

GTFS Examples

Version 1.41 (last updated 7-July-2016)

About

This document shows examples of [General Transit Feed Specification \(GTFS\)](#) data to illustrate how the Specification can be used to describe various configurations of transit service. Various schedule and fare configurations are presented. In many cases, there can be multiple ways of presenting the same transit service, and these examples aim to illustrate some of those approach options. Feel free to add comments to this document with questions or suggestions. Request edit permissions if you would like to add examples or make enhancements to the document.

A Chinese-translated version of version 1.2 of this document is available from the World Bank's [link repository for international GTFS training materials](#).

Document versions

- **1.41** (7-July-2016) Add World Bank's link repository for international GTFS training materials
- **1.4** (10-January-2016) Change background link from Appropedia to TransitWiki; added example 11 with linked dataset
- **1.3** (30-October-2014): Add "Background / Overview of the Specification" section to the document
- **1.2** (1-Dec-2013): Add example 4, which shows multiple records in frequencies.txt associated with one trip_id.

Table of contents

[Background / Overview of the Specification](#)

[Example 1: relationships between files to define schedules.](#)

[Example 2: using calendar.txt to define seasonal schedules.](#)

[Example 3: using frequencies.txt with stop_times.txt](#)

[Example 4: another look at frequencies.txt; headways intervals vary throughout the day.](#)

[Example 5: another look at frequencies.txt; headways intervals and travel times vary throughout the day.](#)

[Example 6: Trip variations: express and short trips](#)

[Example 7: Different but similar travel patterns are defined as separate routes](#)

[Example 8: Fares, a combination of free and discounted transfers](#)

[Example 9: Fares are defined by zones](#)

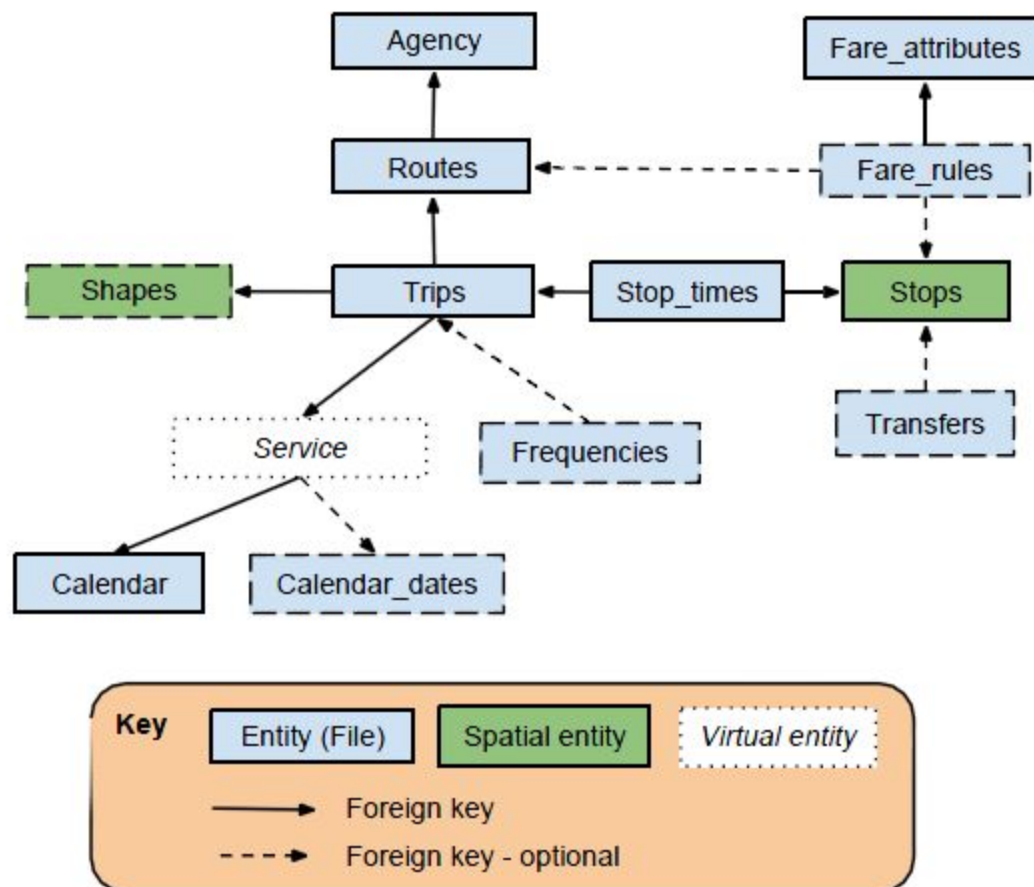
[Example 10: Fare structure includes free transfers, but only for specific routes](#)

[Example 11: A full example, with trips, frequencies, and fares](#)

[Document contributors](#)

Background / Overview of the Specification

- TransitWiki provides a approachable and complete [background on GTFS](#).
- GTFS datasets consist of multiple spreadsheet-like files, in a comma-separated values (CSV) format. Understanding these examples, and GTFS, requires understanding the basic definition of a CSV file (see Wikipedia, [“Comma-separated values”](#))
- The CSV files in a GTFS dataset are “relational”. This means that multiple files contain related information, stored as tables of rows (records) and columns (fields), and allowing a link to be established between separate files that have a matching field ([relational database definition](#)).
- While not required to understand these examples, we recommend at least skimming the [“General Transit Feed Specification” definition document](#).



(Data model of the GTFS file format, created by Martin Davis, as per blog post <http://lin-ear-th-inking.blogspot.com.au/2011/09/data-model-diagrams-for-gtfs.html>.)

Example 1: relationships between files to define schedules.

Schedules defined using stop_times.txt only. Only winter weekday service is defined. Two round-trips occur between downtown and the airport every weekday, also stopping at the railway station.

Required file not shown: agency.txt

calendar.txt									
service_id	start_date	end_date	monday	tuesday	wednesday	thursday	friday	saturday	sunday
winter_weekday	20130921	20140619	1	1	1	1	1	0	0

routes.txt			
route_id	route_short_name	route_long_name	route_type
route_1	1	Downtown/Airport	3

trips.txt				
trip_id	route_id	service_id	direction_id	trip_headsign
trip_1	route_1	winter_weekday	0	Airport
trip_2	route_1	winter_weekday	1	Downtown
trip_3	route_1	winter_weekday	0	Airport
trip_4	route_1	winter_weekday	1	Downtown

stops.txt			
stop_id	stop_name	stop_lat	stop_lon
stop_1	Main and 1st St.	28.8	115.9
stop_2	Railway Station	28.9	116
stop_3	Airport	29	116.1

stop_times.txt				
trip_id	stop_sequence	stop_id	arrival_time	departure_time
trip_1	1	stop_1	9:00:00	9:00:00
trip_1	2	stop_2	9:10:00	9:10:00
trip_1	3	stop_3	9:30:00	9:30:00
trip_2	1	stop_3	9:30:00	9:30:00
trip_2	2	stop_2	9:50:00	9:50:00
trip_2	3	stop_1	10:00:00	10:00:00
trip_3	1	stop_1	10:00:00	10:00:00
trip_3	2	stop_2	10:10:00	10:10:00
trip_3	3	stop_3	10:30:00	10:30:00
trip_4	1	stop_3	10:30:00	10:30:00
trip_4	2	stop_2	10:50:00	10:50:00
trip_4	3	stop_1	11:00:00	11:00:00

Example 2: using calendar.txt to define seasonal schedules.

Schedules defined using stop_times.txt only. Only winter weekday, and summer service (Tuesdays and Thursdays only) is defined. Two-round trips per day occur in the winter. There is one round-trip in the summer, with shorter travel time. Data is color-coded according to seasonal schedules (service_id).

Required file not shown: agency.txt

calendar.txt									
service_id	start_date	end_date	monday	tuesday	wednesday	thursday	friday	saturday	sunday
winter_weekday	20160921	20170619	1	1	1	1	1	0	0
summer_tuesdays_thursdays	20170620	20170920	0	1	0	1	0	0	0

routes.txt			
route_id	route_short_name	route_long_name	route_type
route_1	1	Downtown/Airport	3

trip_id	route_id	service_id	direction_id	trip_headsign
trip_1	route_1	winter_weekday	0	Airport
trip_2	route_1	winter_weekday	1	Downtown
trip_3	route_1	winter_weekday	0	Airport
trip_4	route_1	winter_weekday	1	Downtown
trip_5	route_1	summer_tuesdays_thursdays	0	Airport
trip_6	route_1	summer_tuesdays_thursdays	1	Downtown

stops.txt			
stop_id	stop_name	stop_lat	stop_lon
stop_1	Main and 1st St.	28.8	115.9
stop_2	Railway Station	28.9	116
stop_3	Airport	29	116.1

stop_times.txt				
trip_id	stop_sequence	stop_id	arrival_time	departure_time
trip_1	1	stop_1	9:00:00	9:00:00
trip_1	2	stop_2	9:10:00	9:10:00
trip_1	3	stop_3	9:30:00	9:30:00
trip_2	1	stop_3	9:30:00	9:30:00
trip_2	2	stop_2	9:50:00	9:50:00
trip_2	3	stop_1	10:00:00	10:00:00
trip_3	1	stop_1	10:00:00	10:00:00
trip_3	2	stop_2	10:10:00	10:10:00
trip_3	3	stop_3	10:30:00	10:30:00
trip_4	1	stop_3	10:30:00	10:30:00
trip_4	2	stop_2	10:50:00	10:50:00
trip_4	3	stop_1	11:00:00	11:00:00
trip_5	1	stop_1	9:00:00	9:00:00
trip_5	2	stop_2	9:08:00	9:08:00
trip_5	3	stop_3	9:20:00	9:20:00
trip_6	1	stop_3	9:20:00	9:20:00
trip_6	2	stop_2	9:32:00	9:32:00
trip_6	3	stop_1	9:40:00	9:40:00

Example 3: using frequencies.txt with stop_times.txt

Route 1 runs every 1 hour from 9:00 to 11:00 on winter weekdays. With frequencies.txt, service periods defined in frequencies.txt override specific departure_time and arrival_time values. Travel intervals are provided by stop_times.txt. The service defined here is the same as what is defined in Example 2. Both options are correct GTFS and will be readable by GTFS applications.

Required file not shown: agency.txt

calendar.txt										
service_id	start_date	end_date	monday	tuesday	wednesday	thursday	friday	saturday	sunday	
winter_weekday	20130921	20140619	1	1	1	1	1	0	0	
summer_tuesdays_thursdays	20130620	20130920	0	1	0	1	0	0	0	

routes.txt			
route_id	route_short_name	route_long_name	route_type
route_1	1	Downtown/Airport	3

trip_id	route_id	service_id	direction_id	trip_headsign
trip_1	route_1	winter_weekday	0	Airport
trip_2	route_1	winter_weekday	1	Downtown
trip_3	route_1	summer_tuesdays_thursdays	0	Airport
trip_4	route_1	summer_tuesdays_thursdays	1	Downtown

stops.txt			
stop_id	stop_name	stop_lat	stop_lon
stop_1	Main and 1st St.	28.8	115.9
stop_2	Railway Station	28.9	116
stop_3	Airport	29	116.1

trip_id	stop_sequence	stop_id	arrival_time	departure_time
trip_1	1	stop_1	0:00:00	0:00:00
trip_1	2	stop_2	0:10:00	0:10:00
trip_1	3	stop_3	0:30:00	0:30:00
trip_2	1	stop_3	0:00:00	0:00:00
trip_2	2	stop_2	0:20:00	0:20:00
trip_2	3	stop_1	0:30:00	0:30:00
trip_3	1	stop_1	9:00:00	9:00:00
trip_3	2	stop_2	9:08:00	9:08:00
trip_3	3	stop_3	9:20:00	9:20:00
trip_4	1	stop_3	9:20:00	9:20:00
trip_4	2	stop_2	9:32:00	9:32:00
trip_4	3	stop_1	9:40:00	9:40:00

frequencies.txt			
trip_id	headway_secs	start_time	end_time
trip_1	3600	9:00:00	10:00:00
trip_2	3600	9:30:00	10:30:00

Example 4: another look at frequencies.txt; headways intervals vary throughout the day.

Route 1 operates service in both directions with 5 minute average headways from 7:00 to 12:00. From

12:00 to 22:00, Route 1 operates with 10 minute average headways.
Colors match trips_ids.
Required file not shown: agency.txt

calendar.txt									
service_id	start_date	end_date	monday	tuesday	wednesday	thursday	friday	saturday	sunday
winter_weekday	20130921	20140619	1	1	1	1	1	0	0

routes.txt			
route_id	route_short_name	route_long_name	route_type
route_1	1	Downtown/Airport	3

trips.txt				
trip_id	route_id	service_id	direction_id	trip_headsign
trip_1	route_1	winter_weekday	0	Airport
trip_2	route_1	winter_weekday	1	Downtown

stops.txt			
stop_id	stop_name	stop_lat	stop_lon
stop_1	Main and 1st St.	28.8	115.9
stop_2	Railway Station	28.9	116
stop_3	Airport	29	116.1

stop_times.txt				
trip_id	stop_sequence	stop_id	arrival_time	departure_time
trip_1	1	stop_1	0:00:00	0:00:00
trip_1	2	stop_2	0:10:00	0:10:00
trip_1	3	stop_3	0:30:00	0:30:00
trip_2	1	stop_3	0:00:00	0:00:00
trip_2	2	stop_2	0:10:00	0:10:00
trip_2	3	stop_1	0:30:00	0:30:00

frequencies.txt			
trip_id	headway_secs	start_time	end_time
trip_1	300	7:00:00	12:00:00
trip_1	600	12:00:00	22:00:00
trip_2	300	7:00:00	12:00:00
trip_2	600	12:00:00	22:00:00

Example 5: another look at frequencies.txt; headways intervals and travel times vary throughout the day.

Route 1 operates service in both directions with 5 minute average headways from 7:00 to 12:00. From 12:00 to 22:00, Route 1 operates with 10 minute average headways, but service is faster (less travel time).

Colors match trips_ids.

Required file not shown: agency.txt

calendar.txt									
service_id	start_date	end_date	monday	tuesday	wednesday	thursday	friday	saturday	sunday
winter_weekday	20130921	20140619	1	1	1	1	1	0	0

routes.txt			
route_id	route_short_name	route_long_name	route_type
route_1	1	Downtown/Airport	3

trips.txt				
trip_id	route_id	service_id	direction_id	trip_headsign
trip_1	route_1	winter_weekday	0	Airport
trip_2	route_1	winter_weekday	1	Downtown
trip_3	route_1	winter_weekday	0	Airport
trip_4	route_1	winter_weekday	1	Downtown

stops.txt			
stop_id	stop_name	stop_lat	stop_lon
stop_1	Main and 1st St.	28.8	115.9
stop_2	Railway Station	28.9	116
stop_3	Airport	29	116.1

stop_times.txt				
trip_id	stop_sequence	stop_id	arrival_time	departure_time
trip_1	1	stop_1	0:00:00	0:00:00
trip_1	2	stop_2	0:10:00	0:10:00
trip_1	3	stop_3	0:30:00	0:30:00
trip_2	1	stop_3	0:00:00	0:00:00
trip_2	2	stop_2	0:10:00	0:10:00
trip_2	3	stop_1	0:30:00	0:30:00
trip_3	1	stop_1	0:00:00	0:00:00
trip_3	2	stop_2	0:08:00	0:08:00
trip_3	3	stop_3	0:20:00	0:20:00
trip_4	1	stop_3	0:00:00	0:00:00
trip_4	2	stop_2	0:12:00	0:12:00
trip_4	3	stop_1	0:20:00	0:20:00

frequencies.txt			
trip_id	headway_secs	start_time	end_time
trip_1	300	7:00:00	12:00:00
trip_2	300	7:00:00	12:00:00
trip_3	600	12:00:00	22:00:00
trip_4	600	12:00:00	22:00:00

Example 6: Trip variations: express and short trips

In addition to trips that serve the full route, Route 1 service includes express trips to the airport that skip the railway station, and shortened trips that end at the railway station. A different trip_headsign value indicates the difference in trip patterns.

Required file not shown: agency.txt

calendar.txt									
service_id	start_date	end_date	monday	tuesday	wednesday	thursday	friday	saturday	sunday
winter_weekday	20130921	20140619	1	1	1	1	1	0	0

routes.txt			
route_id	route_short_name	route_long_name	route_type
route_1	1	Downtown/Airport	3

trips.txt				
trip_id	route_id	service_id	direction_id	trip_headsign
trip_1	route_1	winter_weekday	0	Airport
trip_2	route_1	winter_weekday	1	Downtown
trip_3	route_1	winter_weekday	0	Airport (Express)
trip_4	route_1	winter_weekday	1	Downtown (Express)
trip_5	route_1	winter_weekday	0	Railway Station
trip_6	route_1	winter_weekday	1	Downtown

stops.txt			
stop_id	stop_name	stop_lat	stop_lon
stop_1	Main and 1st St.	28.8	115.9
stop_2	Railway Station	28.9	116
stop_3	Airport	29	116.1

stop_times.txt				
trip_id	stop_sequence	stop_id	arrival_time	departure_time
trip_1	1	stop_1	0:00:00	0:00:00
trip_1	2	stop_2	0:10:00	0:10:00
trip_1	3	stop_3	0:30:00	0:30:00
trip_2	1	stop_3	0:00:00	0:00:00
trip_2	2	stop_2	0:10:00	0:10:00
trip_2	3	stop_1	0:30:00	0:30:00
trip_3	1	stop_1	9:00:00	9:00:00
trip_3	2	stop_3	9:22:00	9:22:00
trip_4	1	stop_3	9:22:00	9:22:00
trip_4	2	stop_1	9:42:00	9:42:00
trip_5	1	stop_1	10:00:00	10:00:00
trip_5	2	stop_2	10:10:00	10:10:00
trip_6	1	stop_2	10:10:00	10:10:00
trip_6	2	stop_1	10:20:00	10:20:00

frequencies.txt			
trip_id	headway_secs	start_time	end_time
trip_1	3600	9:00:00	10:30:00
trip_2	3600	9:30:00	11:00:00

Example 7: Different but similar travel patterns are defined as separate routes

The schedule defined here is the same as is defined in example 5. However, in this case, variations are presented to customers as different routes (1, 1A, and 1B). Colors match to route_id.

Required file not shown: agency.txt

calendar.txt									
service_id	start_date	end_date	monday	tuesday	wednesday	thursday	friday	saturday	sunday
winter_weekday	20130921	20140619	1	1	1	1	1	0	0

routes.txt			
route_id	route_short_name	route_long_name	route_type
route_1	1	Downtown/Airport	3
route_1a	1A	Airport Express	3
route_1b	1B	Downtown/Railway Station	3

trips.txt				
trip_id	route_id	service_id	direction_id	trip_headsign
trip_1	route_1	winter_weekday	0	Airport
trip_2	route_1	winter_weekday	1	Downtown
trip_3	route_1a	winter_weekday	0	Airport
trip_4	route_1a	winter_weekday	1	Downtown
trip_5	route_1b	winter_weekday	0	Railway Station
trip_6	route_1b	winter_weekday	1	Downtown

stops.txt			
stop_id	stop_name	stop_lat	stop_lon
stop_1	Main and 1st St.	28.8	115.9
stop_2	Railway Station	28.9	116
stop_3	Airport	29	116.1

stop_times.txt				
trip_id	stop_sequence	stop_id	arrival_time	departure_time
trip_1	1	stop_1	0:00:00	0:00:00
trip_1	2	stop_2	0:10:00	0:10:00
trip_1	3	stop_3	0:30:00	0:30:00
trip_2	1	stop_3	0:00:00	0:00:00
trip_2	2	stop_2	0:10:00	0:10:00
trip_2	3	stop_1	0:30:00	0:30:00
trip_3	1	stop_1	9:00:00	9:00:00
trip_3	2	stop_3	9:22:00	9:22:00
trip_4	1	stop_3	9:22:00	9:22:00
trip_4	2	stop_1	9:42:00	9:42:00
trip_5	1	stop_1	10:00:00	10:00:00
trip_5	2	stop_2	10:10:00	10:10:00
trip_6	1	stop_2	10:10:00	10:10:00
trip_6	2	stop_1	10:20:00	10:20:00

frequencies.txt			
trip_id	headway_secs	start_time	end_time
trip_1	3600	9:00:00	10:30:00
trip_2	3600	9:30:00	11:00:00

Example 8: Fares, a combination of free and discounted transfers

One ride is 2 RMB, but customers may purchase a transfer for an additional 1 RMB. This fare structure is applied across the system.

fare_attributes.txt

fare_id	price	currency_type	payment_method	transfers	transfer_duration
one_ride	2	CNY	1	0	
transfer_fare	3	CNY	1	1	

Example 9: Fares are defined by zones

Travel within the central district (zone_a) is less expensive than travel between zones. Coloring according to fare_id.

stops.txt

stop_id	stop_name	stop_lat	stop_lon	zone_id
stop_1	Main and 1st St.	28.8	115.9	zone_a
stop_2	Railway Station	28.9	116	zone_a
stop_3	Airport	29	116.1	zone_b

fare_attributes.txt

fare_id	price	currency_type	payment_method	transfers	transfer_duration
one_zone	2	CNY	1		
two_zones	3	CNY	1		

fare_rules.txt

fare_id	origin_id	destination_id
one_zone	zone_a	zone_a
two_zones	zone_a	zone_b
two_zones	zone_b	zone_a

Example 10: Fare structure includes free transfers, but only for specific routes

Routes 1 and 2 cost 1 RMB to ride. Transfers are available between those routes for 0.5 RMB of extra cost (transfers are valid for 1 hour). Routes 3 and 4 cost 5 RMB to ride. Discounted transfers are not available between those routes. However, free transfers are available to Routes 1 and 2. Notice that route_3_fare and route_4_fare are applied for route_1 and route_2 in fare_rules.txt. Still, when the rider only uses Route 1 or 2, the less expensive fare will be returned: when software reads the GTFS and determines the fare, it will always select the least expensive fare if multiple rules match.

Coloring according to fare_id.

fare_attributes.txt

fare_id	price	currency_type	payment_method	transfers	transfer_duration
cheap_fare	1	CNY	1		
cheap_fare_transfer	1.5	CNY	1		3600
route_3_fare	5	CNY	1		3600
route_4_fare	5	CNY	1		3600

fare_rules.txt

fare_id	route_id
cheap_fare	route_1
cheap_fare	route_2
cheap_fare_transfer	route_1
cheap_fare_transfer	route_2
route_3_fare	route_3
route_3_fare	route_1
route_3_fare	route_2
route_4_fare	route_4
route_4_fare	route_1
route_4_fare	route_2

Example 11: A full example, with trips, frequencies, and fares

- Mon-Fri service in winter, 2 trips in each direction per day
- Mon-Sun service in the summer
 - 10 min headway before noon
 - 15 min headway in afternoon
- Zone-based fares
 - \$3 fare to/from Stagecoach Hotel & Casino
 - \$2 fare for other trips

This dataset can be downloaded from <http://data.trilliumtransit.com/gtfs/deathvalley-demo-ca-us/>

agency.txt						
agency_id	agency_name	agency_url	agency_timezone	agency_phone	agency_fare_url	agency_lang
249	Demo Transit Authority	http://gtfsdemo-transit.org	America/Los_Angeles	503-567-8422	http://gtfsdemo-transit.org/fares-and-tickets	en

stops.txt				
stop_id	stop_name	stop_lat	stop_lon	zone_id
stop_1	E Main St and S Irving St	36.905697	-116.76218	zone_a
stop_2	North Ave at D Ave N	36.914893	-116.76821	zone_a
stop_3	Stagecoach Hotel & Casino	36.915682	-116.751677	zone_b

calendar.txt									
service_id	monday	tuesday	wednesday	thursday	friday	saturday	sunday	start_date	end_date
winter_weekday	1	1	1	1	1	0	0	20151101	20160430
summer_daily	1	1	1	1	1	1	1	20160501	20161031

routes.txt			
route_id	route_short_name	route_long_name	route_type
route_1	1	Beatty Local	3

trips.txt				
trip_id	route_id	service_id	direction_id	trip_headsign
trip_1	route_1	winter_weekday	0	Stagecoach Hotel & Casino
trip_2	route_1	winter_weekday	1	E Main St and S Irving St (Express)

trip_3	route_1	winter_weekday	0	Stagecoach Hotel & Casino (Express)
trip_4	route_1	winter_weekday	1	E Main St and S Irving St
trip_5	route_1	summer_daily	0	Stagecoach Hotel & Casino
trip_6	route_1	summer_daily	1	E Main St and S Irving St
trip_7	route_1	summer_daily	0	Stagecoach Hotel & Casino
trip_8	route_1	summer_daily	1	E Main St and S Irving St

stop_times.txt

trip_id	stop_sequence	stop_id	arrival_time	departure_time
trip_1	1	stop_1	10:00:00	10:00:00
trip_1	2	stop_2	10:10:00	10:10:00
trip_1	3	stop_3	10:30:00	10:30:00
trip_2	1	stop_3	10:30:00	10:30:00
trip_2	3	stop_1	11:00:00	11:00:00
trip_3	1	stop_1	15:00:00	15:00:00
trip_3	3	stop_3	15:30:00	15:30:00
trip_4	1	stop_3	15:30:00	15:30:00
trip_4	2	stop_2	15:50:00	15:50:00
trip_4	3	stop_1	16:00:00	16:00:00
trip_5	1	stop_1	7:00:00	7:00:00
trip_5	2	stop_2	7:10:00	7:10:00
trip_5	3	stop_3	7:30:00	7:30:00
trip_6	1	stop_3	7:30:00	7:30:00
trip_6	2	stop_2	7:50:00	7:50:00
trip_6	3	stop_3	8:00:00	8:00:00
trip_7	1	stop_1	12:00:00	12:00:00
trip_7	2	stop_2	12:08:00	12:08:00
trip_7	3	stop_3	12:25:00	12:25:00
trip_8	1	stop_3	12:25:00	12:25:00
trip_8	2	stop_2	12:37:00	12:42:00
trip_8	3	stop_1	12:50:00	12:50:00

frequencies.txt

trip_id	headway_secs	start_time	end_time
trip_5	600	7:00:00	12:00:00
trip_6	600	7:30:00	12:30:00
trip_7	900	12:00:00	22:00:00
trip_8	900	12:25:00	22:25:00

fare_attributes.txt

fare_id	price	currency_type	payment_method	transfers	transfer_duration
one_zone	2	USD	1		
two_zones	3	USD	1		

fare_rules.txt

fare_id	origin_id	destination_id
one_zone	zone_a	zone_a
two_zones	zone_a	zone_b
two_zones	zone_b	zone_a

Document contributors

Aaron Antrim

[Trillium Solutions, Inc.](#)

Portland, Oregon USA

aaron@trilliumtransit.com

Twitter: [@trilliumtransit](#)

(503) 567-8422

Holly Krambeck

The World Bank

Washington, DC USA

Phone: +1 (202) 473-2282

Email: hkrambeck@worldbank.org

Blog: <http://blogs.worldbank.org/transport/blogs/holly-krambeck>

Linghong Zou

Columbia University, GSAPP, Master of Science in Urban Planning Candidate 2014

linghongzou0116@gmail.com

Dr. Li Qu

The World Bank

Washington, DC USA

lqu@worldbank.org

Chris Perry

[Trillium Solutions, Inc.](#)

Portland, Oregon USA

Kalon Thomas

Trillium Solutions, Inc.

Portland, Oregon USA