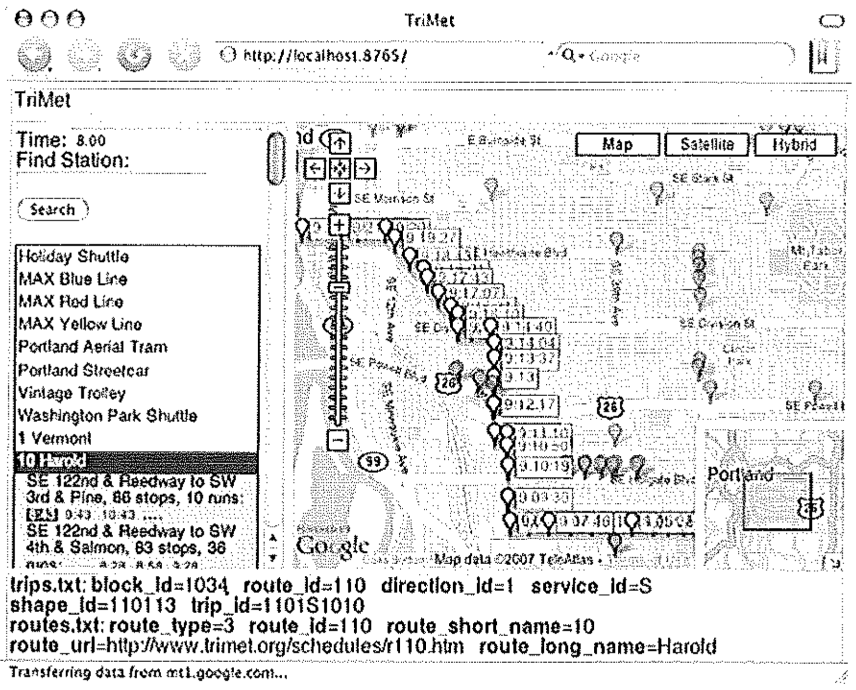
ScheduleViewer

bdferris edited this page on Sep 20 2014 · 1 revision



Holiday Shuttle
MAX Blue Line
MAX Red Line
MAX Yellow Line
Portland Aerial Tram
Portland Streetcar
Vintage Trolley
Washington Park Shuttle
1 Vermont
10 Harold
SE 122nd & Reedway to SW 3rd & Pine, 86 stops, 10 runs: 9:43 10:43 ....
SE 122nd & Reedway to SW 4th & Salmon, 83 stops, 36 runs: ... 4:24 4:54 5:24

trips.txt: block_id=1034 route_id=110 direction_id=1 service_id=S shape_id=110113 trip_id=1101S1010
routes.txt: route_type=3 route_id=110 route_short_name=10 route_url=http://www.trimet.org/schedules/r110.htm route_long_name=Harold

Transferring data from mt1.google.com...

Clone this wiki locally

https://github.com/google/trai

Clone in Desktop

Introduction

Schedule Viewer is a Python program for viewing the contents of a GTFS feed on a map. It's a diagnostic program intended for those creating a feed, and as such doesn't include trip planning or other features that are useful to transit riders.

Installation

The Schedule Viewer is part of the Transit Feed Distribution. See the [TransitFeedDistribution page about the transit feed distribution] for instructions on installing either the Windows executable or python source.

? Adding Circle Stop GPS Coordinates

To run you need:

- (Optional) If you want to access the schedule viewer from another machine you need `<google_maps_api_key>`, a Google Maps API key for your machine `http://my-hostname:8765`, available from the [http://www.google.com/apis/maps/signup.html Google Maps API signup page]
- `<feed_file_or_directory>` a transit feed file or a directory containing an expanded feed file. If you don't have a feed download one from the PublicFeeds page or read

the [http://code.google.com/transit/spec/transit_feed_specification.htm Google Transit Feed Specification] to create your own.

Running from source

```
python schedule_viewer.py [--key <google_maps_api_key>] --feed_filename <feed_file_
```

On Windows the python executable is normally installed as `C:\Python25\python`

Running Windows exe

If you will only access the schedule viewer from your local computer (this is safest and most common case) double click on `schedule_viewer.exe` . When prompted `Enter Feed Location:` drag the GTFS file into the window and press enter.

Otherwise run the following command

```
schedule_viewer.exe [--key <google_maps_api_key>] --feed_filename <feed_file_or_dir
```

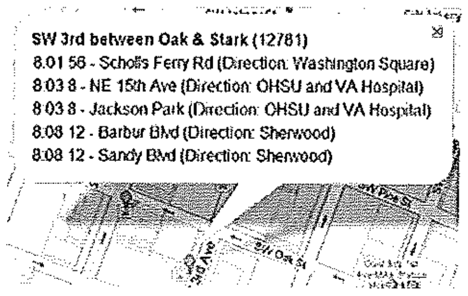
If you run it from Start -> Run include the full path of schedule_viewer.exe and if you are in a command prompt change into the directory containing schedule_viewer.exe.

Explore your data

Once schedule_viewer has finished loading the feed file it prints a URL. Ignore that URL because it won't work with your Google Maps API key. Instead go to http://localhost:8765/ in your web browser and a page similar to the screen shot above should load.

Exploring stops

After you drag the map to a new location with your mouse it will display stops in the current area as semi-transparent blue markers. Multiple stops at the same location will stack up making a darker blue marker. If stops.txt defines `location_type` attribute to distinguish between stations and stops, the stations have red markers and stops have blue ones. Note that the FeedValidator warns that stops within 2m of each other should be merged. If there are lots of stops in the map area only a sample will be displayed. To see all the stops you need to zoom in. The map zooms in when double clicked. If you click on a marker it opens a window listing the "stop_name (stop_id)" and the next trips to visit the stop after "Time:" (HH:MM at top left of screen). If the stop_times table doesn't contain an exact time for a trip the viewer interpolates a time and displays it with a ~. Click on a trip to display it (See "Viewing a trip").


SW 3rd between Oak & Stark (12781)
8:01 56 - Scholls Ferry Rd (Direction: Washington Square)
8:03 8 - NE 15th Ave (Direction: OHSU and VA Hospital)
8:03 8 - Jackson Park (Direction: OHSU and VA Hospital)
8:08 12 - Barbur Blvd (Direction: Sherwood)
8:08 12 - Sandy Blvd (Direction: Sherwood)

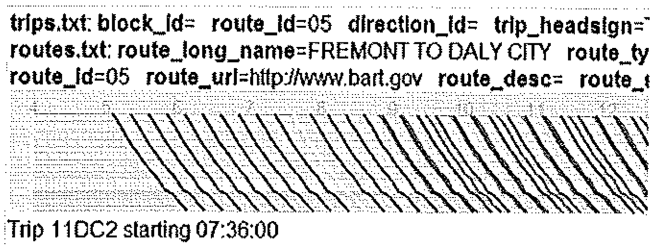
The "Find Station:" field (top left of window) will search for all stops that have a name or stop_id containing the string you enter. The stops will be displayed in yellow. If there is a single match the station's info window will open. If there are multiple matches you may need to zoom out to see them all.

Exploring routes

The left pane lists all the routes in the feed. Click on one of the route names to see the trip patterns (a group of trips that visit the same stops in the same order) for that route. Gray background of pattern indicates that it has a trip with non-zero trip_type value (an unusual trip). The first trip will automatically be selected, and shown on the map. For each trip pattern the number of stops, number of trips and start times are displayed. For patterns that have many trips only a few start times are displayed. To see the start times of more trips edit "Time:" (HH:MM at top left of screen) and click on the route name again.

Viewing a trip

After you click on a trip in a stop marker's window or on a start time in the route list it is displayed in the map. Every stop of the trip is shown as a yellow marker. A semi-transparent blue marker (See "Exploring stops") on top of a yellow looks gray. Timepoints have the time displayed on the map. Some of the source data for that trip and route will be displayed at the bottom of the window. Below the source data is a Marey Graph ([http://www.communication.org.au/publications/case-histories/Timetables-people-and-Tufte/61,0 CRI, Timetables people and Tufte] and [http://www.drones.com/bart.html Graphical Timetables for BART]).


trips.txt: block_id= route_id=05 direction_id= trip_headsign=
routes.txt: route_long_name=FREMONT TO DALY CITY **route_ty**
route_id=05 **route_url=**http://www.bart.gov **route_desc= route_**
Trip 11DC2 starting 07:36:00

The Marey Graph shows all the trips with the same pattern as the selected trip. Different colors representing different service periods (See issue 79). Time increases to the right with a scale in hours since midnight at the top. The distance between stops is on the Y axis but not labeled. A steeper line represents faster movement. In the screen shot above you can see fairly fast service with stops far apart in the East Bay, followed by a big gap without stops in the tunnel under the bay, followed by frequent stops in the city of San Francisco. If you place your mouse over a trip it is selected in white and the trip_id appears below the graph. You see a single service id provides the only service, at a constant headway, until about 9am.

SLADE FEST for COMMUTER CHECK

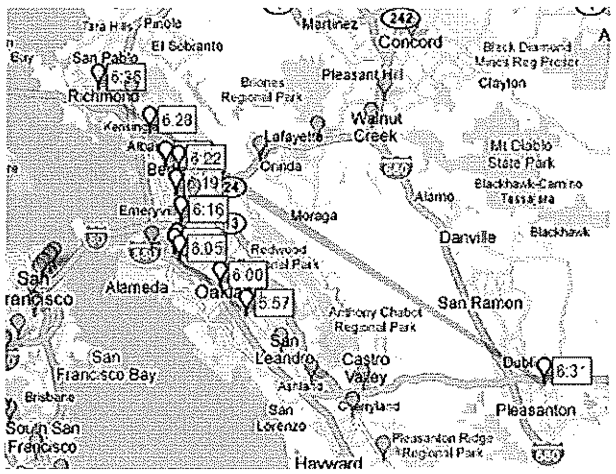
Inspecting for problems

When you first load the schedule viewer the map is zoomed out to contain all stops. If this is much larger than your service area there is probably a stop in the wrong place. Move the service are out of displayed map area and the distant stop will probably appear.

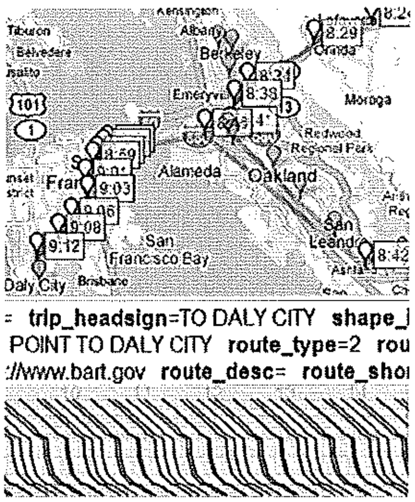
Check the location of a stop by zooming in until you can see the street names. Click on the stop marker to open the window and make sure the marker is in the correct location. For rail stations you may see the real location more easily with satellite view. Repeat this for stops in different parts of the service area.

Click through all the routes. In each route look at one trip of each pattern. Watch out for unreasonably fast or slow legs. If the polyline has any jumps there may be a stop with the incorrect geo location or incorrect information in the stop_times table. If the agency has published a route map it can help you confirm that the trip in GTFS is correct.

Here are some examples of buggy data:

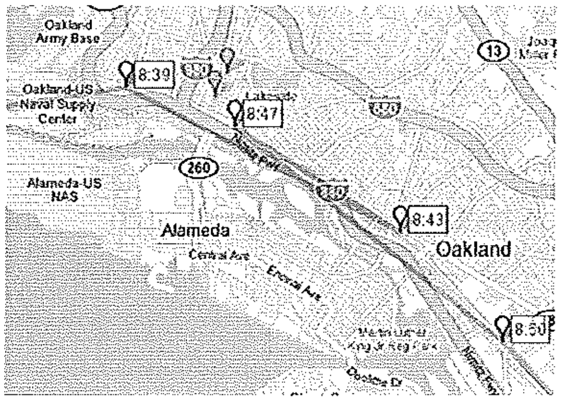


A stop with a bad geo location in stops.txt or using the wrong stop_id in stop_times.txt can cause a big jump.

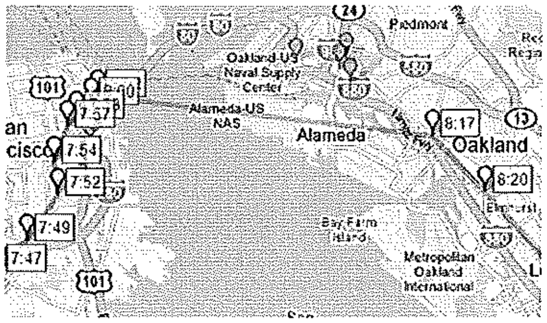


= **trip_headsign**=TO DALY CITY **shape_**
POINT TO DALY CITY **route_type**=2 **rou**
://www.bart.gov **route_desc**= **route_sho**

Big jumps also stick out as a very fast (almost vertical) line in the Marey Graph. Note that a route with a long express section may look similar in the Marey Graph.



A zig-zag can be caused by bad geo locations in stops.txt or an incorrect sequence of stop_id values in stop_times.txt.



Someone familiar with BART may notice that this skips two stops.

Bugs and feature requests

[http://code.google.com/p/googletransitdatafeed/issues/list?q=label:App-ScheduleViewer Vote for known ScheduleViewer issues] so the developers know what is important.

If you find an unreported bug or have a feature request please file a new issue and add the label `App-ScheduleViewer` .