Overview of GTFS-Realtime and Comparison to TSIP-RTDP

December 20, 2011

Prepared for:

New York State Department of Transportation (NYSDOT)

Prepared by:

Scott Altman
Consensus Systems Technologies

Contents

1 Introduction

The General Transit Feed Specification (GTFS) is a specification released and maintained by Google which defines a standard format for maintaining public transit schedule data. Data in this format is used by the users of Google Maps for transit trip planning, as well as with third party applications. GTFS data is widely used by many transit agencies throughout the world to relay transit data. The GTFS-realtime feed is an extension of the GTFS feed designed for relaying up to the minute transit information to consumers. GTFS-realtime relays three categories of information to users: trip updates, service alerts, and vehicle positions. The GTFS-realtime feed is a specification for transporting data; it is not a prediction engine. GTFS-realtime stores data in Google's Protocol Buffer (.pb) format. At this time GTFS-realtime is not very widely used. GTFS-realtime is currently in version 1.0.

The Real Time Status and Data Profile (RTDP) is part of the New York State Department of Transportation (NYSDOT) Transit Service Information Portal (TSIP) project. The RTDP is one of four data profiles that make up TSIP, the others being the Schedule Data Profile (SDP), the Planning Data Profile (PDP), and the Fare Calculator Data Profile (FCDP). The SDP is analogous to the GTFS, and is currently used by the Transit Trip Planner of 511NY, NYSDOT's 5-1-1 system. The RTDP is analogous to GTFS-realtime, but is not yet in regular use. Like GTFS-realtime, the RTDP is not a prediction engine.

This document will describe the GTFS-realtime specification, and compare it to the RTDP in terms of architecture, data elements, API's/interfaces, technology, and translation between the two formats.

2 Description of GTFS-realtime Feed

2.1 Overview

The GTFS-realtime feed specification is designed to transfer transit status data in real time to end users. There are three different feed types, each relaying a different type of information. The Trip Updates feed type is for conveying deviations from the schedule for a specific trip. The Service Alerts feed type is for conveying larger scale disruptions, such as delays on all busses on a specific route, or a cancellation of all services for a specific agency. Finally the Vehicle Positions feed type conveys data about the specific location of a transit vehicle at a given time. A specific feed can only contain one feed type (e.g. a trip update and vehicle location would have to be disseminated in separate files).

2.2 History

The GTFS-realtime specifications were developed as an extension of the GTFS specification, which has existed since 2006. GTFS-realtime was released in August, 2011, and developed by Google and a consortium that included the transit agencies in San Francisco (BART), Boston (MBTA), Madrid (Metro de Madrid), Portland (Trimet), San Diego (SDMTS), and Torino (GTT).

2.3 Details of Feed Types

2.3.1 Trip Updates

The Trip Updates feed type is for relaying deviations from a previously released schedule. This feed can be used to relay stop delay information, additions to or subtractions from the schedule, or that a transit trip is on time (no deviation). The specification states that a trip update should only occur once per trip, though it does not explicitly prohibit additional updates. It also states that the lack of a trip update feed for a trip does not indicate that service is on time, but rather that the status is unknown.

The Trip Updates feed provides a flexible mechanism for updating the data for a trip. The feed states that stop time updates should be listed sequentially in the feed. A deviation from the schedule may be in terms of an exact arrival or departure time, or a delay. When one stop time is updated, all subsequent stops are updated with the same delay, unless a new delay is specified. For example, if the first stop is listed as delayed by 2 minutes, and the fourth stop is listed as delayed by 5 minutes, the second stop and third stops are assumed to be delayed by 2 minutes, and the fifth stop is assumed to be delayed by 5 minutes. A range of uncertainty can also be included in the stop prediction.

2.3.2 Service Alerts

The Service Alerts feed type is used to classify service issues on a broader scale than trip updates. The Service Alerts feed describes updates that affect an entire agency, route, route type, trip, or specific stop. For each entity requiring a service alert, the cause and effect can be described. For example, a service alert can be issued for a specific route where construction is causing service cancellations. A detailed description, or URL to a description, can be provided to users to describe the problem.

2.3.3 Vehicle Positions

The Vehicle Position feed type is used to classify specific details about a vehicle currently in service. This feed is not intended to provide updates to a trip, like the Trip Updates Feed. This feed provides position information (including location, speed, and direction), congestion level, which is traffic status information, vehicle stop status, which provides vehicle location relative to the nearest or current stop, and vehicle description information, which physically describes a vehicle. A timestamp is included in this feed to reference the time when the data was collected. The position relative to a stop can be referenced by either a GTFS stop_id, or the stop sequence of the current trip.

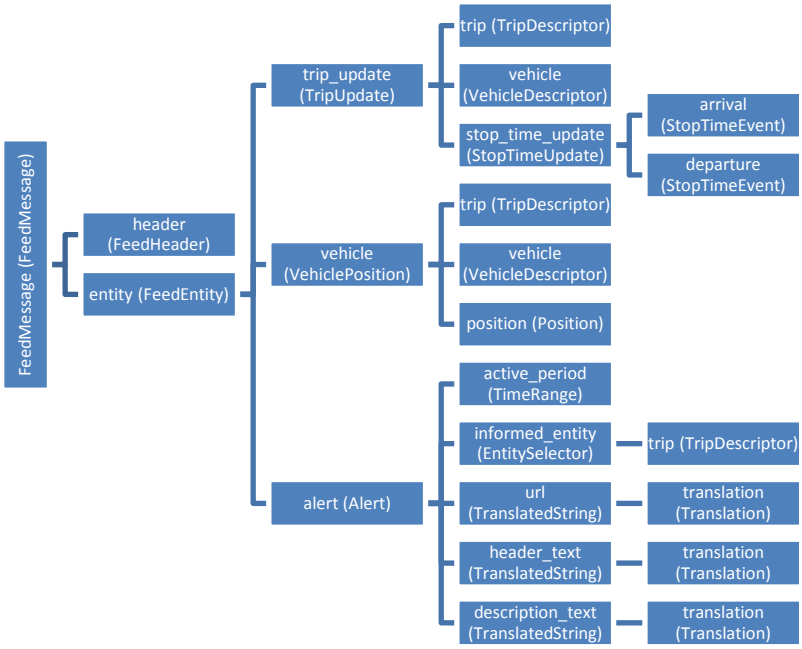
2.4 Feed Structure

2.4.1 Description

The GTFS-realtime feed is structured is a hierarchal feed type, structured in Google's Protocol Buffer format. The feed defines two types of elements: *message* and *enum.* Message describes an entity within a GTFS-realtime feed that is a specific structure of data, similar to a complex-type in an XML schema. A message contains multiple fields, which may be a data unit (a string), or another message. *Enum* describes a type of field that may be in a message, similar to an enumerated code in an XML schema. An enum has several discreet options that may be used to fill in a field.

2.4.2 High Level Structure

The chart below shows the high level structure of the GTFS-realtime feed. Each entity shows the name of the entity, and the corresponding *message* type in parenthesis.



More detailed structure definitions can be found in Appendix A.

2.5 Technology

2.5.1 Feed Format (Protocol Buffers)

GTFS-realtime data is stored in Google's Protocol Buffer format (.pb files). The protocol buffer language is similar to XML in terms of its hierarchal nature. Protocol buffers use curly braces ({}) instead of tags (<XXX>). Similar to an XML schema (.xsd file), a protocol buffer definition file (.proto file) can be made to define the necessary structure of the data file. GTFS-realtime uses a definition file entitled gtfs-realtime.proto. With the definition file, a class structure can then be defined in the programming language being used to generate data. At the present time, only C++, Java, and Python are officially supported by Google for this purpose. However, outside developers have projects in progress to make protocol buffers work with other programming languages. Since Protocol Buffers are an open format, there is no official barrier to a developer using the format with an unsupported language. See the example below of a simple protocol buffer message file and a comparable XML file.

Protocol Buffer Example

```
StateList{
        State{
                name: “New Jersey”
                abbreviation: “NJ”
                capital: “Trenton”
        }
        State{
                name: “New York”
                abbreviation: “NY”
                capital: “Albany”
        }
}
```

XML Example

```
<StateList>
        <State>
                <name>New Jersey</name>
                <abbreviation>NJ</abbreviation>
                <capital>Trenton</Trenton>
        </State>
        <State>
                <name>New York</name>
                <abbreviation>NY</abbreviation>
                <capital>Albany</capital>
        </State>
</StateList>
```

2.5.2 Feed Transfer

Currently, the GTFS-realtime specification does not define transfer protocols for the GTFS-realtime feed. GTFS-realtime data is transferred via HTTP. The transit agency providing the feed is responsible for either setting the feed to automatically POST, or for setting up a URL for submitting a GET request. No API is defined for requesting data as part of the specification.

2.6 Tools

The primary use of GTFS-realtime is with Google Maps. From the Google Maps interface, public transit riders can view realtime transit information from participating agencies based on the provided GTFS-realtime feeds.

Since the specification for GTFS-realtime is an open specification, it is easy for developers to develop applications and tools using a GTFS-realtime feed. Several transit agencies using GTFS-realtime, such as BART, make the API for accessing their feed available to developers. An open source development project known as the Public Transport Engine has included the ability to read GTFS-realtime feeds.

3 Comparison to RTDP

3.1 Architecture Differences

GTFS-realtime and RTDP differ in terms of architecture definition. The RTDP requirements provide a detailed definition of the necessary architecture systems and flows. There are three information flows described by the RTDP: Schedule and Daily Changes Flow, Current Locations and Disruptions Flow, and Customer Access and API Flow.

In contrast to this, architecture and information flows are mostly undefined by the GTFS-realtime specifications. The specifications merely imply a direct flow of data between the transit agency, or data provider, and the application requesting the data, or data user. While the GTFS-realtime feed accounts

for schedule changes, vehicle locations, and API's, there is no direct definition of how this information flows. It is known, however, that the GTFS-realtime feed distinguishes between schedule updates, alerts, and vehicle positions, and requires separate feeds for each of these.

3.2 Data Element Differences

In general, the data elements of both the RTDP and GTFS-realtime specifications are designed with the intent of conveying similar information. There are, however, some differences between the two formats. One difference is that the GTFS-realtime feed is designed to work in tandem with a specific GTFS schedule feed. The RTDP, while based on a specific SDP feed, already contains the relevant schedule data within its structure.

GTFS-realtime and RTDP both have a slightly different structure. RTDP contains structures for Bus Location, Rail Location, Prediction Results, Travel Time Results, Agency information, Service Information, Block Information, Route Information, and Errors. This is compared to the GTFS structure, which contains only three major structure types: Service Updates, Alerts, and Vehicle Position. Additionally, there are structures for requesting information defined in the XML schema, something not defined by GTFS-realtime. Due to the differences in structure, the same information may be placed differently in the two feeds. For example, in the RTDP, service notices can be provided in the Bus and Rail location sections, whereas GTFS-realtime provides this information in a separate alert structure.

Some elements of the RTDP do not exist at all in GTFS-realtime. One example of this is travel time information. While not explicitly in GTFS-realtime, this data could be derived by querying prediction data that does exist. The GTFS-realtime specification also does not directly carry the ability to support next arrival data for a specific stop. This data would have to also be analyzed by looking at the predicted stop times for all scheduled trips at a given stop.

Two important elements of comparing the feeds involve comparing location and time data. Location is expressed the same way between the two, using latitude and longitude. Time is specified differently between the two feeds. RTDP specifies time in terms of seconds past midnight on a given date. In GTFS-realtime, time is specified in seconds past midnight on January 1, 1970. This eliminates the need to include fields for date. Note that GTFS schedule data (non realtime) does utilize separate date and time fields.

See Appendix B for a direct comparison of bus and real location data elements of the two feeds

3.3 API/Interfaces

In the RTDP, Application Programming Interfaces (API's) are defined with great detail. The RTDP requirements specify several different types of requests to be supported. Each request, and the corresponding data that is returned, is described and defined in the RTDP requirements.

The GTFS-realtime specifications do not define the associated API's with the feed. The definition of API's is left to the transit agencies that are providing the data. In their guidelines for transit agencies, Google describes only that transit agencies should either make their feed available to be "pulled" using a GET

request, as defined by HTTP, or "pushed", using the HTTP defined POST request to upload the feed to a user. The structures of the specific API's are not defined any further.

3.4 Technology Differences

3.4.1 Storage Structure

The main technological difference between the GTFS-realtime feed and the RTDP is in the storage of data. The RTDP feed is stored in an XML file. The structure of this XML file is defined by the RTDP schema, which includes the SDP schema as well. The use of an XML schema provides a hierarchal structure for the RTDP.

The GTFS-realtime specification defines that data be stored in a Protocol Buffer file. This is an open storage method designed by Google. The Protocol Buffer format is similar to XML in that it is also hierarchal in structure. It does not include the use of opening and closing tags, but rather curly braces. Protocol Buffers and XML are similar enough that conversion between the two is likely to be possible.

3.4.2 Feed Transfer

The RTDP defines the transfer of data in terms of the Web Service Definition Language (WSDL) standard, with the use of Hypertext Transfer Protocol (HTTP) and Representative State Transfer (REST). The GTFS-realtime specification makes no reference to WSDL, or REST specifically, but does refer to the use of HTTP. Both specifications specifically mention the HTTP functions of GET and POST.

3.5 Ease of Translation

Since many of the data elements in these two feed specifications can be directly mapped to each other, a translation between the two feed types would be possible, with some difficulties. In order for this to be done, static schedule data (e.g. SDP and GTFS feeds) would need to be pre converted, since these are time consuming processes, and GTFS-realtime feeds do not already contain static schedule data. There are also some entities that are contained in the RTDP specification, but not the GTFS-realtime specification. Examples of this are travel time and next service predictions. If moving from GTFS-realtime to RTDP, a process would need to exist to query the appropriate data from the GTFS-realtime feed.

A more feasible option than direct translation would be to make both feeds accessible separately. In general, the same sort of information is conveyed between the two specifications, just with a different structure. For example, assume a train is delayed five minutes due to construction. It would be straightforward for a transit agency's prediction engine to separately push out a trip update with this delay in GTFS-realtime and respond to a next service prediction request in RTDP for a given stop on that train's route. A prediction engine would use the same input to output in two different formats.

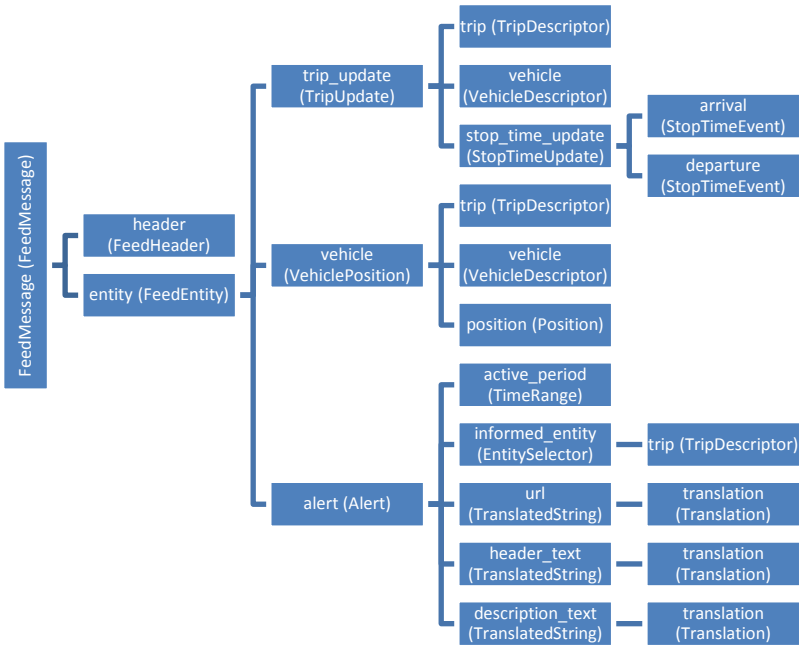
Since the GTFS-realtime specification does not define API requests, translation of requests would not be possible.

4 Conclusion

The RTDP and GTFS-realtime formats are structured quite differently. A key high level difference is in the simplicity of the GTFS-realtime format when compared to the complexity of the RTDP format. While GTFS-realtime is designed to utilize fewer computing resources, RTDP allows for a more flexible data transfer. The requirements of RTDP also define the necessary transfer protocols, which GTFS-realtime does not define. Both formats are similar in that they define the necessary data structures to convey real time transit information. The complex nature of the RTDP, however, allows that format to more fully display real time transit information.

Appendix A: Detailed Feed Structure.

Hierarchy



Entity Descriptions

Message Types

message FeedMessage

Field Name	Type	Cardinality	Description
header	*message* FeedHeader	Required	Metadata about this feed and feed message
entity	*message* FeedEntity	Required	Contents of the feed

message FeedHeader

Field Name	Type	Cardinality	Description
gtfs_realtime_version	string	required	Version of the feed specification
incrementality	*enum* incrementality	optional	Determines whether the current fetch is incremental
timestamp	int	optional	This timestamp identifies the moment when the content of this feed has been created

message FeedEntity

Field Name	Type	Cardinality	Description
id	string	required	Feed-unique identifier for this entity
is_deleted	bool	optional	Whether this entity is to be deleted
trip_update	*message* TripUpdate	optional	Data about the realtime departure delays of a trip
vehicle	*message* VehiclePosition	optional	Data about the realtime position of a vehicle
alert	*message* Alert	optional	Data about the realtime alert

message TripUpdate

Field Name	Type	Cardinality	Description
trip	*message*TripDescriptor	required	The Trip that this message applies to
vehicle	*message* VehicleDescriptor	optional	Additional information on the vehicle that is serving this trip
stop_time_update	*message* StopTimeUpdate	repeated	Updates to StopTimes for the trip

message StopTimeEvent

Field Name	Type	Cardinality	Description
delay	int	optional	Delay (in seconds)
time	int	optional	Event as absolute time
uncertainty	int	optional	Roughly specifies the expected error in true delay

message StopTimeUpdate

Field Name	Type	Cardinality	Description
stop_sequence	int	optional	Same as stop_sequence in stop_times.txt in the corresponding GTFS feed
stop_id	string	optional	Same as stop_id in stops.txt in the corresponding GTFS feed
arrival	*message* StopTimeEvent	optional	Timing information for a single predicted arrival event
departure	*message* StopTimeEvent	optional	Timing information for a single predicted departure event
schedule_relationship	*enum* ScheduleRelationship	optional	The relation between this StopTime and the static schedule

message VehiclePosition

Field Name	Type	Cardinality	Description
trip	*message* TripDescriptor	optional	The Trip that this vehicle is serving
vehicle	*message* VehicleDescriptor	optional	Additional information on the vehicle that is serving this trip
position	*message* Position	optional	Current position of this vehicle
current_stop_sequence	int	optional	The stop sequence index of the current stop
stop_id	string	optional	Identifies the current stop
current_status	*enum* VehicleStopStatus	optional	he exact status of the vehicle with respect to the current stop
timestamp	int	optional	Moment at which the vehicle's position was measured
congestion_level	*enum* CongestionLevel	optional	Congestion level that is affecting this vehicle

message Alert

Field Name	Type	Cardinality	Description
active_period	*message* TimeRange	repeated	Time when the alert should be shown to the user
informed_entity	*message* EntitySelector	repeated	Time when the alert should be shown to the user
cause	*enum* Cause	optional	Cause of this alert
effect	*enum* Effect	optional	The effect of this problem on the affected entity
url	*message* TranslationString	optional	The URL which provides additional information about the alert
header_text	*message* TranslationString	optional	Header for the alert
description_text	*message* TranslationString	optional	Description for the alert

message TimeRange

Field Name	Type	Cardinality	Description
start	int	optional	Start time
end	int	optional	End time

message Position

Field Name	Type	Cardinality	Description
latitude	float	required	Degrees North
longitude	float	required	Degrees East
bearing	float	optional	Bearing, in degrees, clockwise from True North
odometer	double	optional	Odometer value, in meters
speed	float	optional	Momentary speed measured by the vehicle, in meters per second

message TripDescriptor

Field Name	Type	Cardinality	Description
trip_id	string	optional	The trip_id from the GTFS feed that this selector refers to
route_id	string	optional	The route_id from the GTFS that this selector refers to
start_time	string	optional	The scheduled start time of this trip instance.
start_date	string	optional	The scheduled start date of this trip instance
schedule_relationship	*enum* ScheduleRelationship	optional	The relation between this trip and the static schedule

message VehicleDescriptor

Field Name	Type	Cardinality	Description
id	string	optional	Internal system identification of the vehicle
label	string	optional	User visible label
license_plate	string	optional	The license plate of the vehicle

message EntitySelector

Field Name	Type	Cardinality	Description
agency_id	string	optional	agency_id from GTFS (non realtime)
route_id	string	optional	route_id from GTFS (non realtime)
route_type	int	optional	route_type from GTFS (non realtime)
trip	*message* TripDescriptor	optional	A descriptor that identifies an instance of a GTFS trip, or all instances of a trip along a route
stop_id	string	optional	stop_id from GTFS (non realtime)

message TranslatedString

Field Name	Type	Cardinality	Description
translation	*message* Translation	repeated	A localized string mapped to a language

message Translation

Field Name	Type	Cardinality	Description
text	string	required	A UTF-8 string containing the message
language	string	optional	BCP-47 language code

Enum Types

enum Incrementality

Value	Description
FULL_DATASET	This feed update will overwrite all preceding realtime information for the feed
DIFFERENTIAL	Currently, this mode is unsupported and behavior is unspecified for feeds that use this mode

enum ScheduleRelationship (refers to ScheduleRelationsip used in *message* StopTimeUpdate)

Value	Description
SCHEDULED	The vehicle is proceeding in accordance with its static schedule of stops, although not necessarily according to the times of the schedule
SKIPPED	The stop is skipped
NO_DATA	No data is given for this stop

enum VehicleStopStatus

Value	Description
INCOMING_AT	The vehicle is just about to arrive at the stop
STOPPED_AT	The vehicle is standing at the stop
IN_TRANSIT_TO	The vehicle has departed the previous stop and is in transit

enum CongestionLevel

Value
UNKNOWN_CONGESTION_LEVEL
RUNNING_SMOOTHLY
STOP_AND_GO
CONGESTION
SEVERE_CONGESTION

enum Cause

Value
UNKNOWN_CAUSE
OTHER_CAUSE
TECHNICAL_PROBLEM
STRIKE
DEMONTRATION
ACCIDENT
HOLIDY
WEATHER
MAINTENANCE
CONSTRUCTION
POLICE_ACTIVITY
MEDICAL_EMERGENCY

enum Effect

Value
NO_SERVICE
REDUCED_SERVICE
SIGNIFICANT_DELAYS
DETOUR
ADDITIONAL_SERVICE
MODIFIED_SERVICE
OTHER_EFFECT
STOP_MOVED

enum ScheduleRelationship (refers to ScheduleRelationship used in *message* TripDescriptor)

Value	Description
SCHEDULED	Trip that is running in accordance with its GTFS schedule, or is close enough to the scheduled trip to be associated with it
ADDED	An extra trip that was added in addition to a running schedule, for example, to replace a broken vehicle or to respond to sudden passenger load
UNSCHEDULED	A trip that is running with no schedule associated to it, for example, if there is no schedule at all
CANCELED	A trip that existed in the schedule but was removed
REPLACEMENT	A trip that replaces a portion of the static schedule

The official detailed descriptions of each entity can be found at http://code.google.com/transit/realtime/docs/gtfs-realtime-reference.html.

Appendix B: Direct Data Comparison

RTDP Bus Location/Status compared to GTFS-Realtime

RTDP Element	GTFS-Realtime Element	Comments (if applicable)
r-time	timestamp	GTFS-RT Uses timestamp here to mark when data was taken
a-time	timestamp	GTFS-RT uses timestamp here to mark when data was processed on server
agencyID	agency_id	
vehicleID	id	
latitude	latitude	
longitude	longitude	
blockID	block_id	In GTFS (non realtime)
routeID	route_id	
routeDirection		direction_id can be taken from trips.txt GTFS file
tripID	trip_id	
lastStopID		not explicitly in GTFS-Realtime feed, can be interpreted from combination of stop_id or current_stop_sequence and current_status
departed	current_status	
nextStopID		not explicitly in feed, can be interpreted from combination of stop_id or current_stop_sequence and current_status
serviceStatus		Handled by Alert or Trip_Update feed in GTFS-realtime
headsign		Taken from headsign in trips.txt GTFS file.
scheduleVersionID		scheduleVersionID not in GTFS
revision		revision not in GTFS
routeDepotVersion		not accounted for in GTFS
publicNotice		handled by message Alert
avgSpeed	(speed)	not explicitly in feed, can be calculated by average speed values
distanceTraveled	(odometer)	not explicitly in feed, can be calculated by comparing odometer values
Health		Not accounted for in GTFS
Quality		Not accounted for in GTFS

RTDP Rail Location/Status compared to GTFS-Realtime

Rail Location/Status Data	GTFS-Realtime Element	Comments (if applicable)
r-time	timestamp	GTFS-RT Uses timestamp here to mark when data was taken
a-time	timestamp	GTFS-RT uses timestamp here to mark when data was processed on server

agencyID	agency_id	
vehicleID	id	
latitude	latitude	
longitude	longitude	
routeID	route_id	
tripID	trip_id	
lastStopID		not explicitly in feed, can be interpreted from combination of stop_id or current_stop_sequence and current_status
departed	current_status	
nextStopID		not explicitly in feed, can be interpreted from combination of stop_id or current_stop_sequence and current_status
nextScheduleStopTime		handled by Trip_Update feed
serviceStatus		Handled by message Alert feed in GTFS-realtime
headsign		Not explicitly in GTFS-realtime, taken from headsign in trips.txt GTFS file.
consistSize		consist size not accounted for in GTFS
scheduleVersionID		scheduleVersionID not in GTFS
revision		revision not in GTFS
routeDepotVersion		not accounted for in GTFS
publicNotice		handled by Alert feed
detour	(enum ScheduleRelationship)	not a direct translation
agencyProvidingService		not explicitly specified, but handled by Alert feed
detour-revision		not explicitly specified
delayTimeLo	delay - uncertainity	Requires use of two GTFS-realtime elements
delayTimeHi	delay + uncertainty	Requires use of two GTFS-realtime elements
Health		not in GTFS-realtime
Quality		not in GTFS-realtime

Appendix C: Acronyms

511NY	New York's 5-1-1 System
API	Application Programming Interface
bool	Boolean (true/false) value
FCDP	Fare Calculator Data Profile
GTFS	General Transit Feed Specification
HTTP	Hypertext Transfer Protocol
int	integer
NYSDOT	New York State Department of Transportation
PB	Protocol Buffer
PDP	Planning Data Profile
REST	Representational State Transfer
RTDP	Real Time and Status Data Profile
SDP	Schedule Data Profile
TSIP	Transit Service Information Portal
WSDL	Web Services Description Language
XML	Extensible Markup Language

Appendix D: References

General Transit Feed Specification http://code.google.com/transit/spec/transit_feed_specification.html

GTFS-realtime Specification http://code.google.com/transit/realtime/docs/introduction.html

Protocol Buffers http://code.google.com/apis/protocolbuffers/

Public Transport Data Engine http://publictransport.horizon-host.com/doc/engine/branch-gtfs/index.html

Real time BART information now available on Google Maps
http://www.bart.gov/news/articles/2011/news20110608.aspx

TSIP Real Time and Service Status Information Requirements
http://ngtsip.pbworks.com/w/page/12503408/Real%20Time%20and%20Service%20Information