U.S. Department of Transportation
Office of the Assistant Secretary for Research and Technology
Module 14 Part 2:

Applying GTFS-realtime to your Agency

```json
entity:
  id: "1001"
  is_deleted: false
  trip_update {
    trip {
      trip_id: "64"
      start_time: "14:15:00"
      start_date: "20151130"
      schedule_relationship: SCHEDULED
      route_id: "2"
      direction_id: 1
    }
    stop_time_update {
      stop_sequence: 2
      arrival {
        delay: 0
        time: 1448892960000
      }
      departure {
        delay: 0
        time: 1448892960000
      }
      stop_id: "106"
      schedule_relationship: SCHEDULED
    }
    stop_time_update {
      stop_sequence: 3
      arrival {
        delay: 0
        time: 1448893320000
      }
      departure {
        delay: 0
        time: 1448893320000
      }
      stop_id: "154"
      schedule_relationship: SCHEDULED
    }
    stop_time_update {
      stop_sequence: 4
      arrival {
        delay: 0
        time: 1448893440000
      }
      departure {
        delay: 0
        time: 1448893440000
      }
      stop_id: "19"
      schedule_relationship: SCHEDULED
    }
    stop_time_update {
      stop_sequence: 5
```

[Map of Lindbergh Field & Terminal 1]
Instructor

Scott Altman

Technical Staff
ConSysTec

Course Co-Developer: Bruce Eisenhart
Learning Objectives

Define the scope, uses for, and users of the GTFS-realtime specification

Apply transit source applications to GTFS-realtime

Implementation of GTFS-realtime
Learning Objective 1

Define the scope, uses for, and users of the GTFS-realtime specification
Background of GTFS-realtime

User Story

Current Time: 7:28 AM

Source: Google Maps
Background of GTFS-realtime

Overview

- **Real time** version of the General Transit Feed Specification (GTFS)
- Launched in 2011 - 6 cities initially
- Maintained by Google
- Specification, not a standard
- Information/Feed Types Included
  - **Trip Update** - When will the vehicle arrive/depart?
  - **Vehicle Position** - Where is the vehicle?
  - **Alerts** - Are any planned or unplanned events affecting service?
What GTFS-realtime is and is not

- GTFS-realtime is:
  - Format for sharing real-time transit information
  - Set of rules for encoding such data

- GTFS-realtime is not:
  - Software that provides real-time information
  - Software that uses real-time information
Background of GTFS-realtime

Specification Update Process

- Specification maintained by Google
- No formal standardization process
- Focuses on online discussion forum
- Advantages and Disadvantages to this process
Background of GTFS-realtime

Specification Update Process

(1) Proposal

(2) Feedback

(3) Testing

(4) Final Comments

(5) Implement Change
Background of GTFS-realtime

Alternatives to GTFS-realtime

- Transit Communications Interface Profiles (TCIP)
  - American Public Transportation Association (APTA)
  - ITS Transit Standards Modules 3 and 4

- Service Interface for Real Time Information (SIRI)
  - European Committee for Standardization (CEN)
  - Several deployments in the United States
Benefits and Uses of GTFS-realtime

Uses and Users

- **Uses**
  - Customer facing information
  - Performance Measurement

- **Using the three feed types**
  - Trip Update
  - Vehicle Position
  - Alerts

- **Users**
  - 3rd Party Developers
  - Agency Staff, Consultants, and Systems
Benefits and Uses of GTFS-realtime

GTFS-realtime and the National ITS Architecture

APTS01 – Transit Vehicle Tracking

Source: USDOT/ National ITS Architecture
Benefits and Uses of GTFS-realtime

GTFS-realtime and the National ITS Architecture

Source: USDOT/ National ITS Architecture
Benefits and Uses of GTFS-realtime

Benefits of Implementing GTFS-realtime

- Provide **real time information** to customers
- Required **data inputs** often available
- **Open source** tools available
- Becoming **widely-accepted and deployed**
Target Devices for Real Time Information

Applications

- Customer applications
- Smartphone and desktop
- Electronic signage
- Performance analysis tools
- Google Maps Transit

Source: Google Maps
ACTIVITY
Which of the following groups may participate in developing the GTFS-realtime specification?

Answer Choices

a) Transit Agency Staff
b) Planning Tool Users
c) Smartphone Application Developers
d) All of the Above
Review of Answers

a) Transit Agency Staff

Incorrect. This is only one group that may participate.

b) Planning Tool Users

Incorrect. This is only one group that may participate.

c) Smartphone Application Developers

Incorrect. This is only one group that may participate.

d) All of the Above

Correct! Anyone who can claim involvement with GTFS-realtime may participate.
Learning Objective 2

Apply transit source applications to GTFS-realtime
Data Necessary to Create a GTFS-realtime Feed

Data Types

- Three feed types in GTFS-realtime:
  - Trip Update
  - Vehicle Position
  - Alerts

- Why is this information important?
  - It answers key questions about the current status of transit service
Data Necessary to Create a GTFS-realtime Feed

Trip Update

- WHEN will Trip/Vehicle A arrive at location X?
- WHAT is the delay of the trip?

Source: http://openclipart.org
Data Necessary to Create a GTFS-realtime Feed

Vehicle Position

- WHERE is vehicle B right now?
- To answer this need, we ask:
  - WHAT are the GPS coordinates of the vehicle? OR
  - WHAT stop is the vehicle en route to/approaching/physically stopped at?

Source: http://openclipart.org
Data Necessary to Create a GTFS-realtime Feed

Alert

- WHAT is happening?
- WHERE is it happening?
- WHEN is it happening?
- HOW does this affect service?

Source: http://openclipart.org
Data Necessary to Create a GTFS-realtime Feed

GTFS-realtime Data Sources

- Computer Aided Dispatch (CAD)
- Automatic Vehicle Location (AVL)
- Incident Management Systems (IMS)
- Existing Data Feeds
- Reference Data (GTFS)
Issues with Providing Stop Time Update (ETA) Data

Predicting Estimated Time of Arrival

- Current Location
- Current Delay
- Events/Incidents
- Static Schedule
- Historical Travel Time
Issues with Providing Stop Time Update (ETA) Data

ETA Issues

- Stop Time Updates disseminate predicted arrival/departure times
- Predictions depend on data available
- Data should only be shown if it is reliable
- High reliability = high confidence in accuracy
- Current delay may be a better field to display

Source: Google Maps
GTFS-realtime Content and Structure

Protocol Buffers

- GTFS-realtime is encoded in Protocol Buffer format
- “Binary String”- Not human readable
- Must translate into a text format to appear readable
- “.proto” files define the structure
Protocol Buffers

entity {
  id: "V1001"
  is_deleted: false
  vehicle {
    trip {
      trip_id: "121"
      start_time: "15:23:00"
      start_date: "20160512"
      schedule_relationship: SCHEDULED
      route_id: "2"
      direction_id: 1
    }
    current_stop_sequence: 19
    current_status: IN_TRANSIT_TO
    timestamp: 1463080565700
    stop_id: "96"
  }
}

Source: TRANSCOM
GTFS-realtime Content and Structure

Organization of a GTFS-realtime Feed

- GTFS-realtime is a series of **messages**
- Messages contain **fields**
- Fields may be
  - Another message
  - Single value
- Does not support request/response
- Broadcast messages
- Can be extended

<table>
<thead>
<tr>
<th>Message1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field1: “xyyyyyzzzzz”</td>
</tr>
<tr>
<td>Field2: “12345678”</td>
</tr>
<tr>
<td>Field3:</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>MessageA</td>
</tr>
<tr>
<td>FieldA: “aaa”</td>
</tr>
<tr>
<td>FieldB: “bbb”</td>
</tr>
</tbody>
</table>
GTFS-realtime Content and Structure

GTFS-realtime Field Details

- Message
- Series of fields
- Type of fields
- Required, Optional, Repeated
- Description
- POSIX TIME- Seconds since Jan 1, 1970
- Experimental

POSIX
12:00:00 AM on January 1, 2016 EST
1464104462 sec since 12:00 AM January 1, 1970 GMT
## GTFS-realtime Data Types

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>message</td>
<td>departure{</td>
</tr>
<tr>
<td></td>
<td>time: 8432342345</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
<tr>
<td>enum</td>
<td>SCHEDULED</td>
</tr>
<tr>
<td>string</td>
<td>“express1”</td>
</tr>
<tr>
<td>uint32, uint64</td>
<td>120</td>
</tr>
<tr>
<td>int32, int64</td>
<td>-20</td>
</tr>
<tr>
<td>float/double</td>
<td>57.4232425</td>
</tr>
<tr>
<td>bool</td>
<td>true</td>
</tr>
</tbody>
</table>
GTFS-realtime Content and Structure

GTFS-realtime Specification Structure

GTFS-realtime Feed
msg FeedMessage

header
msg FeedHeader

entity
msg FeedEntity

trip_update
msg TripUpdate

vehicle
msg VehiclePosition

alert
msg Alert
## GTFS-realtime Content and Structure

### FeedMessage

<table>
<thead>
<tr>
<th>Field</th>
<th>Required/Optional</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>header</td>
<td>required</td>
<td>msg FeedHeader</td>
</tr>
<tr>
<td>entity</td>
<td>repeated</td>
<td>msg FeedEntity</td>
</tr>
</tbody>
</table>
GTFS-realtime Content and Structure

FeedMessage

```json
header {
  gtfs_realtime_version: "1"
  timestamp: 1461619328
}
entity {
  id: "351"
  trip_update {
    trip {
      route_id: "2"
    }
    stop_time_update {
      departure {
        delay: 0
        time: 1461619440
      }
      stop_id: "68"
    }
    vehicle {
      label: "351"
    }
  }
}
```

Source: TRANSCOM
## GTFS-realtime Content and Structure

### FeedHeader

<table>
<thead>
<tr>
<th>Field</th>
<th>Required/Optional</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>gtfs_realtime_version</td>
<td>required</td>
<td>string</td>
</tr>
<tr>
<td>incrementality</td>
<td>optional</td>
<td>enum Incrementality: FULL_DATASET, DIFFERENTIAL</td>
</tr>
<tr>
<td>timestamp</td>
<td>integer</td>
<td>uint64</td>
</tr>
</tbody>
</table>

```json
header {
  gtfs_realtime_version: "1.0"
  incrementality: FULL_DATASET
  timestamp: 1463080565700
}
```

*Source: TRANSCOM*
# GTFS-realtime Content and Structure

## FeedEntity

<table>
<thead>
<tr>
<th>Field</th>
<th>Required/Optional</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>required</td>
<td>string</td>
</tr>
<tr>
<td>is_deleted</td>
<td>optional</td>
<td>bool</td>
</tr>
<tr>
<td>trip_update</td>
<td>optional</td>
<td>msg TripUpdate</td>
</tr>
<tr>
<td>vehicle</td>
<td>optional</td>
<td>msg VehiclePosition</td>
</tr>
<tr>
<td>alert</td>
<td>optional</td>
<td>msg Alert</td>
</tr>
</tbody>
</table>
FeedEntity

```json
entity {
    id: "V1001"
    is_deleted: false
    vehicle {
        trip {
            trip_id: "121"
            start_time: "15:23:00"
            start_date: "20160512"
            schedule_relationship: SCHEDULED
            route_id: "2"
            direction_id: 1
        }
        current_stop_sequence: 19
        current_status: IN_TRANSIT_TO
        timestamp: 1463080565700
        stop_id: "96"
    }
}
```
## GTFS-realtime Content and Structure

### TripUpdate

<table>
<thead>
<tr>
<th>Field</th>
<th>Required/Optional</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>trip</td>
<td>required</td>
<td>msg TripDescriptor</td>
</tr>
<tr>
<td>vehicle</td>
<td>optional</td>
<td>msg VehicleDescriptor</td>
</tr>
<tr>
<td>stop_time_update</td>
<td>repeated</td>
<td>msg StopTimeUpdate</td>
</tr>
<tr>
<td>timestamp</td>
<td>optional</td>
<td>uint64</td>
</tr>
<tr>
<td>delay</td>
<td>optional</td>
<td>int32</td>
</tr>
</tbody>
</table>
TripUpdate

trip_update {
    trip {
        trip_id: "058450_1..N02R"
        start_date: "20151104"
        route_id: "1"
    }
    stop_time_update {
        arrival {
            time: 1446651757
        }
        departure {
            time: 1446652057
        }
        stop_id: "103N"
    }
    stop_time_update {
        arrival {
            time: 1446652147
        }
        stop_id: "101N"
    }
}

Source: NYCT
### GTFS-realtime Content and Structure

#### TripDescriptor

<table>
<thead>
<tr>
<th>Field</th>
<th>Required/Optional</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>trip_id*</td>
<td>optional</td>
<td>string</td>
</tr>
<tr>
<td>route_id*</td>
<td>optional</td>
<td>string</td>
</tr>
<tr>
<td>direction_id*</td>
<td>optional</td>
<td>uint32 (0 or 1)</td>
</tr>
<tr>
<td>start_time</td>
<td>optional</td>
<td>string (not POSIX time)</td>
</tr>
<tr>
<td>start_date</td>
<td>optional</td>
<td>string</td>
</tr>
<tr>
<td>schedule_relationship</td>
<td>optional</td>
<td>enum ScheduleRelationship</td>
</tr>
</tbody>
</table>

Values in fields with asterisk (*) should match corresponding GTFS feed exactly
TripDescriptor

```json
trip {
    trip_id: "121"
    start_time: "15:23:00"
    start_date: "20160512"
    schedule_relationship: SCHEDULED
    route_id: "2"
    direction_id: 1
}
```

Source: TRANSCOM
GTFS-realtime Content and Structure

StopTimeUpdate

<table>
<thead>
<tr>
<th>Field</th>
<th>Required/Optional</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>stop_sequence*</td>
<td>optional</td>
<td>uint32</td>
</tr>
<tr>
<td>stop_id*</td>
<td>optional</td>
<td>string</td>
</tr>
<tr>
<td>arrival</td>
<td>optional</td>
<td>msg StopTimeEvent</td>
</tr>
<tr>
<td>departure</td>
<td>optional</td>
<td>msg StopTimeEvent</td>
</tr>
<tr>
<td>schedule_relationship</td>
<td>optional</td>
<td>enum ScheduleRelationship</td>
</tr>
</tbody>
</table>

Values in fields with asterisk (*) should match corresponding GTFS feed exactly
StopTimeUpdate

```json
stop_time_update {
    stop_sequence: 20
    arrival {
        delay: 0
        time: 1463066580000
    }
    departure {
        delay: 0
        time: 1463066580000
    }
    stop_id: "34"
    schedule_relationship: SCHEDULED
}
```

Source: TRANSCOM
## StopTimeEvent

<table>
<thead>
<tr>
<th>Field</th>
<th>Required/Optional</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>delay</td>
<td>optional</td>
<td>int32</td>
</tr>
<tr>
<td>time</td>
<td>optional</td>
<td>int64</td>
</tr>
<tr>
<td>uncertainty</td>
<td>optional</td>
<td>int32</td>
</tr>
</tbody>
</table>

```json
departure {
    delay: 0
    time: 1463066580000
}
```

Source: TRANSCOM
## GTFS-realtime Content and Structure

### VehiclePosition

<table>
<thead>
<tr>
<th>Field</th>
<th>Required/Optional</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>trip</td>
<td>optional</td>
<td>msg TripDescriptor</td>
</tr>
<tr>
<td>vehicle</td>
<td>optional</td>
<td>msg VehicleDescriptor</td>
</tr>
<tr>
<td>position</td>
<td>optional</td>
<td>msg Position</td>
</tr>
<tr>
<td>current_stop_sequence*</td>
<td>optional</td>
<td>uint32</td>
</tr>
<tr>
<td>stop_id*</td>
<td>optional</td>
<td>string</td>
</tr>
<tr>
<td>current_status</td>
<td>optional</td>
<td>enum VehicleStopStatus</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INCOMING_AT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>STOPPED_AT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IN_TRANSIT_TO</td>
</tr>
<tr>
<td>timestamp</td>
<td>optional</td>
<td>uint64</td>
</tr>
</tbody>
</table>

Values in fields with asterisk (*) should match corresponding GTFS feed exactly
## VehiclePosition

<table>
<thead>
<tr>
<th>Field</th>
<th>Required/Optional</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>congestion_level</td>
<td>optional</td>
<td>enum CongestionLevel&lt;br&gt;UNKNOWN_CONGESTION_LEVEL&lt;br&gt;RUNNING_SMOOTHLY&lt;br&gt;STOP_AND_GO&lt;br&gt;CONGESTION&lt;br&gt;SEVERE_CONGESTION</td>
</tr>
<tr>
<td>occupancy_status</td>
<td>optional</td>
<td>enum OccupancyStatus&lt;br&gt;EMPTY&lt;br&gt;MANY_SEATS_AVAILABLE&lt;br&gt;FEW_SEATS_AVAILABLE&lt;br&gt;STANDING_ROOM_ONLY&lt;br&gt;CRUSHED_STANDING_ROOM_ONLY&lt;br&gt;FULL&lt;br&gt;NOT_ACCEPTING_PASSENGERS</td>
</tr>
</tbody>
</table>
VehiclePosition

```json
vehicle {
  trip {
    trip_id: "30118425"
    start_date: "20160516"
    route_id: "1"
    direction_id: 1
  }
  position {
    latitude: 42.372686768
    longitude: -71.1155471802
    bearing: 220.0
  }
  current_stop_sequence: 4
  current_status: STOPPED_AT
  timestamp: 1463433348
  stop_id: "2167"
  vehicle {
    id: "y2294"
    label: "2294"
  }
}
```
GTFS-realtime Content and Structure

VehicleDescriptor

<table>
<thead>
<tr>
<th>Field</th>
<th>Required/Optional</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>optional</td>
<td>string</td>
</tr>
<tr>
<td>label</td>
<td>optional</td>
<td>string</td>
</tr>
<tr>
<td>license_plate</td>
<td>optional</td>
<td>string</td>
</tr>
</tbody>
</table>

vehicle {
  id: "y2294"
  label: "2294"
}

Source: MBTA
## GTFS-realtime Content and Structure

### Position

<table>
<thead>
<tr>
<th>Field</th>
<th>Required/Optional</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>latitude</td>
<td>required</td>
<td>float</td>
</tr>
<tr>
<td>longitude</td>
<td>required</td>
<td>float</td>
</tr>
<tr>
<td>bearing</td>
<td>optional</td>
<td>float (0-360)</td>
</tr>
<tr>
<td>odometer</td>
<td>optional</td>
<td>double</td>
</tr>
<tr>
<td>speed</td>
<td>optional</td>
<td>float</td>
</tr>
</tbody>
</table>

```json
position {
    latitude: 42.372686768
    longitude: -71.1155471802
    bearing: 220.0
}
```

Source: MBTA
GTFS-realtime Content and Structure

**Alert**

<table>
<thead>
<tr>
<th>Field</th>
<th>Required/Optional</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>active_period</td>
<td>repeated</td>
<td>msg TimeRange</td>
</tr>
<tr>
<td>informed_entity</td>
<td>repeated</td>
<td>msg EntitySelector</td>
</tr>
<tr>
<td>cause</td>
<td>optional</td>
<td>enum Cause</td>
</tr>
<tr>
<td>effect</td>
<td>optional</td>
<td>enum Effect</td>
</tr>
<tr>
<td>url</td>
<td>optional</td>
<td>msg TranslatedString</td>
</tr>
<tr>
<td>header_text</td>
<td>optional</td>
<td>msg TranslatedString</td>
</tr>
<tr>
<td>description_text</td>
<td>optional</td>
<td>msg TranslatedString</td>
</tr>
</tbody>
</table>
### GTFS-realtime Content and Structure

#### Alert

<table>
<thead>
<tr>
<th>enum Cause</th>
<th>enum Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNKNOWN_CAUSE</td>
<td>NO_SERVICE</td>
</tr>
<tr>
<td>OTHER_CAUSE</td>
<td>REDUCED_SERVICE</td>
</tr>
<tr>
<td>TECHNICAL PROBLEM</td>
<td>SIGNIFICANT_DELAYS</td>
</tr>
<tr>
<td>STRIKE</td>
<td>DETOUR</td>
</tr>
<tr>
<td>DEMONSTRATION</td>
<td>ADDITIONAL_SERVICE</td>
</tr>
<tr>
<td>ACCIDENT</td>
<td>MODIFIED_SERVICE</td>
</tr>
<tr>
<td>HOLIDAY</td>
<td>OTHER_EFFECT</td>
</tr>
<tr>
<td>WEATHER</td>
<td>UNKNOWN_EFFECT</td>
</tr>
<tr>
<td>MAINTENANCE</td>
<td>STOP_MOVED</td>
</tr>
<tr>
<td>CONSTRUCTION</td>
<td></td>
</tr>
<tr>
<td>POLICE_ACTIVITY</td>
<td></td>
</tr>
<tr>
<td>MEDICAL_EMERGENCY</td>
<td></td>
</tr>
</tbody>
</table>
Alert

```json
alert {
    active_period {
        start: 1464020433
        end: 1464035632
    }
    informed_entity {
        agency_id: "1"
        route_id: "36"
        route_type: 3
    }
    cause: TECHNICAL_PROBLEM
    effect: OTHER_EFFECT
    header_text {
        translation {
            text: "Moderate Route 36 delay"
            language: "en"
        }
    }
    description_text {
        translation {
            text: "Route 36 experiencing moderate delays due to disabled bus."
            language: "en"
        }
    }
}
```

Source: MBTA
GTFS-realtime Content and Structure

TimeRange

<table>
<thead>
<tr>
<th>Field</th>
<th>Required/Optional</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>start</td>
<td>optional</td>
<td>uint64</td>
</tr>
<tr>
<td>end</td>
<td>optional</td>
<td>uint64</td>
</tr>
</tbody>
</table>

```javascript
active_period {
    start: 1464020433
    end: 1464035632
}

active_period {
    start: 1464020433
}

active_period {
    end: 1464035632
}
```

Source: MBTA
## GTFS-realtime Content and Structure

### EntitySelector

<table>
<thead>
<tr>
<th>Field</th>
<th>Required/Optional</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>agency_id*</td>
<td>optional</td>
<td>string</td>
</tr>
<tr>
<td>route_id*</td>
<td>optional</td>
<td>string</td>
</tr>
<tr>
<td>route_type*</td>
<td>optional</td>
<td>integer</td>
</tr>
<tr>
<td>trip</td>
<td>optional</td>
<td>msg TripDescriptor</td>
</tr>
<tr>
<td>stop_id*</td>
<td>optional</td>
<td>string</td>
</tr>
</tbody>
</table>

Values in fields with asterisk (*) should match corresponding GTFS feed exactly.

```python
informed_entity {
    agency_id: "1"
    route_id: "36"
    route_type: 3
}
```

Source: MBTA
# GTFS-realtime Content and Structure

## TranslatedString

<table>
<thead>
<tr>
<th>Field</th>
<th>Required/Optional</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>translation</td>
<td>repeated</td>
<td>msg Translation</td>
</tr>
</tbody>
</table>

```json
description_text {
  translation {
    text: "Route 36 experiencing moderate delays due to disabled bus."
    language: "en"
  }
}
```

*Source: MBTA*
**GTFS-realtime Content and Structure**

## Translation

<table>
<thead>
<tr>
<th>Field</th>
<th>Required/Optional</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>text</td>
<td>required</td>
<td>string</td>
</tr>
<tr>
<td>language**</td>
<td>optional</td>
<td>string</td>
</tr>
</tbody>
</table>


```json
translation {
  text: "Route 43 experiencing minor delays due to traffic."
  language: "en"
}
```

*Source: MBTA*
Translating Source Application Data to GTFS-realtime files

Data Translation Process

1. Evaluate
2. Export
3. Translate
4. Reference

GTFS
Related Tools for GTFS-realtime

Tools

- No commonly used stand-alone off-the-shelf tools exist
- Open source tools can be integrated with other software
- CAD/AVL systems can include necessary software
- Custom software may be required

```python
if field == "route_id": # tests to ensure a route_id appears in routes.txt and that it matches the reference trip_id
    if value in routesLookup: # if the route_id is in routes.txt:
        if key == "": # if no trip_id, can't verify the trip-route match
            warn = True
            refOk = True
            comment = "WARNING: route_id %s exists in routes.txt, but no corresponding trips_id to check against. Entity: %s" % (value,entID)
        elif tripsLookup.has_key(key) == False: # if the trip_id does not exist in trips.txt, can't verify trip_route match
            error = True
            refOk = False
            comment = "ERROR. trip_id %s does not exist. Therefore, route_id %s can not be verified. Entity: %s" %(key,value,entID)
```
Which of the following formats is used for encoding a GTFS-realtime feed?

**Answer Choices**

a) Extensible Markup Language (XML)
b) JavaScript Object Notation (JSON)
c) Protocol Buffers
d) Comma Separated Values (CSV)
Review of Answers

a) Extensible Markup Language (XML)

*Incorrect. GTFS-realtime files are not encoded in XML.*

b) Javascript Object Notation (JSON)

*Incorrect. GTFS-realtime files are not encoded in JSON.*

c) Protocol Buffers

*Correct! Protocol Buffers are the format used for GTFS-realtime.*

d) Comma Separated Values (CSV)

*Incorrect. GTFS-realtime files are not encoded in CSV.*
Which of the following is NOT a way to show location in GTFS-realtime

Answer Choices

a) Latitude/Longitude
b) Stop sequence on a trip
c) Stop identifier
d) Distance to destination
a) Latitude and Longitude

Incorrect. The fields latitude and longitude are used for this.

b) Stop sequence on a trip

Incorrect. The field current_stop_sequence is used for this.

c) Stop identifier

Incorrect. The field stop_id is used for this.

d) Distance to destination

Correct! There is no field for this in GTFS-realtime.
Learning Objective 3

Implementation of GTFS-realtime
Testing GTFS-realtime Files and GTFS-realtime Validation Tools

- **Why is testing important?**
  - Need to ensure high quality data

- **Types of Testing**
  - Specification Conformance
  - Is the data Accurate?
  - Appropriate references

Source: Metro North Railroad
Testing GTFS-realtime Files and GTFS-realtime Validation Tools

- No off-the-shelf GTFS-realtime testing tools, some open source tools in development
- Testing typically built into production tools or consumption tools
- Google’s open source software modules provide rudimentary testing

```
record: 533 >> WARNING: optional field uncertainty IS NOT present. Entity: T1009
record: 537 >> WARNING: optional field uncertainty IS NOT present. Entity: T1009
record: 546 >> WARNING: optional field uncertainty IS NOT present. Entity: T1009
record: 550 >> WARNING: optional field uncertainty IS NOT present. Entity: T1009
record: 559 >> WARNING: optional field uncertainty IS NOT present. Entity: T1009
record: 563 >> WARNING: optional field uncertainty IS NOT present. Entity: T1009
record: 572 >> WARNING: optional field uncertainty IS NOT present. Entity: T1009
record: 576 >> WARNING: optional field uncertainty IS NOT present. Entity: T1009
record: 585 >> WARNING: optional field uncertainty IS NOT present. Entity: T1009
record: 589 >> WARNING: optional field uncertainty IS NOT present. Entity: T1009
record: 592 >> WARNING: optional field delay IS NOT present. Entity: T1009
record: 606 >> WARNING: optional field vehicle IS NOT present. Entity: V1009
record: 607 >> WARNING: optional field position IS NOT present. Entity: V1009
record: 613 >> WARNING: optional field congestion_level IS NOT present. Entity: V1009
record: 614 >> WARNING: optional field occupancy_status IS NOT present. Entity: V1009
record: 629 >> WARNING: optional field vehicle IS NOT present. Entity: T1011
record: 637 >> WARNING: optional field uncertainty IS NOT present. Entity: T1011
record: 641 >> WARNING: optional field uncertainty IS NOT present. Entity: T1011
record: 650 >> WARNING: optional field uncertainty IS NOT present. Entity: T1011
record: 654 >> WARNING: optional field uncertainty IS NOT present. Entity: T1011
record: 663 >> WARNING: optional field uncertainty IS NOT present. Entity: T1011
record: 667 >> WARNING: optional field uncertainty IS NOT present. Entity: T1011
```
How an Agency Implements GTFS-realtime

Implementation Checklist

- Determine need for real time information
- Inventory existing systems and data
- Determine feasibility
- Procure and/or Develop
- Produce and provide data
How an Agency Implements GTFS-realtime

Should GTFS-realtime be implemented?

- Established need: Identify usage
  - Will customers benefit from real-time information?
  - Will archiving real-time information be useful?
  - Do downstream applications exist?

- Determine feasibility
  - Can existing systems be leveraged?
  - Does funding exist for procurement if needed?
  - Can a system that produces GTFS-realtime feeds be maintained?

- Is GTFS-realtime the correct format to use?
How an Agency Implements GTFS-realtime

Procurement

▪ Use Systems Engineering Process
▪ User needs
  ▫ *Ex. Downstream users need to receive trip update information*
▪ Functional Requirements
  ▫ *Ex. The system shall export a GTFS-realtime feed containing the TripUpdate, VehiclePosition, and Alert messages*
▪ Performance Requirements
  ▫ *Ex. The system shall provide an updated GTFS-realtime feed every 30 seconds*
How an Agency Implements GTFS-realtime

Data Lifecycle Requirements and Strategies

1. Collect Data
2. Merge Data
3. Export Data
4. Provide Data

Repeat at defined frequency (e.g. every 30 seconds)

Test Data
How an Agency Implements GTFS-realtime

Making Data Accessible

- Fixed location
- Agencies define Application Programming Interface (API) for requests
- Typically HTTP request
- Access often must be granted
- Need to make static GTFS accessible
Agency Case Study

TRANSCOM

- TRANSCOM (NY/NJ/CT Metro Region)
- 16 member agencies
- Consolidating real time transit data
- Providing GTFS-realtime
- Data provided to downstream users
- Developing transit dashboard
Agency Case Study

TRANS.COM

Source: TRANS.COM /ConSysTec
Agency Case Study

TRANSCOM

- Feeds
  - NJ Transit Rail (Year 1)- Custom XML
  - NJ Transit Bus (Year 2)- Custom XML
  - MTA NYCT Subway (Year 2)- GTFS-realtime
  - MTA Long Island Rail Road (Year 2)- GTFS-realtime
  - MTA Metro North Railroad (Year 2)- GTFS-realtime
  - MTA NYCT Bus (Year 3)- SIRI

- Lessons learned
  - Document the meaning of the data
  - Understand the meaning of data when integrating multiple sources
ACTIVITY
Which of the following methods is most commonly used to access GTFS-realtime feeds?

Answer Choices

a) Email
b) FTP
c) HTTP
d) Telephone
Review of Answers

a) Email
   Incorrect. Email is too slow.

b) FTP
   Incorrect. FTP is not typically used.

c) HTTP
   Correct! HTTP is commonly used to share GTFS-realtime feeds.

d) Telephone
   Incorrect. This is not an electronic format.
Module Summary

What We Have Learned

1. The **background** of the GTFS-realtime specification, its **benefits** and its **uses**.

2. How GTFS-realtime feeds are **structured**, the **content** of the feeds, and how the feeds are **created**.

3. How agencies **test and implement** GTFS-realtime feeds.

This module taught us about the GTFS-realtime specification and how it can be used.
Thank you for completing this module.

Feedback

Please use the Feedback link below to provide us with your thoughts and comments about the value of the training.

Thank you!