GTFS-Flex v2.1 extensions proposal

Including GTFS-FlexibleTrips and GTFS-BookingRules

Warning: This document contains a proposal for an extension for GTFS which hasn't been adopted yet. The following content may evolve based on community feedback, and any implementation of it may have to be updated when the final version is adopted. If you have any questions, please reach out to hello@mobilitydata.org.

Overview

Transit operators provide a variety of demand-responsive service which cannot be currently represented in GTFS because the services pick up or drop off riders at a location and/or time chosen by the rider. The following two extensions aim to address this need:

GTFS-FlexibleTrips describes flexible services that operate according to some schedule, but which will, on request, during scheduled service, perform certain actions to suit the particular needs of individual riders, such as deviate to a requested address or go to one of a number of predefined stops.

GTFS-BookingRules provides booking information for rider-requested services using **GTFS-FlexibleTrips**, such as how far in advance booking should occur or a phone number that should be called.

A few examples give sample of dataset in different use cases:

- [E1] STM Baie-D'Urfé PM: from stop to set of stops
- [E2] Informal bus line: hail-and-ride frequency based
- [E3] Citymapper SmartRide: on-demand on main streets
- [E4] RCT with continuous stops & route deviation

GTFS-FlexibleTrips

Goals

This extension describes services that operate according to a schedule, but also include one or more flexible features, such as:

- **Dial-a-ride service**: the vehicle serves a zone where pickups and drop offs are allowed during certain service hours.
- **Route deviation services**: the vehicle serves a fixed route and ordered set of stops, and may detour to pick up or drop off a passenger between stops (e.g. [E8] BVG Doorstep Service).
- **Point-to-zone service**: the rider can board at a fixed stop such as a train station, and then alight anywhere within an area, or vice versa (e.g. [E9] Assier TAD). Departures from some locations are scheduled or timed with other services.

- **Point deviation or checkpoint service**: the rider can board at a fixed stop, and then alight anywhere among an unordered list of stops, or the opposite. The driver only serves stops at which a request is made (e.g. [E2] STM Baie-D'Urfé PM, [E3] STM Industrial Park PM).
- **Hail-and-ride services**: the vehicle stays along a fixed path, but the rider can request a stop anywhere along the path to board or alight (e.g. [E5] MBTA Snow Route 11, [E6] Informal bus line).

GTFS-FlexibleTrips describes the times when and locations where flexible service can be requested.

Overview

This extension

- **Describe locations and groups of locations where riders can request pickup or drop off**: these locations are included in new files called `location_groups.txt` and `locations.geojson`
- **Indicates the times when services are available at on demand locations and the expected travel times**: new fields in `stop_times.txt` provide ranges that equate to service hours and expected travel times on demand trips
- **Clarifies elements of the current specification necessary to inform data consumers of how to interpret the above files and fields added**: new fields in `stop_times.txt` provide ranges that equate to service hours or expected traversal times for locations

Requirements

None. Extends the core CSV GTFS.

Warning: In order for a trip planner to provide a user with information about how to request many flexible services, data producers must also provide information according to the **GTFS-BookingRules** extension.

Files extended or added

File Name	State	Defines
`location_groups.txt`	Added	Adds location groups. Location groups are groups of stops and GeoJSON locations, which allow predetermined groups of these features to be specified on individual rows of `stop_times.txt`.
`locations.geojson`	Added	Adds GeoJSON locations, which are `LineString`, `MultiLineString`, `Polygon` and `MultiPolygon` features that indicate groups of lat/lon coordinates where riders can request either pickup or drop off.

stop_times.txt	Extended and modified	

File Definitions

location_groups.txt (file added)

Field Name	Details
location_group_id	(ID, Required) Defines an identifier for the location group. A location group is a group of stops and/or GeoJSON locations that together indicate various locations where a rider may request pickup or drop off. The location_group_id belongs to the same namespace as stop_id in stops.txt and id in locations.geojson, called “stop locations”. Multiple rows in location_group.txt can have the same location_group_id.
location_id	(ID, Optional) Adds this stop location to the location group. Refers either to: - a stop_id from stops.txt - an id of a location in locations.geojson.
location_group_name	(Text, Optional) Defines the name of the location group. For one location_group_id, either only one entry should have a location_group_name or each must contain the same value.

locations.geojson (file added)

- This file uses a subset of the GeoJSON format, described in RFC 7946.
- The locations.geojson file must contain a FeatureCollection.
- A FeatureCollection defines various stop locations where riders may request pickup or drop off.
- Only features with types LineString, MultiLineString, Polygon and MultiPolygon can be defined. Individual stops should be defined in stops.txt and not in locations.geojson.
- Every GeoJSON Feature must have an id. The id belongs to the same namespace as stop_id in stops.txt and location_group_id in location_groups.txt, called “stop locations”.
- Every GeoJSON Feature may have an properties object, with the following keys:

Key	Value

stop_name	(Text, Optional) Defines the name of the location. The name should be the same, which is used in customer communication, eg. the name of the village where the service stops.
stop_desc	(Text, Optional) Description of the location that provides useful, quality information. Can contain a textual representation of the geometry for the location.
zone_id	(ID, Optional) Identifies the fare zone for a stop. This field is required if providing fare information using fare_rules.txt, otherwise it is optional.
stop_url	(URL, Optional) URL of a web page about the location. This should be different from the `agency.agency_url` and the `routes.route_url` field values.

stop_times.txt (file extended)

Field Name	**Details**
`stop_id`	[... existing spec...] If service is on request, a GeoJSON location or location group can also be referenced: - Field `id` from `locations.geojson` - Field `location_group_id` from `location_groups.txt`
`arrival_time`	[... Current definition …, altering the last sentence “An arrival time must be specified for the first and the last stop time in a trip, unless that stop time refers to a GeoJSON location or location group.”] **Forbidden** when the `stop_id` refers to a GeoJSON location or location group.
`departure_time`	[... Current definition …] **Forbidden** when the `stop_id` refers to a GeoJSON location or location group.

`start_pickup_dropoff_window`
 	
and
 	
`end_pickup_dropoff_window`
	(Time, Conditionally Required) Service to a GeoJSON location or location group happens within a time frame. Fields `start_pickup_dropoff_window` and `end_pickup_dropoff_window` define the beginning and end of the time frame.
	
	For times occurring after midnight on the service day, enter the time as a value greater than 24:00:00 in HH:MM:SS for the day on which the trip schedule begins. A single stop time with a `start_pickup_dropoff_window` of 00:00:00 and a `end_pickup_dropoff_window` of 24:00:00 represent service that is available continuously at all times, if service exists on preceding and

	subsequent days. `start_pickup_dropoff_window` must be earlier than `end_pickup_dropoff_window`. **Forbidden** when the `stop_id` refers to a single stop in stops.txt, but **Conditionally Required** if `stop_id` refers to a GEOJSON location or location group.
`pickup_type`	[...] GeoJSON locations and location groups: - Cannot have `pickup_type=0`. - Can have `pickup_type=1` or `pickup_type=2`. - Can have `pickup_type=3` only if they are a single GeoJSON LineString, since it implies that the passenger is able to "coordinate with the driver" for pick up, and therefore the driver has to serve all of the location groups to see the potential passenger waiting on the sidewalk and hailing them. It implies that the vehicle will entirely serve the path defined by the LineString, in the order in which it is defined in the GeoJSON.
`drop_off_type`	[...] GeoJSON locations and location groups: - Cannot have `drop_off_type=0`. - Can have `drop_off_type=1`, `drop_off_type=2` or `drop_off_type=3`.

`mean_duration_factor`
 	
and
 	
`mean_duration_offset`
	(Float, **Conditionally Required**) Fields `mean_duration_factor` and `mean_duration_offset` allow the estimation of how fast a rider's trip will take during the use of service in a GeoJSON location or location group. Below is a description of the calculation data consumers are expected to make, when estimating the travel time within a GeoJSON location or location group.
	
	Use `mean_duration_factor` and `mean_duration_offset` to calculate the `MeanTravelDuration` based on the `DrivingDuration`. The `MeanTravelDuration` is given by the following formula:
	
	`MeanTravelDuration = mean_duration_factor × DrivingDuration + mean_duration_offset`
	
	The `DrivingDuration` is the time it would take in a car to travel the distance being calculated for flexible service, as defined by the data consumer. The `MeanTravelDuration` is the calculated average time one expects the service to travel the same trip.

	The `MeanTravelDuration` may be calculated for the time and the day of the trip to take into account traffic; in other words the consumer is expected to know that `DrivingDuration` is dynamic. Producers should thus provide values which reflect increases in `DrivingDuration` due to additional pickups and drop offs beyond that of the passenger. A downtown TNC will likely always have a `mean_duration_factor` of 1, with or without traffic, since it goes with the flow. But a shared service can have a factor of 2 or more if many additional pickups and drop offs are expected. `mean_duration_offset` can be utilized to increase travel times of shorter trips relatively more than times for longer trips. While traveling through undefined space between GeoJSON locations or location groups, it is assumed that: `MeanTravelDuration = DrivingDuration` **Conditionally Required:** - **Forbidden** if `stop_id` does not refer to a GeoJSON location or location group. - Optional otherwise.

`safe_ duration_ factor`
 	
and
 	
`safe_ duration_ offset`
	(Float, **Conditionally Required**) Fields `safe_duration_factor` and `safe_duration_offset` allow calculation of the `SafeTravelDuration` based on the `DrivingDuration`. The `SafeTravelDuration` is a way for data consumers to calculate the likely “worst case scenario” for a rider when estimating an on demand trip.
	
	The `SafeTravelDuration` is given by the following formula:
	
	`SafeTravelDuration = safe_duration_factor × DrivingDuration + safe_duration_offset`
	
	The `SafeTravelDuration` should reflect the longest time one expects the service to require, in 95% of trips. A rider can account for this much time to use the service, and have a very good chance of having the trip take this much time or less.
	
	Conditionally Required:
	- **Forbidden** if stop_id does not refer to a GeoJSON location or location group.
	- Optional otherwise.

GTFS-BookingRules

Goals

Many flexible services included in the **GTFS-FlexibleTrips** extension must be booked in advance and/or by using a phone or the internet. This extension provides the rider with information about how to request service.

Requirements

[None. Extends the core CSV GTFS.]

Files extended or added

File Name	State	Defines
stop_times.txt	Extended	Adds links to booking rules.
booking_rules.txt	Added	Defines the booking rules

Tables Definitions

stop_times.txt (file extended)

Field Name	Details
`pickup_booking_rule_id`	(ID, Optional) Defines the boarding booking rules for this stop time. It is strongly recommended to have a booking rule defined (either at the trip or the stop time level) when `pickup_type=2`.
`drop_off_booking_rule_id`	(ID, Optional) Defines the alighting booking rules for this stop time. It is strongly recommended to have a booking rule defined (either at the trip or the stop time level) when `drop_off_type=2`.

booking_rules.txt (file added)

Field Name	Details
`booking_rule_id`	(ID, **Required**) Defines an ID for the booking rule, which will be referenced from `stop_times.txt`.
`booking_type`	(Enum, **Required**) Defines how much in advance the booking can be made. Value are: - 0: Real-time booking only

	- 1: Up to same-day booking, with advance notice - 2: Up to prior day(s) booking
`prior_notice_duration_min`	(Integer, **Conditionally Required**) Minimum number of minutes of advance time necessary before travel to make a booking request. **Conditionally Required**: (The timing must be provided by either a duration or a last time & day) - **Required** for up-to-same-day booking (`booking_type=1`).. - **Forbidden** otherwise.
`prior_notice_duration_max`	(Integer, Conditionally Optional) Maximum number of minutes of advance time necessary before travel to make a booking request. Conditionally Optional: - Optional for up-to-same-day booking (`booking_type=1`) - **Forbidden** otherwise.
`prior_notice_last_day`	(Integer, **Conditionally Required**) Latest day on which a booking can be made. Defined as an offset, so the number of service days in advance of the booking. Example: “Ride must be booked 1 day in advance before 5PM” will be encoded as `prior_notice_last_day=1`. **Conditionally Required**: (The timing must be provided by either a duration or a last time & day) - **Required** for up-to-prior-day booking (`booking_type=2`). - **Forbidden** otherwise.
`prior_notice_last_time`	(Time, **Conditionally Required**) Latest time of day on the last day on which a booking can be made. The timezone used is the one defined by `agency.agency_timezone`. Example: “Ride must be booked 1 day in advance before 5PM” will be encoded as `prior_notice_last_time=17:00:00`. **Conditionally Required**: - **Required** if `prior_notice_last_day` is defined. - **Forbidden** otherwise.
`prior_notice_start_day`	(Integer, Conditionally Optional) Earliest day on which a booking can be made. Defined as an offset, so the number of service days in advance of the booking. Example: “Ride can be booked at the earliest one week in advance at midnight” will be encoded as `prior_notice_start_day=7`. Conditionally Optional: - Optional for up-to-prior-day booking (`booking_type=2`).

	- Optional for up-to-same-day booking (`booking_type=1`) if `prior_notice_duration_max` is empty. - **Forbidden** otherwise.
`prior_notice_start_time`	(Time, **Conditionally Required**) Earliest time of the earliest day at which a booking can be made. The timezone used is the one defined by `agency.agency_timezone`. Example: "Ride can be booked at the earliest one week in advance at midnight" will be encoded as `prior_notice_start_time=00:00:00`. **Conditionally Required**: - **Required** if `prior_notice_start_day` is defined. - **Forbidden** otherwise.
`prior_notice_service_id`	(ID from `calendar.txt`, Optional) When `prior_notice_start_day` is used, `prior_notice_service_id` defines a subset of days that count towards the `prior_notice_start_day`. Example: If empty, `prior_notice_start_day=2` will be two calendar days in advance. If defined as a service_id containing only business days (weekdays without holidays), `prior_notice_start_day=2` will be two business days in advance.
`message`	(Text, Optional) Message to passengers utilizing service at a `stop_time` with this boarding rule. This message appears when the rider is booking a demand responsive pickup and drop off. The message is meant to provide minimal information to be transmitted within a user interface about the action a rider must take in order to utilize the service. This text is expected to be a link within user interfaces, whether to a phone number, a url, or a deep link to an app.
`pickup_message`	(Text, Optional) Identical to `message` but used when riders have a demand response pickup only.
`drop_off_message`	(Text, Optional) Identical to `message` but used when riders have a demand response drop off only.
`phone_number`	(Phone number, Optional) Phone number to make the booking. Must follow the E.123 standard (e.g. "`+1 503 238 7433`" for TriMet).
`info_url`	(URL, Optional) The `info_url` field contains a URL providing human readable information about the booking rule.
`booking_url`	(URL, Optional) If a rider can book trips according to this booking rule through an online interface or app, the link to that reservation system or app download page is the `booking_url`.

Examples

[E1] STM Côte-Vertu Ouest: triggered scheduled trip

[Source: http://www.stm.info/en/info/networks/shared-taxibus/cote-vertu-ouest]

Description: Fix route (fix ordered list of stops), fix schedule, but triggered only on demand.

GTFS-Flex features used:

- GTFS-FlexibleTrips: no
- GTFS-BookingRules: **yes**

Data:

`trips.txt`

```
route_id,service_id,trip_id,trip_headsign,booking_rule_id
[...]
ri_CoteVertuOuest,we,ti_1,Cote Vertu Subway Station,bri_1
[...]
```

`booking_rules.txt`

```
booking_rule_id,booking_type,prior_notice_duration_min,phone_number,info_
url
bri_1,1,60,+1 514 636 6666,https://coopouest.accestaxi.com
```

[E1] STM Baie-D'Urfé PM: from stop to set of stops

[Source: http://www.stm.info/en/info/networks/shared-taxibus/baie-durfe]

Description: During evening peak hours, this service picks up at the suburban train station (Gare Baie-d'Urfée) and drops off riders anywhere among a set of stops.

GTFS-Flex features used:

- GTFS-FlexibleTrips: **yes**
- GTFS-BookingRules: **yes** (not represented, same as STM Côte-Vertu Ouest example)

Note: Values in fields `mean_duration_factor` and `safe_duration_factor` are guessed.

Data:

`stops.txt`

```
stop_id,stop_name,stop_lon,stop_lat
si_1,Baie-D'Urfé Station,-73.91556501388548,45.41996153213654
si_2,Morgan / Victoria,-73.91631603240967,45.4152923527824
si_3,Sunny Acres / Victoria,-73.91279697418213,45.41649733925505
si_4,Sunny Acres / Surrey,-73.91215324401855,45.41996153213654
[...]
```

`location_groups.txt`

```
location_group_id,child_id,location_group_name
li_1,si_2,Sainte-Anne-de-Bellevue taxibus stops
li_1,si_3,
li_1,si_4,
[...]
```

`trips.txt`

```
route_id,service_id,trip_id,trip_headsign
[...]
ri_BaieDUrfePM,WorkDays,ti_1556,Sainte-Anne-de-Bellevue
ri_BaieDUrfePM,WorkDays,ti_1557,Sainte-Anne-de-Bellevue
[...]
```

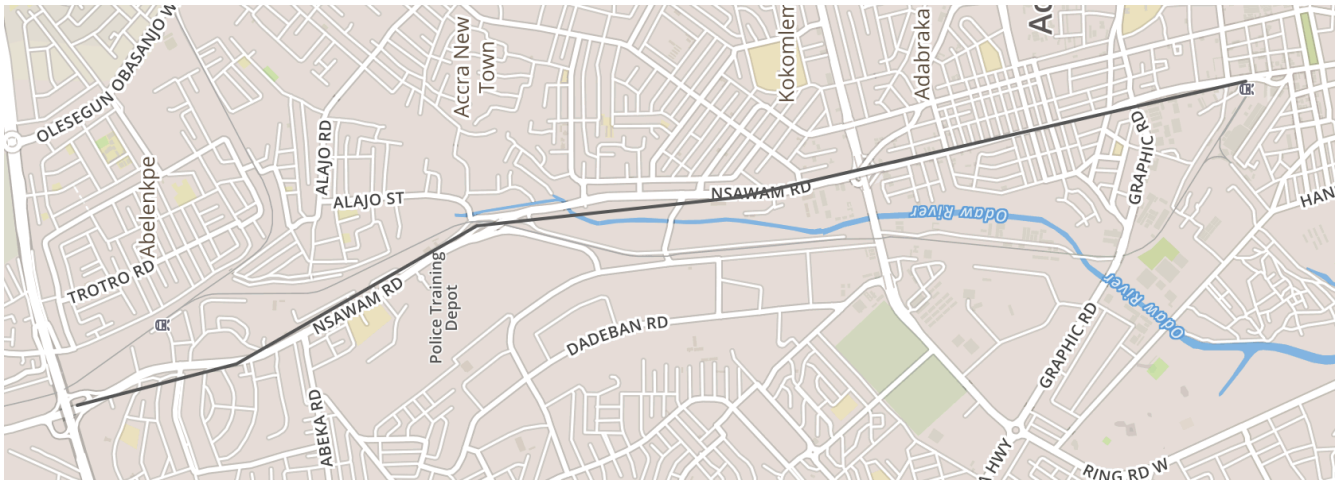
stop_times.txt

```
trip_id,stop_sequence,arrival_time,departure_time,start_pickup_dropoff_window,end_pickup_dropoff_window,stop_id,pickup_type,drop_off_type,pickup_booking_rule_id,drop_off_booking_rule_id,mean_duration_factor,safe_duration_factor
ti_1556,1,15:56:00,15:56:00,,,si_1,0,1,,,,,
ti_1556,2,,,15:56:00,16:29:00,li_1,1,2,,bri_1,2,4
ti_1557,1,16:29:00,16:29:00,,,si_1,0,1,,,,,
ti_1557,2,,,16:29:00,17:06:00,li_1,1,2,,bri_1,2,4
```

booking_rules.txt

```
booking_rule_id,booking_type,prior_notice_duration_min,phone_number,info_url,message
bri_1,1,60,+1 514 636 6666,https://coopouest.accestaxi.com,”Call at least 60 minutes before your trip”
```

[E2] Informal bus line: hail-and-ride frequency based



Description: In Accra (Ghana), a transit service will be operated on a street by drivers owning their vehicles. The addition of the individual service creates an informal hail-and-ride line.

GTFS-Flex features used:

- GTFS-FlexibleTrips: **yes**
- GTFS-BookingRules: no

Data:

stops.txt

```
stop_id,stop_name,stop_lon,stop_lat
si_1,Nsawam Road / N1,-0.22805213928222653,5.611714867868062
si_2,Accra Train Station,-0.2105426788330078,5.549013715023947
```

locations.geojson

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "id": "si_NsawamRd",
      "properties": {},
      "geometry": {
        "type": "LineString",
        "coordinates": [
          [-0.22805213928222653,5.611714867868062],
          [-0.22582054138183594,5.603172871471021],
          [-0.218353271484375,5.590274220204556],
          [-0.21680831909179688,5.576179342540755],
          [-0.2105426788330078,5.549013715023947]
        ]
      }
    }
  ]
}
```

trips.txt

```
route_id,service_id,trip_id,trip_headsign
ri_NsawamRd,Everyday,ti_NsawamRdS,Nsawam to Train Station
```

frequencies.txt

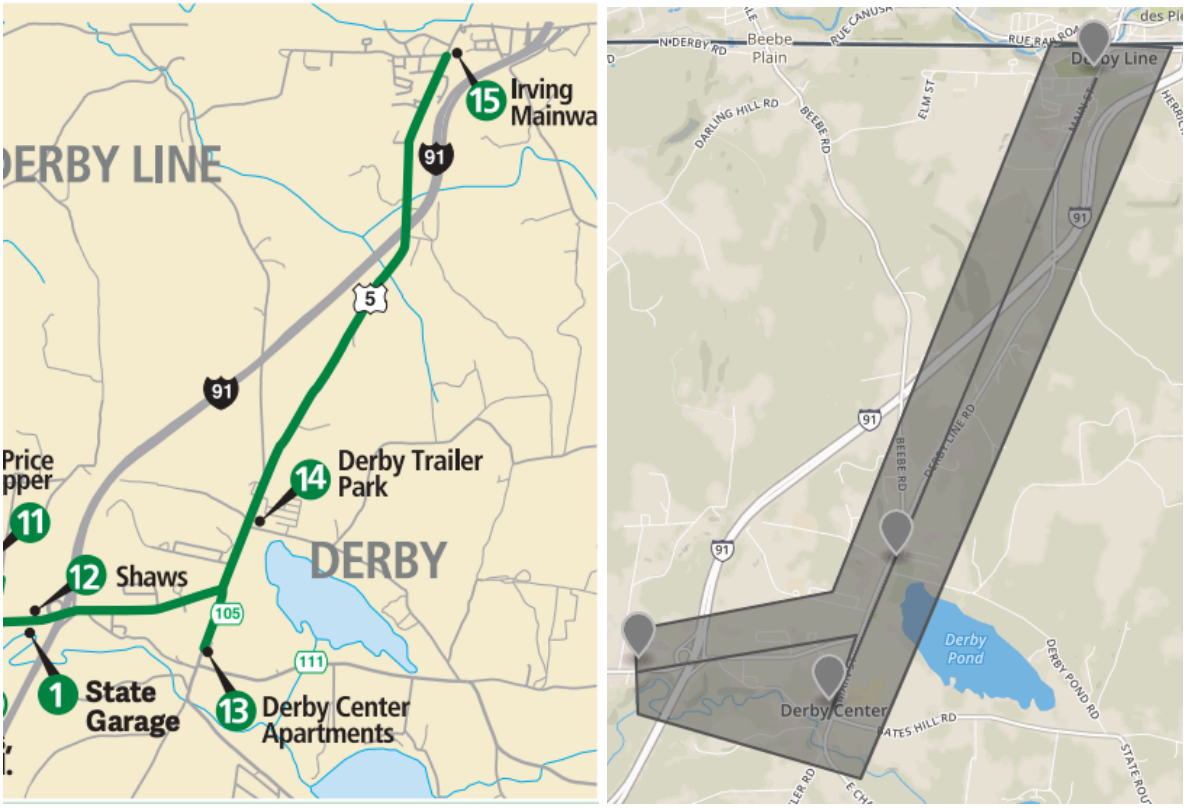
```
trip_id,start_time,end_time,headway_secs
ti_NsawamRdS,04:00:00,06:00:00,600
ti_NsawamRdS,06:00:00,09:00:00,300
ti_NsawamRdS,09:00:00,16:00:00,600
ti_NsawamRdS,16:00:00,19:00:00,300
ti_NsawamRdS,19:00:00,23:00:00,600
```

stop_times.txt

```
trip_id,stop_sequence,arrival_time,departure_time,stop_id,pickup_type,dr
op_off_type,mean_duration_factor,safe_duration_factor
ti_NsawamRdS,1,,,si_1,3,3
ti_NsawamRdS,2,,,si_NsawamRd,3,3,1.5,2
ti_NsawamRdS,3,,,si_2,3,3
[...]
```

[E4] RCT with continuous stops & route deviation

[Source: http://www.riderct.org/wp-content/uploads/2014/08/highlander_817.pdf



Description: This trip combines different types of services:

- stops are scheduled along the way, like a regular GTFS trip
- passenger can alight anywhere between stops
- passenger can book a detour if they live less than ¼ mile off of the shuttle route.

GTFS-Flex features used:

- GTFS-FlexibleTrips: **yes**
- GTFS-BookingRules: **yes**

Data:

stops.txt

```
stop_id,stop_name,stop_lon,stop_lat
si_0012,Shaw’s Park,-72.15751647949219,44.95265089681472
si_0013,Rt 105 & West St.,-72.13417053222656,44.949006285310986
si_0014,Derby Trailer Park,-72.12593078613281,44.961518484706204
si_0015,Irving Mainway,-72.10172653198242,45.003772531931666
```

locations.geojson

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "id": "li_buff",
      "properties": {},
      "geometry": {
        "type": "Polygon",
        "coordinates": [
          [
            [-72.1578598022461,44.95520198716179],
            [-72.15751647949219,44.947912856729495],
            [-72.13022232055664,44.94232389645868],
            [-72.09211349487305,45.00547175576768],
            [-72.10687637329102,45.005957239034736],
            [-72.1336555480957,44.95848179382986],
            [-72.1578598022461,44.95520198716179]
          ]
        ]
      }
    }
  ]
}
```

trips.txt

```
route_id,service_id,trip_id
817,MonSat,ti_817_0715
```

stop_times.txt

```
trip_id,stop_sequence,stop_id,to_stop_sequence,arrival_time,departure_ti
me,start_pickup_dropoff_window,end_pickup_dropoff_window,pickup_type,drop
_off_type,pickup_booking_rule_id,drop_off_booking_rule_id,mean_duration_f
```

```
actor,safe_duration_factor
[...]
ti_817_0715,1,li_0012,08:15:00,08:15:00,,,0,0,,
ti_817_0715,2,li_buff,,,08:15:00,08:50:00,2,2,bri_1,bri_1,1,2
ti_817_0715,3,li_0013,08:20:00,08:20:00,0,0,,
ti_817_0715,4,li_0014,08:25:00,08:25:00,0,0,,
ti_817_0715,5,li_0015,08:30:00,08:30:00,0,0,,
ti_817_0715,6,li_0014,08:40:00,08:40:00,0,0,,
ti_817_0715,7,li_0013,08:45:00,08:45:00,0,0,,
ti_817_0715,8,li_0012,08:50:00,08:50:00,0,0,,
[...]
```

booking_rules.txt

```
booking_rule_id,prior_notice_last_day,prior_notice_last_time,prior_notice
_start_day,prior_notice_start_time,prior_notice_service_id,message,phone_
number
bri_1,1,16:30:00,1,07:30:00,"The Highlander is a deviated shuttle route
that is Wheelchair Accessible, ADA Compliant and open to the General
Public. If you live within 1/4 mile of the shuttle route, have a
disability and would like to ride the shuttle, please call to schedule a
ride.",+1 802 334 0243
```