	no	string	Additional information value.
quantifierType	no	string	Additional information type (e.g., speed, minute(s), severity, percentage).

Location (Parent element: XDIncident)

Container for the Segments located at the incident.

Syntax: <Location
 countryCode = "integer"
 direction = "string"
 biDirectional = "boolean">
 <Segment ... />
 </Location>

Attribute	Required	Type	Usage
countryCode	yes	integer	Country code of segment (North America=1).
direction	yes	string	Segment direction (Northbound, Southbound, Eastbound, Westbound, both ways).
biDirectional	yes	boolean	

Element	Required	Occurs	Description
Segment	yes	1..n	Container for individual segments involved in the incident.

Segment (Parent element: Location)

Container for individual segments involved in the incident.

Syntax: <Segment
 type = "string"

```

    code = "string"
    offset = "string">
</Segment>

```

Attribute	Required	Type	Usage
type	yes	string	Segment type (TMC,XDS).
code	yes	string	Segment code.
offset	yes	string	Offset within segment.

Schedule (Parent element: XDIncident)

Schedule for the incident.

```

Syntax: <Schedule
    planned = "boolean"
    advancedWarning = "boolean">
    <OccurrenceStartTime ... />
    <OccurrenceEndTime ... />
    <Description ... />
</Schedule >

```

Attribute	Required	Type	Usage
planned	yes	boolean	
advanceWarning	yes	boolean	

Element	Required	Occurs	Description
OccurrenceStartTime	no	0..1	Start date/time of the incident.
OccurrenceEndTime	no	0..1	End date/time of the incident.
Description	yes	0..1	Textual description of start and end time in local time.

Description (Parent element: Schedule)

Textual description of start and end date/time in local time.

```

Syntax: <Description
    xml:lang = "string">
    TEXTUAL DESCRIPTION OF START AND END DATE/TIME
</Description>

```

Attribute	Required	Type	Usage
xml:lang	yes	string	Language code (en-US).

Descriptions (Parent element: XDIncident)

Container for textual descriptions of the incident.

```

Syntax: <Descriptions>

```

```

    <Desc ... />
    <ParameterizedDescription .../>
</Descriptions>

```

Element	Required	Occurs	Description
Desc	yes	1..n	Textual descriptions of the incident.
ParameterizedDescription	yes	0..1	Detailed information about the incident specified in the eventCode attribute.

Desc (Parent element: Descriptions)
Textual descriptions of the incident.

```

Syntax: <Desc
        xml:lang = "string"
        type = "string">
        TEXTUAL DESCRIPTION
</Desc>

```

Attribute	Required	Type	Usage
xml:lang	yes	string	Language code (en-US).
type	yes	string	Type of description (short, long, Text-to-Speech).

ParameterizedDescription (Parent element: Descriptions)
Detailed information about the incident specified in the eventCode attribute.

```

Syntax: <ParameterizedDescription
        eventCode = "string">
        <EventText ... />
        <RoadName ... />
        <Direction ... />
        <FromLocation ... />
        <ToLocation ... />
        <Crossroad1 ... />
        <Crossroad2 ... />
        <Position1 ... />
        <Position2 ... />
</ParameterizedDescription>

```

Attribute	Required	Type	Usage
eventCode	yes	string	Identifies the type of incident.

Element	Required	Occurs	Description
EventText	no	1	Textual description of the type of incident.
RoadName	no	1	Name of the road or venue where the incident occurred.

Direction	no	1	Direction of travel on road for which incident occurred.
FromLocation	no	1	City that the road is going from.
ToLocation	no	1	City that the road is going towards.
Crossroad1	no	1	Name (and TMCId, if TMCs are requested) of the crossroad where the incident occurred.
Crossroad2	no	1	Name (and TMCId, if TMCs are requested) of the crossroad where the incident occurred.
Position1	no	1	Positional relationship between RoadName and Crossroad1.
Position2	no	1	Positional relationship between RoadName and Crossroad2.

Crossroad1 (Parent element: ParameterizedDescription)

Name (and TMCId, if TMCs are requested) of the crossroad where the incident occurred.

Syntax: <Crossroad1 tmcCode = "string">
 NAME OF CROSSROAD
 </Crossroad1>

Attribute	Required	Type	Usage
tmcCode	no	string	TMCId of crossroad.

Crossroad2 (Parent element: ParameterizedDescription)

Name (and TMCId, if TMCs are requested) of the crossroad where the incident occurred.

Syntax: <Crossroad2 tmcCode = "string">
 NAME OF CROSSROAD
 </Crossroad2>

Attribute	Required	Type	Usage
tmcCode	no	string	TMCId of crossroad.

Head (Parent element: XDIncident)

Latitude/longitude of the furthest point on the road where the incident starts.

Syntax: <Head
 latitude = "double "
 longitude = "double " >
 </ Head >

Attribute	Required	Type	Usage
latitude	yes	double	Latitude of the furthest point on the road where the incident starts.

longitude	yes	double	Longitude of the furthest point on the road where the incident starts.
-----------	-----	--------	--

Tails (Parent element: XDIncident)
Container for Tail information.

Syntax: <Tails>
 <Tail ... />
</Tails>

Element	Required	Occurs	Description
Tail	yes	1..n	Latitude/longitude of the point behind the head where the incident ends.

Tail (Parent element: Tails)
Latitude/longitude of the point behind the head where the incident ends.

Syntax: <Tail
 latitude=" double "
 longitude =" double " >
</ Tail>

Attribute	Required	Type	Usage
latitude	yes	double	Latitude of the point behind the head where the incident ends.
longitude	yes	double	Longitude of the point behind the head where the incident ends.

LastDetourPoints (Parent element: XDIncident)
Container for LastDetourPoint information.

Syntax: < LastDetourPoints>
 < LastDetourPoint ... />
</ LastDetourPoints>

Element	Required	Occurs	Description
LastDetourPoint	yes	1..n	Latitude/longitude of the last detour point to avoid the incident.

LastDetourPoint (Parent element: LastDetourPoints)
Latitude/longitude the last detour point to avoid the incident.

Syntax: < LastDetourPoint
 latitude=" double "
 longitude =" double " >
</ LastDetourPoint>

Attribute	Required	Type	Usage
latitude	yes	double	Latitude of the last detour point to avoid the incident.
longitude	yes	double	Longitude of the last detour point to avoid the incident.

DLRs (Parent element: XDIncident)

Container for the XD Segments located at the incident. Specified using Dynamic Location Referencing (AGORA-C).

Syntax: <DLRs type = "string">
 <DLR ... />
 </DLRs>

Attribute	Required	Type	Usage
type	yes	string	Identifies XD Segments are returned.

Element	Required	Occurs	Description
DLR	yes	1..n	Data for individual XD Segments involved in the incident.

DLR (Parent element: DLRs)

Data for individual XD Segments involved in the incident.

Syntax: <DLR offset = "string">
 XD SEGMENT ID
 </DLR>

Attribute	Required	Type	Usage
offset	yes	string	Offset within segment.

RDS (Parent element: XDIncident)

Radio Data System data.

Syntax: <RDS
 alertcMessage = "hexadecimal"
 tmcLocation = "integer"
 tmcCountry = "integer"
 tmcRegion = "integer"
 direction = "integer"
 extent = "integer"
 duration = "integer"
 diversion = "boolean"
 directionalityChanged = "boolean">
 <EventCodes ... />
 </RDS>

Attribute	Required	Type	Usage
alertcMessage	yes	hexadecimal	Alert C encoded message that contains the incident information.
tmcLocation	no	integer (four digits)	TMC location code that represents an intersection (if TMCs are requested).
tmcCountry	no	Integer (0 or 1)	Country code for the TMC (if TMCs are requested): 0=United States 1=United Kingdom.
tmcRegion	no	integer	Region code for the TMC (if TMCs are requested). There are multiple regions per country.
direction	yes	integer (0 or 1)	Direction of queue growth: 0=queue growth is positive and flow growth is negative 1=queue growth is negative and flow growth is positive
extent	yes	Integer (0 – 31)	Number of TMC Segments affected by incident: 0=just this TMC Segments is affected 1=the incident also affects the adjacent TMC Segments, and so on.
duration	yes	integer	Expected remaining duration of incident, in minutes. This value is often just an estimate. When encoded, the duration value is often set to zero, meaning unknown.
diversion	yes	Boolean (0 or 1)	Whether or not drivers are recommended to avoid the area if possible: 0=avoiding is not recommended 1=avoiding is recommended
directionalityChanged	yes	boolean	Whether the directionality is different than the default event code directionality.

Element	Required	Occurs	Description
EventCodes	yes	1	Collection of EventCode elements.

EventCodes (Parent element: RDS)

A container for RDS EventCode elements.

Syntax: <EventCodes>

```
<EventCode ... />
</EventCodes>
```

Element	Required	Occurs	Description
EventCode	yes	1..n	Represents a single RDS Alert C event code.

EventCode (Parent element: EventCodes)

Represents a single Radio Data System Alert C event code.

```
Syntax: <EventCode
  value = "integer"
  primary = "boolean">
</EventCode>
```

Attribute	Required	Type	Usage
value	yes	integer	EventCode value, as described in document ISO/FDIS 14819-2.
primary	yes	boolean	If there are multiple event codes, indicates whether this is the primary event code.

DelayImpact (Parent element: XDIncident)

Information about the delay caused by the incident.

```
Syntax: <DelayImpact
  fromTypicalMinutes=" double "
  fromFreeFlowMinutes=" double "
  distance=" double "
  abnormal = "boolean">
</ DelayImpact >
```

Attribute	Required	Type	Usage
fromTypicalMinutes	yes	double	The increase in travel time across the road extent affected by the incident relative to the travel time under average conditions for the given day of week and time of day.
fromFreeFlowMinutes	yes	double	The increase in travel time across the road extent affected by the incident relative to the travel time under free-flowing conditions.
distance	yes	double	The length in miles of the road extent affected by the incident.
abnormal	no	boolean	Provided for flow incidents.

DESCRIPTION

Retrieves traffic-related incidents for a Segment Set, consisting of XD and/or TMC Segments.

EXAMPLE CALL REQUEST

http://na.api.inrix.com/Traffic/Inrix.ashx?Action=GetXDIncidentsInSet&Token=joCApAj71QCfID*7XS6*I5ZM2qkX3mX8UfwbTuJEcFA|&SegmentSetId=2061742903&IncidentOutputFields=All&LocRefMethod=TMC,XD

EXAMPLE CALL RESPONSE

```
<Inrix docType="GetXDIncidentsInSet" copyright="Copyright INRIX Inc." versionNumber="10.5" createdDate="2017-08-10T20:32:49Z" statusId="0" statusText="" responseId="12551856-e11f-4011-93c4-538539287e96">
  <XDIncidents>
    <XDIncident id="112422" version="2" type="1" severity="2" latitude="36.477728" longitude="-78.237807" impacting="N" status="active">
      <Messages>
        <AlertCMessageCodes>
          <EventCode level="Primary" quantifierType="0">701</EventCode>
          <EventCode level="Secondary" quantifierType="0">701</EventCode>
        </AlertCMessageCodes>
        <InrixMessage>
          <InrixCode type="Effect">701</InrixCode>
          <InrixCode type="Information">292</InrixCode>
          <InrixCode type="Cause">1037</InrixCode>
        </InrixMessage>
      </Messages>
      <Location countryCode="1" direction="both ways" biDirectional="true">
        <Segment type="XDS" code="1386708016" offset="279,289"/>
        <Segment type="XDS" code="1386877300" offset="679,689"/>
        <Segment type="TMC" code="125+05471" offset="3444,3454"/>
        <Segment type="TMC" code="125-05470" offset="679,689"/>
      </Location>
      <Schedule planned="true" advanceWarning="false">
        <OccurrenceStartTime>2015-04-28T11:00:00Z</OccurrenceStartTime>
        <OccurrenceEndTime>2019-07-01T21:00:00Z</OccurrenceEndTime>
        <Description xml:lang="en-US">
          Starts at 4/28/2015 7:00 AM, ends at 7/1/2019 5:00 PM.
        </Description>
      </Schedule>
      <Descriptions>
        <Desc xml:lang="en-US" type="short">
          I-85: intermittent lane closures at I-85 Exit 229 / Oine Rd
        </Desc>
        <Desc xml:lang="en-US" type="long">
```

Intermittent lane closures due to construction work on I-85 both ways at I-85 Exit 229 / Oine Rd.
Expect delays.

```

</Desc>
<Desc xml:lang="en-US" type="Text-to-Speech">
  There are intermittent lane closures because of construction work on I-85 both ways at I-85 Exit 229.
  Expect delays.
</Desc>
<ParameterizedDescription eventCode="701">
  <EventText>
    Intermittent Lane Closures, expect Delays, construction work
  </EventText>
  <RoadName>I-85</RoadName>
  <Direction>both ways</Direction>
  <ToLocation>Oine</ToLocation>
  <Crossroad2 tmcCode="125P00000">I-85 Exit 229 / Oine Rd</Crossroad2>
  <Position1>at</Position1>
  <Position2/>
</ParameterizedDescription>
</Descriptions>
<Head latitude="36.477728" longitude="-78.237807"/>
<DLRs type="XDSegment">
  <DLR offset="279,289">1386708016</DLR>
  <DLR offset="679,689">1386877300</DLR>
</DLRs>
<RDS alertcMessage="00C2BD155F004957A000000000000000000000000000000000000000000000000" tmcLocation="5471"
tmcCountry="1" tmcRegion="25" direction="1" extent="0" duration="996670" diversion="false" directionality
yChanged="false">
  <EventCodes>
    <EventCode value="701" primary="true"/>
    <EventCode value="701" primary="false"/>
  </EventCodes>
</RDS>
<DelayImpact fromTypicalMinutes="0.00" fromFreeFlowMinutes="0.00" distance="0.00"/>
</XDIncident>
<XDIncident id="52092006" version="4" type="3" severity="1" latitude="35.919298" longitude="-
80.597163" impacting="Y" status="active">
  <Messages>
    <AlertCMessageCodes>
      <EventCode level="Primary">74</EventCode>
    </AlertCMessageCodes>
    <InrixMessage>
      <InrixCode type="Effect" quantifierData="1" quantifierType="minute(s),
minute(s)">989</InrixCode>
      <InrixCode type="Information" quantifierData="229" quantifierType="severity
value">1000</InrixCode>
      <InrixCode type="Information" quantifierData="-0.08" quantifierType="percentage,
%">998</InrixCode>
  </Messages>

```

```

<InrixCode type="Information" quantifierData="-0.06" quantifierType="percentage,
%">997</InrixCode>
<InrixCode type="Information" quantifierData="-3.76" quantifierType="speed,
mph">996</InrixCode>
<InrixCode type="Information" quantifierData="650" quantifierType="severity
value">995</InrixCode>
<InrixCode type="Information" quantifierData="54.18 / 38.9 / 1" quantifierType="speed,
mph">991</InrixCode>
<InrixCode type="Information" quantifierData="25.48" quantifierType="speed,
mph">11</InrixCode>
<InrixCode type="Effect" quantifierData="54" quantifierType="speed, mph">228</InrixCode>
<InrixCode type="Effect" quantifierData="27" quantifierType="speed, mph">228</InrixCode>
</InrixMessage>
</Messages>
<Location countryCode="1" direction="Eastbound" biDirectional="false">
  <Segment type="XDS" code="1386749437" offset="844,1061"/>
  <Segment type="XDS" code="1386667645" offset="0,729"/>
  <Segment type="XDS" code="1386667627" offset="0,886"/>
  <Segment type="TMC" code="125P05046" offset="844,1061"/>
  <Segment type="TMC" code="125+05047" offset="0,1616"/>
</Location>
<Schedule planned="false" advanceWarning="false">
  <OccurrenceStartTime>2017-08-10T19:38:00Z</OccurrenceStartTime>
  <OccurrenceEndTime>2017-08-10T21:06:35Z</OccurrenceEndTime>
  <Description xml:lang="en-US">
    Starts at 8/10/2017 3:38 PM, ends at 8/10/2017 5:06 PM.
  </Description>
</Schedule>
<Descriptions>
  <Desc xml:lang="en-US" type="short">
    I-40 E/B: delays of one minute between Exit 168 US-64 and Exit 170 US-601
  </Desc>
  <Desc xml:lang="en-US" type="long">
    Delays of one minute on I-40 Eastbound between Exit 168 US-64 and Exit 170 US-601. Average
    speed 25 mph.
  </Desc>
  <Desc xml:lang="en-US" type="Text-to-Speech">
    There are delays of one minute on I-40 Eastbound between Exit 168 US-64 and Exit 170 US-601.
    Average speed 25 mph.
  </Desc>
  <ParameterizedDescription eventCode="74">
    <EventText>XDFI Delays, xDFI Average Speed</EventText>
    <RoadName>I-40</RoadName>
    <Direction>Eastbound</Direction>
    <Crossroad1 tmcCode="125P05046">I-40 Exit 168 / US-64</Crossroad1>
    <Crossroad2 tmcCode="125P05047">I-40 Exit 170 / US-601 / US-601 Yadkinville Rd</Crossroad2>
    <Position1>between</Position1>
    <Position2>and</Position2>
  </ParameterizedDescription>

```

```

    </ParameterizedDescription>
  </Descriptions>
  <Head latitude="35.919298" longitude="-80.597163"/>
  <Tails>
    <Tail latitude="35.910921" longitude="-80.614690"/>
  </Tails>
  <LastDetourPoints>
    <LastDetourPoint latitude="35.907333" longitude="-80.622139"/>
  </LastDetourPoints>
  <DLRs type="XDSegment">
    <DLR offset="844,1061">1386749437</DLR>
    <DLR offset="0,729">1386667645</DLR>
    <DLR offset="0,886">1386667627</DLR>
  </DLRs>
  <RDS alertcMessage="08484A13B7" tmcLocation="5047" tmcCountry="1" tmcRegion="25" direction="1" extent="1" duration="38" diversion="false" directionalityChanged="false">
    <EventCodes>
      <EventCode value="74" primary="true"/>
    </EventCodes>
  </RDS>
  <DelayImpact fromTypicalMinutes="1.38" fromFreeFlowMinutes="1.38" fromNas="1" distance="1.14" abnormal="true"/>
</XDIncident>
<XDIncident id="52085033" version="3" type="4" severity="3" latitude="35.607084" longitude="-78.564137" impacting="Y" status="active">
  <Messages>
    <AlertCMessageCodes>
      <EventCode level="Primary" quantifierType="0">500</EventCode>
      <EventCode level="Secondary" quantifierType="0">229</EventCode>
    </AlertCMessageCodes>
    <InrixMessage>
      <InrixCode type="Effect">10</InrixCode>
      <InrixCode type="Cause">2</InrixCode>
      <InrixCode type="Effect">228</InrixCode>
    </InrixMessage>
  </Messages>
  <Location countryCode="1" direction="Westbound" biDirectional="false">
    <Segment type="XDS" code="1386752356" offset="12,22"/>
    <Segment type="TMC" code="125P05120" offset="512,522"/>
  </Location>
  <Schedule planned="false" advanceWarning="false">
    <OccurrenceStartTime>2017-08-10T18:55:29Z</OccurrenceStartTime>
    <OccurrenceEndTime>2017-08-10T20:39:43Z</OccurrenceEndTime>
    <Description xml:lang="en-US">
      Starts at 8/10/2017 2:55 PM, ends at 8/10/2017 4:39 PM.
    </Description>
  </Schedule>
</Descriptions>

```

```

<Desc xml:lang="en-US" type="short">I-40 W/B: lane closed at Exit 312 NC-42</Desc>
<Desc xml:lang="en-US" type="long">
  Lane closed and slow traffic due to accident on I-40 Westbound at Exit 312 NC-42.
</Desc>
<Desc xml:lang="en-US" type="Text-to-Speech">
  There is a lane closed and slow traffic because of an accident on I-40 Westbound at Exit 312 NC-42.
</Desc>
<ParameterizedDescription eventCode="500">
  <EventText>Lane (or Lanes) closed, accident, slow traffic</EventText>
  <RoadName>I-40</RoadName>
  <Direction>Westbound</Direction>
  <ToLocation>Edmondson</ToLocation>
  <Crossroad2 tmcCode="125N00000">I-40 Exit 312 / NC-42</Crossroad2>
  <Position1>at</Position1>
  <Position2/>
</ParameterizedDescription>
</Descriptions>
<Head latitude="35.607084" longitude="-78.564137"/>
<Tails>
  <Tail latitude="35.611668" longitude="-78.566387"/>
</Tails>
<LastDetourPoints>
  <LastDetourPoint latitude="35.611668" longitude="-78.566387"/>
</LastDetourPoints>
<DLRs type="XDsegment">
  <DLR offset="12,22">1386752356</DLR>
</DLRs>
<RDS alertcMessage="0081F4140000491CA00000000000000000000000000000000" tmcLocation="5120" t
mcCountry="1" tmcRegion="25" direction="0" extent="0" duration="35" diversion="false"directionalityChan
ged="false">
  <EventCodes>
    <EventCode value="500" primary="true"/>
    <EventCode value="229" primary="false"/>
  </EventCodes>
</RDS>
<DelayImpact fromTypicalMinutes="0.00" fromFreeFlowMinutes="0.00" distance="0.00"/>
</XDIncident>
</XDIncidents>
</Inrix>

```

SYNTAX

ServerPath

?Action=GetXDIncidentsInSet

&Token=*string*

&SegmentSetId=*integer*

&IncidentType=*string*

&Severity=*string*
 &IncidentOutputFields=*string*
 &LocRefMethod=*string*

PARAMETERS

ServerPath (Type: string, Required)

The URL of the server to which requests should be addressed as returned in the ServerPath element of the response to a GetSecurityToken request. (Should use the API path.)

Token (Type: string, Required)

A valid security token as returned in the AuthToken element of the response to a GetSecurityToken request.

SegmentSetId (Type: integer, Required)

Segment Set Id for which to get data. Segment Set can include XDIds, TMCIIds, or both. Any duplicate Ids are ignored.

IncidentType (Type: string, Optional)

The types of incidents to return. Multiple incident types can be specified in a comma-delimited list. If this parameter is not specified, all fields are returned.

Incidents	All unusual incidents that may slow down traffic such as a car accident.
Construction	Construction incidents.
Events	Unusual events slated for the area such as a major sporting event.
Flow	Reports about the slowing down of traffic on route.

Severity (Type: string, Optional)

Filters incident reports based on severity level. This value can be in the range of 0-4, with 4 indicating the highest severity. Multiple severity values can be specified in a comma-delimited list. If this parameter is not specified, all severity levels are returned

IncidentOutputFields (Type: string, Optional)

The incident fields to be returned in the response. Multiple fields can be specified in a comma-delimited list. If this parameter is not specified, all fields except "RDS", "DelayImpact" and "HeadTail" are returned. To return all fields, including new fields added in the future, specify "All".

All	Returns all fields.
ID	Unique identifier of an incident.
Version	Version number of the incident report, incremented each time an incident report is updated.
Type	Type of the incident (Incidents, Construction, Events, Flow). Incidents can be determined from the Alert-C event code: Construction indicates the presence of road construction; Events can be weather-related or a scheduled sporting/public event; and Flow indicates a blocking incident.
Severity	Severity of the incident. This value can be in the range of 0-4, with 4 indicating the highest severity.
LatLong	Latitude and longitude of the incident.

Impacting	Whether the incident impacts traffic flow. This field is set to “yes” if the appearance of the incident changes the traffic flow below a certain percentage from that which is normally expected for the given segment of road at that time, given the current conditions.
TMCs	TMCs or XD Segments located at the incident. (The type is determined by the LocRefMethod parameter.)
StartTime	Starting time of the incident.
EndTime	Ending time of the incident.
ShortDescription	Short textual description of the incident including language code.
FullDescription	Longer textual description of the incident including language code.
ParameterizedDescription	Detailed information about the event specified by the eventCode.
HeadTail	Location of the furthest point on the road where the incident starts, location of the point behind the head where the incident ends, and location of the last detour point to avoid the incident.
RDS	Radio Data System data.
DelayImpact	Information about the delay caused by the event.

LocRefMethod (Type: string, Optional)

The location reference method returned for the incidents. Multiple types can be specified in a comma-delimited list. If this parameter is not specified, it defaults to TMC.

TMC (default)	Specify TMCs located at the incident.
XD	Specify XD Segments located at the incident.

RETURN VALUES

Each XML response to this API call will contain two basic elements, each having several attributes and/or child elements.

- *Inrix* – The root element in the response for all APIs is always <Inrix>. It contains response data in child elements and information about the request in the following attributes:
 - *docType* identifies the API called and the schema used for the response.
 - *copyright* identifies the INRIX, Inc. copyright.
 - *versionNumber* refers to the version number of the INRIX traffic services platform that produced the response.
 - *createdDate* is the timestamp indicating when the response was created. The timestamp specifies a DateTime value, formatted according to the ISO 8601 standard and expressed in UTC time. The format of the attribute is YYYY-MM-DDTHH:MM:SSZ. The “T” character separates the date from the time, and the “Z” character specifies that UTC time is used.
 - *statusId* indicates whether an API request succeeded or failed. Each *statusId* value is associated with a unique *statusText* value. A *statusId* is intended to be convenient to handle in code. *statusId*=0 is a successful call. [Appendix K](#) contains a list of common statusID responses.
 - *statusText* corresponds to the *statusId*. A *statusText* is intended to be convenient for humans to interpret.
 - *responseID* is logged by INRIX for future use in assisting customer inquiries. Service Requestors can use *responseID* as a way for INRIX to identify the specific API call that was made. It is recommended that Service Requestors log this information to expedite technical support incidents.
- *XDIncidents* – This child element is a container for all of the requested incident data.

RESPONSE ELEMENTS

XDIncidents (Parent element: Inrix)

A container for incident data.

Syntax: <XDIncidents>
 <XDIncident ... />
 </XDIncidents>

Element	Required	Occurs	Description
XDIncident	no	0..n	Represents a single traffic incident.

XDIncident (Parent element: XDIncidents)

Represents a single traffic incident.

Syntax: <Incident
 id = "string"
 version = "version"
 type = "integer"
 severity = "integer"
 latitude = "double"
 longitude = "double"
 impacting = "boolean"
 status = "string">
 <Messages ... />
 <Location ... />
 <Schedule ... />
 <Descriptions ... />
 <Head ... />
 <Tails ... />
 < LastDetourPoints ... />
 <DLRs ... />
 <RDS ... />
 < DelayImpact ... />

</Incident>

Attribute	Required	Type	Usage
id	no	string	Unique identifier associated with the incident
version	no	version	Version of the incident report
type	no	integer	Type of incident: 1=Construction 2=Event 3=Flow 4=Incident 6 = Police

severity	no	integer	Severity of the incident: 0=Minimal impact 1=Low impact 2=Moderate impact 3=High impact 4=Severe impact
latitude	no	double	Latitude of the incident
longitude	no	double	Longitude of the incident
impacting	no	boolean	If the incident is impacting traffic
status	yes	string	Status of the incident: Active, Cleared, Inactive

Element	Required	Occurs	Description
Messages	yes	0..1	Container for message codes describing the incident.
Location	no	0..1	Container for the Segments located at the incident.
Schedule	yes	0..1	Schedule for the incident.
Descriptions	no	0..1	Container for textual descriptions of the incident.
Head	no	0..1	Latitude/longitude of the furthest point on the road where the incident starts.
Tails	no	0..1	Latitude/longitude of the point behind the head where the incident ends.
LastDetourPoints	no	0..1	Latitude/longitude of the last detour point to avoid the incident.
DLRs	yes	0..1	Container for the XD Segments located at the incident.
RDS	no	0..1	Radio Data System data.
DelayImpact	no	0..1	Information about the delay caused by the incident.

Messages (Parent element: XDIncident)

Container for message codes describing the incident.

```
Syntax: <Messages>
        <AlertCMessageCodes ... />
        <InrixMessage ... />
</Messages>
```

Element	Required	Occurs	Description
AlertCMessageCodes	yes	0..1	Container for Alert C message codes associated with the incident.
InrixMessage	yes	0..1	Container for INRIX message codes describing the incident.

AlertCMessageCodes (Parent element: Messages)

Container for Alert C message codes associated with the incident.

```
Syntax: <AlertCMessageCodes>
        <EventCode ... />
</AlertCMessageCodes>
```

Element	Required	Occurs	Description
EventCode	yes	1..n	Alert C event code associated with the incident.

EventCode (Parent element: AlertCMessageCodes)

Alert C event code associated with the incident.

```
Syntax: <EventCode
        level = "string"
        quantifierType = "integer">
        ALERTC EVENT CODE
</EventCode>
```

Attribute	Required	Type	Usage
level	yes	string	Primary, secondary.
quantifierType	no	Integer	Additional information.

InrixMessage (Parent element: Messages)

Container for INRIX message codes describing the incident.

```
Syntax: <InrixMessage>
        <InrixCode ... />
</InrixMessage>
```

Element	Required	Occurs	Description
InrixCode	yes	1..n	INRIX message code describing the incident.

InrixCode (Parent element: InrixMessage)

INRIX message code describing the incident.

```
Syntax: <InrixCode
  type = "string"
  quantifierData = "string">
  quantifierType = "string">
  INRIX CODE
</ InrixCode >
```

Attribute	Required	Type	Usage
type	yes	string	Cause, Effect, Information.
quantifierData	no	string	Additional information value.
quantifierType	no	string	Additional information type (e.g., speed, minute(s), severity, percentage).

Location (Parent element: XDIncident)

Container for the Segments located at the incident.

```
Syntax: <Location
  countryCode = "integer"
  direction = "string"
  biDirectional = "boolean">
  <Segment ... />
</Location>
```

Attribute	Required	Type	Usage
countryCode	yes	integer	Country code of segment (North America=1).
direction	yes	string	Segment direction (Northbound, Southbound, Eastbound, Westbound, both ways).
biDirectional	yes	boolean	

Element	Required	Occurs	Description
Segment	yes	1..n	Container for individual segments involved in the incident.

Segment (Parent element: Location)

Container for individual segments involved in the incident.

```
Syntax: <Segment
  type = "string"
  code = "string"
  offset = "string">
</Segment>
```

Attribute	Required	Type	Usage
type	yes	string	Segment type (TMC,XDS).
code	yes	string	Segment code.
offset	yes	string	Offset within segment.

Schedule (Parent element: XDIncident)
Schedule for the incident.

Syntax: <Schedule
 planned = "boolean"
 advancedWarning = "boolean">
 <OccurrenceStartTime ... />
 <OccurrenceEndTime ... />
 <Description ... />
 </Schedule >

Attribute	Required	Type	Usage
planned	yes	boolean	
advanceWarning	yes	boolean	

Element	Required	Occurs	Description
OccurrenceStartTime	no	0..1	Start date/time of the incident.
OccurrenceEndTime	no	0..1	End date/time of the incident.
Description	yes	0..1	Textual description of start and end time in local time.

Description (Parent element: Schedule)
Textual description of start and end date/time in local time.

Syntax: <Description
 xml:lang = "string">
 TEXTUAL DESCRIPTION OF START AND END DATE/TIME
 </Description>

Attribute	Required	Type	Usage
xml:lang	yes	string	Language code (en-US).

Descriptions (Parent element: XDIncident)
Container for textual descriptions of the incident.

Syntax: <Descriptions>
 <Desc ... />
 <ParameterizedDescription .../>
 </Descriptions>

Element	Required	Occurs	Description
Desc	yes	1..n	Textual descriptions of the incident.
ParameterizedDescription	yes	0..1	Detailed information about the incident specified in the eventCode attribute.

Desc (Parent element: Descriptions)
Textual descriptions of the incident.

Syntax: <Desc
 xml:lang = "string"
 type = "string">
 TEXTUAL DESCRIPTION
 </Desc>

Attribute	Required	Type	Usage
xml:lang	yes	string	Language code (en-US).
type	yes	string	Type of description (short, long, Text-to-Speech).

ParameterizedDescription (Parent element: Descriptions)
Detailed information about the incident specified in the eventCode attribute.

Syntax: <ParameterizedDescription
 eventCode = "string">
 <EventText ... />
 <RoadName ... />
 <Direction ... />
 <FromLocation ... />
 <ToLocation ... />
 <Crossroad1 ... />
 <Crossroad2 ... />
 <Position1 ... />
 <Position2 ... />
 </ParameterizedDescription>

Attribute	Required	Type	Usage
eventCode	yes	string	Identifies the type of incident.

Element	Required	Occurs	Description
EventText	no	1	Textual description of the type of incident.
RoadName	no	1	Name of the road or venue where the incident occurred.
Direction	no	1	Direction of travel on road for which incident occurred.

FromLocation	no	1	City that the road is going from.
ToLocation	no	1	City that the road is going towards.
Crossroad1	no	1	Name (and TMCId, if TMCs are requested) of the crossroad where the incident occurred.
Crossroad2	no	1	Name (and TMCId, if TMCs are requested) of the crossroad where the incident occurred.
Position1	no	1	Positional relationship between RoadName and Crossroad1.
Position2	no	1	Positional relationship between RoadName and Crossroad2.

Crossroad1 (Parent element: ParameterizedDescription)

Name (and TMCId, if TMCs are requested) of the crossroad where the incident occurred.

Syntax: <Crossroad1 tmcCode = "string">
 NAME OF CROSSROAD
 </Crossroad1>

Attribute	Required	Type	Usage
tmcCode	no	string	TMCId of crossroad.

Crossroad2 (Parent element: ParameterizedDescription)

Name (and TMCId, if TMCs are requested) of the crossroad where the incident occurred.

Syntax: <Crossroad2 tmcCode = "string">
 NAME OF CROSSROAD
 </Crossroad2>

Attribute	Required	Type	Usage
tmcCode	no	string	TMCId of crossroad.

Head (Parent element: XDIncident)

Latitude/longitude of the furthest point on the road where the incident starts.

Syntax: <Head
 latitude = " double "
 longitude = " double " >
 </ Head >

Attribute	Required	Type	Usage
latitude	yes	double	Latitude of the furthest point on the road where the incident starts.

longitude	yes	double	Longitude of the furthest point on the road where the incident starts.
-----------	-----	--------	--

Tails (Parent element: XDIncident)
Container for Tail information.

Syntax: <Tails>
 <Tail ... />
</Tails>

Element	Required	Occurs	Description
Tail	yes	1..n	Latitude/longitude of the point behind the head where the incident ends.

Tail (Parent element: Tails)
Latitude/longitude of the point behind the head where the incident ends.

Syntax: <Tail
 latitude=" double "
 longitude =" double " >
</ Tail>

Attribute	Required	Type	Usage
latitude	yes	double	Latitude of the point behind the head where the incident ends.
longitude	yes	double	Longitude of the point behind the head where the incident ends.

LastDetourPoints (Parent element: XDIncident)
Container for LastDetourPoint information.

Syntax: < LastDetourPoints>
 < LastDetourPoint ... />
</ LastDetourPoints>

Element	Required	Occurs	Description
LastDetourPoint	yes	1..n	Latitude/longitude of the last detour point to avoid the incident.

LastDetourPoint (Parent element: LastDetourPoints)
Latitude/longitude the last detour point to avoid the incident.

Syntax: < LastDetourPoint
 latitude=" double "
 longitude =" double " >
</ LastDetourPoint>

Attribute	Required	Type	Usage
latitude	yes	double	Latitude of the last detour point to avoid the incident.
longitude	yes	double	Longitude of the last detour point to avoid the incident.

DLRs (Parent element: XDIncident)

Container for the XD Segments located at the incident. Specified using Dynamic Location Referencing (AGORA-C).

Syntax: <DLRs type = "string">
 <DLR ... />
 </DLRs>

Attribute	Required	Type	Usage
type	yes	string	Identifies XD Segments are returned.

Element	Required	Occurs	Description
DLR	yes	1..n	Data for individual XD Segments involved in the incident.

DLR (Parent element: DLRs)

Data for individual XD Segments involved in the incident.

Syntax: <DLR offset = "string">
 XD SEGMENT ID
 </DLR>

Attribute	Required	Type	Usage
offset	yes	string	Offset within segment.

RDS (Parent element: XDIncident)

Radio Data System data.

Syntax: <RDS
 alertcMessage = "hexadecimal"
 tmcLocation = "integer"
 tmcCountry = "integer"
 tmcRegion = "integer"
 direction = "integer"
 extent = "integer"
 duration = "integer"
 diversion = "boolean"
 directionalityChanged = "boolean">
 <EventCodes ... />
 </RDS>

Attribute	Required	Type	Usage
alertcMessage	yes	hexadecimal	Alert C encoded message that contains the incident information.
tmcLocation	no	integer (four digits)	TMC location code that represents an intersection (if TMCs are requested).
tmcCountry	no	Integer (0 or 1)	Country code for the TMC (if TMCs are requested): 0=United States 1=United Kingdom.
tmcRegion	no	integer	Region code for the TMC (if TMCs are requested). There are multiple regions per country.
direction	yes	integer (0 or 1)	Direction of queue growth: 0=queue growth is positive and flow growth is negative 1=queue growth is negative and flow growth is positive
extent	yes	Integer (0 – 31)	Number of TMC Segments affected by incident: 0=just this TMC Segments is affected 1=the incident also affects the adjacent TMC Segments, and so on.
duration	yes	integer	Expected remaining duration of incident, in minutes. This value is often just an estimate. When encoded, the duration value is often set to zero, meaning unknown.
diversion	yes	Boolean (0 or 1)	Whether or not drivers are recommended to avoid the area if possible: 0=avoiding is not recommended 1=avoiding is recommended
directionalityChanged	yes	boolean	Whether the directionality is different than the default event code directionality.

Element	Required	Occurs	Description
EventCodes	yes	1	Collection of EventCode elements.

EventCodes (Parent element: RDS)

A container for RDS EventCode elements.

Syntax: <EventCodes>
 <EventCode ... />
 </EventCodes>

Element	Required	Occurs	Description
EventCode	yes	1..n	Represents a single RDS Alert C event code.

EventCode (Parent element: EventCodes)

Represents a single Radio Data System Alert C event code.

Syntax: <EventCode
 value = "integer"
 primary = "boolean">
 </EventCode>

Attribute	Required	Type	Usage
value	yes	integer	EventCode value, as described in document ISO/FDIS 14819-2.
primary	yes	boolean	If there are multiple event codes, indicates whether this is the primary event code.

DelayImpact (Parent element: XDIncident)

Information about the delay caused by the incident.

Syntax: <DelayImpact
 fromTypicalMinutes=" double "
 fromFreeFlowMinutes=" double "
 distance=" double "
 abnormal = "boolean">
 </ DelayImpact >

Attribute	Required	Type	Usage
fromTypicalMinutes	yes	double	The increase in travel time across the road extent affected by the incident relative to the travel time under average conditions for the given day of week and time of day.
fromFreeFlowMinutes	yes	double	The increase in travel time across the road extent affected by the incident relative to the travel time under free-flowing conditions.
distance	yes	double	The length in miles of the road extent affected by the incident.
abnormal	no	boolean	Provided for flow incidents.

GetSpeedBucket

DESCRIPTION

Returns a Speed Bucket definition. If no SpeedBucketID is provided, returns all definitions available to the user.

EXAMPLE CALL REQUEST

```
http://na.api.inrix.com/Traffic/Inrix.ashx?Action=GetSpeedBucket&Token=*Xm5OrM4rtfJoj50nYUugkKAaz*hvWsyqBM  
MJLSHyw8|&SpeedBucketID=1
```

EXAMPLE CALL RESPONSE

```
<Inrix docType="GetSpeedBucket" copyright="Copyright INRIX Inc." versionNumber="11.1" createDate="2017-12-  
04T22:05:06Z" statusId="0" statusText="" responseId="008f2897-9ff7-4f42-a541-c967cd3bdb12">  
  <SpeedBuckets>  
    <SpeedBucket type="1" id="1" name="Inrix Default" description="Inrix Default" private="false"  
      createTime="2008-04-11T04:10:00Z">  
      <RoadClosure penStyle="2" backColor="brown" foreColor="white" hatchStyle="Percent50"/>  
      <Buckets>  
        <Bucket num="0" min="0" max="31" penStyle="1" backColor="black"/>  
        <Bucket num="1" min="32" max="62" penStyle="1" backColor="Red"/>  
        <Bucket num="2" min="63" max="92" penStyle="1" backColor="Yellow"/>  
        <Bucket num="3" min="93" max="100" penStyle="1" backColor="ForestGreen"/>  
      </Buckets>  
    </SpeedBucket>  
  </SpeedBuckets>  
</Inrix>
```

SYNTAX

ServerPath

?Action=GetSpeedBucket

&Token=*string*

&SpeedBucketID=*integer*

PARAMETERS

ServerPath (Type: string, Required)

The URL of the server to which requests should be addressed as returned in the ServerPath element of the response to a GetSecurityToken request. (Should use the API path.)

Token (Type: string, Required)

A valid security token as returned in the AuthToken element of the response to a GetSecurityToken request.

SpeedBucketID (Type: integer, Optional)

The ID of the Speed Bucket to retrieve, returned from a previous call to CreateSpeedBucket. INRIX default Speed Buckets (SpeedBucketIDs 1-6) are available to all users. Only one SpeedBucketID can be specified at a time. If no SpeedBucketID is provided, returns all definitions available to the user.

RETURN VALUES

Each XML response to this API call will contain two basic elements, each having several attributes and/or child elements.

- **Inrix** – The root element in the response for all APIs is always <Inrix>. It contains response data in child elements and information about the request in the following attributes:
 - *docType* identifies the API called and the schema used for the response.
 - *copyright* identifies the INRIX, Inc. copyright.
 - *versionNumber* refers to the version number of the INRIX traffic services platform that produced the response.
 - *createdDate* is the timestamp indicating when the response was created. The timestamp specifies a DateTime value, formatted according to the ISO 8601 standard and expressed in UTC time. The format of the attribute is YYYY-MM-DDTHH:MM:SSZ. The “T” character separates the date from the time, and the “Z” character specifies that UTC time is used.
 - *statusId* indicates whether an API request succeeded or failed. Each *statusId* value is associated with a unique *statusText* value. A *statusId* is intended to be convenient to handle in code. statusId=0 is a successful call. [Appendix K](#) contains a list of common statusID responses.
 - *statusText* corresponds to the statusId. A statusText is intended to be convenient for humans to interpret.
 - *responseID* is logged by INRIX for future use in assisting customer inquiries. Service Requestors can use *responseID* as a way for INRIX to identify the specific API call that was made. It is recommended that Service Requestors log this information to expedite technical support incidents.
- **SpeedBuckets** – This child element is a container for all of the requested speed bucket data.

RESPONSE ELEMENTS

SpeedBuckets (Parent element: Inrix)

A container for all of the requested speed bucket data.

Syntax: <SpeedBuckets>
 <SpeedBucket.../>
 <SpeedBucket.../>
 </SpeedBuckets>

Element	Required	Occurs	Description
SpeedBucket	yes	1..n	Defines how speed data is grouped and how it is displayed on Traffic Tiles.

SpeedBucket (Parent element: SpeedBuckets)

Defines how speed data is grouped and how it is displayed on Traffic Tiles.

Syntax: <SpeedBucket
 type= “integer”
 id= “string”

```

name= "string"
description= "string"
private= "Boolean"
createTime= "DateTime">
  <RoadClosure.../>
  <Buckets.../>
</SpeedBucket>

```

Attribute	Required	Type	Usage
type	yes	integer	The type of speed bucket: 1=Congestion 2=Speed 3=Relative Speed 4=Relative Congestion
id	yes	integer	The ID of the Speed Bucket.
name	yes	string	The name of the Speed Bucket.
description	no	string	A description of the Speed Bucket.
private	yes	Boolean	Whether the Speed Bucket is private or available to the public.
createTime	yes	DateTime	When the Speed Bucket was created.

Element	Required	Occurs	Description
RoadClosure	yes	1	Information for how a road closure is displayed on Traffic Tiles.
Buckets	yes	1	A collection of buckets that defines how speeds are grouped and displayed.

RoadClosure (Parent element: SpeedBucket)

Information for how a road closure is displayed.

```

Syntax: <RoadClosure
  penStyle= "integer"
  backColor= "string"
  foreColor= "string"
  hatchStyle= "string"
  penWidthPercent= "integer">
  <Pen.../>
</RoadClosure>

```

Attribute	Required	Type	Usage
penStyle	yes	integer	The style of the pen used to draw the section of road: 1=solid 2=hatched
backColor	yes	string	The color for solid pens and the background color for hatched pens.
foreColor	no	string	The accent color for hatched pens.
hatchStyle	no	string	The style for hatched pens.
penWidthPercent	no	integer	The percentage of the pen width used for drawing the road.

Element	Required	Occurs	Description
Pen	no	1..n	Information for how road closures are displayed for different road classes. If not specified, all road classes are drawn as specified in the <RoadClosure> attributes.

Buckets (Parent element: SpeedBucket)

A collection of buckets that defines how speeds are grouped and displayed.

Syntax: <Buckets>
 <Bucket ... />
 <Bucket ... />
 </Buckets >

Element	Required	Occurs	Description
Bucket	yes	1..n	Information for how speed data is grouped and displayed.

Bucket (Parent element: Buckets)

Information for how speed data is grouped and displayed.

Syntax: <Bucket
 num= "integer"
 min= "integer"
 max= "integer"
 penStyle= "integer"
 backColor= "string"
 foreColor= "string"
 hatchStyle= "string"
 penWidthPercent= "integer">
 <Pen ... />
 </Bucket>

Attribute	Required	Type	Usage
num	yes	integer	The index of the bucket in the series of bucket elements.
min	yes	integer	The minimum value for data that goes into this bucket.
max	yes	integer	The maximum value for data that goes into this bucket.
penStyle	yes	integer	The style of the pen used to draw the section of road: 1=solid 2=hatched
backColor	yes	string	The color for solid pens and the background color for hatched pens.
foreColor	no	string	The accent color for hatched pens.
hatchStyle	no	string	The style for hatched pens.
penWidthPercent	no	integer	The percentage of the pen width used for drawing the road.

Element	Required	Occurs	Description
Pen	no	1..n	Information for how speed data in this bucket is displayed for different road classes. If not specified, all road classes are drawn as specified in the <Bucket> attributes.

Pen (Parent elements: RoadClosure, Bucket)
Information for how speed data is displayed for different road classes.

```
Syntax: <Pen
    roadClass= "integer"
    penStyle= "integer"
    backColor= "string"
    foreColor= "string"
    hatchStyle= "string"
    penWidthPercent= "integer"
/>
```

Attribute	Required	Type	Usage
roadClass	yes	integer	Federal Road Classification (FRC) code: 1= First class roads, such as national highway network roads 2= Second class roads, such as state highway network roads 3= Third class roads, such as state interconnecting network roads 4= Fourth class roads, such as major connecting roads 5= Fifth class roads, such as minor roads connecting suburbs 6= Sixth class roads, such as destination and destination collector roads 7= Seventh class roads, such as destination dead-end roads
penStyle	yes	integer	The style of the pen used to draw the section of road: 1=solid 2=hatched
backColor	yes	string	The color for solid pens and the background color for hatched pens.
foreColor	no	string	The accent color for hatched pens.
hatchStyle	no	string	The style for hatched pens.
penWidthPercent	no	integer	The percentage of the pen width used for drawing the road.

NOTES

- Pen color can be:
 - [.NET color name](#) as a string (for example, "ForestGreen")
 - ARGB color value in a standard web format with the colors specified as hex values in order a,r,g,b (for example, "#00112233")
 - ARGB color values can be used to create opacity when defining the color attribute of a speed bucket.
 - For example: To create an approximately 50% opaque red, do not use the pre-defined web color "red." Instead, use the ARGB color value "#80FF0000". The alpha value #80 is halfway between 00 and FF, thus specifying 50% opacity, and the red color is specified by the FF red value, 00 green value, and 00 blue value.
- Hatched pen style can be:
 - [.NET hatch styles](#) as a string (for example, "SmallCheckerBoard")

CreateSpeedBucket

DESCRIPTION

Generates a new Speed Bucket definition and returns a unique SpeedBucketID value.

EXAMPLE CALL REQUEST

```
http://na.api.inrix.com/Traffic/Inrix.ashx?Action=CreateSpeedBucket&Token=qQuYkdZPM*I02eVZ98y5EOaGQZe06J*s1VugKPBkDNg|&Name=TestCreateSpeedBucketAPI&SpeedBucketXML=%3CSpeedBucket%20type=%221%22%3E%3CRoadClosure%20penStyle%3D%222%22%20backColor%3D%22brown%22%20foreColor%3D%22white%22%20hatchStyle%3D%22Percent50%22%20%2F%3E%3CBuckets%3E%0A%3CBucket%20num%3D%220%22%20min%3D%220%22%20max%3D%2231%22%20penStyle%3D%221%22%20backColor%3D%22black%22%20penWidthPercent%3D%2290%22%2F%3E%0A%3CBucket%20num%3D%221%22%20min%3D%2232%22%20max%3D%2262%22%20penStyle%3D%221%22%20backColor%3D%22Red%22%20%2F%3E%0A%3CBucket%20num%3D%222%22%20min%3D%2263%22%20max%3D%2292%22%20penStyle%3D%221%22%20backColor%3D%22Yellow%22%20%2F%3E%0A%3CBucket%20num%3D%223%22%20min%3D%2293%22%20max%3D%22100%22%20penStyle%3D%221%22%20backColor%3D%22ForestGreen%22%20%2F%3E%0A%3C%2FBuckets%3E%3C/SpeedBucket%3E
```

In this example, the SpeedBucketXML parameter value, not URL-encoded, is:

```
<SpeedBucket type="1">
  <RoadClosure penStyle="2" backColor="brown" foreColor="white" hatchStyle="Percent50" />
  <Buckets>
    <Bucket num="0" min="0" max="31" penStyle="1" backColor="black" penWidthPercent="90"/>
    <Bucket num="1" min="32" max="62" penStyle="1" backColor="Red" />
    <Bucket num="2" min="63" max="92" penStyle="1" backColor="Yellow" />
    <Bucket num="3" min="93" max="100" penStyle="1" backColor="ForestGreen" />
  </Buckets>
</SpeedBucket>
```

EXAMPLE CALL RESPONSE

```
<Inrix docType="CreateSpeedBucket" copyright="Copyright INRIX Inc." versionNumber="5.21" createdDate="2014-10-06T17:46:06Z" statusId="0" statusText="" responseId="cb9ed694-06ce-465f-9caf-679ba743325d">
```

```
<SpeedBuckets>
  <SpeedBucket type="1" id="1508344873" name=" TestCreateSpeedBucketAPI" description="" private="false"
  createTime="2014-10-06T23:44:37Z">
    <RoadClosure penStyle="2" backColor="brown" foreColor="white" hatchStyle="Percent50"/>
    <Buckets>
      <Bucket num="0" min="0" max="31" penStyle="1" backColor="black" penWidthPercent="90"/>
      <Bucket num="1" min="32" max="62" penStyle="1" backColor="Red"/>
      <Bucket num="2" min="63" max="92" penStyle="1" backColor="Yellow"/>
      <Bucket num="3" min="93" max="100" penStyle="1" backColor="ForestGreen"/>
    </Buckets>
  </SpeedBucket>
</SpeedBuckets>
</Inrix>
```

SYNTAX

```

ServerPath
?Action=CreateSpeedBucket
&Token=string
&Name=string
&Description=string
&Private=boolean
&SpeedBucketXML=string

```

PARAMETERS

ServerPath (Type: string, Required)

The URL of the server to which requests should be addressed as returned in the ServerPath element of the response to a GetSecurityToken request. (Should use the API path.)

Token (Type: string, Required)

A valid security token as returned in the AuthToken element of the response to a GetSecurityToken request.

Name (Type: string, Required)

The name of the Speed Bucket.

Description (Type: string, Optional)

A textual description of the Speed Bucket.

Private (Type: Boolean, Optional)

Specifies whether the bucket is accessible by all vendors in the group, or only the defining vendor. The default is false.

SpeedBucketXML (Type: string, Required)

The "type" attribute for the Speed Bucket must be specified; the other attributes are generated when the Speed Bucket is created. The attributes for the "roadClosure" and "Buckets" elements must also be specified. (See the SpeedBucket response element for a detailed description of the attributes and elements.)

RETURN VALUES

Each XML response to this API call will contain two basic elements, each having several attributes and/or child elements.

- *Inrix* – The root element in the response for all APIs is always <Inrix>. It contains response data in child elements and information about the request in the following attributes:
 - *docType* identifies the API called and the schema used for the response.
 - *copyright* identifies the INRIX, Inc. copyright.
 - *versionNumber* refers to the version number of the INRIX traffic services platform that produced the response.
 - *createdDate* is the timestamp indicating when the response was created. The timestamp specifies a DateTime value, formatted according to the ISO 8601 standard and expressed in UTC time. The format of the attribute is YYYY-MM-DDTHH:MM:SSZ. The "T" character separates the date from the time, and the "Z" character specifies that UTC time is used.

- *statusId* indicates whether an API request succeeded or failed. Each *statusId* value is associated with a unique *statusText* value. A *statusId* is intended to be convenient to handle in code. *statusId*=0 is a successful call. [Appendix K](#) contains a list of common *statusID* responses.
- *statusText* corresponds to the *statusId*. A *statusText* is intended to be convenient for humans to interpret.
- *responseID* is logged by INRIX for future use in assisting customer inquiries. Service Requestors can use *responseID* as a way for INRIX to identify the specific API call that was made. It is recommended that Service Requestors log this information to expedite technical support incidents.
- *SpeedBuckets* – This child element is a container for all of the speed bucket data.

RESPONSE ELEMENTS

SpeedBuckets (Parent element: Inrix)

A container for all of the requested speed bucket data.

```
Syntax: <SpeedBuckets>
         <SpeedBucket.../>
         <SpeedBucket.../>
</SpeedBuckets>
```

Element	Required	Occurs	Description
SpeedBucket	yes	1..n	Defines how speed data is grouped and how it is displayed on Traffic Tiles.

SpeedBucket (Parent element: SpeedBuckets)

Defines how speed data is grouped and how it is displayed on Traffic Tiles.

```
Syntax: <SpeedBucket
         type= "integer"
         id= "string"
         name= "string"
         description= "string"
         private= "Boolean"
         createTime= "DateTime">
         <RoadClosure.../>
         <Buckets.../>
</SpeedBucket>
```

Attribute	Required	Type	Usage
type	yes	integer	The type of speed bucket: 1=Congestion 2=Speed 3=Relative Speed 4=Relative Congestion
id	yes	integer	The ID of the Speed Bucket.
name	yes	string	The name of the Speed Bucket.

description	no	string	A description of the Speed Bucket.
private	yes	Boolean	Whether the Speed Bucket is private or available to the public.
createTime	yes	DateTime	When the Speed Bucket was created.

Element	Required	Occurs	Description
RoadClosure	yes	1	Information for how a road closure is displayed on Traffic Tiles.
Buckets	yes	1	A collection of buckets that defines how speeds are grouped and displayed.

RoadClosure (Parent element: SpeedBucket)

Information for how a road closure is displayed.

```
Syntax: <RoadClosure
    penStyle= "integer"
    backColor= "string"
    foreColor= "string"
    hatchStyle= "string"
    penWidthPercent= "integer">
    <Pen.../>
</RoadClosure>
```

Attribute	Required	Type	Usage
penStyle	yes	integer	The style of the pen used to draw the section of road: 1=solid 2=hatched
backColor	yes	string	The color for solid pens and the background color for hatched pens.
foreColor	no	string	The accent color for hatched pens.
hatchStyle	no	string	The style for hatched pens.
penWidthPercent	no	integer	The percentage of the pen width used for drawing the road.

Element	Required	Occurs	Description
Pen	no	1..n	Information for how road closures are displayed for different road classes. If not specified, all road classes are drawn as specified in the <RoadClosure> attributes.

Buckets (Parent element: SpeedBucket)

A collection of buckets that defines how speeds are grouped and displayed.

```
Syntax: <Buckets>
        <Bucket ... />
        <Bucket ... />
</Buckets >
```

Element	Required	Occurs	Description
Bucket	yes	1..n	Information for how speed data is grouped and displayed.

Bucket (Parent element: Buckets)

Information for how speed data is grouped and displayed.

```
Syntax: <Bucket
        num= "integer"
        min= "integer"
        max= "integer"
        penStyle= "integer"
        backColor= "string"
        foreColor= "string"
        hatchStyle= "string"
        penWidthPercent= "integer">
        <Pen ... />
</Bucket>
```

Attribute	Required	Type	Usage
num	yes	integer	The index of the bucket in the series of bucket elements.
min	yes	integer	The minimum value for data that goes into this bucket.
max	yes	integer	The maximum value for data that goes into this bucket.
penStyle	yes	integer	The style of the pen used to draw the section of road: 1=solid 2=hatched
backColor	yes	string	The color for solid pens and the background color for hatched pens.
foreColor	no	string	The accent color for hatched pens.
hatchStyle	no	string	The style for hatched pens.
penWidthPercent	no	integer	The percentage of the pen width used for drawing the road.

Element	Required	Occurs	Description
Pen	no	1..n	Information for how speed data in this bucket is displayed for different road classes. If not specified, all road classes are drawn as specified in the <Bucket> attributes.

Pen (Parent elements: RoadClosure, Bucket)
 Information for how speed data is displayed for different road classes.

```
Syntax: <Pen
    roadClass= "integer"
    penStyle= "integer"
    backColor= "string"
    foreColor= "string"
    hatchStyle= "string"
    penWidthPercent= "integer"
```

/>

Attribute	Required	Type	Usage
roadClass	yes	integer	Federal Road Classification (FRC) code: 1= First class roads, such as national highway network roads 2= Second class roads, such as state highway network roads 3= Third class roads, such as state interconnecting network roads 4= Fourth class roads, such as major connecting roads 5= Fifth class roads, such as minor roads connecting suburbs 6= Sixth class roads, such as destination and destination collector roads 7= Seventh class roads, such as destination dead-end roads
penStyle	yes	integer	The style of the pen used to draw the section of road: 1=solid 2=hatched
backColor	yes	string	The color for solid pens and the background color for hatched pens.
foreColor	no	string	The accent color for hatched pens.

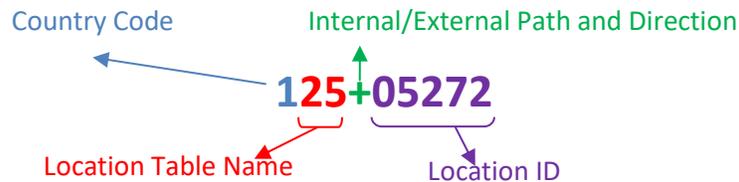
hatchStyle	no	string	The style for hatched pens.
penWidthPercent	no	integer	The percentage of the pen width used for drawing the road.

NOTES

- Pen color can be:
 - [.NET color name](#) as a string (for example, "ForestGreen")
 - ARGB color value in a standard web format with the colors specified as hex values in order a,r,g,b (for example, "#00112233")
 - ARGB color values can be used to create opacity when defining the color attribute of a speed bucket.
 - For example: To create an approximately 50% opaque red, do not use the pre-defined web color "red." Instead, use the ARGB color value "#80FF0000". The alpha value #80 is halfway between 00 and FF, thus specifying 50% opacity, and the red color is specified by the FF red value, 00 green value, and 00 blue value.
- Hatched pen style can be:
 - [.NET hatch styles](#) as a string (for example, "SmallCheckerBoard")

Interpreting TMC Codes

A TMC Code consists of nine characters that define a unique segment and direction of roadway in North America. The TMC Code detailed here provides the following information:



- **Digit 1** (“1” in this example) refers to the Country Code. The United States has the Country Code “1”, and Canada has “C”.
- **Digits 2-3** (“25” in this example) refer to the Location Table Name. North America is covered with thirty-five distinct tables as shown in the picture below. This configuration allows for the final six digits of the TMC Code to be reused, as well as allowing for the table coverage to grow in density while still maintaining the geographic integrity of the coding scheme.



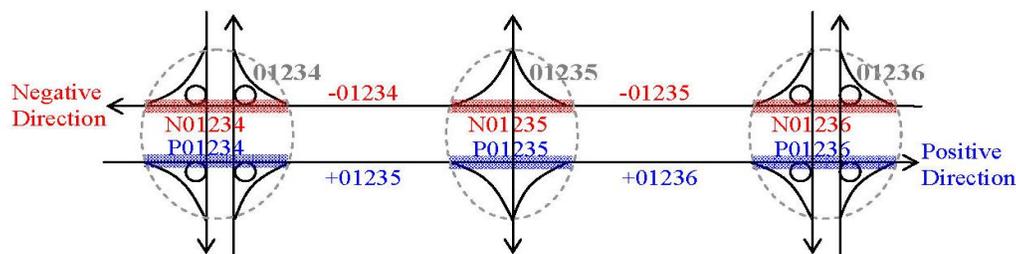
- **Digit 4** (“+” in this example) refers to both the direction of travel and if the TMC Segment type is “internal” or “external.” The meanings of the possible values (“+,” “-,” “P,” and “N”) are described in the following section.

- **Digits 5-9** (“05272” in this example) refer to the specific Location ID in the Location Table that is tied to a specific interchange, intersection, boundary, or decision point. In most cases there will be four distinct TMC Segments near one another on the road network that share this same five-digit Location ID, each preceded by either “+,” “-,” “P,” or “N.”

INTERNAL AND EXTERNAL TMC SEGMENTS

TMC location coding describes two segment types for every Location ID—internal and external. The internal segment refers to the area just past the decision point or intersection at which the Location ID was placed (for example, a freeway off ramp), while the external segment refers to the section of the road leading up the decision point. In order to maximize possible data precision, INRIX considers the internal segment and external segment as individual spatial locations; i.e., speed is reported separately for the two segments. Traffic congestion is often caused by traffic exiting at a decision point; thus, the external and internal segments corresponding to that Location ID might have substantially different speed profiles.

The figure below illustrates the difference between internal (labeled with ‘P’ or ‘N’) and external (labeled with a ‘+’ or ‘-’) TMC Segments.

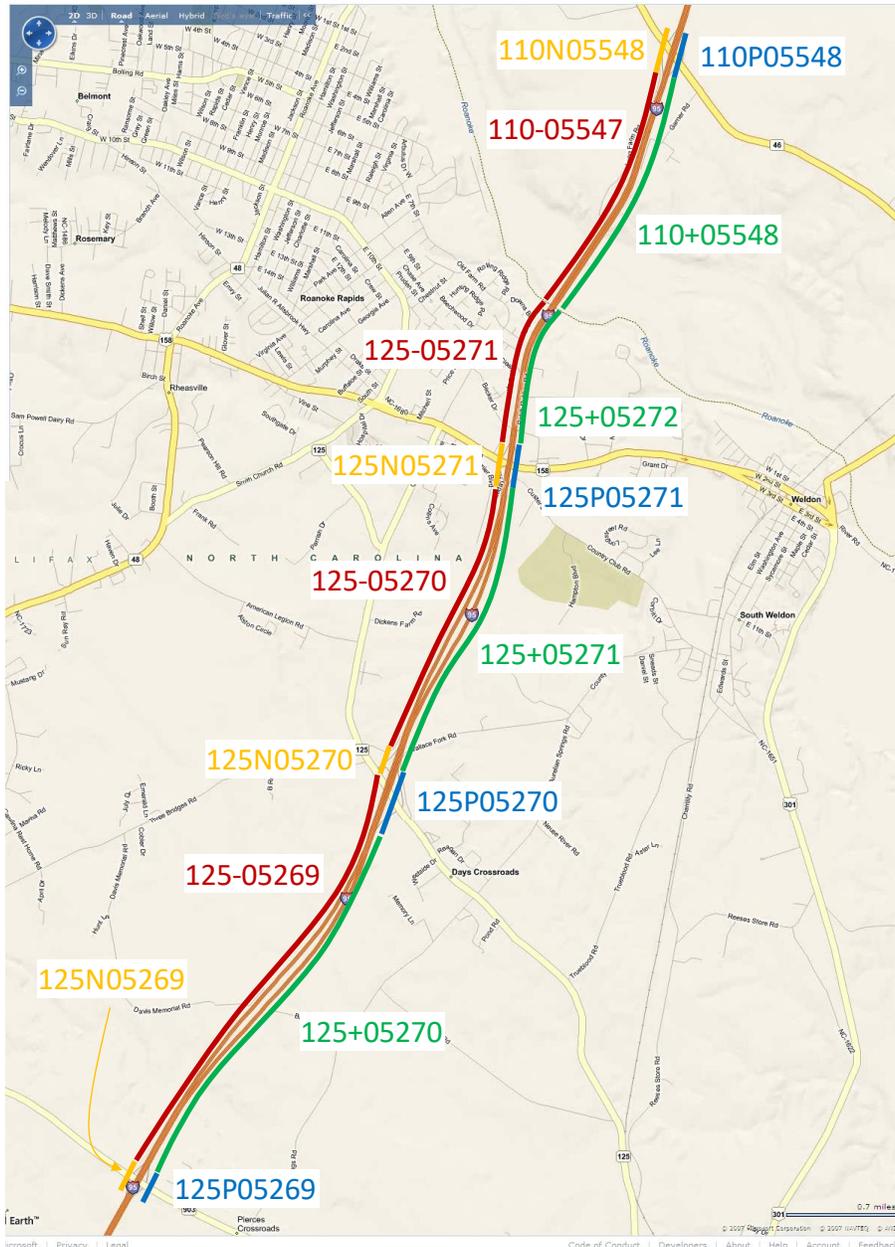


While there are several situations where this does not apply (for example, inner and outer loop road geometry), the typical convention for identifying the travel direction for the segment and whether the segment is internal or external is:

- “P” = Northbound or Westbound, internal segments
- “N” = Southbound or Eastbound, internal segments
- “+” = Northbound or Westbound, external segments
- “-” = Southbound or Eastbound, external segments

Sample TMC Segments

The following figure shows a map of an eight mile stretch of I-95 in northern North Carolina, the region approximately from the exit at mile marker 178 to the exit at mile marker 186. This corridor is defined by eight TMC Segments in each direction or sixteen in total. Note that each interchange is treated as an “internal” segment with a “P” or an “N” in the TMC Code. Each segment between interchanges is treated as an “external” segment with a “+” or a “-” in the TMC Code. This is generally true of the TMC implementation across road networks.



It's important to note that not every five-digit Location ID has both an internal and an external TMC Segment associated with it. Two external TMC Segments can be adjacent to each other. For example, at the northern end of this corridor there is a border between Halifax and Northampton counties at the Roanoke River. As is standard, the Location Table contains a five-digit Location ID at the border. Since the border is essentially zero length, there is no “internal” segment at the border, but rather a transition between two “external” segments in each direction.

Other TMC configurations include the following:

- Smaller roads will cross at a point and therefore have only “external” TMC Segments.
- Some complex interchanges create situations where “internal” TMC Segments overlap.
- Where highways merge, there will often be two sets of TMC Segments (one for each highway) over the same stretch of road.

TMC Segment Reference Tables

INRIX provides users with files containing the information necessary to interpret TMC Segments. The files contain data for all road segments in the region that have defined TMC Segments, and the TMC Segments that are covered in this project are a subset of the segments in these files. Each time INRIX introduces a new version of the TMC Segments based on a new TMC Table version, updated files will be distributed. Information from these files can only be used for the purposes of geo-locating data provided by this project.

The table on the following page shows the fields that will be provided for all TMC Segments. Note that the TMC Segments in the table have been arranged to show adjacent segments in order, which helps illustrate that the end latitude and longitude of a previous TMC Segment generally equals the start latitude and longitude of the next TMC Segment. Note, however, no sorting order is guaranteed in the actual table to be provided.

As a user determines which TMC Set(s) it will utilize in its application, the full TMC Segment Table can be filtered for only the TMC Segments included in the applicable TMC set(s). This filtered table will then provide the information necessary to reference the TMC Segment locations.

TMC SEGMENT TABLE DEFINITIONS

The column headings for the TMC Segment Table are defined below. Many of the columns are included as an aid to understand the codes; only a few of the columns are needed to reference the TMC Segment location:

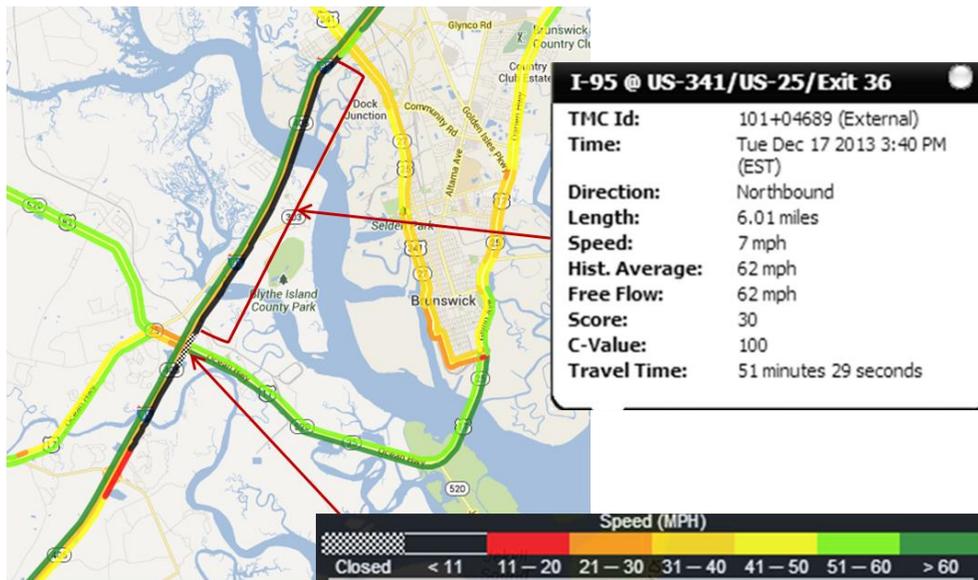
- TMC – The unique 9-digit value identifying the TMC Segment, as described previously in the Interpreting TMC Codes section. The TMC attribute is identical to the value of the TMC column.
- Type – The type of TMC Code. “P1” is the typical TMC Code. “P3” indicates national, state, and county boundaries, rest areas, toll plazas, major bridges, etc. “P4” is for ramps.
- RoadNumber – The roadway number, for road segments on numbered roadways.
- RoadName – The local name of the road, for road segments that have either no road number or both a name and number.
- FirstName – The cross street and/or interchange associated with the internal segment of 5-digit Location ID.
- LinearTMC – A reference to the “Linear TMC” that includes the TMC Segment. Typically, several TMC Segments are part of a Linear TMC, which usually represents a road corridor through a single county. The purpose of this column is to provide assistance for filtering and locating TMC Segments and simplifying the process of linking consecutive TMC Segments. Note that care must be taken when linking consecutive TMC Segments in the exception cases where multiple TMC Codes refer to the same stretch of road.
- Country – The country in which the TMC Segment is located.
- State – The postal abbreviation of the state to which the TMC interchange is assigned.
- County – The county to which the TMC interchange is assigned.
- Zip – The 5-digit zip code to which the TMC interchange is assigned.
- Direction – The direction of travel along a chain of TMCs that the TMC Segment defines.
- StartLat – The low-resolution latitude of the beginning of the TMC Segment.
- StartLong – The low-resolution longitude of the beginning of the TMC Segment.
- EndLat – The low-resolution latitude of the end of the TMC Segment.
- EndLong – The low-resolution longitude of the end of the TMC Segment.
- Miles – The length of the TMC Segment along the road in miles.

TMC SEGMENT TABLE EXAMPLE

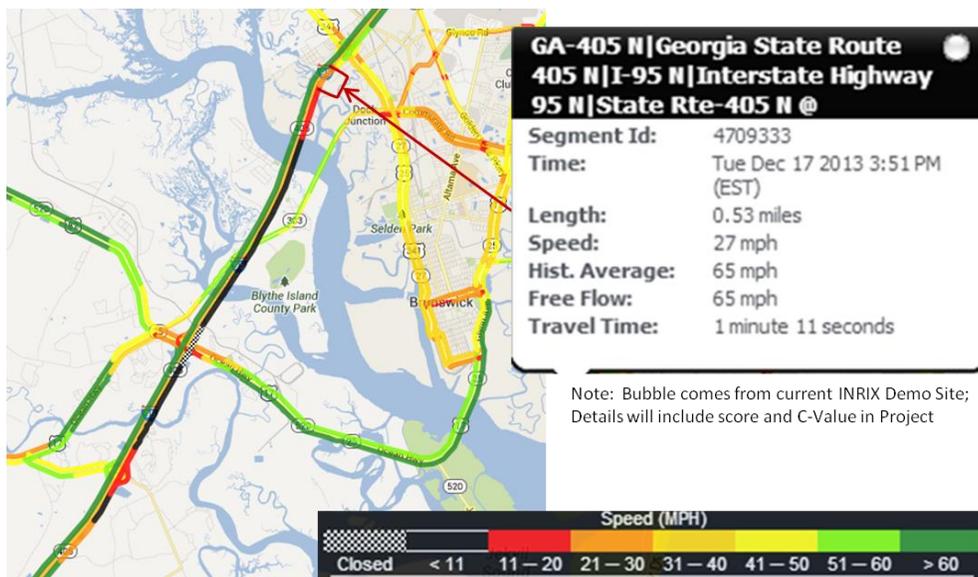
TMC	Type	RoadNumber	RoadName	First Name	LinearTMC	Country	State	County	ZIP	Direction	StartLong	StartLat	EndLong	EndLat	Miles
125F05269	P1	195	HWY 903/EXIT 168		12500118	USA	NC	HALIFAX	27870	NORTHBOUND	-77.67429	36.36146	-77.67092	36.36661	0.40198
125+05270	P1	195	HWY 125/EXIT 171		12500118	USA	NC	HALIFAX	27870	NORTHBOUND	-77.67092	36.36661	-77.64708	36.39645	2.45490
125F05270	P1	195	HWY 125/EXIT 171		12500118	USA	NC	HALIFAX	27870	NORTHBOUND	-77.64708	36.39645	-77.64245	36.40361	0.56853
125+05271	P1	195	US 158/EXIT 173		12500118	USA	NC	HALIFAX	27890	NORTHBOUND	-77.64245	36.40361	-77.63114	36.42800	1.80912
125F05271	P1	195	US 158/EXIT 173		12500118	USA	NC	HALIFAX	27890	NORTHBOUND	-77.63114	36.42800	-77.62920	36.43304	0.37657
125+05272	P3	195	HALIFAX/NORTHAMPTON COUNTY LINE		12500118	USA	NC	HALIFAX	27870	NORTHBOUND	-77.62920	36.43304	-77.62576	36.44408	0.86238
110+05548	P1	195	HWY 46/EXIT 176		11000199	USA	NC	NORTHAMPTON	27831	NORTHBOUND	-77.62470	36.44533	-77.61282	36.46558	1.62204
110F05548	P1	195	HWY 46/EXIT 176		11000199	USA	NC	NORTHAMPTON	27831	NORTHBOUND	-77.61282	36.46558	-77.61009	36.47082	0.36917
110N05548	P1	195	HWY 46/EXIT 176		11000199	USA	NC	NORTHAMPTON	27831	SOUTHBOUND	-77.60987	36.47130	-77.61276	36.46574	0.42230
110+05547	P3	195	NORTHAMPTON/HALIFAX COUNTY LINE		11000199	USA	NC	NORTHAMPTON	27831	SOUTHBOUND	-77.61276	36.46574	-77.62470	36.44533	1.60203
125-05271	P1	195	US 158/EXIT 173		12500118	USA	NC	HALIFAX	27890	SOUTHBOUND	-77.62576	36.44408	-77.62931	36.43268	0.89357
125N05271	P1	195	US 158/EXIT 173		12500118	USA	NC	HALIFAX	27890	SOUTHBOUND	-77.62931	36.43268	-77.63110	36.42810	0.33953
125-05270	P1	195	HWY 125/EXIT 171		12500118	USA	NC	HALIFAX	27870	SOUTHBOUND	-77.63110	36.42810	-77.64165	36.40534	1.68082
125N05270	P1	195	HWY 125/EXIT 171		12500118	USA	NC	HALIFAX	27870	SOUTHBOUND	-77.64165	36.40534	-77.64375	36.40081	0.33830
125-05269	P1	195	HWY 903/EXIT 168		12500118	USA	NC	HALIFAX	27870	SOUTHBOUND	-77.64375	36.40081	-77.67107	36.36641	2.83961
125N05269	P1	195	HWY 903/EXIT 168		12500118	USA	NC	HALIFAX	27870	SOUTHBOUND	-77.67107	36.36641	-77.67437	36.36132	0.39813

APPENDIX B – INRIX XD SEGMENTS

INRIX XD Segments, first introduced in late 2013, mark a key upgrade for INRIX and its customers. While similar to TMC Segments in that they delineate a specific section of roadway, XD Segments are defined and maintained by INRIX. This allows INRIX to provide more road coverage (all FRC 1, 2, and 3 level roads, plus any higher FRC roads covered by TMCs), better segment resolution with no overlaps or gaps, and the ability to cover new roads more quickly. As an example, the first map below uses INRIX data with TMC Segments. It shows two backups around a road closure. The second map, which uses INRIX XD Segments, includes more queue detail and more road coverage, including interchanges.

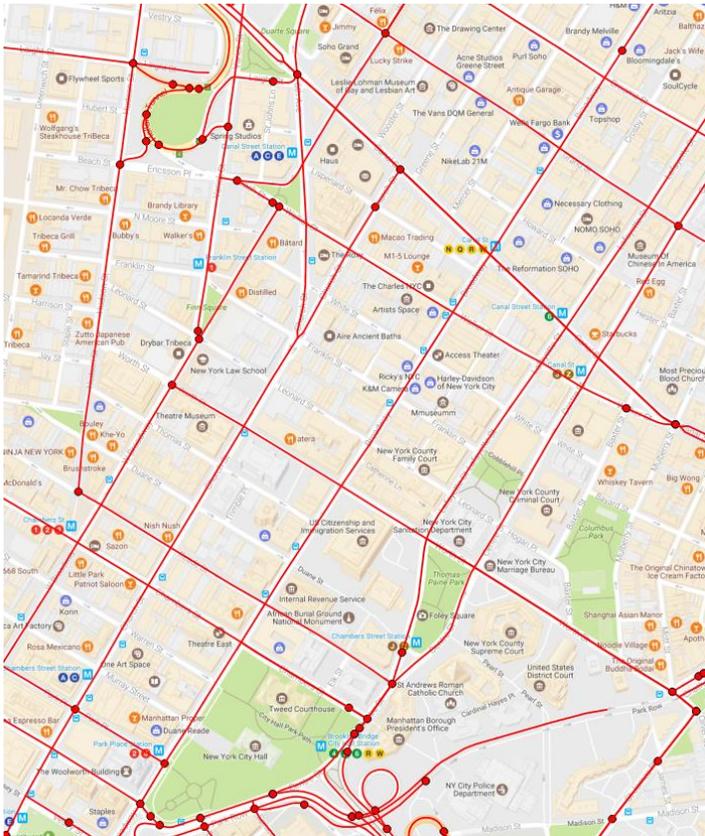


TMC Segments

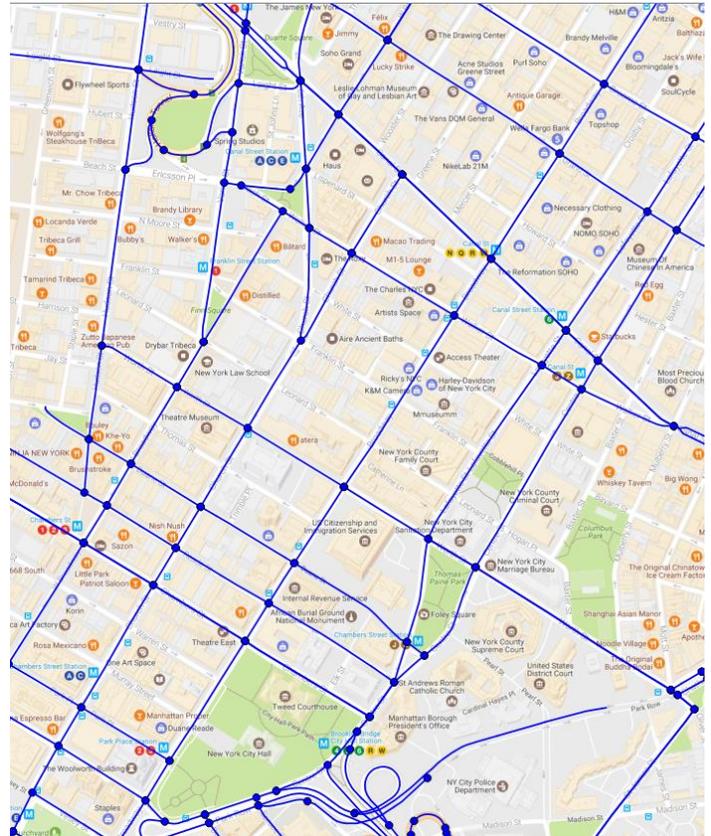


INRIX XD Segments

New XD functionality introduced in 2017, known as XD v2, improves the utility of data on arterial networks with two key enhancements: the intentional termination of segments at junctions/intersections/interchanges and an increase in the granularity of segments, particularly in urban signalized corridors. The maps below of the Soho area in lower Manhattan illustrate the difference between XD v1 and XD v2 segmentation.



XD v1

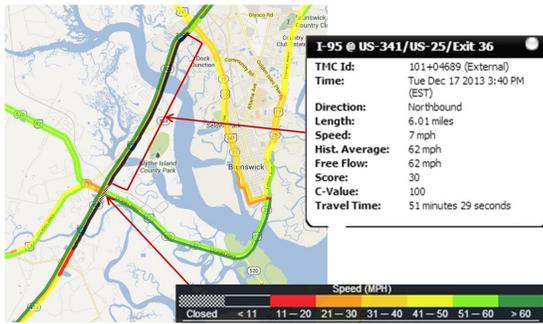


XD v2

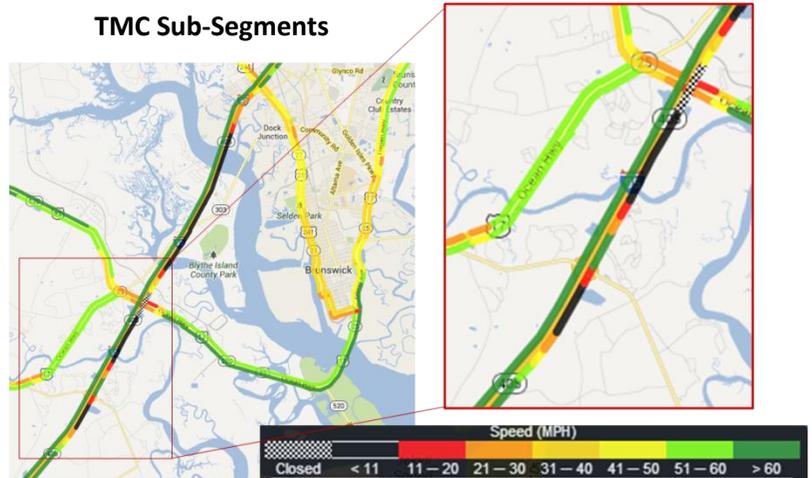
APPENDIX C – SUB-SEGMENTS

Sub-segment options for INRIX XD Segments and TMC Segments also further improve granularity, enabling greater resolution on any road covered by INRIX XD Segments or TMC Segments. The following maps illustrate the difference for TMC Segments and for INRIX XD Segments.

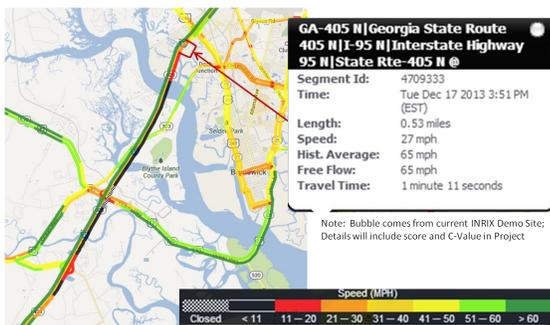
TMC Segments



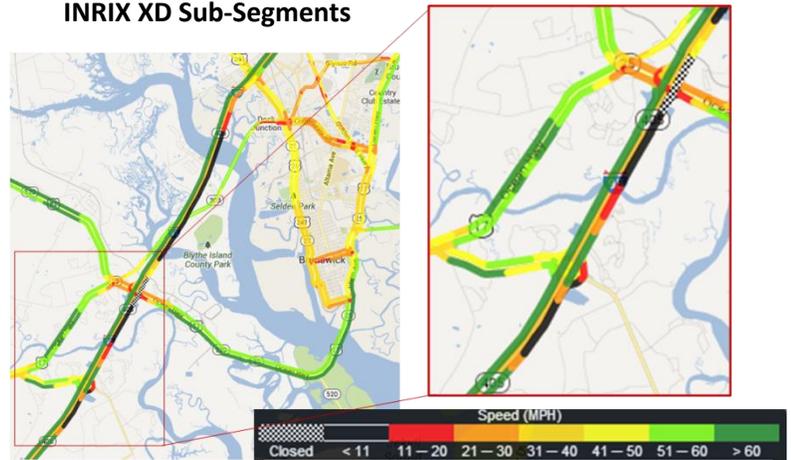
TMC Sub-Segments



INRIX XD Segments



INRIX XD Sub-Segments



Both TMC and XD segments will be accessible in “sub-segment” granularity, with flexibility depending upon the use case and desired integration sophistication. Some API calls have optional parameters to allow the return of sub-segment speeds nested within the values returned for each segment. Sub-segment length can also be specified.

Another parameter will allow the user to determine when sub-segment speed values are returned based on the variance of the sub-segment speed from the full segment speed. Setting this parameter to zero mph would return all sub-segments. Setting it to 5 mph (for example) would only return sub-segment speeds where the difference from the segment speed is greater than 5 mph. This allows a user to determine the most effective way to return sub-segment data based on their particular use case. Returning all sub-segment data will create “fixed” segments that are easier to integrate; however, the size of the returned payload will also increase significantly.

APPENDIX D – INRIX-MANAGED SET FILES

Project File Architecture

Since a project area may ultimately span tens of thousands of road miles and many states, INRIX employs a modular implementation to reduce file sizes and the complexity of requests and responses. The overall roadway coverage is subdivided into discrete lists (or “sets”) of segments, specified by a SetId. Segment Sets can also be comma separated within a call to return more than one set within a single response.

INRIX initially prepares set files for each state containing all contract road mileage by state broken out by freeway, arterial, ramp, and all coverage.

(Note: INRIX also offers users the ability to create and manage their own sets where subsets of the data can be called with a SetID that is created independent of INRIX.)

Coverage Expansion

If and when roadway coverage is added, it could be in a new state or in a state where coverage already exists. The process for expanding coverage under these two scenarios differs slightly. For each new state, additional set files would be created, just as for the original core states. However, when coverage is expanded in a state where coverage currently exists, the new road segments will be added to the existing sets as part of the mobilization process. The original SetIDs will be maintained but may return additional segments. Coverage may also be added with each map update, as more roads become available in the TMC tables and underlying map data. Integrators should plan on the possibility that the number of segments within a set could change if coverage is added in specific states.

Map Updates

INRIX updates the underlying map data for which speeds are reported approximately twice per year. This allows INRIX to report speeds on the most current/accurate road geometry and coverage. There will be annual communications from INRIX specific to the updating of map data and how any update may affect those using INRIX data.

Note that when INRIX performs a map data update, some TMC and XD locations and lengths may change and information about changes will be included in INRIX communications about digital map data updates.

This project will facilitate many kinds of applications, some of which will be map-based. There is a wide variety of vendors for map data, and the choice of vendor has significant ramifications on how TMC data can be integrated with maps. Some map vendors provide integrated support for TMC Segments, while others require the user to implement their own TMC Segment rendering solution. As a result, there are many different environments that will exist, each having different levels of integration complexity. While it is not possible to provide detailed information for all scenarios, some guidance on integration for likely scenarios is provided below.

- *Integration with TomTom Maps* – This is perhaps the simplest integration scenario. If a user has a TomTom map license, the license should include TMC layer information that will facilitate a simple translation from TMC paths to TomTom map Edge Ids.
- *Integration with HERE (previously NAVTEQ) Maps* – If a user has a current HERE map license, the license should include TMC layer information that will facilitate a simple translation from TMC Segments to HERE map Link Ids.
- *Integration with KML-based Internet Maps* – Keyhole Mark-up Language (KML) based maps, such as those supplied by Google or Microsoft, are increasingly being used to support map-based applications. While these Internet-based mapping services offer high quality maps, usually derived from TomTom or HERE (previously NAVTEQ) map databases, they currently do not expose TMC Codes or Segments. Thus, any effort to overlay traffic data onto these maps will require a one-time rendering of TMC Segments to create a traffic data layer suitable for use as an overlay. The TMC Segment Table contains start and end points for the TMC Segments, but the information necessary to follow the roadway geometry requires map data INRIX does not own and is not licensed to provide. A user must create detailed roadway geometry manually. Alternatively, if the user has a license for TomTom or HERE map data, it may be possible to automate the one-time rendering of TMC Segments to a traffic data layer. For visualization purposes, please see the sections specific to INRIX Traffic Tiles, which will be helpful in overlaying INRIX Traffic Tiles on a base map.
- *Integration using User- Developed Base Maps* – Since user-developed maps may not be based on currently licensed TomTom or HERE (previously NAVTEQ) commercial map data, creation of a traffic data overlay will need to create either a map layer of TMCs or an association of TMCs to an existing map layer to allow visualization, overlay and/or analysis.
- *Integration with Stylized Maps* – Stylized maps are not location referenced. In this scenario, mapping TMC paths to the stylized map will be required.
- *Applications Not Using Maps* – Some applications will not require maps or geo-referencing. Examples include travel time and event alerts for dynamic message signs, 511 telephone services, and/or web sites. For corridor travel time applications, summing the *travelTimeMinutes* attribute for a list of selected TMCs yields the cumulative travel time. For applications that convert traffic data into anecdotal alert messages, a list of TMC Segments can be monitored so that slowdowns that traverse multiple segments can be combined into a single alert. It is possible to utilize information provided in the TMC Segment Table to establish the TMC Sets, including RoadName, Direction, and the start and end latitudes and longitudes. Information provided in the TMC Segment Table (such as RoadName, direction, LinearTMC, etc.) can also aid in automatically creating alerts.

INRIX uses the concept of a Speed Bucket to separate data into groups (or “buckets”), based on speed or congestion level. There are four types of Speed Bucket:

1. Congestion, which displays the current speed as a percentage of free flow or reference speed
2. Speed, which displays the absolute current reported speed
3. Relative Speed, which displays the absolute difference between current reported speed and typical speed
4. Relative Congestion, which displays the percentage difference between current congestion (current speed versus free flow speed) and typical congestion (typical speed versus free flow speed)

The Speed Bucket definitions are most useful on Traffic Tiles providing a means to specify how different speeds are depicted. The pen style, color, hatching, and width of the lines associated with each bucket, as well as with a road closure, can be defined. Users can also specify different line styles for different road classifications, allowing skinnier lines for smaller roads, for example.

All users have access to six pre-defined INRIX Speed Buckets, represented by SpeedBucketIds 1-6. Definitions of these buckets can be retrieved using the GetSpeedBucket API call. Users also have the ability to define their own Speed Buckets, using the CreateSpeedBucket API call.

Details for the API requests and responses can be found in the API Calls section of this document. (Click on the API names to jump directly to the referenced locations.)

REQUESTS

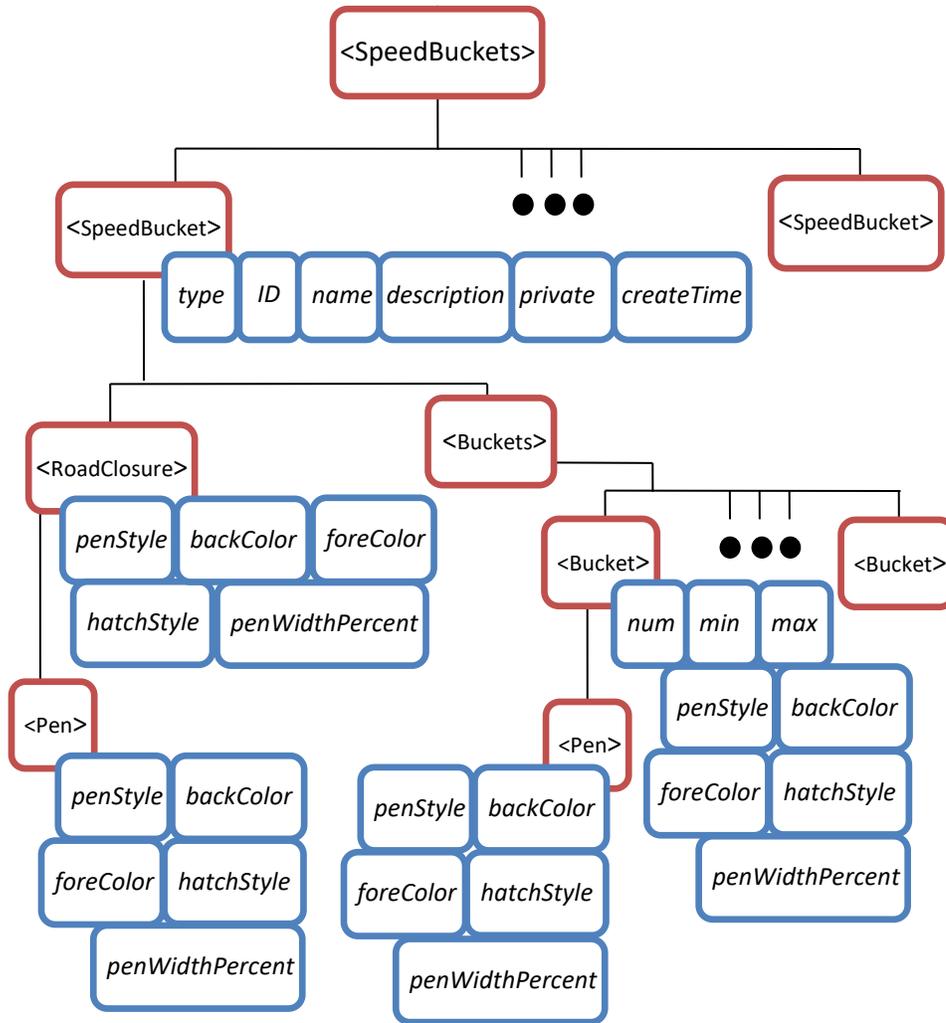
[GetSpeedBucket](#)– Retrieves the definition of one (or all) INRIX-defined or user-created Speed Buckets, identified by SpeedBucketIds.

[CreateSpeedBucket](#)– Generates a new user-defined Speed Bucket.

RESPONSES

Speed Bucket responses are delivered in XML format, similar to other basic traffic data. In general, each XML response contains a common root element <Inrix>, which has several attributes (*docType, copyright, versionNumber, createdDate, statusId, statusText, responseId*). It also contains a <SpeedBuckets> child element that serves as a container for the Speed Bucket data requested in the API call and has the following structure:

- The specifications for each defined Speed Bucket (identified by a SpeedBucketId) are contained in one or more child elements called <SpeedBucket>. Each of these elements contains some or all of the following attributes: *type, ID, name, description, private, and createTime*. They also contain a <RoadClosure> element with attributes specifying how a road closure is to be displayed, as well as a <Buckets> element to define how groups of speeds are to be displayed.
- Each <Buckets> element contains one <Bucket> child element for each range of speeds to be used.
- Each <Bucket> element will include the *num, max* and *min* attributes to identify what speeds map into the bucket, in addition to attributes describing how to display the segments in that bucket.



Both Speed Bucket APIs return the same type of data, defining the Speed Buckets and detailing how to display the associated lines on a map. The reply to CreateSpeedBucket essentially echoes back the parameters specified in the call.

APPENDIX G – SCORE

The *score* is used to determine the source of the speed data returned in an API call.

Score = 30 Real Time Data

- Any segment that has adequate data, at any time of day, will report real-time data.

Score = 20 Historical Average

- Between 4 am and 10 pm, any segment without sufficient real-time data will show the historical average for that segment during that day/time period (15-minute granularity).

Score = 10 Reference Speed

- From 10 pm to 4 am, any segment without sufficient real-time data will show the reference speed for that segment.
- Any segment that does not have calculated historical averages will show the reference speed 24 hours a day if there is not sufficient real time data.

The *score* provides a means for users to easily select only real-time data, if that is what they require for their application or to meet their internal standards.

APPENDIX H – CONFIDENCE (C-VALUE)

The *c-value*, or confidence value, is designed to provide supplemental information to the *score* attribute to best identify the type and confidence of the data being sent by INRIX.

The *score* is used to determine the type of data, and the *c-value* should be used to add commentary to the *score*, or a confidence of the real-time data (where *score*=30). The *c-value* alone is not intended to convey whether the data is purely real-time, nor is it intended to map to specific thresholds relating to use. Rather, when used in conjunction with *score*, the *c-value* allows agencies to make an independent determination as to the type and confidence of the data based on individual use cases.

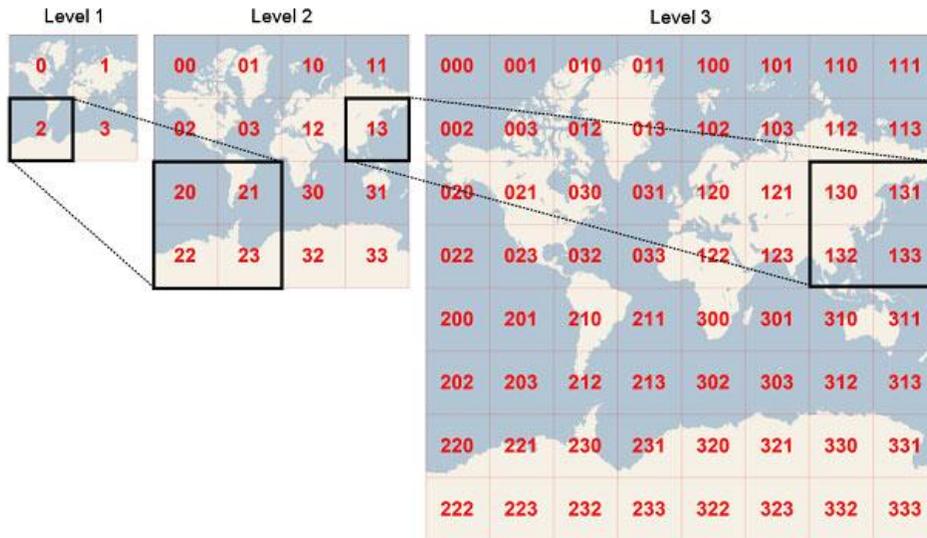
The I-95 Corridor Coalition, in joint effort with the University of Maryland, has prepared a [white paper](#) specific to the *c-value* and its suggested use. The following points should be considered when using the *c-value*:

- Real-time speeds have variability, which is a function of driver behavior and congestion state.
- A model of confidence must capture this variability and identify speed estimates that are inconsistent with “expectations.”
- Our model of “expectations” is based on three time-scales that are used within the INRIX calculation to produce the single *c-value* (it is a blended calculation based on all three time-scales).
 - Short-Term:
 - Probe data is a reliable estimator of current speeds if density is sufficient.
 - If sufficient real-time data points exist, then confidence value is 100.
 - Mid-Term:
 - Compute a model of speed distributions within the last 45 minutes.
 - Is the current speed probable given mid-term speeds?
 - Confidence value is proportional to probability.
 - Long-Term:
 - Historic data can be used to create a very dense model of average behavior. Within a broader range, these speeds can validate current speeds.
 - If current speed is within confidence interval about the historic average, then confidence value is 100.
- The *c-value* is expected to be returned only when a sufficient amount of real-time data exists for which to calculate confidence (in other words, when *score* is “30”).
- *c-value* of 100 means that there was high data density, the current data is very consistent with the data over the past 45 minutes, and/or the current data is very consistent with the historical data.
- The *c-value* was jointly developed between INRIX and the University of Maryland, for which benchmarking has been done.
- Validation testing by the University of Maryland has determined that there is a linear relationship between the *c-value* and average error— the higher the *c-value*, the lower the average error.

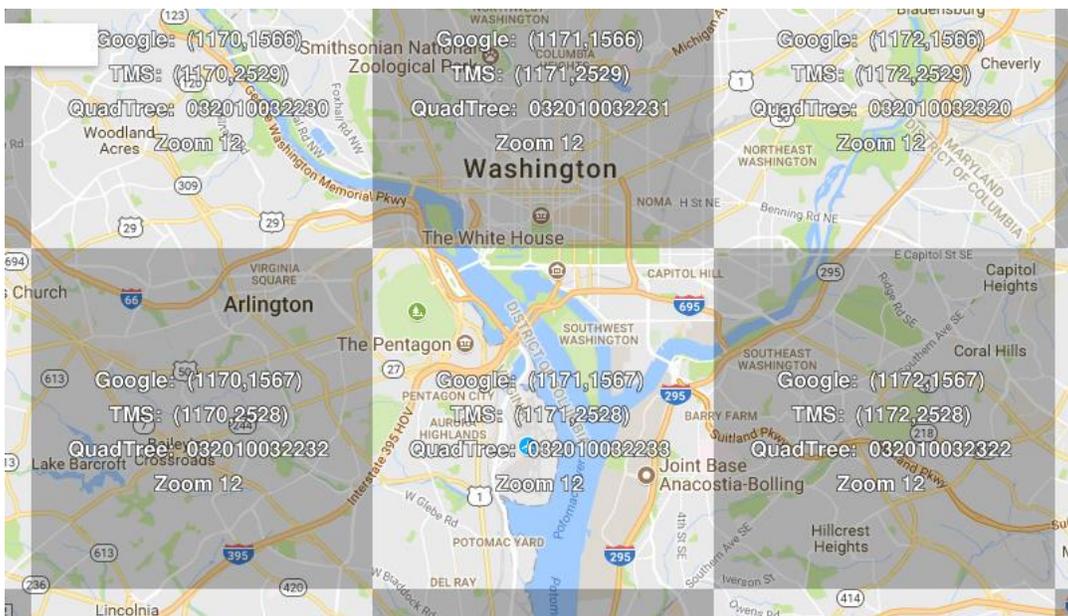
APPENDIX I – QUADKEYS

INRIX recommends the use of quadkeys for obtaining traffic tiles (although a bounding box may be used to better define custom areas).

Quadkeys define a specific section on a map at a specific level of detail. As illustrated below, a longer quadkey value indicates a smaller map section and a higher level of detail. More information on the use of Bing Maps Tile System quadkeys to define map tiles can be found at <https://msdn.microsoft.com/en-us/library/bb259689.aspx>.



An easy way to determine the quadkey for a region is to use the application pictured below, found at <http://www.maptiler.org/google-maps-coordinates-tile-bounds-projection/>. This app allows users to search and/or zoom on the world map to a desired region and read off the associated quadkey (identified as “QuadTree”). The numerical boundaries for the tile can also be displayed by clicking on the tile image.



PARAMETERS

appld (Type: GUID, Required)

A unique ID assigned by INRIX to each application.

hashToken (Type: string, Required)

This is generated by the following sequence:

1. Create a string with a value of AppId|AppKey – The AppKey is a value provided by INRIX and is separate from the AppId value (also provided for each customer). The string is the AppId joined to the Appkey separated by a | symbol.
2. Lowercase the string value.
3. Encode the string into a byte array using UTF-8 encoding.
4. Use SHA1 to hash the bytes.
5. Write the resulting bytes in hexadecimal for the hashToken value.

RETURN VALUES

Each response to this API call will contain:

- *copyright* identifies the INRIX, Inc. copyright.
- *createdDate* is the timestamp indicating when the response was created. The timestamp specifies a DateTime value, formatted according to the ISO 8601 standard and expressed in UTC time. The format of the attribute is YYYY-MM-DDTHH:MM:SSZ. The “T” character separates the date from the time, and the “Z” character specifies that UTC time is used.
- *responseID* is logged by INRIX for future use in assisting customer inquiries. Service Requestors can use *responseID* as a way for INRIX to identify the specific API call that was made. It is recommended that Service Requestors log this information to expedite technical support incidents.
- *result* contains the requested token and its expiry date/time.
- *versionNumber* refers to the version number of the INRIX platform that produced the response.

Tile API

DESCRIPTION

Generates a Traffic Tile overlay with an area specified by a Bing Maps Tile System quadkey. Segments included in the returned tile can be limited by road segment type (XD or TMC) and/or FRC level, but Segments or Segment Sets cannot be specified.

EXAMPLE CALL REQUEST

```
http://tile-api.inrix.com/v1/tiles/0320120020?accessToken=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhbnR5cCI6ImVjOWM3ZDAwLWUwNjUtNDc1OS05M2UyLWl3MWJhMmQ0MTAyYSIsIkV4cGlyeSI6IjIwMTUyMDYtMDVUMjI6NTg6MzkuMDcwMzQ5N1oiLCJleHAiOiE0MzI1NDUxMTksInR5cCI6IkpXVCJ2aWNlIn0.5fL8TL9z6hq2smrpbFhZE3S2CATTvhYNHgzFa5TQtzE
```

EXAMPLE CALL RESPONSE



SYNTAX

```
http://tile-api.inrix.com/v1/tiles/quadkey=string
?accessToken=string
&roadSegmentType=string
&FRCLLevel=string
&penWidth=float
&opacity=string
&speedBucketID=integer
```

PARAMETERS

accessToken (Type: string, Required)

A valid access token as returned in the appToken API response.

quadkey (Type: string, Required)

A Bing Maps Tile System quadkey that corresponds to the area for which to get data. The minimum quadkey length is 6 digits and the maximum is 17. (For more information about using quadkeys, see Appendix I.)

roadSegmentType (Type: string, Optional)

The Road Segment Type (Location Referencing method) for the traffic tile to be.

TMC (default)	Returns Traffic Tiles based on TMCs.
XDS	Returns Traffic Tiles based on XD Segments.

FRCLevel (Type: string, Optional)

The Federal Road Classification code of the roads to report. Multiple codes can be specified in a comma-delimited list.

All (default)	All road types.
1	First class roads, such as national highway network roads.
2	Second class roads, such as state highway network roads.
3	Third class roads, such as state interconnecting network roads.
4	Fourth class roads, such as major connecting roads.
5	Fifth class roads, such as minor roads connecting suburbs
6	Sixth class roads, such as destination and destination collector roads.
7	Seventh class roads, such as destination dead-end roads.

penWidth (Type: float, Optional)

The pen width of the lines on the Traffic Tile overlay, in pixels. The default is 4.

opacity (Type: string, Optional)

The pen opacity, as a percentage value between 0 and 100. The default is 100.

speedBucketID (Type: integer, Optional)

A Speed Bucket is a range of speeds or percentages that is used to categorize data and specify how it should be displayed on a Traffic Tile. All Speed Buckets are identified by a unique number called a SpeedBucketID.

If no Speed Bucket is specified, the default is the INRIX-created Speed Bucket with SpeedBucketID=1, which is defined as:

Color=black, if current speed is 0-31% of reference speed

Color=red, if current speed is 32-62% of reference speed

Color=yellow, if current speed is 63-92% of reference speed

Color=green, if current speed is 92-100% of reference speed

(For more information about creating custom Speed Buckets, see [Appendix E.](#))

RETURN VALUES

A PNG image whose size, bounds, and contents are determined by the parameters specified in API call.

APPENDIX K – COMMON STATUS IDS

StatusID	StatusText	HttpResponseCode	Discussion
0	OK	200	normal processing
1	no vendorId	403	missing vendorId parameter
2	vendorId not found	403	vendorId not in database
3	vendor forbidden	403	vendor is not authorized to perform the action
4	no action	400	missing action parameter
5	bad action	400	malformed action parameter
8	unknown service error	500	failure in underlying service
13	subscription expired	400	the consumer's subscription has expired, according to Inrix data
17	bad consumerId	403	malformed consumerId
18	no consumerId	403	missing consumerId parameter
19	consumerId not found	403	consumerId not in database
20	vendorGroup not found	404	vendorGroup not in database
41	NoToken	400	missing Token parameter
42	BadToken	400	malformed Token parameter
43	TokenExpired	401	Token has expired

APPENDIX L – CONTACTS AND RESOURCES

Points of Contact

UNIVERSITY OF MARYLAND/I-95 CORRIDOR COALITION

Contacts for the following areas can be found on the Vehicle Probe Project page of the Coalition website (<http://i95coalition.org/projects/vehicle-probe-project/>):

- General Project Questions
- Contracting Issues and Data Use Agreement (DUA)
- Data Validation
- Probe Data Analytics Suite (formerly the VPP Suite)

INRIX

General Support Questions:

Email: I95support@inrix.com

Phone: 425-284-3870

Technical Lead, Integration and Support/Public Sector Technical Account Manager (PSTAM):

Email: PSTAM@inrix.com

Project Manager: (All electronic inquiries should also be copied to Mr. Schuman.)

Rick Schuman

Email: rick@inrix.com

Phone: 407-572-5584

The procedures for contacting INRIX for real-time data services support issues are provided separately, as these procedures differ markedly from assisting in integration and should be followed to ensure proper treatment of an issue in a timely fashion.

Resources Available Through INRIX

EMAIL DISTRIBUTION LIST

To be added to the email distribution list that is used for project updates, scheduled maintenance notifications, and other related communications, please send an email to i95support@inrix.com requesting that appropriate contacts be added to the distribution. Please include the agency and email to be added.

I-95 MONITORING SITE

All member agencies that have signed a DUA with the University of Maryland have access to the I-95 Monitoring Site. To obtain access once a DUA has been signed, please visit the site at www.i95.inrix.com and follow the registration link. Please register using your agency email address. Your account will be approved within 24-48 hours of registration.

Approval is manual and is not immediate.

HISTORICAL DATA

INRIX can provide access to historical archived speed data for roads that are under contract. The Historical Data is not available via API, but rather by state and road type within a .csv file that is returned after a request is made on the I-95 Monitoring Site hosted by INRIX. Data archive requests may take up to 48 hours to process, and users will need to return to the monitoring site to obtain the results.

INTEGRATION RESOURCES AND ASSISTANCE

While the Coalition and its member agencies may have several resource options from internal staff on-call or contracted consulting/integration support, the INRIX team also has consulting and integration resources available that can assist agencies in planning and executing integration. These resources are available via task order under the INRIX contract. Should you require additional assistance please contact: i95support@inrix.com.

GENERAL INFORMATION

For general information and the latest news related to INRIX data services, please visit www.inrix.com.