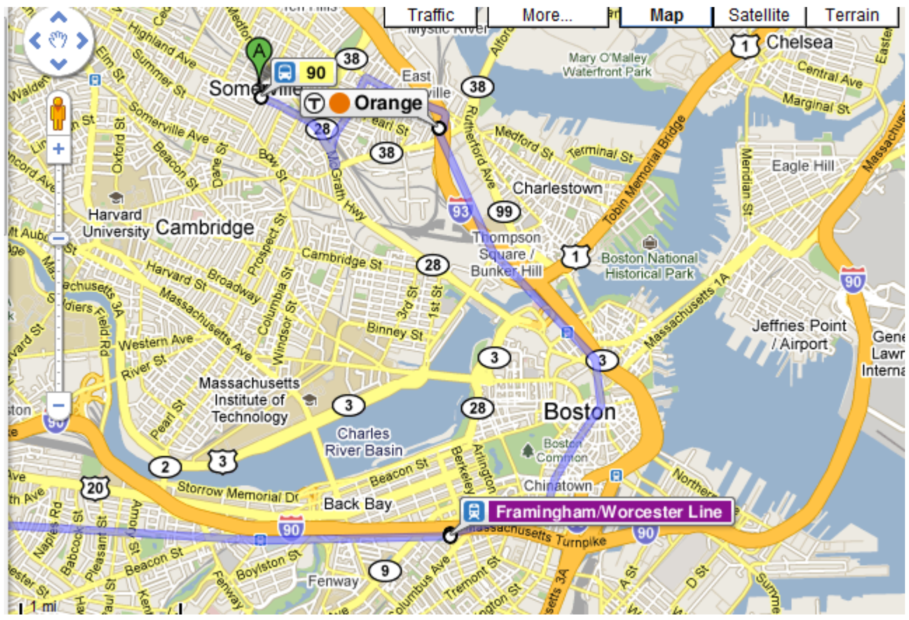
Google Transit Feed Specification: A Primer



Daniel Moraff
Massachusetts Department of Transportation
November 14, 2009
dmorafftransit@gmail.com

Contents

Working with the Google Transit Feed Specification: the basics

Resources

- http://code.google.com/transit/spec/transit_feed_specification.html (the specification)
- http://code.google.com/p/googletransitdatafeed/downloads/list (feedvalidator, Schedule Viewer, and other tools)
- http://groups.google.com/group/googletransit/files (more tools for working with GTFS, including Bob Heitzman's invaluable XLS tools)
- http://earth.google.com/

Getting Started

1. Make sure you have Izarc or some other tool to handle .zip files.
2. Extract the .zip file into a folder.
3. From Bob Heitzman's XLS tools, open GTFSImportExport, and save it into that same folder.
4. Using this self-explanatory excel document, import your feed into Excel. When you are done with your feed in Excel, you can use this tool again to export the data into .txt files.

The Structure of the Feed

This is an overview of how a Google Transit feed works. It will not explain every single aspect, and will leave out the more self-explanatory or unnecessary ones—see the specification for more details. It is intended for agencies which already have a completed field, and wish to learn how to maintain and work with it themselves. Some fields in the feed are not portrayed at all—these should be left blank. Some intuitive parts of the feed or those which are explained very well in the Google Transit Feed Specification (such as agencies.txt) are not explained here.

The Google Transit Feed is a .zip file composed of 7-12 .txt files containing multiple comma-separated fields, depending on the agency. The structure of all of these files is laid out in the specification, but the most important file is **trips.txt**. In this file, each trip made in a given agency (i.e., the #12 bus departing outbound on weekdays at 10:00 AM) has its own entry. It is useful to think of the entire feed as being based around the trips.txt file.

For example, a line in trips depicting the Silver Line leaving at 8:00 towards Logan Airport on weekdays will look something like this:

route_id,service_id,trip_id,trip_headsign,shape_id
741_-1058,Bus-Weekday,11462304,Logan Airport,SL1OUT

--route_id: This defines what route, exactly, the trip is running on. It refers to the **routes.txt** file. A sample line:

route_id,agency_id,route_long_name,route_desc,route_type,
741_-1058,MBTA,Silver Line SL1,,3

Google Transit will know that this trip runs on route 741_-1058, which refers to the "Silver Line SL1", run by the agency MBTA (defined in **agencies.txt**, which is self-explanatory), and is of type 3 (referring to a bus—see the specification for more details). Route_desc is optional, but can be used to inform riders that the bus will stop anywhere—the text you put here will be displayed in Google Transit.

--**service_id:** This defines when, exactly, the route will run. It references the **calendar.txt** file. A sample line:

service_id,monday,tuesday,wednesday,thursday,friday,saturday,sunday,start_date, end_date
Bus-Weekday,1,1,1,1,1,0,0,20090601,20100630

Google Transit will know that this trip runs on service_id Bus-Weekday, which runs (as indicated by the value "1") on Monday, Tuesday, Wednesday, Thursday, and Friday, but does not run (as indicated by the value "0") on Saturday and Sunday. The trip only operates between June 1, 2009, and July 30, 2010—this is how you deal with defined scheduling periods (i.e. fall, winter, etc.)

For dealing with holidays and other schedule irregularities, please see the "Holidays" section.

--**trip_id:** this is the unique ID referring to the trip name, referenced in many other text files. Trip IDs **cannot** be repeated.

--**trip_headsign:** This is the destination of the bus, typically whatever is displayed on the buses (i.e. "Haverhill", "Logan Airport"). This helps riders identify the direction in which they want to go, and depends on your agency's naming conventions. Using trip_headsign is optional.

--**shape_id:** See the shapes section for a full rundown on dealing with shapes.

There are two other main files on actually scheduling a trip: **stops.txt** and **stop_times.txt**.

stops.txt: This gives the exact location of all your stops. Sample line:

stop_id,stop_name, stop_lat,stop_lon,zone_id
place-south,South Station,42.395428,-71.142483,1

stop_id is a unique stop identifier, stop_name will be the name displayed in the trip planner, and stop_lat and stop_lon give the location for the stop. For help in working with latitude and longitude, see “Stops and Shapes in Google Earth”, and for information on zone_id, see “Fares”.

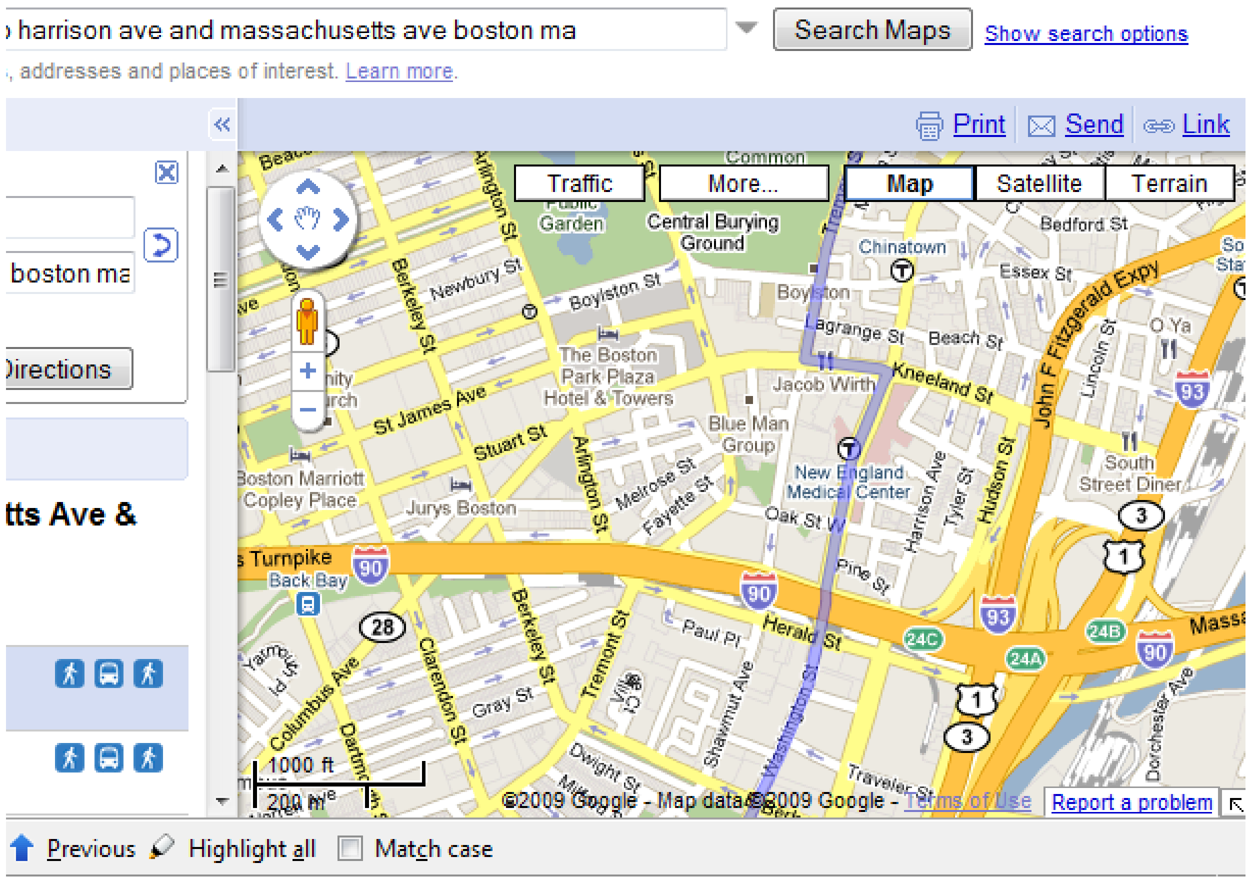
stop_times.txt: This gives the stops and the times for any given trip. For example:

```
trip_id,arrival_time,departure_time,stop_id,stop_sequence
11462304,13:50:00,13:50:00,74611,1
11462304,13:52:00,13:52:00,74612,2
11462304,13:54:00,13:54:00,74613,3
11462304,13:57:00,13:57:00,74624,4
11462304,14:05:00,14:05:00,17091,5
```

For a given trip_id, we have our arrival times and departure times (which will almost always be the same) for all stops in that trip, in the order specified by stop_sequence. Times should be formatted as seen above, h:mm:ss—in Excel, highlight the cells, right-click and go to “Format Cells,” and then pick the correct option (this will vary slightly, based on your version of Excel.)

Stops and Shapes in Google Earth

You should download Google Earth to best understand this section.


harrison ave and massachusetts ave boston ma
Search Maps
Show search options
, addresses and places of interest. Learn more.
Print
Send
Link
Traffic
More...
Map
Satellite
Terrain
Directions
tts Ave &
Previous
Highlight all
Match case
©2009 Google - Map data ©2009 Google - Terms of Use
Report a problem

On Google Transit, you may notice that for some agencies, the routes are shown directly on the streets they cover. For many RTAs, this is particularly important. The transit feed **only** includes timepoints given on schedules, and so the shapes are a way to reflect routes that stop on demand.

Each trip has a "shape" matched with it, referring to **shapes.txt**. Sample line:

```
shape_id,shape_pt_lat,shape_pt_lon,shape_pt_sequence
SL1OUT,42.352578,-71.05536,1
SL1OUT,42.353237,-71.053756,2
```

```
SL1OUT,42.353427,-71.052845,3
SL1OUT,42.353447,-71.052673,4
SL1OUT,42.353457,-71.052514,5
SL1OUT,42.353468,-71.052415,6
```

Each shape is effectively a series of points in a certain direction. All Silver Line One outbound trips, for example, will use the same shape.

However, it is not necessary to manually enter every point when you want to create or edit a shape. The best tool to work with shapes is to use Google Earth, as follows:

1. Open the folder containing your feed (in a .zip file)
2. Open the Feed Validator folder
3. Drag and drop the feed into kmlwriter
4. This will automatically generate a kml file, which you can then open with Google Earth.

Using the path tool (consult earth.google.com for help with Google Earth), you can view and edit each individual shape. With this done, using Bob Heitzman's excellent XLS tools (use GTFS_KML_GPX_Import for this task), you can then convert this kml file into GTFS format within Excel. You can use this same tool to export your own shapes straight from Excel to Google Earth.

IMPORTANT: note that there is a separate shape for inbound and outbound. One shape cannot cover both. It is all right if the shape starts or ends beyond where the actual trip starts or ends.

You can use this same method to move around existing stops or to create new ones.

Holidays and Other Service Irregularities

You can deal with holidays and service irregularities in the **calendar_dates.txt** file. Sample line:

```
service_id,date,exception_type
WEEKDAY,20100215,2
```

Exception_type can have a value of 2 (meaning that service does not run for the given service_id on the given date) or 1 (meaning that service does run). For example, this indicates that all trips operating on service_id WEEKDAY do not run on February 15, 2010 (Presidents' Day). If, instead, all trips on service_id SATURDAY were to run, it would look like:

```
WEEKDAY,20100215,2
SATURDAY,20100215,1
```

Fares

Fares are optional, and are often very difficult to do—this is one of the deficiencies of the Google Transit Feed Specification. Fares are governed by **fare_rules.txt** and **fare_attributes.txt**.

```
fare_id,route_id,origin_id,destination_id,contains,id
BUS,downeaster2,9,10,
```

Fare_id denotes a unique fare name. Route_id tells which routes the fare_id applies to. **Note that if one fare applies to multiple routes, you will need multiple fare_id entries.**

Origin_id, destination_id, and contains_id are optional, and refer to the **zone** field in stops.txt. For example, if you have two zones, each stop will be assigned a zone_id value of either 1 or 2. Say that one bus route 5, it costs $1 to travel within zone one, $2 to travel within zone two, and $2.50 to travel between zones. Fare_rules would look like this:

```
BUS1,5,1,1,
BUS2,5,2,2,
BUS3,5,1,2,
BUS3,5,2,1,
```

Alternatively, if any bus traveling through a certain zone has a single fare, you can use **contains_id**.

Now you will define the fare amount in **fare_rules.txt**, which looks like this:

```
fare_id,agency_id,price,currency_type,payment_method,transfers
BUS1,YOURAGENCY,1,USD,1,0
BUS2,YOURAGENCY, 2,USD,1,0
BUS3,YOURAGENCY, 2.5,USD,1,0
```

See the feed specification for information on payment_method. Transfers refers to the amount of free transfers allowed within a fare—**leaving this blank will allow unlimited free transfers.**

Examples

This provides a guide for common maintenance you will need to perform on the feed.

Seasonal change, no schedule change: All you will need to do is update the start and end dates in calendar.txt, and the holidays in calendar_dates.txt.

Change in a trip: Locate the relevant trip, and edit the times in stop_times.txt.

New trip: Add any new shapes stops to stops.txt, using Google Earth and Bob Heitzman's XLS tools. Construct the trips in trips.txt and the schedule in stop_times.txt. Create new routes in routes.txt if necessary.