

Iowa DOT Traffic Operations Open Data Service

**User Guide and Software
Requirements Specification**

Version 1.2, April 2018



IOWA STATE UNIVERSITY
Institute for Transportation

Sponsored by
Iowa Department of Transportation
(InTrans Project 16-587)
Iowa Statewide Transportation
Innovation Council
Federal Highway Administration

About InTrans

The mission of the Institute for Transportation (InTrans) at Iowa State University is to develop and implement innovative methods, materials, and technologies for improving transportation efficiency, safety, reliability, and sustainability while improving the learning environment of students, faculty, and staff in transportation-related fields.

About CTRE

The mission of the Center for Transportation Research and Education (CTRE) at Iowa State University is to develop and implement innovative methods, materials, and technologies for improving transportation efficiency, safety, and reliability while improving the learning environment of students, faculty, and staff in transportation-related fields.

ISU Non-Discrimination Statement

Iowa State University does not discriminate on the basis of race, color, age, ethnicity, religion, national origin, pregnancy, sexual orientation, gender identity, genetic information, sex, marital status, disability, or status as a U.S. veteran. Inquiries regarding non-discrimination policies may be directed to Office of Equal Opportunity, 3410 Beardshear Hall, 515 Morrill Road, Ames, Iowa 50011, Tel. 515-294-7612, Hotline: 515-294-1222, email eooffice@iastate.edu.

Disclaimer Notice

The contents of this document reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. The opinions, findings and conclusions expressed in this publication are those of the authors and not necessarily those of the sponsors.

The sponsors assume no liability for the contents or use of the information contained in this document. This document does not constitute a standard, specification, or regulation.

The sponsors do not endorse products or manufacturers. Trademarks or manufacturers' names appear in this report only because they are considered essential to the objective of the document.

Iowa Department of Transportation Statements

Federal and state laws prohibit employment and/or public accommodation discrimination on the basis of age, color, creed, disability, gender identity, national origin, pregnancy, race, religion, sex, sexual orientation or veteran's status. If you believe you have been discriminated against, please contact the Iowa Civil Rights Commission at 800-457-4416 or the Iowa Department of Transportation affirmative action officer. If you need accommodations because of a disability to access the Iowa Department of Transportation's services, contact the agency's affirmative action officer at 800-262-0003.

The preparation of this document was financed in part through funds provided by the Iowa Department of Transportation through its "Second Revised Agreement for the Management of Research Conducted by Iowa State University for the Iowa Department of Transportation" and its amendments.

The opinions, findings, and conclusions expressed in this publication are those of the authors and not necessarily those of the Iowa Department of Transportation or the U.S. Department of Transportation.

Technical Report Documentation Page

1. Report No. InTrans Project 16-587		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle Iowa DOT Traffic Operations Open Data Service				5. Report Date April 2018	
				6. Performing Organization Code	
7. Author(s) Skylar Knickerbocker (orcid.org/0000-0002-0202-5872), Vamsi Krishna Jagarlamudi (orcid.org/0000-0002-0205-8340), Neal Hawkins (orcid.org/0000-0003-0618-6275), and Anuj Sharma (orcid.org/0000-0001-5929-5120)				8. Performing Organization Report No. InTrans Project 16-587	
9. Performing Organization Name and Address Center for Transportation Research and Education Iowa State University 2711 South Loop Drive, Suite 4700 Ames, IA 50010-8664				10. Work Unit No. (TRAIS)	
				11. Contract or Grant No.	
12. Sponsoring Organization Name and Address Iowa Statewide Transportation Innovation Council Iowa Department of Transportation 800 Lincoln Way Ames, IA 50010				13. Type of Report and Period Covered User Guide and Software Requirements Specification – Version 1.2	
				14. Sponsoring Agency Code STP-000-S(916)--2C-00	
15. Supplementary Notes Visit www.intrans.iastate.edu for color pdfs of this and other research reports.					
16. Abstract <p>The goal of this project was to establish the Iowa Department of Transportation (DOT) open traffic data service, which will allow vendors and agencies to provide near real-time, proactive alerts to commercial drivers regarding traffic conditions along their routes. This open data service provides operations-related data to any commercial, public or private entity to develop their own applications or integrate these data into their existing hardware, software, and logistics systems.</p> <p>The purpose of this document is to give a detailed description of the requirements for the open data service maintained by the REACTOR Lab at the Institute for Transportation on behalf of the Iowa Department of Transportation (DOT). The report explains the purpose and complete declaration for the development of the system, as well as the system constraints, interface, and interactions with other external applications.</p> <p>The open data service makes transportation related data accessible to both internal DOT and external third-party users. A combination of high-performance computing (HPC) on premise and on the cloud was used to create data feeds which provide clean data streams, alerts for traffic congestion using machine learning, performance reporting, and data analytics.</p>					
17. Key Words data feeds—open data—traffic operations				18. Distribution Statement No restrictions.	
19. Security Classification (of this report) Unclassified.		20. Security Classification (of this page) Unclassified.		21. No. of Pages 70	22. Price NA

IOWA DOT TRAFFIC OPERATIONS OPEN DATA SERVICE

User Guide and Software Requirements Specification
Version 1.2
April 2018

Principal Investigator

Skylar Knickerbocker, Research Engineer
Center for Transportation Research and Education, Iowa State University

Co-Principal Investigators

Neal Hawkins, Associate Director
Institute for Transportation, Iowa State University

Anuj Sharma, Research Scientist
Center for Transportation Research and Education, Iowa State University

Authors

Skylar Knickerbocker, Vamsi Krishna Jagarlamudi, Neal Hawkins, and Anuj Sharma

Sponsored by
Iowa Statewide Transportation Innovation Council,
Iowa Department of Transportation, and
Federal Highway Administration

Preparation of this report was financed in part
through funds provided by the Iowa Department of Transportation
through its Research Management Agreement with the
Institute for Transportation
(InTrans project 16-587)

A report from
Institute for Transportation
Iowa State University
2711 South Loop Drive, Suite 4700
Ames, IA 50010-8664
Phone: 515-294-8103 / Fax: 515-294-0467
www.intrans.iastate.edu

TABLE OF CONTENTS

ACKNOWLEDGMENTS	ix
GLOSSARY	x
DOCUMENT REVISION HISTORY	x
INTRODUCTION	1
Purpose.....	1
Document Conventions.....	2
Project Scope	2
References.....	2
OVERALL DESCRIPTION.....	3
Product Perspective.....	3
Product Functions	3
Data Archival	3
Work Zone Congestion Detection	5
Data Feeds.....	5
User Class and Characteristics.....	6
Operating Environment.....	6
Design and Implementation Constraints.....	7
User Documentation	7
Assumptions and Dependencies	7
EXTERNAL INTERFACE REQUIREMENTS.....	8
User Interfaces	8
Console Application.....	8
Hardware Interfaces	9
Software Interfaces	9
Communications Interfaces	10
FUNCTIONAL REQUIREMENTS.....	11
Data Archival	11
Traffic Detector Inventory Data.....	11
Detector Traffic Data	11
INRIX Data.....	12
Waze Data.....	12
DMS Inventory Data.....	13
Work Zone Congestion Events	13
INRIX Daily Performance Measures.....	14
INRIX Monthly Performance Measures	14
INRIX Historical Performance Measures.....	15
ATMS Event Data.....	15
Camera Inventory.....	16
Waze Cluster Data	16
LCPT Monthly Sensor Data.....	17

Sensor Stations Data	18
Sensor Stations Data for LCPT	18
Snowplow AVL Data	19
Work Zone Congestion Detection	19
Work Zone Metadata	19
Work Zone Events	20
Work Zone Schedulers	20
Text Alert Content Generation	21
SMS Gateway Interaction	21
Data Feeds	22
Joint Sensor Data	22
Work Zone Alert Feed	23
DMS Inventory History Feed	23
INRIX Daily Performance Measure Feed	24
INRIX Monthly Performance Measure Feed	24
INRIX Historical Performance Measure Feed	25
Work Zone Congestion Events	25
ATMS Events Feed	26
Waze Alert Feed	26
LCPT Monthly Sensor Data Feed	27
Stations Feed	27
LCPT Stations Feed	28
API DOCUMENTATION	29
Global Configuration	29
DATA DESCRIPTION AND QUALITY ASSURANCE	39
Joint Sensor Feed	39
Context Data	39
Detector Data	39
Quality Assurance	41
Work Zone Alert Feed	42
Quality Assurance	42
Assigning Slow or Stopped Conditions	43
DMS Inventory History Feed	44
Quality Assurance	45
INRIX Daily Performance Measures	45
Quality Assurance	48
INRIX Monthly Performance Measures	49
INRIX Historical Performance Measures	49
Work Zone Congestion Events	49
Quality Assurance	50
ATMS Event Feed	51
Quality Assurance	54
Waze Alert Feed	54
Quality Assurance	55
LCPT Monthly Sensor Data	55

Quality Assurance.....	56
Stations Feed.....	57
Quality Assurance.....	58
LCPT Stations.....	58
Quality Assurance.....	59

LIST OF FIGURES

Figure 1. Overview of open data service	1
Figure 2. High-level architecture of open data service	3
Figure 3. Console application for sensors	8
Figure 4. Editing sensors in console application	9
Figure 5. Data smoothing for work zone alert	43
Figure 6. Historical data analysis and decision boundaries for work zone alert.....	44
Figure 7. Bottleneck definition	48

LIST OF TABLES

Table 1. Response codes	29
Table 2. Joint sensor feed.....	30
Table 3. Work zone alert feed.....	31
Table 4. DMS inventory history feed	31
Table 5. INRIX daily performance measures	32
Table 6. INRIX monthly performance measures	33
Table 7. INRIX historical performance measures	34
Table 8. Work zone congestion events	35
Table 9. ATMS event feed.....	36
Table 10. Waze alert feed	36
Table 11. LCPT monthly sensor data	37
Table 12. Stations feed.....	38
Table 13. LCPT stations	38

ACKNOWLEDGMENTS

This project was sponsored by the Iowa Statewide Transportation Innovation Council and Iowa Department of Transportation using Federal Highway Administration surface transportation program funding.

GLOSSARY

AADT: Average annual daily traffic

ATMS: Advanced Traffic Management System

DMS: Dynamic Message Sign

DOT: Department of Transportation

HCM: Highway Capacity Manual

LCPT: Lane Closure Planning Tool (A tool developed by the Iowa DOT to determine when a lane can be closed based on volume data)

LRS: Linear Referencing System

RAMS: Roadway Asset Management System (The Iowa DOT's management system to maintain assets on the LRS)

TCP: Traffic Critical Project program (A program by the Iowa DOT to monitor work zones that will impact traffic)

TMC: Traffic Management Center

DOCUMENT REVISION HISTORY

Name	Date	Description of Changes	Version
Vamsi Krishna Jagarlamudi	9-6-2017	Initial document	1.0
Vamsi Krishna Jagarlamudi	2-2-2018	Updated document with the LCPT, stations, Waze alert feeds, and snowplow archival	1.1
Skylar Knickerbocker	2-10-2018	Added user interface, updated functional requirements, added API documentation, and added description and quality assurance	1.2

INTRODUCTION

Purpose

The purpose of this document is to give a detailed description of the requirements for the open data service maintained by the Real-time Analytics of Transportation Data (REACTOR) Lab at the Institute for Transportation (InTrans) on behalf of the Iowa Department of Transportation (DOT). It will illustrate the purpose and complete declaration for the development of the system. It will also explain system constraints, interface, and interactions with other external applications.

The open data service makes transportation related data accessible to both internal DOT and external third-party users. A combination of high-performance computing (HPC) on premise and on the cloud is used to create data feeds that provide clean data streams, alerts for traffic congestion using machine learning, performance reporting, and data analytics. Figure 1 shows a high-level view of how the open data service ingests data (on the left) and provides these as openly available feeds and services for a variety of use cases (on the right).

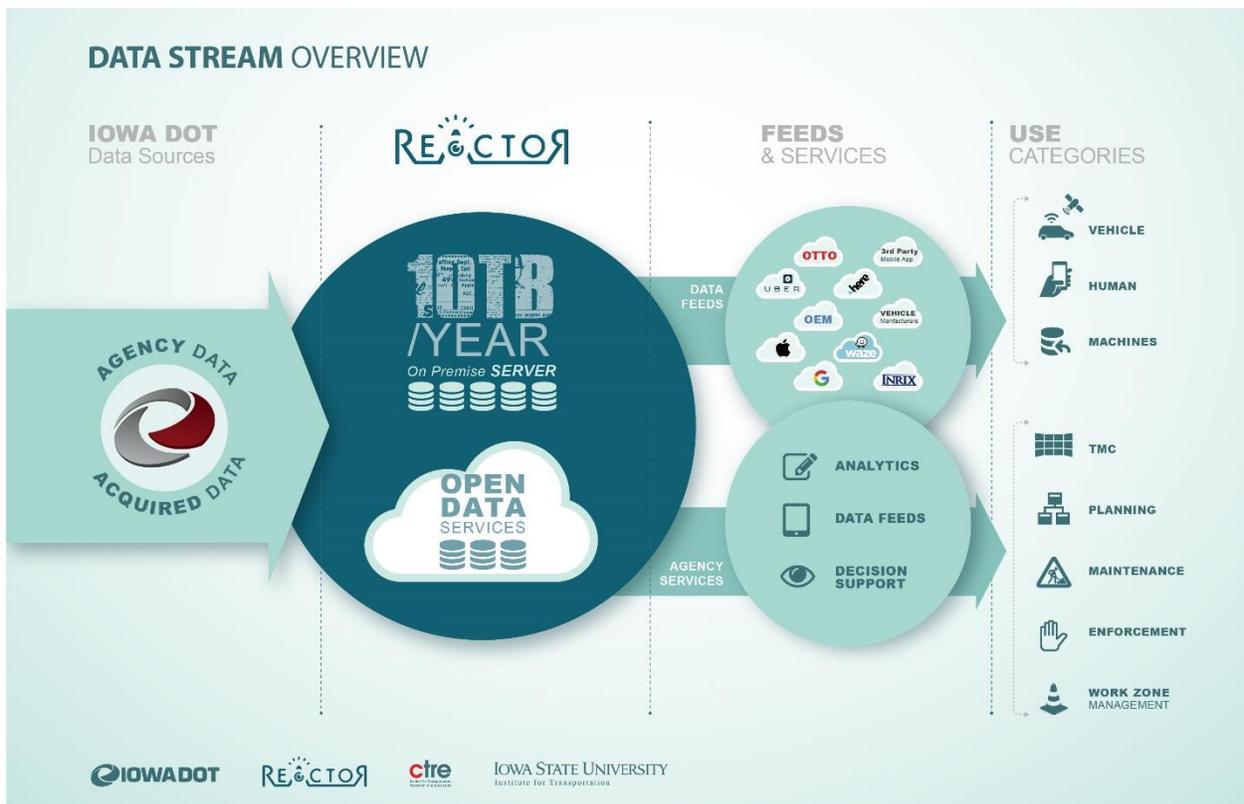


Figure 1. Overview of open data service

Document Conventions

This document features some terminology which readers may be unfamiliar with. See Appendix A (Glossary) for a list of terms and definitions.

Project Scope

The open data service is composed of three main components: data archival, data feeds and the text alert module. The data archival module supports the storing of transportation related data from various service providers. The archived data is used for research and development of other projects/feeds. The data feeds module provides a value added feeds to the consumers. The text alert module provides a mechanism to notify the users with the text alerts when triggered (for example by work zone congestion).

References

- <http://reactorfeeds.org>
- http://205.221.97.102/Iowa.Sims.AllSites.C2C/IADOT_SIMS_AllSites_C2C.asmx
- <https://www.twilio.com/sms/api>
- <https://azure.microsoft.com/en-us/services/virtual-machines/>

OVERALL DESCRIPTION

Product Perspective

The open data service is expected to provide a variety of data feeds for both DOT- and non-DOT users. While this application acts as a producer that generates the data feeds, there is also a consumer part that gets the data from other providers, and converts the data into a structured format, and then archives that data for further research and application. Apart from the data feeds, there is also another component that detects the congestion bottlenecks in the work zone constructions, and immediately notifies the traffic engineers with text alerts. The scope of this project combines data generation, data archival and the work zone text alerts. All these components are covered in detail in this document. Figure 2 illustrates the high-level architecture of the application.

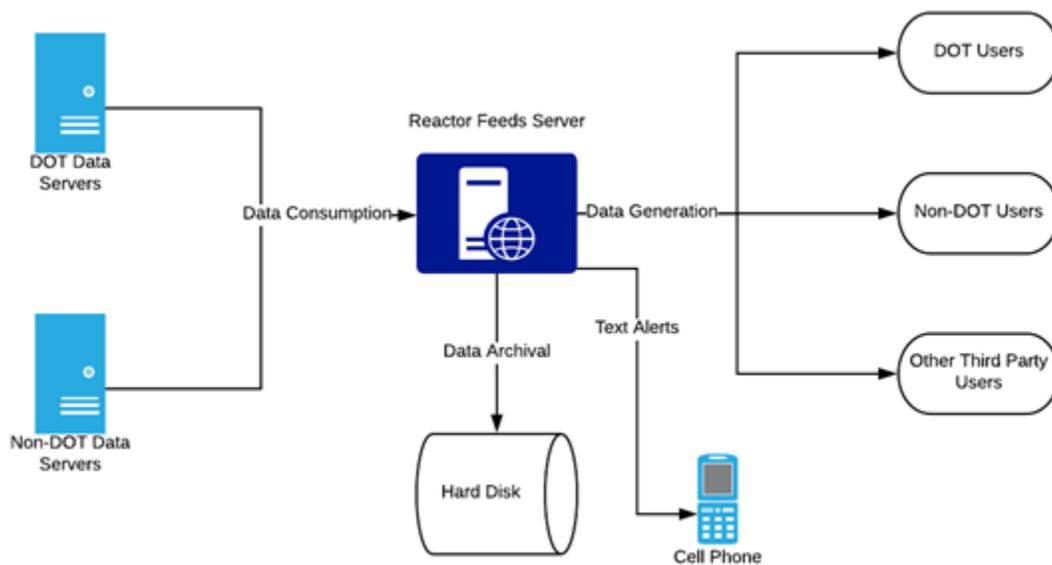


Figure 2. High-level architecture of open data service

Product Functions

The following list outlines the main features and functionalities of the open data service. The features are split into three major categories: data archival, work zone congestion alerts, and the data feeds.

Data Archival

The data archiving serves multiple purposes in the open data service and is typically the first process completed for any data feeds. The first purpose of the data archiving is the ability to provide real-time data feeds that have gone through a quality assurance process. Some data archived on the open data service are purely for historical purposes and the ability to query

historical data. Data archiving also ensures that data is available for research or the development of other applications/data feeds if needed. The following list outlines the data that is being archived on the open data service:

- Traffic detector inventory data: downloads and archives the detector inventory data (location of detectors) once per day.
- Traffic detector data: downloads and archives the traffic data for every 20 seconds.
- INRIX data: archives the INRIX data every minute.
- Waze data: archives the Waze data every five minutes.
- Dynamic Message Sign (DMS) inventory: archives the DMS inventory (location of DMS) data once per day.
- Work zone congestion events: archives the data whenever a work zone congestion event is created.
- INRIX daily performance measures: archives the daily performance measures for each INRIX XD segment data once per day. This information is processed on the HPC in the REACTOR Lab.
- DMS monthly performance measures: archives the monthly performance measures for each INRIX XD segment data once per month. This information is processed on the HPC in the REACTOR Lab.
- DMS historical performance measures: archives the INRIX historical data generated in 2015-2016 years. This should be a one-time process and will no longer be updated.
- Advanced traffic management system (ATMS) event data: archives the ATMS data generated by the Iowa DOT Traffic Management Center (TMC) once per day.
- Waze cluster data: archives the Waze cluster data whenever the Waze Alert System produces an alert.
- Lane closure planning tool (LCPT) monthly performance data: archives the LCPT performance metric each month.
- Stations data: downloads and merges the detector stations data every day. The data gets deleted at the end of the month.

- LCPT stations data: downloads and appends the stations each month.
- Snowplow automatic vehicle location (AVL): archives the snowplow AVL every minute.
- Road weather information system (RWIS) traffic data: archives the RWIS traffic data every five minutes. (in progress)

Work Zone Congestion Detection

These functions are used in the quality assurance and alerting process of the work zone congestion events. This process ensures that only events that meet a defined criteria are sent using the text alerting. The following list outlines the major functions of the work zone congestion detection on the open data service:

- Provides representational state transfer (REST) service to receive the status of work zone sensors.
- Traffic data analysis for the work zone sensors and creates congestion events.
- Sends text alerts for new or cleared congestion events.

Data Feeds

The data feeds use the data that is archived or generated from the previous categories. These feeds are the final product of the open data service, which can be used by DOT- or non-DOT users to access transportation-related data from the Iowa DOT. The following list outlines the data feeds which are being generated through the open data service:

- Joint sensor data: joint aggregate data for detector location and detector traffic data that includes integration of RWIS traffic data
- Work zone alert feed: real-time work zone congestion bottlenecks
- DMS inventory history
- INRIX daily performance measures
- INRIX monthly performance measures
- INRIX historical performance measures

- All work zone congestion events
- ATMS events
- Waze alert feed
- LCPT monthly performance feed
- Stations feed
- LCPT stations feed

User Class and Characteristics

The research team can classify open data service users based on the data consumed from the application. There are three different types of users that interact with the system: data archival consumers, data feed consumers, and work zone personnel (resident construction engineer, district construction staff, engineers, traffic operations staff, DOT consultants, and work zone research staff).

Data archival users have access to a significant amount of information which can be used to analyze trends and to identify historical performance measures (data analysis).

Data feed users receive the data from the open data service for integration into their systems. To receive the data, users must make hypertext transfer protocol (HTTP) REST calls to the application. This will require some familiarity with basic HTTP REST implementation. An example user is the Iowa DOT TMC that continuously receives the work zone alert feeds and displays bottlenecks in its operations dashboard.

Work zone personnel users have little or no interaction with the system directly. Instead, they receive text alerts when specific conditions are met to trigger a work zone congestion alert.

Operating Environment

The software is primarily developed using the Java programming language. Since the application archives a significant amount of data daily, it is recommended that a good amount of disk space be available to run the application. Most of these application are not process intensive but run multiple scheduled jobs. It is expected that at least 1 GB of random-access memory (RAM) will be allocated for the application.

The application is hosted in Microsoft Azure Virtual Machine (VM) and deployed inside a Tomcat Servlet Container. The VM is operating with the Red Hat Enterprise Linux Operating

(RHEL) 6 operating system (OS) with 2 GB of RAM and 2 TB of the storage disk. The machine is assigned with a public domain ("reactorfeeds.org"). The application uses a MongoDB database server.

Design and Implementation Constraints

The open data service archive will grow over time with the main design constraint being data storage. The dependency, and accessibility, of third-party servers are an additional application constraint as is the capacity of the database servers and speed of the processors.

User Documentation

Online documentation can be found at <http://reactorfeeds.org>.

Assumptions and Dependencies

This application depends heavily on third-party servers (to get the data). This assumes access for all the external components with a list of external dependencies summarized as follows:

- Wavetronix Data application programming interface (API) (maintained by Iowa DOT/Iowa DOT contractor)
- INRIX API (maintained by INRIX contracted through Iowa DOT)
- Waze data API (maintained by Waze)
- DMS inventory API (maintained by Iowa DOT/Iowa DOT contractor)
- File transfer protocol (FTP) accessibility for ATMS data. (maintained by Iowa DOT/Iowa DOT contractor)
- External application for identifying the work zone traffic status. (maintained by REACTOR Lab)
- Twilio (Short Message Service) SMS Gateway (maintained by REACTOR Lab)
- External applications for identifying the Waze clusters
- Traffic Inventory Data API for preparing the stations
- SkyHawk database accessibility for archiving the snowplow AVL

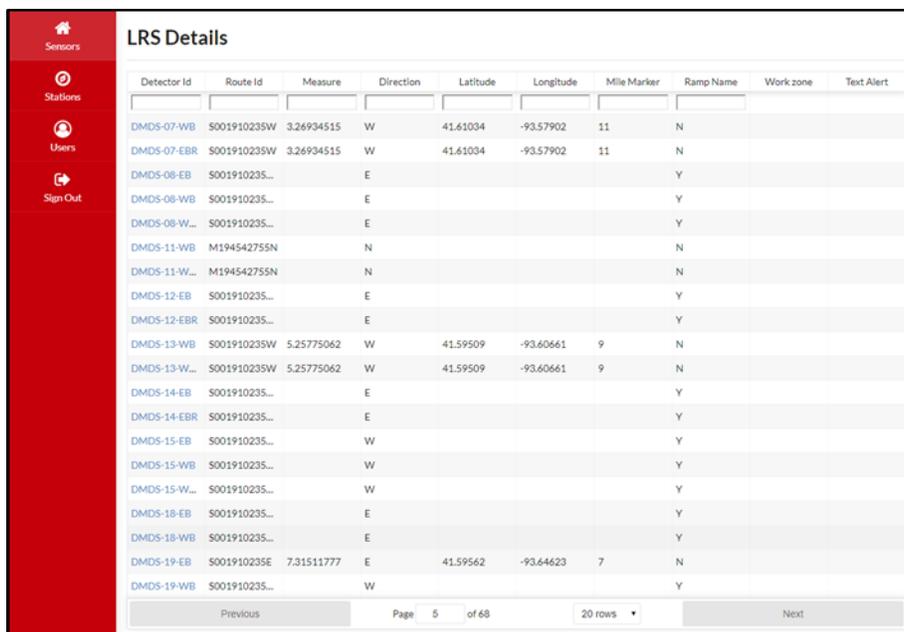
EXTERNAL INTERFACE REQUIREMENTS

User Interfaces

Open data service has no user interface as of now. There is a simple web page that provides the documentation for the feeds and a console application has been developed to maintain the sensor information used in various data feeds.

Console Application

The console application is maintained by the REACTOR Lab at InTrans and is used to maintain and correct issues related to the Iowa DOT's traffic sensors. The console application, shown in Figure 3, maintains a list of all traffic sensor detectors including sensors that have been removed from the data feed. This is to ensure a record of sensor location for future reference. The console application displays the results of the linear referencing system (LRS) conflation processes to the Iowa DOT's LRS roadway asset management system (RAMS). The information in the console application is used when producing any data related to the Iowa DOT's traffic sensors.



The screenshot shows a web application interface titled "LRS Details". On the left is a red sidebar with navigation options: Sensors, Stations, Users, and Sign Out. The main content area displays a table with the following columns: Detector Id, Route Id, Measure, Direction, Latitude, Longitude, Mile Marker, Ramp Name, Work zone, and Text Alert. The table contains 20 rows of data, with the first row being: DMDS-07-WB, S001910235W, 3.26934515, W, 41.61034, -93.57902, 11, N, and an empty Text Alert field. The table is paginated, showing "Page 5 of 68" and "20 rows" per page.

Detector Id	Route Id	Measure	Direction	Latitude	Longitude	Mile Marker	Ramp Name	Work zone	Text Alert
DMDS-07-WB	S001910235W	3.26934515	W	41.61034	-93.57902	11	N		
DMDS-07-EBR	S001910235W	3.26934515	W	41.61034	-93.57902	11	N		
DMDS-08-EB	S001910235...		E				Y		
DMDS-08-WB	S001910235...		E				Y		
DMDS-08-W...	S001910235...		E				Y		
DMDS-11-WB	M194542755N		N				N		
DMDS-11-W...	M194542755N		N				N		
DMDS-12-EB	S001910235...		E				Y		
DMDS-12-EBR	S001910235...		E				Y		
DMDS-13-WB	S001910235W	5.25775062	W	41.59509	-93.60661	9	N		
DMDS-13-W...	S001910235W	5.25775062	W	41.59509	-93.60661	9	N		
DMDS-14-EB	S001910235...		E				Y		
DMDS-14-EBR	S001910235...		E				Y		
DMDS-15-EB	S001910235...		W				Y		
DMDS-15-WB	S001910235...		W				Y		
DMDS-15-W...	S001910235...		W				Y		
DMDS-18-EB	S001910235...		E				Y		
DMDS-18-WB	S001910235...		E				Y		
DMDS-19-EB	S001910235E	7.31511777	E	41.59562	-93.64623	7	N		
DMDS-19-WB	S001910235...		W				Y		

Figure 3. Console application for sensors

Information is provided specific to whether a sensor is within a work zone or if designated as part of a text alert system. The information is used in a variety of other data feeds, including the LCPT and work zone alerting so that only these sensors are used in the text alerting system.

The sensor inventory is checked once a day and any updates will be added to the console application. Primary changes include when there are new sensors added to the data feed or if a

current sensor has updated location coordinates. Either change generates a text alert which prompts a verification of the LRS information.

All sensors in the console application can be edited by selecting the detector-ID in the first column. Once the detector-ID is selected, the editing function is displayed as shown in Figure 4. Any errors in the conflation process can be corrected in an editing session. These corrections will then be maintained for future use. The editing function also provides the ability to identify if a sensor is located in a work zone or is part of the text alerting system.

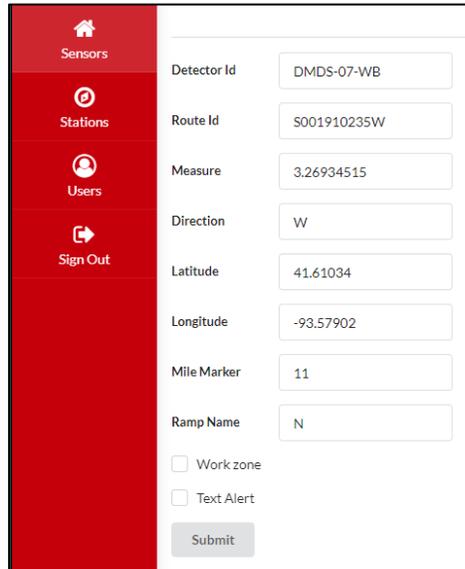


Figure 4. Editing sensors in console application

Hardware Interfaces

The application doesn't require any designated hardware, so it doesn't have any hardware interfaces. All feed information is sent using the network with transmission control protocol/internet protocol (TCP/IP) and HTTP protocol.

Software Interfaces

1. Database Interface

- The Spring Data MongoDB APIs are used to interact with the database servers.
- The Spring Data Java Persistence API (JPA) implementation is used to access the SkyHawk MySQL database.

2. Incoming and outgoing data

- Incoming data consists of all data archived from the third-party servers (DOT servers, non-DOT servers, REACTOR Lab servers).
- All the incoming data is stored either in the file system or the database.
- Outgoing data consists of the value-added feeds generated by REACTOR staff.
- The Spring model–view–controller (MVC) framework is used to consume and produce the data over the HTTP.

3. Multimedia messages

- The SMS gateway providers are used to send multimedia messages.

Communications Interfaces

The incoming and outgoing data transfers happen over the network interface using standard TCP/IP and HTTP protocol. The communication between the application and the SMS gateway service provider occurs via the HTTP, however, from the SMS gateway to the end user, the gateway uses the standard SMS communication protocols.

FUNCTIONAL REQUIREMENTS

The features are split into three major categories: data archival, work zone congestion alerts, and the data feeds. Each section includes the requirements that specify all the fundamental actions of the software system.

Data Archival

Traffic Detector Inventory Data

- ID: FR1
- TITLE: Download traffic detector inventory data
- DESC: Application should be able to access the Wavetronix Data API and download the inventory data using the HTTP GET or POST method. The system should convert the downloaded extensible markup language (XML) data into comma-separated values (CSV) format and archive that CSV file into the file system.
- FREQUENCY: Once per day.
- PRIORITY: High
- RAT: In order for the system to archive the inventory data.
- AUTHORITATIVE SOURCE: Iowa DOT Office of Traffic Operations
- DEPENDENCY: Outside source (TransCore – DOT Contractor)

Detector Traffic Data

- ID: FR2
- TITLE: Download traffic data
- DESC: Application should be able to access the Wavetronix Data API and download the detector data using the HTTP GET or POST method. The system should convert the downloaded XML data into CSV format and archive that CSV file into the file system.
- FREQUENCY: The download should happen for every 20 seconds.

- PRIORITY: High
- RAT: In order for the system to archive the traffic data.
- AUTHORITATIVE SOURCE: Iowa DOT Office of Traffic Operations
- DEPENDENCY: Outside source (TransCore – DOT Contractor)

INRIX Data

- ID: FR3
- TITLE: Download INRIX XD segment data
- DESC: Application should be able to access the Iowa INRIX Data API and download the INRIX XD segment data using the HTTP GET method. The HTTP call should provide proper authentication token. The system should be able to get a new token whenever an old token expires. System should convert the downloaded XML data into CSV format.
- FREQUENCY: The download should happen for every one minute. The system should consolidate all CSV files for one day into a single file and archive that into the file system.
- PRIORITY: High
- RAT: In order for the system to archive the INRIX data.
- AUTHORITATIVE SOURCE: INRIX (contracted through Iowa DOT)
- DEPENDENCY: Outside source (INRIX)

Waze Data

- ID: FR4
- TITLE: Download Waze data
- DESC: Application should be able to access the Waze Data API and download the Waze data using the HTTP GET method. The HTTP call should provide proper parameters and authentication.

- **FREQUENCY:** The frequency is every five minutes. The downloaded data should be archived into the file system.
- **PRIORITY:** High
- **RAT:** In order for the system to archive the Waze data.
- **AUTHORITATIVE SOURCE:** Waze
- **DEPENDENCY:** Outside source (Waze)

DMS Inventory Data

- **ID:** FR5
- **TITLE:** Download DMS inventory data
- **DESC:** Application should be able to access the TransCore Data API and download the DMS Inventory data using the HTTP GET method. The downloaded data should be converted into JavaScript Object Notation (JSON) format and saved into the MongoDB database.
- **FREQUENCY:** This process should happen once per day.
- **PRIORITY:** Medium
- **RAT:** In order for the system to archive the DMS inventory data.
- **AUTHORITATIVE SOURCE:** Iowa DOT Office of Traffic Operations
- **DEPENDENCY:** Outside source (TransCore – DOT Contractor)

Work Zone Congestion Events

- **ID:** FR6
- **TITLE:** Provide an API to consume the work zone congestion status
- **DESC:** The application should provide a secure API such that the other application (traffic detection algorithm) sends the work zone congestion statuses to the server. The work zone congestion status is provided by REACTOR Lab.

- FREQUENCY: The download should happen every minute.
- PRIORITY: High
- RAT: In order for the system to receive the work zone status.
- AUTHORITATIVE SOURCE: REACTOR Lab
- DEPENDENCY: REACTOR Lab

INRIX Daily Performance Measures

- ID: FR7
- TITLE: Provide an API to consume INRIX daily performance measures
- DESC: The application should provide an HTTP POST API such that everyday performance measures will be uploaded to the server. Once the file is uploaded, the application should pre-process the file and generate the JSON data and store into the MongoDB database. The data uploaded will be provided by REACTOR Lab.
- FREQUENCY: Data should be posted by REACTOR Lab once a day.
- PRIORITY: High
- RAT: In order for the system to archive the INRIX daily performance measures.
- AUTHORITATIVE SOURCE: REACTOR Lab
- DEPENDENCY: REACTOR Lab

INRIX Monthly Performance Measures

- ID: FR8
- TITLE: Provide an API to consume INRIX monthly performance measures
- DESC: The application should provide an HTTP POST API such that monthly performance measures will be uploaded to the server. Once the file is uploaded, the application should

pre-process the file, generate the JSON data, and store this into the MongoDB database. The data uploaded will be provided by REACTOR Lab.

- FREQUENCY: Data should be posted by REACTOR Lab once a month.
- PRIORITY: High
- RAT: In order for the system to archive the INRIX monthly performance measures.
- AUTHORITATIVE SOURCE: REACTOR Lab
- DEPENDENCY: REACTOR Lab

INRIX Historical Performance Measures

- ID: FR9
- TITLE: Provide an API to consume INRIX historical performance measures
- DESC: The application should provide an HTTP POST API such that the old INRIX performance measures will be uploaded into the server and stored into a different collection in MongoDB.
- FREQUENCY: Data will no longer be posted. Only for historical use.
- PRIORITY: High
- RAT: In order for the system to archive the INRIX historical performance measures.
- AUTHORITATIVE SOURCE: REACTOR Lab
- DEPENDENCY: None

ATMS Event Data

- ID: FR10
- TITLE: Provide an API to consume ATMS event data

- DESC: The application should provide an HTTP POST API such that every day ATMS incident events will be uploaded to the server. Once the file is uploaded, the application should pre-process the file and generate the JSON data and store into the MongoDB database.
- FREQUENCY: This process should happen once per day.
- PRIORITY: High
- RAT: In order for the system to archive the ATMS event data.
- AUTHORITATIVE SOURCE: Iowa DOT Office of Traffic Operations
- DEPENDENCY: Outside source (TransCore – DOT Contractor)

Camera Inventory

- ID: FR11
- TITLE: Maintain the latest camera inventory data
- DESC: The system should be able to access the TransCore closed-circuit television (CCTV) information APIs using the HTTP GET method. The application should convert the downloaded XML file into a JSON data and save into MongoDB. At any time, the database should have the latest information, meaning that every time new data is downloaded old entries are removed and the new values are stored.
- FREQUENCY: The CCTV data should get update once per day
- PRIORITY: High
- RAT: In order for the system to maintain the latest CCTV information.
- AUTHORITATIVE SOURCE: Iowa DOT Office of Traffic Operations
- DEPENDENCY: Outside source (TransCore – DOT Contractor)

Waze Cluster Data

- ID: FR12

- TITLE: Provide an API to consume the Waze Clusters
- DESC: The application should provide a secure API such that the external applications (Waze alerting system) sends the Waze clusters and the alert data. The design should follow the producer-consumer pattern. The application should process the received data and convert them into the proper format such that this can be used for later feeds. The transformed JSON data should be stored into the MongoDB.
- FREQUENCY: The cluster data is stored every 5 minutes.
- PRIORITY: High
- RAT: In order for the system to receive the Waze clusters.
- AUTHORITATIVE SOURCE: REACTOR Lab
- DEPENDENCY: REACTOR Lab

LCPT Monthly Sensor Data

- ID: FR13
- TITLE: Provide an API to consume the LCPT monthly performance measures
- DESC: The application should provide an HTTP POST API such that monthly performance measures will be uploaded to the server. Once the file gets uploaded, the application should pre-process the file, generate the JSON data, and store this in the MongoDB database.
- FREQUENCY: The LCPT monthly sensor data will be updated once per month.
- PRIORITY: High
- RAT: In order for the system to receive the LCPT monthly sensor data.
- AUTHORITATIVE SOURCE: REACTOR Lab
- DEPENDENCY: REACTOR Lab

Sensor Stations Data

- ID: FR14
- TITLE: Prepare sensors station data
- DESC: The system should prepare stations data from the traffic inventory data. All sensors should be grouped into the stations based on their station id. The prepared stations data should be compared with the previous day stations data that was stored on the database. If any new sensor is added, that system should add this sensor to the database and also update the average location coordinates. If any sensor is removed, that should be logged on to the system. This process should be scheduled for every day. At the beginning of the new month, all records should be cleared from the database. Before clearing the collection records, all this data should be transferred to the LCPT stations handler.
- FREQUENCY: The station data is downloaded and processed daily then cleared once a month.
- PRIORITY: High
- RAT: In order for the system to maintain latest stations data.
- AUTHORITATIVE SOURCE: REACTOR Lab
- DEPENDENCY: FR1

Sensor Stations Data for LCPT

- ID: FR15
- TITLE: Prepare stations data for LCPT
- DESC: This system will receive the latest stations data at the end of the month from FR14 and process them to remove the sensors located on ramps. The LCPT stations data will be archived into MongoDB each month.
- FREQUENCY: The station data is appended once a month.
- PRIORITY: High
- RAT: In order for the system to maintain archive of LCPT stations data.

- AUTHORITATIVE SOURCE: REACTOR Lab
- DEPENDENCY: FR14

Snowplow AVL Data

- ID: FR16
- TITLE: Download snowplow AVL Data
- DESC: The system will access the SkyHawk database remotely and execute the stored procedure by passing the proper parameters.
- FREQUENCY: The scheduler will call the stored procedure every minute. The application will create a new CSV file at the beginning of the day, transform the result to CSV format, and append the flat file.
- PRIORITY: High
- RAT: In order for the system to archive the snowplow AVL data.
- AUTHORITATIVE SOURCE: Iowa DOT Office of Maintenance
- DEPENDENCY: Outside source (SkyHawk – DOT Contractor)

Work Zone Congestion Detection

Work Zone Metadata

- ID: FR17
- TITLE: Maintain work zone metadata
- DESC: Includes different configurations related to work zones such as: sensor name and nearest camera mapping, work zone and project names, mile markers, and the list of the recipients who will receive text alerts. This list is maintained by REACTOR Lab on behalf of the Iowa DOT Office of Traffic Operations for the Traffic Critical Program (TCP).
- FREQUENCY: Updated as needed

- PRIORITY: High
- RAT: In order for the system to maintain the work zone metadata.
- AUTHORITATIVE SOURCE: REACTOR Lab
- DEPENDENCY: REACTOR Lab

Work Zone Events

- ID: FR18
- TITLE: Process work zone status and create events
- DESC: The system will process the work zone statuses and identify the current scenario for the given work zone and sensor. If there is no event created for the work zone and yet there is a SLOW or STOP condition received, then the system will create a new event for that work zone with this sensor. If there is already an event available for this work zone, then there are three different possibilities: updating the current severity, maintaining in the same severity, or closing the current event for that sensor.
- FREQUENCY: The process runs every minute based on the data from FR6.
- PRIORITY: High
- RAT: In order for the system to generate the work zone events.
- AUTHORITATIVE SOURCE: REACTOR Lab
- DEPENDENCY: FR6

Work Zone Schedulers

- ID: FR19
- TITLE: Work zone schedulers to trigger text alerts
- DESC: Scheduler jobs run every minute to identify the STOP conditions in the work zones. Whenever a new STOP event persists for more than five minutes, the alert scheduler identifies that work zone and triggers the SMS module. Similarly, whenever a STOP event

gets cleared, and everything goes back to normal for more than five minutes, the clearance scheduler triggers the SMS module.

- FREQUENCY: The process runs every minute based on the data from FR12.
- PRIORITY: High
- RAT: In order for the system to trigger the text alerts.
- AUTHORITATIVE SOURCE: REACTOR Lab
- DEPENDENCY: FR18

Text Alert Content Generation

- ID: FR20
- TITLE: Create text alert content from work zone metadata
- DESC: There are two different text formats for the text messages. The “Alert” message includes the latest image from the closest camera, the route ID, work zone project name, congestion mile marks range, and when the event was created, along with the current average speeds. The “Clearance” message includes the latest image from the closest camera, the route ID, work zone project name, and the total event duration.
- FREQUENCY: As needed based on inputs from FR13.
- PRIORITY: High
- RAT: In order for the system to generate text content.
- AUTHORITATIVE SOURCE: REACTOR Lab
- DEPENDENCY: FR11, FR17, FR19

SMS Gateway Interaction

- ID: FR21
- TITLE: Triggering the text alert messages using the SMS gateway

- **DESC:** The system sends the text alerts using the SMS gateway providers. Twilio is currently in use as the service provider; however, this process has now been moved to the DOT. This function will remain as a testing system for new alerts being developed. Given that the scheduler triggers the SMS module, the system will generate the content and interact with the TWILIO SMS gateway using the REST API calls.
- **FREQUENCY:** As needed
- **PRIORITY:** High
- **RAT:** In order for the system to send text messages.
- **AUTHORITATIVE SOURCE:** REACTOR Lab
- **DEPENDENCY:** FR17, FR19, FR20, Twilio

Data Feeds

Joint Sensor Data

- **ID:** FR22
- **TITLE:** Joint Sensor Data Feed
- **DESC:** The system continuously generates the joint data feed by combining the Wavetronix sensor locations and the traffic data. Given that the traffic data are downloaded every 20 seconds, the feed will aggregate the data for the last minute and will calculate the detector average speed and volume metrics. The generated data will be updated every minute.
- **FREQUENCY:** The data is updated every minute.
- **PRIORITY:** High
- **RAT:** In order for the system to generate the aggregate data for detector location and the traffic data.
- **AUTHORITATIVE SOURCE:** REACTOR Lab
- **DEPENDENCY:** FR1, FR2

Work Zone Alert Feed

- ID: FR23
- TITLE: Work zone congestion alert feed
- DESC: The system provides a feed that lists the current congestion events “SLOW” or “STOP” which are happening in work zones. The feed will list the information about the work zone name, event start time, current average speeds, and the mile marker range.
- FREQUENCY: The data is updated every minute.
- PRIORITY: High
- RAT: In order for the system to provide the live congestion bottlenecks on work zones.
- AUTHORITATIVE SOURCE: REACTOR Lab
- DEPENDENCY: FR18, FR20

DMS Inventory History Feed

- ID: FR24
- TITLE: API to search and download the DMS Inventory history.
- DESC: The system provides a REST API that enables the users to download the DMS inventory history. The API will support the time range parameters such that when given the time range, it gets all the archived inventory history on those dates.
- FREQUENCY: The data is updated once a day.
- PRIORITY: Medium
- RAT: In order for the system to provide DMS inventory history.
- AUTHORITATIVE SOURCE: REACTOR Lab
- DEPENDENCY: FR5

INRIX Daily Performance Measure Feed

- ID: FR25
- TITLE: API to search and download INRIX daily performance measures
- DESC: The system provides a REST API that enables the users to download the INRIX daily performance measure data. The API should support search filters, where a user can filter the data by providing the date ranges and/or INRIX code. This service has daily performance measure for INRIX XD segments starting on May 29th, 2017.
- FREQUENCY: The data is updated once a day.
- PRIORITY: High
- RAT: In order for the system to provide INRIX daily performance measure data.
- AUTHORITATIVE SOURCE: REACTOR Lab
- DEPENDENCY: FR7

INRIX Monthly Performance Measure Feed

- ID: FR26
- TITLE: API to search and download INRIX monthly performance measures
- DESC: The system provides a REST API that enables the users to download the INRIX monthly performance measure data. The API will support search filters, where a user can filter the data by providing the month ranges and/or INRIX code. This service has monthly performance measure for INRIX XD segments starting in June 2017.
- FREQUENCY: The data is updated once a month.
- PRIORITY: High
- RAT: In order for the system to provide INRIX monthly performance measure data.
- AUTHORITATIVE SOURCE: REACTOR Lab

- DEPENDENCY: FR8

INRIX Historical Performance Measure Feed

- ID: FR27
- TITLE: API to search and download INRIX historical performance measures
- DESC: The system provides a REST API that enables the users to download the INRIX historical performance measure data for TMC segments. The API will support search filters, where a user can filter the data by providing the month ranges and/or INRIX code. This service has monthly performance measure for INRIX TMC segments starting from January 2013 to May 2017.
- FREQUENCY: The data is no longer being updated and contains historical data from January 2013 to May 2017.
- PRIORITY: High
- RAT: In order for the system to provide INRIX historical performance measure data.
- AUTHORITATIVE SOURCE: REACTOR Lab
- DEPENDENCY: FR9

Work Zone Congestion Events

- ID: FR28
- TITLE: API to download all work zone events
- DESC: The application provides an API that dumps all work zone event data. This API will be secured by authentication.
- FREQUENCY: This returns all work zone congestion events.
- PRIORITY: High
- RAT: In order for the system to provide work zone events data.

- AUTHORITATIVE SOURCE: REACTOR Lab
- DEPENDENCY: FR18

ATMS Events Feed

- ID: FR29
- TITLE: API to download ATMS events feed
- DESC: The application provides an API that gives the ATMS event data. The REST API will support search filters, such that user can request the data for a particular day or with a specific event ID. This API should be secured with authentication.
- FREQUENCY: The data is updated once a day.
- PRIORITY: High
- RAT: In order for the system to provide work zone events data.
- AUTHORITATIVE SOURCE: REACTOR Lab
- DEPENDENCY: FR10

Waze Alert Feed

- ID: FR30
- TITLE: Waze alert feed in clustering format
- DESC: The system provides a REST API that gives the current alert information from the Waze cluster data. The feed must include unique cluster ID for each cluster and also list down the information about the type of incident, linear-reference of the cluster paths, and the start time of the event.
- FREQUENCY: The feed provides the most current Waze clusters.
- PRIORITY: High
- RAT: In order for the system to provide Waze alert feed.

- AUTHORITATIVE SOURCE: REACTOR Lab
- DEPENDENCY: FR12

LCPT Monthly Sensor Data Feed

- ID: FR31
- TITLE: API to search and download LCPT monthly performance measures
- DESC: The system provides a REST API that enables the users to download the LCPT monthly sensor data. The API will support search filters, where a user can filter the data by providing the month and/or sensor name.
- FREQUENCY: The data is updated once a month.
- PRIORITY: High
- RAT: In order for the system to provide LCPT monthly sensor data.
- AUTHORITATIVE SOURCE: REACTOR Lab
- DEPENDENCY: FR13

Stations Feed

- ID: FR32
- TITLE: API to download latest stations data
- DESC: The application supports an API that gives the latest stations data that are in the data feed for the current month. Primarily used internally to provide sensor data for each month.
- FREQUENCY: The data feeds is updated daily then cleared at the end of the month.
- PRIORITY: High
- RAT: In order for the system to provide latest stations feed.
- AUTHORITATIVE SOURCE: REACTOR Lab

- DEPENDENCY: FR14

LCPT Stations Feed

- ID: FR33
- TITLE: API to search and download LCPT stations data
- DESC: The system provides a REST API that enables the user to download the LCPT stations data. The API will support search filters, where a user can filter the data by providing the month.
- FREQUENCY: The data is updated once a month.
- PRIORITY: High
- RAT: In order for the system to provide LCPT monthly station data.
- AUTHORITATIVE SOURCE: REACTOR Lab
- DEPENDENCY: FR15

API DOCUMENTATION

Global Configuration

- Base URL: <http://reactorfeeds.org>
- All the GET operations returns the data in either text/xml or application/json format.
- To get the data in text/xml format ->
 1. Append “.xml” in the URL (or)
 2. Use “text/xml” as response content type
- To get the data in application/json format ->
 - Append “.json” in the URL (or)
 - Use “application/json” as response content type

Note: All examples provided in this document are in JSON format.

Table 1. Response codes

Code	Description
	OK
200	Successfully retrieves the requested data.
	UNAUTHORIZED
401	Requested user is not authenticated, either by not providing the password or providing the wrong password.
	FORBIDDEN
403	The user doesn't have enough privileges to retrieve this data.
	NOT FOUND
404	The requested resource is not found on the server.
	INTERNAL SERVER ERROR
500	Server failed to process the request. If you receive this error, please report to the technical support.

Table 2. Joint sensor feed

Joint Sensor Feed Information	
URL	http://reactorfeeds.org/feeds/sensorfeed
Request Method	GET
Authentication	None
Query Parameters	None
Response Type	text/xml (or) application/json
Sample Request:	Sample Response Structure:
http://reactorfeeds.org/feeds/sensorfeed.json	<pre> { "detectorCollection": [{ "detectionTime": { "localDate": "string", "localTime": "string", "utcOffset": "string" }, "detectorList": [{ "approachDirection": "string", "detectionReports": [{ "count": 0, "lanes": [{ "count": 0, "laneId": 0, "largeCount": 0, "mediumCount": 0, "occupancy": 0, "smallCount": 0, "speed": 0 }], "largeCount": 0, "mediumCount": 0, "occupancy": 0, "smallCount": 0, "speed": 0 }], "detectorId": "string", "detectorLocation": { "datum": "string", "latitude": "string", "longitude": "string" }, "detectorType": "string", "lanesType": "string", "linearReference": { "approachDirection": "string", "latitude": "string", "longitude": "string", "measure": "string", "mileMarker": "string", "rampName": "string", "routeId": "string" }, "linkOwnership": "string", "routeDesignator": "string", "status": "string", "workzone": true }], "endTime": "string", "startTime": "string" }], "networkId": "string", "organizationId": "string" } </pre>

Table 3. Work zone alert feed

Work Zone Alert Feed Information	
URL	http://reactorfeeds.org/feeds/alertfeed
Request Method	GET
Authentication	None
Query Parameters	None
Response Type	text/xml (or) application/json
Sample Request:	Sample Response Structure:
http://reactorfeeds.org/feeds/alertfeed.json	<pre>{ "alerts": [{ "avgSpeed": 0, "direction": "string", "endTime": "2018-02-01T22:54:20.992Z", "id": 0, "measure": "string", "milepost": "string", "projectName": "string", "route": "string", "routeId": "string", "severity": "BLANK", "startTime": "2018-02-01T22:54:20.992Z", "workzone": "string" }] }</pre>

Table 4. DMS inventory history feed

DMS Inventory History Feed Information	
URL	http://reactorfeeds.org/feeds/dmsinventory
Request Method	GET
Authentication	None
Query Parameters	start: Start date in “MM-DD-YYYY” format. end: End date in “MM-DD-YYYY” format.
Response Type	text/xml (or) application/json
Sample Request:	Sample Response Structure:
http://reactorfeeds.org/feeds/dmsinventory.json?start=08-09-2017&end=08-12-2017	<pre>{ "dmsInventoriesArchive": [{ "archiveTime": 0, "dmsInventoryList": [{ "beaconType": "string", "charHeight": "string", "charWidth": "string", "deviceId": 0, "deviceName": "string", "lastUpdateTime": "string", "latitude": "string", "linearReference": "string", "linkDirection": "string", "longitude": "string", "networkId": "string", "organizationId": "string", "organizationName": "string", "routeDesignator": "string", "signHeight": "string", "signTechnology": "string", "signType": "string", "signWidth": "string" }] }] }</pre>

Table 5. INRIX daily performance measures

INRIX Daily Performance Measures Information	
URL	http://reactorfeeds.org/feeds/inrix/daily
Request Method	GET
Authentication	None
Query Parameters	<p>start: Gives results from this start date. The date should be provided in “MM-DD-YYYY” format.</p> <p>end: Gives results until this end date. The date should be provided in “MM-DD-YYYY” format.</p> <p>code: Gives results for the given particular code.</p>
Response Type	text/xml (or) application/json
Sample Request:	Sample Response Structure:
http://reactorfeeds.org/feeds/inrix/daily.json?code=1450418477&start=08-15-2017	<pre>[{ "bottleNecks": { "avgDuration": 0, "avgMaxLength": 0, "bottleNeckCount": 0 }, "code": "string", "dataQuality": { "cValue": 0, "count10": 0, "count20": 0, "count30": 0, "totalCount": 0 }, "date": "2018-02-01T22:11:27.577Z", "events": { "count": 0 }, "thirtyMeasures": { "avgDelay": 0, "avgSpeed": 0, "refDelay": 0, "refSpeed": 0, "speed": 0, "travelTime": 0 }, "twentyMeasures": { "avgDelay": 0, "avgSpeed": 0, "refDelay": 0, "refSpeed": 0, "speed": 0, "travelTime": 0 } }]</pre>

Table 6. INRIX monthly performance measures

INRIX Monthly Performance Measures Information	
URL	http://reactorfeeds.org/feeds/inrix/monthly
Request Method	GET
Authentication	None
Query Parameters	<p>start: Gives results from this start date. The date should be provided in “MM-YYYY” format.</p> <p>end: Gives results until this end date. The date should be provided in “MM-YYYY” format.</p> <p>code: Gives results for the given particular code.</p>
Response Type	text/xml (or) application/json
Sample Request:	Sample Response Structure:
http://reactorfeeds.org/feeds/inrix/monthly.json?code=1450418477&start=08-2017	<pre>[{ "bottleNecks": { "avgDuration": 0, "avgMaxLength": 0, "bottleNeckCount": 0 }, "code": "string", "dataQuality": { "cValue": 0, "count10": 0, "count20": 0, "count30": 0, "totalCount": 0 }, "date": "2018-02-01T22:11:27.577Z", "events": { "count": 0 }, "thirtyMeasures": { "avgDelay": 0, "avgSpeed": 0, "refDelay": 0, "refSpeed": 0, "speed": 0, "travelTime": 0 }, "twentyMeasures": { "avgDelay": 0, "avgSpeed": 0, "refDelay": 0, "refSpeed": 0, "speed": 0, "travelTime": 0 } }]</pre>

Table 7. INRIX historical performance measures

INRIX Historical Performance Measures Information	
URL	http://reactorfeeds.org/feeds/inrix/historical
Request Method	GET
Authentication	None
Query Parameters	<p>start: Gives results from this start date. The date should be provided in “MM-YYYY” format.</p> <p>end: Gives results until this end date. The date should be provided in “MM-YYYY” format.</p> <p>code: Gives results for the given particular code.</p>
Response Type	text/xml (or) application/json
Sample Request:	Sample Response Structure:
http://reactorfeeds.org/feeds/inrix/historical.json?code=118-04493&start=04-2017	<pre>[{ "bottleNecks": { "avgDuration": 0, "avgMaxLength": 0, "bottleNeckCount": 0 }, "code": "string", "dataQuality": { "cValue": 0, "count10": 0, "count20": 0, "count30": 0, "totalCount": 0 }, "date": "2018-02-01T22:11:27.577Z", "events": { "count": 0 }, "thirtyMeasures": { "avgDelay": 0, "avgSpeed": 0, "refDelay": 0, "refSpeed": 0, "speed": 0, "travelTime": 0 }, "twentyMeasures": { "avgDelay": 0, "avgSpeed": 0, "refDelay": 0, "refSpeed": 0, "speed": 0, "travelTime": 0 } }]</pre>

Table 8. Work zone congestion events

Work Zone Congestion Events Information	
URL	http://reactorfeeds.org/feeds/wzevents
Request Method	GET
Authentication	HTTP Basic Authentication
Query Parameters	None
Response Type	text/xml (or) application/json
Sample Request:	Sample Response Structure:
To get all the events to date	[
http://reactorfeeds.org/feeds/wzevents	{
	"id": "string",
	"workzone": "string",
	"eventId": 0,
	"sensorEvents": {
	"name": "string",
	"alert": 0,
	"avgSped": 0
	"completed": 0
	"startTime": 0
	"endTime": 0
	"lastUpdateTime": 0
	},
	"completed": "string",
	"alertSmsSent": "string",
	"clearanceSmsSent": 0,
	"longEventSmsSent": "string",
	"startTime": "string",
	"endTime": 0,
	"lastUpdateTime": 0,
	}
]

Table 9. ATMS event feed

ATMS Event Feed Information	
URL	http://reactorfeeds.org/feeds/secure/atms
Request Method	GET
Authentication	HTTP Basic Authentication
Query Parameters	When accessed without any search filters, it provides the list of all events that happened last day.
	<u>Query Parameters:</u>
	date: If provided, returns only the events that are completed on the given date. The date should be provided in “MM-DD-YYYY” format.
	eventId: If provided, returns only the event that is unique to the given event ID.
Response Type	text/xml (or) application/json
Sample Request:	Sample Response Structure:
To get all the events that are completed on 09/05/2017	[
http://reactorfeeds.org/feeds/secure/atms.json?date=09-05-2017	{
	"archiveDate": "1513807271421",
	"eventData": {
	"EventType": "STALLED VEHICLE",
	"Latitude": "41.592148",
	"Event_ID": "157951",

	}
	}]

Table 10. Waze alert feed

Waze Alert Feed Information	
URL	http://reactorfeeds.org/feeds/alerts/waze
Request Method	GET
Authentication	None
Query Parameters	None
Response Type	text/xml (or) application/json
Sample Request:	Sample Response Structure:
http://reactorfeeds.org/feeds/alerts/waze.json	[
n	{
	"clusterId": "string",
	"linear-reference": [
	{
	"end": 0,
	"geoRSSType": "POINT",
	"routeId": "string",
	"start": 0
	}
],
	"startTime": "2018-02-01T23:06:45.480Z",
	"type": "string"
	}]

Table 11. LCPT monthly sensor data

LCPT Monthly Sensor Data Information	
URL	http://reactorfeeds.org/feeds/lcpt/monthly
Request Method	GET
Authentication	None
Query Parameters	When accessed without any query parameters, it provides the LCPT data for the previous month. name: To get the data for a particular sensor. month: If provided, returns only the LCPT data corresponding to that month. The month value should be provided in "MM-YYYY" format. The default value is the previous month. day: To get the data for a particular day in the month. Value should be the first three letters of the day, like ("Mon", "Tue", etc.)
Response Type	text/xml (or) application/json
Sample Request:	Sample Response Structure:
<ol style="list-style-type: none"> 1. Get all sensors LCPT data on 07-2017 http://reactorfeeds.org/feeds/lcpt/monthly?month=07-2017 2. Get LCPT data on 07-2017 for sensor "I-235 EB West of University" http://reactorfeeds.org/feeds/lcpt/monthly?month=07-2017&name=I-235%20EB%20West%20of%20University <i>Note:</i> In the HTTP query parameters, spaces should be replaced with %20 symbol. 3. Get LCPT data on 07-2017 for sensor "I-235 EB West of University" on Wednesday http://reactorfeeds.org/feeds/lcpt/monthly?month=07-2017&name=I-235%20EB%20West%20of%20University&day=Wed 	<pre>[{ "archiveDate": "2018-02-01T22:26:22.998Z", "day": "string", "lcptMeasures": [{ "hour": 0, "lanes": 0, "pce25": 0, "pce75": 0, "pceAvg": 0, "pceMax": 0, "pceMedian": 0, "pceMin": 0, "truckPercent": 0, "volume": 0 }], "month": 0, "detectorId": "string", "year": 0 }]</pre>

Table 12. Stations feed

Stations Feed Information	
URL	http://reactorfeeds.org/feeds/stations
Request Method	GET
Authentication	None
Query Parameters	None
Response Type	text/xml (or) application/json
Sample Request:	Sample Response Structure:
http://reactorfeeds.org/feeds/stations.json	<pre>[{ "aad": "string", "addtCountYear": "string", "avgLatitude": 0, "avgLongitude": 0, "detectorId": "string", "detectorName": "string", "direction": "string", "district": "string", "lastUpdate": "2018-02-01T22:52:09.858Z", "ownership": "string", "route": "string", "stationName": "string", "type": "string" }]</pre>

Table 13. LCPT stations

LCPT Stations Information	
URL	http://reactorfeeds.org/feeds/lcpt/stations
Request Method	GET
Authentication	None
Query Parameters	month: If provided, returns only the LCPT stations data corresponding to that month. The month value should be provided in "MM-YYYY" format.
Response Type	text/xml (or) application/json
Sample Request:	Sample Response Structure:
http://reactorfeeds.org/feeds/lcpt/stations.js on	<pre>[{ "month": "2018-02-01T22:43:17.267Z", "stations": [{ "aad": "string", "addtCountYear": "string", "avgLatitude": 0, "avgLongitude": 0, "detectorId": "string", "detectorName": "string", "direction": "string", "district": "string", "lastUpdate": "2018-02-01T22:43:17.267Z", "ownership": "string", "route": "string", "stationName": "string", "type": "string" }] }]</pre>

DATA DESCRIPTION AND QUALITY ASSURANCE

Joint Sensor Feed

The following section provides a description of the nodes/fields within the sensor data feed. The section is divided into context data and detector data. The context data is relevant for all of the sensors, which provide organization information and the corresponding time stamps of the data. The detector data section provides unique information and current data for each detector. The unique information describes the location of the sensor, including coordinates and the direction of travel. The current data includes speed, occupancy, count, and classification for each lane, as well as the individual lane-level data.

Context Data

The first field provides the organization and time stamp for the detector data:

- **organization-id:** the organization collecting the data (will always be IADOT-Statewide ITS Management System [SIMS])
- **network-id:** The network within the organization collecting the data (will always be IADOT-SIMS)
- **detection-time-stamp** (XML:parent/JSON:object): This is the parent/array for the following fields containing the data and timestamp of the data:
 - **local-date:** Date for the data requested in the format YYYYMMDD
 - **local-time:** Timestamp for the data requested in the format HHMMSS
 - **local-time:** Provides the offset from Coordinated Universal Time (UTC) that is used for timestamps
- **start-time:** The start time of data collected in the format HHMMSS
- **end-time:** The end time of data collected in format HHMMSS

Detector Data

The remaining fields are collected for each detector across the state and located within the parent-node(XML)/array(JSON) called detector-list:

- **detector-list** (XML:parent/JSON:array): contains a list of all of the detectors statewide and the corresponding data for each detector. The remaining nodes are all within this parent/array.
- **status:** Provides the status of the sensor. Status options are: operational, off, or failed.
- **workzone:** Reserved field for identifying whether the sensor is located within a work zone. Not currently used.
- **detector-id:** The unique name for each detector station.
- **detector-location** (XML:parent/JSON:object): The following fields provide the coordinates and datum for the sensor:
 - **latitude:** Latitude coordinate
 - **longitude:** Longitude coordinate
 - **datum:** Datum used with coordinates
- **link-ownership:** The ownership of the roadway the sensor is detecting sensor data.
- **route-designator:** The roadway the sensor is detecting data for.
- **linear-reference** (XML:parent/JSON:object): The following fields provide the Iowa DOT LRS values:
 - **routeId:** This is a reserved field for the LRS routeId for future use.
 - **measure:** This is a reserved field for the LRS measure value for future use.
- **detector-type:** The type of detector used to collect data
- **approach-direction:** The direction of travel collected by the sensor
- **lanes-type:** The type of lanes collected by the sensors. Some examples include: through lanes, right exit ramp, right entrance ramp, etc.
- **detectionReports** (XML:parent/JSON:array): The remaining fields include the data collected by the sensors, including speed, count, occupancy, and classification.

- **count:** The total number of vehicles detected. To convert to hourly volumes, multiply by 60.
- **occupancy:** The average occupancy of all lanes
- **speed:** The average speeds of all lanes in mi/h
- **small-count:** The total number of small vehicles detected
- **medium-count:** The total number of medium vehicles detected
- **large-count:** The total number of large vehicles detected
- **lanes** (XML:parent/JSON:array): The following fields separate the data collected by the sensor by lane. The number of lanes will vary based on the sensor.
 - **speed:** The average speed of vehicles detected in the lane-ID, identified in mi/h
 - **lane-id:** The lane-ID for the data collected. The number of the lane-IDs will vary based on the detector. Typically lane number will start from the median. So lane-ID 1 will be the left-most lane and continue to the shoulder.
 - **count:** The number of vehicles detected in the lane-ID identified. To convert to hourly volumes, multiply by 60.
 - **occupancy:** The average occupancy in the lane-ID identified
 - **small-count:** The count of small vehicles detected in the lane-ID identified
 - **medium-count:** The count of medium vehicles detected in the lane-ID identified
 - **large-count:** The count of large vehicles detected in the lane-ID identified

Quality Assurance

Sensor data is highly variable and must go through a process to remove erroneous data. Sensor data goes through a quality control process by evaluating the speed and volume of the raw data. The raw sensor data is provided every 20 seconds for each individual lane. With this information, a threshold is defined based on the total number of vehicles that could pass the sensor during a 20-second period for each lane. If the volume is greater than the threshold, the data are removed from the reporting. Thresholds are also defined for speed to remove high speeds and for speeds less than 0 mph. In addition to these checks, a redundant data check is

being implemented to remove sensors that produce replicated data for multiple time periods. This is an indicator that the sensor is not correctly detecting vehicles since the data are not changing.

Work Zone Alert Feed

The following section provides a description of the nodes/fields within the work zone alert feed. The work zone alert feed provides the work zone and direction of any slow or stop conditions across the state. These slow or stopped conditions are defined based on sensors located within the work zone ID. This information is used for text alerting, as well as the operations dashboard at the TMC. The alert feed also provides the milepost where the alert is identified along with the route.

- **id:** The unique ID for the work zone event
- **startTime:** The start timestamp of the event in the format “YYYY-MM-DDThh:mm:ss.sssZ”
- **endTime:** The end timestamp of the event in the format “YYYY-MM-DDThh:mm:ss.sssZ”
- **workzone:** The TCP work zone ID (string)
- **projectName:** The name of the work zone referenced by DOT (string)
- **route:** The route where the work zone is located (string)
- **direction:** The direction of the event for the work zone (string)
- **avgSpeed:** The average speed detected in the work zone
- **severity:** The severity of the work zone event (slow or stopped).
- **routeId:** The Iowa DOT LRS routeId
- **measure:** The Iowa DOT LRS measure value
- **milepost:** The milepost the work zone event is located

Quality Assurance

The work zone alert feed relies on sensors located within or surrounding the work zone. Similar to the sensor feed, this data is highly variable and must be cleaned to eliminate false alerts being

provided to the TMC, DOT staff, and others receiving the alerts. The first step in this process is completed in a pre-processing module, which parses then smooths the raw sensor data. For the data smoothing, a fast wavelet transform is applied to smooth the raw speed and occupancy data from the sensor. An example of the smoothing is shown in Figure 5, where the raw data appears in light blue and the smoothed data as a solid black line.

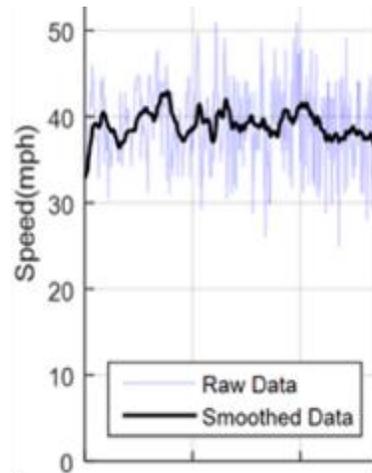


Figure 5. Data smoothing for work zone alert

Assigning Slow or Stopped Conditions

Machine Learning was used to identify slow and stopped conditions from historical work zone data, from previous years, to account for a variety of different traffic characteristics in work zones. K-mean was applied to cluster data into groups, which would later be used to assign proper labels of slow and stopped conditions. Finally, a supervised decision tree algorithm was trained based on the labeled data. The trained results are illustrated in Figure 6, where the horizontal-axis is the smoothed sensor occupancy and the vertical-axis is the smoothed traffic speed. The domain is coded into four different sections as shown in the left chart and the decision regions are represented by color. The right chart displays the data that was used to train the algorithm overlaid on the decision tree. From top to bottom, the decision regions are: normal conditions with no message display (red); “Traffic Delay Possible,” which also has no message display (grey); “Slow Traffic” (light blue) and “Stopped Traffic” (blue). If a sensor’s data was located in any of the decision tree boundaries, it would be classified as that condition. The system would then identify the lowest classified sensor within the work zone for each direction and report that in the feed, only if it was in a slow or stopped condition.

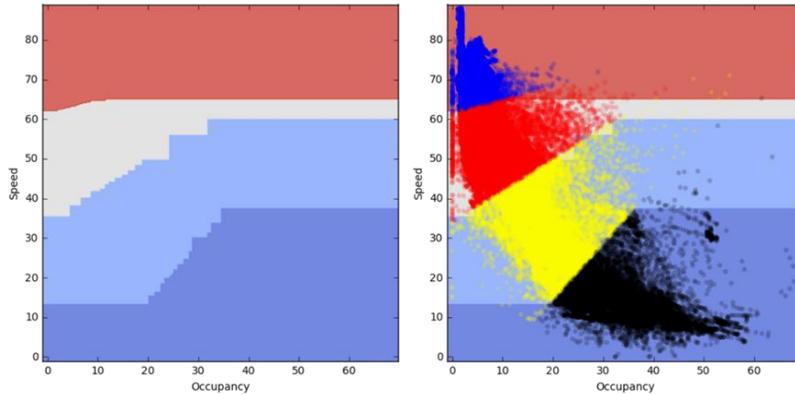


Figure 6. Historical data analysis and decision boundaries for work zone alert

DMS Inventory History Feed

The following section provides a description of the nodes/fields within the DMS inventory history feed. The DOT currently does not track where DMS were historically located specifically for portable DMS, which are used by maintenance garages and in work zones. The feed was intended to provide a historical inventory of where all DMS are located in Iowa once a day. This will allow the DOT to track where DMS have been, how often they are used, moved, etc.

- **archiveTime:** The time that data was archived from the DMS location feed
- **dmsInventoryList** (XML:parent/JSON:array):
 - **deviceId:** The unique device ID for the DMS
 - **deviceName:** The name of the DMS (string)
 - **organizationId:** The organization ID for the DMS
 - **organizationName:** The name of the organization (typically the metro area) (string)
 - **longitude:** The longitude of the DMS
 - **latitude:** The latitude of the DMS
 - **networkId:** The DOT server network the DMS is on (string)
 - **linkDirection:** The direction of roadway the DMS is on (string)

- **linearReference:** The approximate mile marker of the DMS (not related to Iowa DOT LRS)
- **routeDesignator:** The route the DMS is on (string)
- **signType:** The type of DMS sign. Typical option include Variable Message Sign (VMS), or Portable VMS (string)
- **signTechnology:** Unknown
- **signHeight:** Unknown
- **signWidth:** Unknown
- **charHeight:** Unknown
- **charWidth:** Unknown
- **beaconType:** Unknown

Quality Assurance

This service only provides a historical inventory of DMS and does not provide any quality assurance.

INRIX Daily Performance Measures

The following section provides a description of the nodes/fields within the INRIX daily performance measures feed. This feed provides performance measures related to speed, delay, data quality, and bottlenecks for all INRIX segments in Iowa. These performance measures use the XD segmentation provided by INRIX and summarizes these measures daily. INRIX provides a quality score that identifies whether the data is “real-time” or a blend of historical and real-time data. A quality score of 30 (thirtyMeasures) is real-time data and of highest accuracy. A score of 20 (twentyMeasures) is a blend of real-time and historic, which may not always reflect actual traffic conditions but is an alternative if real-time data is unavailable at the segment location.

- **code:** XD segment number
- **twentyMeasures** (XML:parent/JSON:array): measures that are using the confidence score 20 data, which represents the blend of historical and real-time data from INRIX

- **speed:** Average real-time speed (mi/h) for a given XD Segment, date when confidence score is equal to 20
- **avgSpeed:** Average historical average speed (mi/h) for the given XD segment, date when confidence score is equal to 20. This historical average speed is provided by INRIX for day of week/time of day.
- **refSpeed:** Average reference speed (mi/h) for the given XD segment, date when confidence scores is equal to 20. This reference speed is the free-flow speed provided by INRIX.
- **travelTime:** Average travel time (minutes) for the given XD segment, date when confidence score is equal to 20
- **refDelay:** Total delay (hours) with respect to reference speed for the given XD segment, date when confidence score is equal to 20, given by Equation 1. Delay is computed only when the speed observed is lower than the corresponding reference speed.

$$RefDelay(hours) = \sum \frac{traveltime(mins)}{60} - \frac{Length\ of\ XD\ segment(miles)}{Reference\ Speed(mph)}$$

- **avgDelay:** Total delay (hours) with respect to average historical speed for the given XD segment, date when confidence score is equal to 20, given by Equation 2. Delay is computed only when the speed observed is lower than the corresponding average historical speed.

$$AvgDelay(hours) = \sum \frac{traveltime(mins)}{60} - \frac{Length\ of\ XD\ segment(miles)}{Reference\ Speed(mph)}$$

- **thirtyMeasures** (XML:parent/JSON:array): measures that are using the confidence score 30 data, which represents the real-time data from INRIX
 - **speed:** Average real-time speed (mi/h) for the given XD segment, date when confidence score is equal to 30
 - **avgSpeed:** Average historical average speed (mi/h) for the given XD segment, date when confidence score is equal to 30. This historical average speed is provided by INRIX for day of week/time of day.
 - **refSpeed:** Average reference speed (mi/h) for the given XD segment, date when confidence score is equal to 30. This reference speed is the free-flow speed provided by INRIX.

- **travelTime:** Average travel time (minutes) for the given XD segment, date when confidence score is equal to 30
- **refDelay:** Total delay (hours) with respect to reference speed for the given XD segment, date when confidence scores is equal to 30, given by Equation 1. Delay is computed only when the speed observed is lower than the corresponding reference speed.
- **avgDelay:** Total delay (hours) with respect to average historical speed for the given XD segment, date when confidence score is equal to 30, given by Equation 2. Delay is computed only when the speed observed is lower than the corresponding average historical speed.
- **dataQuality** (XML:parent/JSON:array): Performance measures that represent how much real-time data is provided for the given segment.
 - **cValue:** Average CValue for the XD segment, date when the confidence score is equal to 30. Note: CValue is provided by INRIX only when confidence score is 30.
 - **count10:** Fraction of the data for the given XD segment, date when confidence score is equal to 10
 - **count20:** Fraction of the data for the given XD segment, date when confidence score is equal to 20
 - **count30:** Fraction of the data for the given XD segment, date when confidence score is equal to 30
 - **totalCount:** Total date count for the given XD segment, date for all confidence scores
- **bottleNecks** (XML:parent/JSON:array): Measures representing the number of slowdowns observed on the segment
 - **bottleNeckCount:** Total number of bottlenecks observed for the XD segment, date. The bottleneck definition is similar to the one adopted by INRIX. A bottleneck represents a single XD segment or a group of XD segments in which the actual travel speed (for any confidence score) drops below 60% of the reference speed for a period longer than five minutes. The bottleneck is considered closed once the travel speed returns to a value greater than 60% of reference speed and remains there for 10 minutes. Figure 7 shows a basic example of the speed over time plot for a bottleneck, and indicates where the bottleneck's status would be updated.

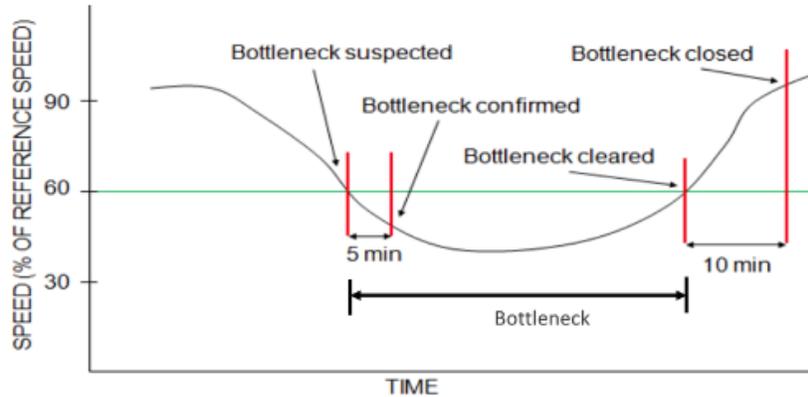


Figure 7. Bottleneck definition

- **avgDuration:** Average duration (minutes) of bottlenecks observed in the given XD segment, date
- **abgMaxLength:** Average maximum length (miles) of all bottlenecks observed in the XD segment, date
- **events** (XML:parent/JSON:array):
 - **count:** The number of ATMS events (non-construction) on the segment
- **date:** Date of the performance measure calculation in the format YYYY-MM-DD

Quality Assurance

INRIX data relies on vehicle probes and the quality of their data depends on how many probes are located on each segment of roadway. A confidence score is provided with the data that designates whether the data are real-time (score 30), a blend of real-time and historical (score 20) or historical only (score 10). The “dataQuality” scores are calculated to provide a score of how much of the data provided is within each quality score for the entire day. This will allow for a percentage to be found to determine how much data throughout the day falls into each category. These measures can then be used to determine whether the “thirtyMeasures” or “twentyMeasures” should be used for the given segment.

With these data quality measures, users can determine what an acceptable amount of real-time data is needed to use the “thirtyMeasures”. The “thirtyMeasures” calculate the performance only when real-time data is available. For example, if real-time data is only available for 50% of the day, only that 50% of the data will be used in performance measure reporting. Real-time data is the most reliable and should be used when available. Outside of interstate locations, the amount of real-time data varies significantly and may be time-of-day dependent based on the number of probes available. Since all locations in the state do not have a high reliability in receiving real-

time data, the “twentyMeasures” were calculated to provide an alternative when real-time data is not reliable. Similar to the “thirtyMeasures,” these measures calculate performance only when the score 20 data is available.

INRIX Monthly Performance Measures

The INRIX monthly performance measures are similar to the INRIX daily performance measures but reported on a monthly basis for each segment. See INRIX daily performance measures for all references and the quality assurance process.

This data feed is intended to report the monthly performance starting in June 2017. XD segmentation is used for performance reporting. Any monthly performance measures before June 2017 should be retrieved from INRIX historical performance measures

INRIX Historical Performance Measures

The INRIX historical performance measures are similar to the INRIX monthly performance measures. See INRIX daily performance measures for all references and the quality assurance process.

This data feed is intended for historical reference of monthly performance between January 2013 to May 2017. TMC segmentation is used for performance reporting. This feed will not be updated and future performance should be retrieved from the INRIX monthly performance measures.

Work Zone Congestion Events

The following section provides a description of the nodes/fields within the work zone congestion events. This feed is a historical record of the causes for the work zone alert feed. This archive provides the detailed causes for all alert feeds starting on June 8th, 2017. This also includes any active alerts present in the work zone alert feed.

In addition to the alert, this feed identifies which sensors are triggering the slow or stopped conditions, when their status has changed, and when the sensor has recovered from the slow or stopped condition. Indicators are also available to identify whether a text alert was sent, if a clearance message was sent, or if a long duration message was sent.

- **id:** Alternative unique id for the work zone event (string)
- **workzone:** The TCP work zone ID (string)
- **eventId:** The unique ID for the work zone event

- **sensorEvents** (XML:parent/JSON:array): The sensors and alerts which triggered the congestion event
 - **name:** The sensor name triggering the alert (string)
 - **alert:** The type of alert either SLOW or STOP
 - **avgSped:** The average speed at the sensor
 - **completed:** Whether the alert was still active or completed (string)
 - **startTime:** Start time of the sensor based alert in milliseconds since Unix Epoch
 - **endTime:** End time of the sensor based alert in milliseconds since Unix Epoch
 - **lastUpdateTime:** The last time the sensor based alert was updated in milliseconds since Unix Epoch
- **completed:** Indicator of whether the alert was completed (TRUE/FALSE)
- **alertSmsSent:** Indicator of whether a text alert was indicating the alert was in a stopped condition for five continuous minutes (TRUE/FALSE)
- **clearanceSmsSent:** Indicator of whether a clearance text alert was sent indicating that the alert was completed, only possible if the alertSmsSent is TRUE (TRUE/FALSE)
- **longEventSmsSent:** Indicator of whether a long event text was sent. This alert is sent if a sensor is in an alert condition for over an hour (TRUE/FALSE).
- **startTime:** Start time of the overall alert in milliseconds since Unix Epoch
- **endTime:** End time of the overall alert in milliseconds since Unix Epoch
- **lastUpdateTime:** The last time the overall alert was updated in milliseconds since Unix Epoch

Quality Assurance

This feed uses the same quality assurance process as the work zone alert feed.

In addition to that process, this feed also serves as a check for the text alerting system. Users can use this feed to check when text alerts were sent, what sensor triggered the text alert, and whether the alert was valid. The system was designed so that text alerts are only sent to DOT staff and other relevant staff when a work zone is in a stopped condition for five continuous minutes. This eliminates the false calls that could be generated for minor slowdowns that can quickly clear.

The system also checks if a sensor has been in a slow condition for a long period of time. If a sensor is in the slow condition for over an hour, a text alert is sent to the REACTOR Lab notifying them that a sensor may have an error that is triggering the slow condition. This can include sensors being programmed incorrectly, sensors detecting the wrong lane configuration, etc.

ATMS Event Feed

The following section provides a description of the nodes/fields within the ATMS event feed. The DOT currently receives a nightly update from their ATMS system that exports all events that were completed the previous day. This includes any accidents, stalled vehicles, construction, maintenance, etc., managed by the TMC.

This feed is intended to provide the archive of all ATMS events starting in August 2015. This feed can be used to locate specific event types and for ATMS performance monitoring.

- **date:** The date of the event in MM/DD/YYYY format
- **Event_ID:** The unique event ID for the ATMS event
- **State:** The state the event occurs in (string)
- **Facility:** The roadway the event occurs on (string)
- **Dir:** The direction the event occurs in (string)
- **Impact:** Unknown
- **Detection:** How the event was detected by the TMC (string)
- **DMS:** The number of DMS that were activated
- **DMSMsgs:** The number of messages posted on the DMS
- **ATISMsgs:** The number of Automatic Terminal Information Service (ATIS) messages sent

- **Comments:** Any comments by the TMC operator
- **Recv:** The time the event was received in MM/DD/YYYY hh:mm format
- **Cleared:** The time the event was cleared in MM/DD/YYYY hh:mm format
- **Duration:** Unknown
- **TotalTime:** The total duration of the event
- **Type:** The type of event (I – Incident, C – Construction)
- **LC1:** Unknown
- **LO1:** Unknown
- **L1Close:** Unknown
- **L1Open:** Unknown
- **L1Tot:** Unknown
- **Latitude:** Latitude coordinates in WGS84
- **Longitude:** Longitude coordinates in WGS84
- **EventType:** The type of event (string)
- **Injuries:** The number of injuries involved in the event
- **Fatalities:** The number of fatalities involved in the event
- **Cars:** The number of cars involved in the event
- **SUVs:** The number of SUVs involved in the event
- **TractorTrailer:** The number of tractor trailers involved in the event
- **Trucks:** The number of trucks involved in the event

- **DOTTrucks:** The number of DOT trucks involved in the event
- **Motorcycle:** The number of motorcycles involved in the event
- **DOT:** Unknown
- **WZ:** Unknown
- **RoadType:** The type of roadway (string)
- **ConfirmedBy:** The operator confirming the event (string)
- **Block01 to Block20:** The number of blocked and open lanes including which lanes, ^ represents an open lane and X represents a closed lane (string)
- **Block01Time to Block20Time:** The time the corresponding lane blockage occurs
- **BlockageLaneMin:** The total number of minutes the lanes were blocked
- **County:** The county the event was located in (string)
- **CrossStreet:** The nearby cross street (string)
- **Weather:** Unknown
- **SegmentId:** The segment ID where the event is located
- **ItisCodes:** Integrated Taxonomic Information System (ITIS) codes for Highway Helper responses (string)
- **Responders:** List of responders arriving to event (string)
- **DayOfWeek:** The day of week the event occurred on (string)
- **FollowUp:** Unknown
- **FollowUpNotes:** Unknown
- **OnlyShouldersClosed:** Indicator if only shoulder is closed (Y/N)

- **MAZone:** Unknown
- **MAOpr:** Unknown
- **AssistedVehicles:** Description of the vehicle assisted (string)
- **PDReportNumber:** Unknown
- **DOTNotes:** Any notes by DOT personnel (string)
- **TrafficNormalAt:** The time traffic returned to normal operating conditions in MM/DD/YYYY hh:mm format
- **SecondarytoEventId:** The unique event ID for the event causing the secondary

Quality Assurance

This service only provides a historical inventory of ATMS events does not provide any quality assurance. The feed is secured, so only authorized users can access.

Waze Alert Feed

The following section provides a description of the nodes/fields within the Waze alert feed. Crowd-sourced data has tremendous value to the DOT that provides citizen reports of slowdowns and accidents. This information can typically be received significantly quicker than other sensors, cameras or other data sources. The problem with crowd-sourced data, such as Waze, is that there are often duplicate reports. Operators do not need to receive these duplicate reports of information they already have, so a procedure was developed to eliminate these reports. In addition, crowd-sourced data are point sources and do not accurately display the extent of a slowdown. This feed provides a “clean” Waze data feed that clusters similar Waze events and represents them as lines to more accurately show the impact area.

- **clusterId:** The unique cluster ID for the Waze event
- **startTime:** The start time of the Waze event in YYYY-MM-DD hh:mm:ss format
- **type:** The type of Waze event (string)
- **confidence:** Placeholder for confidence score of Waze cluster

- **linear-reference** (XML:parent/JSON:array): The Iowa DOT linear reference values identifying the location of the Waze event
 - **routeId**: The route ID for the Iowa DOT LRS (string)
 - **start**: The start measure along the route for the Iowa DOT LRS
 - **end**: The end measure, if applicable, along the route for the Iowa DOT LRS
 - **geoRSSType**: The type of Waze event either a point or line

Quality Assurance

The quality assurance process for Waze attempts to remove redundant reports and show an impact area of the event rather than a point location. Once the Waze events are received, they are sent through the RAMS system to obtain the Iowa DOT LRS measures. Using this information, the Waze events are then clustered using a density-based spatial clustering of applications with noise (DBSCAN) based on time, spatial proximity, roadway, and event types. This process eliminated 61 percent of the Waze reports, which were redundant and were then clustered as one incident. Through this process, the clustered events now provide the start and end locations of reports, which show an impact area. In addition to the clustering, a reliability measure was developed that uses the score provided, by each event, as well as how many reports are received within the cluster. An event that has multiple reports from Waze would therefore have higher reliability than a location with only one or very few crowd-sourced reports.

LCPT Monthly Sensor Data

The following section provides a description of the nodes/fields within the LCPT monthly sensor data feed. This feed is used to populate the hourly volume within the LCPT for the Iowa DOT. Hourly volumes are calculated for all sensors for each month by time of day and day of week. The volumes are also converted to passenger car equivalents based on the truck percentage. This feed provides data for all sensors listed in the joint sensor feed section every month. This feed is primarily used in conjunction with the LCPT monthly sensor data and can be linked through the “detectorId” fields.

- **archiveDate**: The date the data was added to the feed in milliseconds since Unix Epoch
- **detectorId**: The unique ID for each detector station, which can be used to link with the LCPT stations (string)
- **month**: The month for the LCPT data

- **year:** The year for the LCPT data
- **day:** The day of week for the LCPT data (string)
- **lcptMeasures** (XML:parent/JSON:array): The LCPT data measures
 - **hour:** The hour of day for the data
 - **lanes:** The number of lanes in one direction
 - **volume:** Average hourly volume in vehicles/hour. This is the average 1-minute volume in an hour, then multiplied by 60.
 - **truckPercent:** The truck percentage which is a multiplier, not a percentage number. (e.g. 0.01 means 1%). This is the average one-minute truck percentage in an hour.
 - **pceAvg:** The passenger car equivalent volume. Using the highway capacity manual (HCM) 2010 equations. This is the average 1-minute passenger car equivalent (PCE) in an hour, then multiplied by 60.
 - **pceMin:** The passenger car equivalent volume. Using the HCM 2010 equations. This is the minimum 1-minute PCE in an hour, then multiplied by 60.
 - **pce25:** The passenger car equivalent volume. Using the HCM 2010 equations. This is the 25th percentile 1-min PCE in an hour, then multiplied by 60.
 - **pceMedian:** The passenger car equivalent volume. Using the HCM 2010 equations. This is the median 1-minute PCE in an hour, then multiplied by 60.
 - **pce75:** The passenger car equivalent volume. Using the HCM 2010 equations. This is the 75th percentile 1-minute PCE in an hour, then multiplied by 60.
 - **pceMax:** The passenger car equivalent volume. Using the HCM 2010 equations. This is the maximum 1-minute PCE in an hour, then multiplied by 60.

Quality Assurance

The LCPT process goes through a quality assurance process to ensure high quality data is available for use within the Iowa DOT. The process that is used in the sensor feed of removing high speed and volume data is followed to remove erroneous raw data. The LCPT also reduces the amount of sensors by combining the mainline and ramp data. This process is completed by adding the sensor data for all sensors at a given station that are in the same direction of travel.

This provides a reliable estimate in volume that is comparable to the average annual daily traffic (AADT) but broken down by hour.

Stations Feed

The following section provides a description of the nodes/fields within the stations feed. The stations feed is used to identify which sensors should be included in the current month of LCPT data. Since the sensors can vary from month to month, this system allows for a dynamic list of sensors to be included in the LCPT. If a sensor is added or removed at any time during the current month, this feed will retain this information so that all sensors are included in the LCPT data. At the end of each month the process will start again. The feed is primarily for internal use for within the REACTOR Lab in creating feeds.

- **stationName:** The station name that contains multiple detectors (string)
- **detectorId:** The unique ID for each detector station (string)
- **detectorName:** The old detector name for each detector station (string)
- **avgLatitude:** The average latitude for all detectors related to the station name
- **avgLongitude:** The average longitude for all detectors related to the station name
- **route:** The route name the detector is located on (string)
- **direction:** The direction the detector is located (string)
- **district:** The Iowa DOT district the sensor is located in
- **type:** The type of sensor including: intelligent transportation system (ITS), Workzone, Temporary, attenuated total reflectance (ATR), or RWIS (string)
- **ownership:** The ownership of the roadway the sensor is detecting sensor data (string)
- **aadt:** The AADT at the location of the sensor
- **addtCountYear:** The year the AADT data was available for
- **lastUpdate:** When the sensor data was last updated in the format MM-DD-YYYY

Quality Assurance

This service only provides a historical inventory of sensor locations for the current month and does not provide any quality assurance. Sensor locations are verified every day to identify sensors that have been added to the sensor list or have moved to a different location. This feed includes all sensors including ramps.

LCPT Stations

The following section provides a description of the nodes/fields within the LCPT stations feed. This feed provides a monthly snapshot of the sensors that were deployed each month. This provides the capability to change the type of sensors from month to month (change from temporary, work zone, permanent, etc.), as well as retain a dynamic list of sensors for the LCPT. This feeds organizes sensors into stations. Stations represent the location where data is collected, while the sensors represent the different directions of travel. This feed is primarily used in conjunction with the LCPT monthly sensor data and can be linked through the “detectorId” fields.

- **month** (XML:parent/JSON:array): The month of the station data

- **station** (XML:parent/JSON:array): List of stations identification
 - **stationName**: The station name that contains multiple detectors (string)
 - **detectorId**: The unique ID for each detector station that can be used to link with the LCPT monthly sensor data (string)
 - **detectorName**: The old detector name for each detector station (string)
 - **avgLatitude**: The average latitude for all detectors related to the station name
 - **avgLongitude**: The average longitude for all detectors related to the station name
 - **route**: The route name the detector is located on (string)
 - **direction**: The direction the detector is located (string)
 - **district**: The Iowa DOT district the sensor is located in
 - **type**: The type of sensor including: ITS, Workzone, Temporary, ATR, or RWIS (string)
 - **ownership**: The ownership of the roadway the sensor is detecting sensor data (string)

- **aadt:** The AADT at the location of the sensor
- **aadtCountYear:** The year the AADT data was available for
- **lastUpdate:** When the sensor data was last updated in the format MM-DD-YYYY

Quality Assurance

This service provides a historical inventory of sensor locations for use within the LCPT. The station feed is appended to this feed at the end of every month. This feed only contains mainline sensors and removes any ramp sensors that are not included in the LCPT. Any ramp sensors in the stations feed are removed before appending. This results in only mainline sensors (typically one for each direction of travel) that make up a station. The coordinates of the sensors representing the station are averaged to ensure that both sensors are located at the same coordinate within the station for display in the LCPT.

**THE INSTITUTE FOR TRANSPORTATION IS THE FOCAL POINT FOR TRANSPORTATION
AT IOWA STATE UNIVERSITY.**

InTrans centers and programs perform transportation research and provide technology transfer services for government agencies and private companies;

InTrans manages its own education program for transportation students and provides K-12 resources; and

InTrans conducts local, regional, and national transportation services and continuing education programs.



**IOWA STATE
UNIVERSITY**

Visit www.InTrans.iastate.edu for color pdfs of this and other research reports.