Was this helpful?

Reference

On this page

Caution: `GTFS-ContinuousStops` isn't yet available to all partners. Google Transit currently runs the `GTFS-ContinuousStops` specification experimentally with only a few existing cases. Google currently ignores the `GTFS-ContinuousStops` specifications that are in your feed.

This document defines the format and structure of the files that comprise a GTFS dataset.

Table of contents

- `levels.txt`
- `feed_info.txt`
- `translations.txt`
- `attributions.txt`

Term definitions

This section defines terms that are used throughout this document.

- **Dataset** - A complete set of files defined by this specification reference. Altering the dataset creates a new version of the dataset. Datasets should be published at a public, permanent URL, including the zip file name. (e.g., https://www.agency.org/gtfs/gtfs.zip).
- **Record** - A basic data structure comprised of a number of different field values describing a single entity (e.g. transit agency, stop, route, etc.). Represented, in a table, as a row.
- **Field** - A property of an object or entity. Represented, in a table, as a column.
- **Field Value** - An individual entry in a field. Represented, in a table, as a single cell.
- **Required** - The field must be included in the dataset, and a value must be provided in that field for each record. Some required fields permit an empty string as a value (denoted in this specification as empty). To enter an empty string, just omit any text between the commas for that field.
- **Optional** - The field may be omitted from the dataset. If an optional column is included, some of the entries in that field may be empty strings. To enter an empty string, just omit any text between the commas for that field. Note that an omitted field is equivalent to a field that is entirely empty.
- **Conditionally required** - The field or file is required under certain conditions, which are outlined in the field or file description. Outside of these conditions, this field or file is optional.
- **Service day** - A service day is a time period used to indicate route scheduling. The exact definition of service day varies from agency to agency but service days often do not correspond with calendar days. A service day may exceed 24:00:00 if service begins on one day and ends on a following day. For example, service that runs from 08:00:00 on Friday to 02:00:00 on Saturday, could be denoted as running from 08:00:00 to 26:00:00 on a single service day.

Field types

- **Color** - A color encoded as a six-digit hexadecimal number. Refer to https://htmlcolorcodes.com to generate a valid value (the leading "#" is not included).
  Example: `FFFFFF` *for white,* `000000` *for black or* `0039A6` *for the A,C,E lines in NYMTA.*
- **Currency Code** - An ISO 4217 alphabetical currency code. For the list of current currency, refer to https://en.wikipedia.org/wiki/ISO_4217#Active_codes.
  Example: `CAD` *for Canadian dollars,* `EUR` *for euros or* `JPY` *for Japanese yen.*
- **Date** - Service day in the `YYYYMMDD` format. Since time within a service day can be above 24:00:00, a service day often contains information for the subsequent day(s).
  Example: `20180913` *for September 13th, 2018.*
- **Email** - An email address.
  Example: `example@example.com`

- **Enum** - An option from a set of predefined constants defined in the "Description" column.
  Example: The `route_type` *field contains a* `0` *for tram, a* `1` *for subway...*
- **ID** - An ID field value is an internal ID, not intended to be shown to riders, and is a sequence of any UTF-8 characters. Using only printable ASCII characters is recommended. IDs defined in one `.txt` file are often referenced in another `.txt` file.
  Example: The `stop_id` *field in* `stops.txt` *is a ID. The* `stop_id` *field in* `stop_times.txt` *is an ID referencing* `stops.stop_id`.
- **Language Code** - An IETF BCP 47 language code. For an introduction to IETF BCP 47, refer to http://www.rfc-editor.org/rfc/bcp/bcp47.txt and http://www.w3.org/International/articles/language-tags/.
  Example: `en` *for English,* `en-US` *for American English or* `de` *for German.*
- **Latitude** - WGS84 latitude in decimal degrees. The value must be greater than or equal to `-90.0` and less than or equal to `90.0`.
  Example: `41.890169` *for the Colosseum in Rome.*
- **Longitude** - WGS84 longitude in decimal degrees. The value must be greater than or equal to `-180.0` and less than or equal to `180.0`.
  Example: `12.492269` *for the Colosseum in Rome.*
- **Non-negative Float** - A floating point number greater than or equal to `0`.
- **Non-negative Integer** - A integer greater than or equal to `0`.
- **Phone number** - A phone number.
- **Time** - Time in the HH:MM:SS format (H:MM:SS is also accepted). The time is measured from "noon minus 12h" of the service day (effectively midnight except for days on which daylight savings time changes occur. For more information, see the guidelines article). For times occurring after midnight, enter the time as a value greater than 24:00:00 in HH:MM:SS local time for the day on which the trip schedule begins.
  Example: `14:30:00` *for 2:30PM or* `25:35:00` *for 1:35AM on the next day.*
- **Text** - A string of UTF-8 characters, which is aimed to be displayed and which must therefore be human readable.
- **Timezone** - TZ timezone from the https://www.iana.org/time-zones. Timezone names never contain the space character but may contain an underscore. Refer to http://en.wikipedia.org/wiki/List_of_tz_zones for a list of valid values.
  Example: `Asia/Tokyo`, `America/Los_Angeles` *or* `Africa/Cairo`.
- **URL** - A fully qualified URL that includes http:// or https://, and any special characters in the URL must be correctly escaped. See the following http://www.w3.org/Addressing/URL/4_URI_Recommentations.html for a description of how to create fully qualified URL values.

Dataset files

This specification defines the following files:

Filename	Required	Defines
`agency.txt`	Required	Transit agencies with service represented in this dataset.
`stops.txt`	Required	Stops where vehicles pick up or drop off riders. Also defines stations and station entrances.

Filename	Required	Defines
routes.txt	Required	Transit routes. A route is a group of trips that are displayed to riders as a single service.
trips.txt	Required	Trips for each route. A trip is a sequence of two or more stops that occur during a specific time period.
stop_times.txt	Required	Times that a vehicle arrives at and departs from stops for each trip.
calendar.txt	Conditionally required	Service dates specified using a weekly schedule with start and end dates. This file is required unless all dates of service are defined in calendar_dates.txt.
calendar_dates.txt	Conditionally required	Exceptions for the services defined in the calendar.txt. If calendar.txt is omitted, then calendar_dates.txt is required and must contain all dates of service.
fare_attributes.txt	Optional	Fare information for a transit agency's routes.
fare_rules.txt	Optional	Rules to apply fares for itineraries.
shapes.txt	Optional	Rules for mapping vehicle travel paths, sometimes referred to as route alignments.
frequencies.txt	Optional	Headway (time between trips) for headway-based service or a compressed representation of fixed-schedule service.
transfers.txt	Optional	Rules for making connections at transfer points between routes.
pathways.txt	Optional	Pathways linking together locations within stations.
levels.txt	Optional	Levels within stations.
feed_info.txt	Conditionally required	Dataset metadata, including publisher, version, and expiration information.
translations.txt	Optional	Translated information of a transit agency.
attributions.txt	Optional	Specifies the attributions that are applied to the dataset.

File requirements

The following requirements apply to the format and contents of the dataset files:

- All files must be saved as comma-delimited text.
- The first line of each file must contain field names. Each subsection of the Field definitions section corresponds to one of the files in a GTFS dataset and lists the field names that may be used in that file.
- All field names are case-sensitive.
- Field values may not contain tabs, carriage returns or new lines.
- Field values that contain quotation marks or commas must be enclosed within quotation marks. In addition, each quotation mark in the field value must be preceded with a quotation mark. This is consistent with the manner in which Microsoft Excel outputs comma-delimited (CSV) files. For more information on the CSV file format, see http://tools.ietf.org/html/rfc4180. The following example demonstrates how a field value would appear in a comma-delimited file:
    - **Original field value:** `Contains "quotes", commas and text`
    - **Field value in CSV file:** `"Contains ""quotes"", commas and text"`

- Field values must not contain HTML tags, comments or escape sequences.
- Remove any extra spaces between fields or field names. Many parsers consider the spaces to be part of the value, which may cause errors.
- Each line must end with a CRLF or LF linebreak character.
- Files should be encoded in UTF-8 to support all Unicode characters. Files that include the Unicode byte-order mark (BOM) character are acceptable. See http://unicode.org/faq/utf_bom.html#BOM for more information on the BOM character and UTF-8.
- All dataset files must be zipped together.

Field definitions

agency.txt

File: **Required**

Field Name	Type	Required	Description
`agency_id`	ID	Conditionally Required	Identifies a transit brandwhich is often synonymous with a transit agency. Note that in some cases, such as when a single agency operates multiple separate services, agencies and brands are distinct. This document uses the term "agency" in place of "brand". A dataset may contain data from multiple agencies. This field is required when the dataset contains data for multiple transit agencies, otherwise it is optional.
`agency_name`	Text	Required	Full name of the transit agency.
`agency_url`	URL	Required	URL of the transit agency.
`agency_timezone`	Timezone	Required	Timezone where the transit agency is located. If multiple agencies are specified in the dataset, each must have the same `agency_timezone`.
`agency_lang`	Language code	Optional	Primary language used by this transit agency. This field helps GTFS consumers choose capitalization rules and other language-specific settings for the dataset.
`agency_phone`	Phone number	Optional	A voice telephone number for the specified agency. This field is a string value that presents the telephone number as typical for the agency's service area. It can and should contain punctuation marks to group the digits of the number. Dialable text (for example, TriMet's `503-238-RIDE`) is permitted, but the field must not contain any other descriptive text.
`agency_fare_url`	URL	Optional	URL of a web page that allows a rider to purchase tickets or other fare instruments for that agency online.
`agency_email`	Email	Optional	Email address actively monitored by the agency’s customer service department. This email address should be a direct contact point where transit riders can reach a customer service representative at the agency.

stops.txt

File: **Required**

Field Name	Type	Required	Description
`stop_id`	ID	Required	Identifies a stop, station, or station entrance. The term "station entrance" refers to both station entrances and station exits. Stops, stations or station entrances are collectively referred to as locations. Multiple routes may use the same stop.
`stop_code`	Text	Optional	Short text or a number that identifies the location for riders. These codes are often used in phone-based transit information systems or printed on signage to make it easier for riders to get information for a particular location. The `stop_code` can be the same as `stop_id` if it is public facing. This field should be left empty for locations without a code presented to riders.
`stop_name`	Text	Conditionally Required	Name of the location. Use a name that people will understand in the local and tourist vernacular. When the location is a boarding area (`location_type=4`), the `stop_name` should contains the name of the boarding area as displayed by the agency. It could be just one letter (like on some European intercity railway stations), or text like “Wheelchair boarding area” (NYC’s Subway) or “Head of short trains” (Paris’ RER). Conditionally Required: • Required for locations which are stops (`location_type=0`), stations (`location_type=1`) or entrances/exits (`location_type=2`). • Optional for locations which are generic nodes (`location_type=3`) or boarding areas (`location_type=4`).
`stop_desc`	Text	Optional	Description of the location that provides useful, quality information. Do not simply duplicate the name of the location.
`stop_lat`	Latitude	Conditionally Required	Latitude of the location. Conditionally Required: • Required for locations which are stops (`location_type=0`), stations (`location_type=1`) or entrances/exits (`location_type=2`). • Optional for locations which are generic nodes (`location_type=3`) or boarding areas (`location_type=4`).
`stop_lon`	Longitude	Conditionally Required	Longitude of the location. Conditionally Required: • Required for locations which are stops (`location_type=0`), stations (`location_type=1`) or entrances/exits (`location_type=2`). • Optional for locations which are generic

Field Name	Type	Required	Description
			nodes (`location_type=3`) or boarding areas (`location_type=4`).
`zone_id`	ID	Conditionally Required	Identifies the fare zone for a stop. This field is required if providing fare information using `fare_rules.txt`, otherwise it is optional. If this record represents a station or station entrance, the `zone_id` is ignored.
`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.
`location_type`	Enum	Optional	Type of the location: • `0` (or empty): Stop (or Platform). A location where passengers board or disembark from a transit vehicle. Is called a platform when defined within a `parent_station`. • `1`: Station. A physical structure or area that contains one or more platform. • `2`: Entrance/Exit. A location where passengers can enter or exit a station from the street. If an entrance/exit belongs to multiple stations, it can be linked by pathways to both, but the data provider must pick one of them as parent. • `3`: Generic Node. A location within a station, not matching any other `location_type`, which can be used to link together pathways define in `pathways.txt`. • `4`: Boarding Area. A specific location on a platform, where passengers can board and/or alight vehicles.
`parent_station`	ID referencing `stops.stop_id`	Conditionally Required	Defines hierarchy between the different locations defined in `stops.txt`. It contains the ID of the parent location, as followed: • Stop/platform (`location_type=0`): the `parent_station` field contains the ID of a station. • Station (`location_type=1`): this field must be empty. • Entrance/exit (`location_type=2`) or generic node (`location_type=3`): the `parent_station` field contains the ID of a station (`location_type=1`) • Boarding Area (`location_type=4`): the `parent_station` field contains ID of a platform. Conditionally Required: • Required for locations which are entrances (`location_type=2`), generic nodes (`location_type=3`) or boarding areas (`location_type=4`). • Optional for stops/platforms (`location_type=0`). • Forbidden for stations (`location_type=1`).
`stop_timezone`	Timezone	Optional	Timezone of the location. If the location has a parent station, it inherits the parent station's timezone instead of applying its own. Stations

Field Name	Type	Required	Description
			and parentless stops with empty `stop_timezone` inherit the timezone specified by `agency.agency_timezone`. If `stop_timezone` values are provided, the times in `stop_times.txt` should be entered as the time since midnight in the timezone specified by `agency.agency_timezone`. This ensures that the time values in a trip always increase over the course of a trip, regardless of which timezones the trip crosses.
`wheelchair_boarding`	Enum	Optional	Indicates whether wheelchair boardings are possible from the location. Valid options are: For parentless stops: `0` or empty - No accessibility information for the stop. `1` - Some vehicles at this stop can be boarded by a rider in a wheelchair. `2` - Wheelchair boarding is not possible at this stop. For child stops: `0` or empty - Stop will inherit its `wheelchair_boarding` behavior from the parent station, if specified in the parent. `1` - There exists some accessible path from outside the station to the specific stop/platform. `2` - There exists no accessible path from outside the station to the specific stop/platform. For station entrances/exits: `0` or empty - Station entrance will inherit its `wheelchair_boarding` behavior from the parent station, if specified for the parent. `1` - Station entrance is wheelchair accessible. `2` - No accessible path from station entrance to stops/platforms.
`level_id`	ID referencing `levels.level_id`	Optional	Level of the location. The same level can be used by multiple unlinked stations.
`platform_code`	Text	Optional	Platform identifier for a platform stop (a stop belonging to a station). This should be just the platform identifier (eg. `G` or `3`). Words like "platform" or "track" (or the feed's language-specific equivalent) should not be included. This allows feed consumers to more easily internationalize and localize the platform identifier into other languages.

routes.txt

File: **Required**

Field Name	Type	Required	Description
`route_id`	ID	Required	Identifies a route.

Field Name	Type	Required	Description
`agency_id`	ID referencing `agency.agency_id`	Conditionally required	Agency for the specified route. This field is required when the dataset provides data for routes from more than one agency in `agency.txt`, otherwise it is optional.
`route_short_name`	Text	Conditionally required	Short name of a route. This will often be a short, abstract identifier like "32", "100X", or "Green" that riders use to identify a route, but which doesn't give any indication of what places the route serves. Either `route_short_name` or `route_long_name` must be specified, or potentially both if appropriate.
`route_long_name`	Text	Conditionally required	Full name of a route. This name is generally more descriptive than the `route_short_name` and often includes the route's destination or stop. Either `route_short_name` or `route_long_name` must be specified, or potentially both if appropriate.
`route_desc`	Text	Optional	Description of a route that provides useful, quality information. Do not simply duplicate the name of the route. *Example: "A" trains operate between Inwood-207 St, Manhattan and Far Rockaway-Mott Avenue, Queens at all times. Also from about 6AM until about midnight, additional "A" trains operate between Inwood-207 St and Lefferts Boulevard (trains typically alternate between Lefferts Blvd and Far Rockaway).*
`route_type`	Enum	Required	Indicates the type of transportation used on a route. Valid options are: `0` - Tram, Streetcar, Light rail. Any light rail or street level system within a metropolitan area. `1` - Subway, Metro. Any underground rail system within a metropolitan area. `2` - Rail. Used for intercity or long-distance travel. `3` - Bus. Used for short- and long-distance bus routes. `4` - Ferry. Used for short- and long-distance boat service. `5` - Cable tram. Used for street-level rail cars where the cable runs beneath the vehicle, e.g., cable car in San Francisco. `6` - Aerial lift, suspended cable car (e.g., gondola lift, aerial tramway). Cable transport where cabins, cars, gondolas or open chairs are suspended by means of one or more cables. `7` - Funicular. Any rail system designed for steep inclines. `11` - Trolleybus. Electric buses that draw power from overhead wires using poles. `12` - Monorail. Railway in which the track consists of a single rail or a beam.

Field Name	Type	Required	Description
route_url	URL	Optional	URL of a web page about the particular route. Should be different from the agency.agency_url value.
route_color	Color	Optional	Route color designation that matches public facing material. Defaults to white (FFFFFF) when omitted or left empty. The color difference between route_color and route_text_color should provide sufficient contrast when viewed on a black and white screen.
route_text_color	Color	Optional	Legible color to use for text drawn against a background of route_color. Defaults to black (000000) when omitted or left empty. The color difference between route_color and route_text_color should provide sufficient contrast when viewed on a black and white screen.
route_sort_order	Non-negative integer	Optional	Orders the routes in a way which is ideal for presentation to customers. Routes with smaller route_sort_order values should be displayed first.
continuous_pickup	Enum	Optional	Indicates whether a rider can board the transit vehicle anywhere along the vehicle's travel path. The path is described by shapes.txt on every trip of the route. Valid options are: 0 - Continuous stopping pickup. 1 or empty - No continuous stopping pickup. 2 - Must phone an agency to arrange continuous stopping pickup. 3 - Must coordinate with a driver to arrange continuous stopping pickup. The default continuous pickup behavior defined in routes.txt can be overridden in stop_times.txt.
continuous_drop_off	Enum	Optional	Indicates whether a rider can alight from the transit vehicle at any point along the vehicle's travel path. The path is described by shapes.txt on every trip of the route. Valid options are: 0- Continuous stopping drop-off. 1 or empty - No continuous stopping drop-off. 2 - Must phone an agency to arrange continuous stopping drop-off. 3 - Must coordinate with a driver to arrange continuous stopping drop-off. The default continuous drop-off behavior defined in routes.txt can be overridden in stop_times.txt.

trips.txt

File: **Required**

Field Name	Type	Required	Description
`route_id`	ID referencing `routes.route_id`	Required	Identifies a route.
`service_id`	ID referencing `calendar.service_id` or `calendar_dates.service_id`	Required	Identifies a set of dates when ser for one or more routes.
`trip_id`	ID	Required	Identifies a trip.
`trip_headsign`	Text	Optional	Text that appears on signage ider destination to riders. Use this fie between different patterns of ser route. If the headsign changes du `trip_headsign` can be overrid values for the `stop_times.sto`
`trip_short_name`	Text	Optional	Public facing text used to identif for instance, to identify train nur commuter rail trips. If riders do on trip names, leave this field en `trip_short_name` value, if pr uniquely identify a trip within a should not be used for destinatio limited/express designations.

`direction_id`	Enum	Optional	Indicates the direction of travel f field is not used in routing; it pr separate trips by direction when tables. Valid options are:
			
			`0` - Travel in one direction (e.g. `1` - Travel in the opposite directi travel).
			
			Example: The `trip_headsign` `direction_id` *fields could be u assign a name to travel in each d of trips. A* `trips.txt` *file could records for use in time tables:*
			`trip_id,...,trip_headsig`
			`1234,...,Airport,0`
			`1505,...,Downtown,1`

`block_id`	ID	Optional	Identifies the block to which the block consists of a single trip or trips made using the same vehicl shared service days and `block_` can have trips with different serv distinct blocks. See the example
`shape_id`	ID referencing `shapes.shape_id`	Conditionally required	Identifies a geospatial shape that vehicle travel path for a trip. Conditionally required: This field is required if the trip h behavior defined, either at the ro stop time level. Otherwise, it's optional.
`wheelchair_accessible`	Enum	Optional	Indicates wheelchair accessibility are:

Field Name	Type	Required	Description
			`0` or empty - No accessibility inf trip. `1` - Vehicle being used on this pa accommodate at least one rider i `2` - No riders in wheelchairs can on this trip.
`bikes_allowed`	Enum	Optional	Indicates whether bikes are allow are: `0` or empty - No bike informatio `1` - Vehicle being used on this pa accommodate at least one bicycl `2` - No bicycles are allowed on t

Example: Blocks and service day

The example below is valid, with distinct blocks every day of the week.

`route_id`	`trip_id`	`service_id`	`block_id`	*(first stop time)*	*(last stop time)*
`red`	`trip_1`	`mon-tues-wed-thurs-fri-sat-sun`	`red_loop`	22:00:00	22:55:00
`red`	`trip_2`	`fri-sat-sun`	`red_loop`	23:00:00	23:55:00
`red`	`trip_3`	`fri-sat`	`red_loop`	24:00:00	24:55:00
`red`	`trip_4`	`mon-tues-wed-thurs`	`red_loop`	20:00:00	20:50:00
`red`	`trip_5`	`mon-tues-wed-thurs`	`red_loop`	21:00:00	21:50:00

Notes on above table:

- On Friday into Saturday morning, for example, a single vehicle operates `trip_1`, `trip_2`, and `trip_3` (10:00 PM through 12:55 AM). Note that the last trip occurs on Saturday, 12:00 AM to 12:55 AM, but is part of the Friday "service day" because the times are 24:00:00 to 24:55:00.
- On Monday, Tuesday, Wednesday, and Thursday, a single vehicle operates `trip_1`, `trip_4`, and `trip_5` in a block from 8:00 PM to 10:55 PM.

stop_times.txt

File: **Required**

Field Name	Type	Required	Description
`trip_id`	ID referencing `trips.trip_id`	Required	Identifies a trip.
`arrival_time`	Time	Conditionally required	Arrival time at a specific stop for a specific trip on a route. If there are not separate times for arrival and departure at a stop, enter the same value for `arrival_time` and `departure_time`. For times occurring after midnight on the service day, enter the time as a value greater than 24:00:00 in HH:MM:SS local

Field Name	Type	Required	Description
			time for the day on which the trip schedule begins. Scheduled stops where the vehicle strictly adheres to the specified arrival and departure times are timepoints. If this stop is not a timepoint, it is recommended to provide an estimated or interpolated time. If this is not available, `arrival_time` can be left empty. Further, indicate that interpolated times are provided with `timepoint=0`. If interpolated times are indicated with `timepoint=0`, then time points must be indicated with `timepoint=1`. Provide arrival times for all stops that are time points. An arrival time must be specified for the first and the last stop in a trip.
`departure_time`	Time	Conditionally required	Departure time from a specific stop for a specific trip on a route. For times occurring after midnight on the service day, enter the time as a value greater than 24:00:00 in HH:MM:SS local time for the day on which the trip schedule begins. If there are not separate times for arrival and departure at a stop, enter the same value for `arrival_time` and `departure_time`. See the `arrival_time` description for more details about using timepoints correctly. The `departure_time` field should specify time values whenever possible, including non-binding estimated or interpolated times between timepoints.
`stop_id`	ID referencing `stops.stop_id`	Required	Identifies the serviced stop. All stops serviced during a trip must have a record in `stop_times.txt`. Referenced locations must be stops, not stations or station entrances. A stop may be serviced multiple times in the same trip, and multiple trips and routes may service the same stop.
`stop_sequence`	Non-negative integer	Required	Order of stops for a particular trip. The values must increase along the trip but do not need to be consecutive. *Example: The first location on the trip could have a `stop_sequence=1`, the second location on the trip could have a `stop_sequence=23`, the third location could have a `stop_sequence=40`, and so on.*
`stop_headsign`	Text	Optional	Text that appears on signage identifying the trip's destination to riders. This field overrides the default `trips.trip_headsign` when the headsign changes between stops. If the headsign is displayed for an entire trip, use `trips.trip_headsign` instead. A `stop_headsign` value specified for one `stop_time` does not apply to subsequent `stop_times` in the same trip. If you want to override the `trip_headsign` for multiple `stop_times` in the same trip, the

Field Name	Type	Required	Description
			`stop_headsign` value must be repeated in each `stop_time` row.
`pickup_type`	Enum	Optional	Indicates pickup method. Valid options are: `0` or empty - Regularly scheduled pickup. `1` - No pickup available. `2` - Must phone agency to arrange pickup. `3` - Must coordinate with driver to arrange pickup.
`drop_off_type`	Enum	Optional	Indicates drop off method. Valid options are: `0` or empty - Regularly scheduled drop off. `1` - No drop off available. `2` - Must phone agency to arrange drop off. `3` - Must coordinate with driver to arrange drop off.
`continuous_pickup`	Enum	Optional	Indicates whether a rider can board the transit vehicle at any point along the vehicle's travel path. The path is described by `shapes.txt`, from this `stop_time` to the next `stop_time` in the trip's `stop_sequence`. Valid options are: `0` - Continuous stopping pickup. `1` or empty - No continuous stopping pickup. `2` - Must phone an agency to arrange continuous pickup. `3` - Must coordinate with a driver to arrange continuous stopping pickup. The continuous pickup behavior indicated in `stop_times.txt` overrides any behavior defined in `routes.txt`.
`continuous_drop_off`	Enum	Optional	Indicates whether a rider can alight from the transit vehicle at any point along the vehicle's travel path as described by `shapes.txt`, from this `stop_time` to the next `stop_time` in the trip's `stop_sequence`. `0` - Continuous stopping drop off. `1` or empty - No continuous stopping drop off. `2` - Must phone an agency to arrange continuous drop off. `3` - Must coordinate with a driver to arrange continuous stopping drop off. The continuous drop-off behavior indicated in `stop_times.txt` overrides any behavior defined in `routes.txt`.
`shape_dist_traveled`	Non-negative float	Optional	Actual distance traveled along the associated shape, from the first stop to the stop specified in this record. This field specifies how much of the shape to draw between any two stops during a trip. Must be in the same units used in `shapes.txt`. Values used for `shape_dist_traveled` must increase along with `stop_sequence`; they cannot be used to show reverse travel along a route.

Field Name	Type	Required	Description
			Example: If a bus travels a distance of 5.25 kilometers from the start of the shape to the stop, `shape_dist_traveled=5.25`.
`timepoint`	Enum	Optional	Indicates if arrival and departure times for a stop are strictly adhered to by the vehicle or if they are instead approximate and/or interpolated times. This field allows a GTFS producer to provide interpolated stop-times, while indicating that the times are approximate. Valid options are: `0` - Times are considered approximate. `1` or empty - Times are considered exact.

calendar.txt

File: **Conditionally required**

Field Name	Type	Required	Description
`service_id`	ID	Required	Uniquely identifies a set of dates when service is available for one or more routes. Each `service_id` value can appear at most once in a `calendar.txt` file.
`monday`	Enum	Required	Indicates whether the service operates on all Mondays in the date range specified by the `start_date` and `end_date` fields. Note that exceptions for particular dates may be listed in `calendar_dates.txt`. Valid options are: `1` - Service is available for all Mondays in the date range. `0` - Service is not available for Mondays in the date range.
`tuesday`	Enum	Required	Functions in the same way as `monday` except applies to Tuesdays
`wednesday`	Enum	Required	Functions in the same way as `monday` except applies to Wednesdays
`thursday`	Enum	Required	Functions in the same way as `monday` except applies to Thursdays
`friday`	Enum	Required	Functions in the same way as `monday` except applies to Fridays
`saturday`	Enum	Required	Functions in the same way as `monday` except applies to Saturdays.
`sunday`	Enum	Required	Functions in the same way as `monday` except applies to Sundays.
`start_date`	Date	Required	Start service day for the service interval.
`end_date`	Date	Required	End service day for the service interval. This service day is included in the interval.

calendar_dates.txt

File: **Conditionally required**

The `calendar_dates.txt` table can explicitly activate or disable service by date. It can be used in two ways.

- Recommended: Use `calendar_dates.txt` in conjunction with `calendar.txt` to define exceptions to the default service patterns defined in `calendar.txt`. If service is generally regular, with a few changes on explicit dates (for instance, to accommodate special event services, or a school schedule), this is a good approach. In this case `calendar_dates.service_id` is an

ID referencing `calendar.service_id`.

- Alternate: Omit `calendar.txt`, and specify each date of service in `calendar_dates.txt`. This allows for considerable service variation and accommodates service without normal weekly schedules. In this case `service_id` is an ID.

Field Name	Type	Required	Description
`service_id`	ID referencing `calendar.service_id` or ID	Required	Identifies a set of dates when a service exception occurs for one or more routes. Each (`service_id`, `date`) pair can only appear once in `calendar_dates.txt` if using `calendar.txt` and `calendar_dates.txt` in conjunction. If a `service_id` value appears in both `calendar.txt` and `calendar_dates.txt`, the information in `calendar_dates.txt` modifies the service information specified in `calendar.txt`.
`date`	Date	Required	Date when service exception occurs.
`exception_type`	Enum	Required	Indicates whether service is available on the date specified in the `date` field. Valid options are: `1` - Service has been added for the specified date. `2` - Service has been removed for the specified date. *Example: Suppose a route has one set of trips available on holidays and another set of trips available on all other days. One `service_id` could correspond to the regular service schedule and another `service_id` could correspond to the holiday schedule. For a particular holiday, the `calendar_dates.txt` file could be used to add the holiday to the holiday `service_id` and to remove the holiday from the regular `service_id` schedule.*

fare_attributes.txt

File: **Optional**

Field Name	Type	Required	Description
`fare_id`	ID	Required	Identifies a fare class.
`price`	Non-negative float	Required	Fare price, in the unit specified by `currency_type`.
`currency_type`	Currency code	Required	Currency used to pay the fare.
`payment_method`	Enum	Required	Indicates when the fare must be paid. Valid options are: `0` - Fare is paid on board. `1` - Fare must be paid before boarding.
`transfers`	Enum	Required	Indicates the number of transfers permitted on this fare. The fact that this field can be left empty is an exception to the requirement that a Required field must not be empty. Valid options are:

Field Name	Type	Required	Description
			`0` - No transfers permitted on this fare. `1` - Riders may transfer once. `2` - Riders may transfer twice. empty - Unlimited transfers are permitted.
`agency_id`	ID referencing `agency.agency_id`	Conditionally Required	Identifies the relevant agency for a fare. This field is required for datasets with multiple agencies defined in `agency.txt`, otherwise it is optional.
`transfer_duration`	Non-negative integer	Optional	Length of time in seconds before a transfer expires. When `transfers=0` this field can be used to indicate how long a ticket is valid for or it can can be left empty.

fare_rules.txt

File: **Optional**

The `fare_rules.txt` table specifies how fares in `fare_attributes.txt` apply to an itinerary. Most fare structures use some combination of the following rules:

- Fare depends on origin or destination stations.
- Fare depends on which zones the itinerary passes through.
- Fare depends on which route the itinerary uses.

For examples that demonstrate how to specify a fare structure with `fare_rules.txt` and `fare_attributes.txt`, see https://code.google.com/p/googletransitdatafeed/wiki/FareExamples in the GoogleTransitDataFeed open source project wiki.

Field Name	Type	Required	Description
`fare_id`	ID referencing `fare_attributes.fare_id`	Required	Identifies a fare class.

`route_id`	ID referencing `routes.route_id`	Optional	Identifies a route associated with the fare class. If with the same fare attributes exist, create a record `fare_rules.txt` for each route.
			
			Example: If fare class "b" is valid on route "TSW `fare_rules.txt` file would contain these recor class:
			`fare_id,route_id`
			`b,TSW`
			`b,TSE`

`origin_id`	ID referencing `stops.zone_id`	Optional	Identifies an origin zone. If a fare class has multi create a record in `fare_rules.txt` for each or
			
			Example: If fare class "b" is valid for all travel o either zone "2" or zone "8", the `fare_rules.tx` contain these records for the fare class:
			`fare_id,...,origin_id`
			`b,...,2`
			`b,...,8`

`destination_id`	ID referencing	Optional	Identifies a destination zone. If a fare class has m

Field Name	Type	Required	Description

	`stops.zone_id`		zones, create a record in `fare_rules.txt` for e `destination_id`.
			
			Example: The `origin_id` *and* `destination_i` *used together to specify that fare class "b" is valic between zones 3 and 4, and for travel between zor* `fare_rules.txt` *file would contain these recor class:*
			`fare_id,...,origin_id,destination_id`
			`b,...,3,4`
			`b,...,3,5`

`contains_id`	ID referencing `stops.zone_id`	Optional	Identifies the zones that a rider will enter while u class. Used in some systems to calculate correct fa
			
			Example: If fare class "c" is associated with all tr route that passes through zones 5, 6, and 7 the `fa` *would contain these records:*
			`fare_id,route_id,...,contains_id`
			`c,GRT,...,5`
			`c,GRT,...,6`
			`c,GRT,...,7`
			Because all `contains_id` *zones must be matche apply, an itinerary that passes through zones 5 an would not have fare class "c". For more detail, se https://code.google.com/p/googletransitdatafeed/ in the GoogleTransitDataFeed project wiki.*

shapes.txt

File: **Optional**

Shapes describe the path that a vehicle travels along a route alignment, and are defined in the file `shapes.txt`. Shapes are associated with Trips, and consist of a sequence of points through which the vehicle passes in order. Shapes do not need to intercept the location of Stops exactly, but all Stops on a trip should lie within a small distance of the shape for that trip, i.e. close to straight line segments connecting the shape points.

Field Name	Type	Required	Description
`shape_id`	ID	Required	Identifies a shape.
`shape_pt_lat`	Latitude	Required	Latitude of a shape point. Each record in `shapes.txt` represer shape.
`shape_pt_lon`	Longitude	Required	Longitude of a shape point.
`shape_pt_sequence`	Non-negative integer	Required	Sequence in which the shape points connect to form the shape. ` but do not need to be consecutive.  *Example: If the shape "A_shp" has three points in its definition, contain these records to define the shape:* `shape_id,shape_pt_lat,shape_pt_lon,shape_pt_seq` `A_shp,37.61956,-122.48161,0` `A_shp,37.64430,-122.41070,6` `A_shp,37.65863,-122.30839,11`
`shape_dist_traveled`	Non-negative float	Optional	Actual distance traveled along the shape from the first shape poi record. Used by trip planners to show the correct portion of the increase along with `shape_pt_sequence`; they cannot be used

Field Name	Type	Required	Description

			route. Distance units must be consistent with those used in `stop`
			
			Example: If a bus travels along the three points defined above f
			`shape_dist_traveled` *values (shown here in kilometers) wo*
			`shape_id,shape_pt_lat,shape_pt_lon,shape_pt_seq`
			`A_shp,37.61956,-122.48161,0,0`
			`A_shp,37.64430,-122.41070,6,6.8310`
			`A_shp,37.65863,-122.30839,11,15.8765`

frequencies.txt

File: **Optional**

The `frequencies.txt` file represents trips that operate on regular headways (time between trips). This file can be used to represent two different types of service.

- Frequency-based service (`exact_times=0`) in which service does not follow a fixed schedule throughout the day. Instead, operators attempt to strictly maintain predetermined headways for trips.
- A compressed representation of schedule-based service (`exact_times=1`) that has the exact same headway for trips over specified time period(s). In schedule-based service operators try to strictly adhere to a schedule.

Field Name	Type	Required	Description
`trip_id`	ID referencing `trips.trip_id`	Required	Identifies a trip to which the specified headway of service applies.
`start_time`	Time	Required	Time at which the first vehicle departs from the first stop of the trip with the specified headway.
`end_time`	Time	Required	Time at which service changes to a different headway (or ceases) at the first stop in the trip.
`headway_secs`	Non-negative integer	Required	Time, in seconds, between departures from the same stop (headway) for the trip, during the time interval specified by `start_time` and `end_time`. Multiple headways for the same trip are allowed, but may not overlap. New headways may start at the exact time the previous headway ends.
`exact_times`	Enum	Optional	Indicates the type of service for a trip. See the file description for more information. Valid options are: `0` or empty - Frequency-based trips. `1` - Schedule-based trips with the exact same headway throughout the day. In this case the `end_time` value must be greater than the last desired trip `start_time` but less than the last desired trip `start_time` + `headway_secs`.

transfers.txt

File: **Optional**

When calculating an itinerary, GTFS-consuming applications interpolate transfers based on allowable time and stop proximity. The `transfers.txt` file specifies additional rules and overrides for selected transfers.

Field Name	Type	Required	Description
`from_stop_id`	ID referencing `stops.stop_id`	Required	Identifies a stop or station where a connection between routes begins. If this field refers to a station, the transfer rule applies to all its child stops.
`to_stop_id`	ID referencing `stops.stop_id`	Required	Identifies a stop or station where a connection between routes ends. If this field refers to a station, the transfer rule applies to all child stops.
`transfer_type`	Enum	Required	Indicates the type of connection for the specified (`from_stop_id`, `to_stop_id`) pair. Valid options are: `0` or empty - Recommended transfer point between routes. `1` - Timed transfer point between two routes. The departing vehicle is expected to wait for the arriving one and leave sufficient time for a rider to transfer between routes. `2` - Transfer requires a minimum amount of time between arrival and departure to ensure a connection. The time required to transfer is specified by `min_transfer_time`. `3` - Transfers are not possible between routes at the location.
`min_transfer_time`	Non-negative integer	Optional	Amount of time, in seconds, that must be available to permit a transfer between routes at the specified stops. The `min_transfer_time` should be sufficient to permit a typical rider to move between the two stops, including buffer time to allow for schedule variance on each route.

pathways.txt

File: **Optional**

The GTFS-Pathways extension uses a graph representation to describe subway or train, with nodes (the locations) and edges (the pathways).

To go from the entrance (which is a node represented as a location with `location_type=2`) to a platform (which is a node represented as a location with `location_type=0`), the rider will go through walkway, fare gates, stairs, etc (which are edges represented as pathways). The proposal also adds another type of location, a generic one called "generic node", to represent for example a walkway crossing from which different walkways can be taken.

Warning: Pathways must be exhaustive in a station. As consequence, as soon as one platform (as stop), entrance or node belonging to a station has a pathway linked to it, the station is assumed to have exhaustive description of its pathways. Therefore, the following common sense rules apply:

- No dangling location: If any location within a station has a pathway, then all locations should have pathways (except for those platforms that have boarding areas).
- No locked platforms: Each platform must be connected to at least one entrance via some chain of pathways. There are very rare stations in the real life where you cannot go outside.
- No pathways for a platform with boarding areas: A platform that has boarding areas is treated as a parent object, not a point. It may not have pathways assigned. All pathways should be for boarding areas.

Field Name	Type	Required	Description
`pathway_id`	ID	Required	The `pathway_id` field contains an ID that uniquely identifies the pathway. The `pathway_id` is used by systems as an internal identifier of this record (e.g., primary key in database), and therefore the `pathway_id` must be dataset unique. Different pathways can go from the same `from_stop_id` to the same `to_stop_id`. For example, this happens when two escalators are side by side in opposite direction, or when a stair is nearby and elevator and both go from the same place to the same place.
`from_stop_id`	ID referencing `stops.stop_id`	Required	Location at which the pathway begins. It contains a `stop_id` that identifies a platform, entrance/exit, generic node or boarding area from the `stops.txt` file.
`to_stop_id`	ID referencing `stops.stop_id`	Required	Location at which the pathway ends. It contains a `stop_id` that identifies a platform, entrance/exit, generic node or boarding area from the `stops.txt` file.
`pathway_mode`	Enum	Required	Type of pathway between the specified (`from_stop_id`, `to_stop_id`) pair. Valid values for this field are: • `1`: walkway • `2`: stairs • `3`: moving sidewalk/travelator • `4`: escalator • `5`: elevator • `6`: fare gate (or payment gate): A pathway that crosses into an area of the station where a proof of payment is required (usually via a physical payment gate). Fare gates may either separate paid areas of the station from unpaid ones, or separate different payment areas within the same station from each other. This information can be used to avoid routing passengers through stations using shortcuts that would require passengers to make unnecessary payments, like directing a passenger to walk through a subway platform to reach a busway. • `7`: exit gate: Indicates a pathway exiting an area where proof-of-payment is required into an area where proof-of-payment is no longer required.
`is_bidirectional`	Enum	Required	Indicates in which direction the pathway can be used: • `0`: Unidirectional pathway, it can only be used from `from_stop_id` to `to_stop_id`. • `1`: Bidirectional pathway, it can be used in the two directions. Fare gates (`pathway_mode=6`) and exit gates (`pathway_mode=7`) cannot be bidirectional.
`length`	Non-negative Float	Optional	Horizontal length in meters of the pathway from the origin location (defined in `from_stop_id`) to the destination location (defined in `to_stop_id`).

Field Name	Type	Required	Description
			This field is recommended for walkways (`pathway_mode=1`), fare gates (`pathway_mode=6`) and exit gates (`pathway_mode=7`).
`traversal_time`	Positive Integer	Optional	Average time in seconds needed to walk through the pathway from the origin location (defined in `from_stop_id`) to the destination location (defined in `to_stop_id`). This field is recommended for moving sidewalks (`pathway_mode=3`), escalators (`pathway_mode=4`) and elevator (`pathway_mode=5`).
`stair_count`	Non-null Integer	Optional	Number of stairs of the pathway. Best Practices: one could use the approximation of 1 floor = 15 stairs to generate approximative values. A positive `stair_count` implies that the rider walk up from `from_stop_id` to `to_stop_id`. And a negative `stair_count` implies that the rider walk down from `from_stop_id` to `to_stop_id`. This field is recommended for stairs (`pathway_mode=2`).
`max_slope`	Float	Optional	Maximum slope ratio of the pathway. Valid values for this field are: • `0` or (empty): no slope. • A float: slope ratio of the pathway, positive for upwards, negative for downwards. This field should be used only with walkways (`pathway_type=1`) and moving sidewalks (`pathway_type=3`). Example: In the US, 0.083 (also written 8.3%) is the maximum slope ratio for hand-propelled wheelchair, which mean an increase of 0.083m (so 8.3cm) for each 1m.
`min_width`	Positive Float	Optional	Minimum width of the pathway in meters. This field is highly recommended if the minimum width is less than 1 meter.
`signposted_as`	Text	Optional	String of text from physical signage visible to transit riders. The string can be used to provide text directions to users, such as 'follow signs to '. The language text should appear in this field exactly how it is printed on the signs - it should not be translated.
`reversed_signposted_as`	Text	Optional	Same than the `signposted_as` field, but when the pathways is used backward, i.e. from the `to_stop_id` to the `from_stop_id`.

levels.txt

File: **Optional**

Describe the different levels of a station. Is mostly useful when used in conjunction with `pathways.txt`, and is required for elevator (`pathway_mode=5`) to ask the user to take the elevator to the "Mezzanine" or the "Platform" level.

Field Name	Type	Required	Description
`level_id`	ID	Required	Id of the level that can be referenced from `stops.txt`.
`level_index`	Float	Required	Numeric index of the level that indicates relative position of this level in relation to other levels (levels with higher indices are assumed to be located above levels with lower indices). Ground level should have index `0`, with levels above ground indicated by positive indices and levels below ground by negative indices.
`level_name`	Text	Optional	Optional name of the level (that matches level lettering/numbering used inside the building or the station). Is useful for elevator routing (e.g. "take the elevator to level "Mezzanine" or "Platforms" or "-1").

feed_info.txt

File: **Conditionally required**

This file contains information about the dataset itself, rather than the services the dataset describes. In some cases, the publisher of the dataset differs from the agencies. If `translations.txt` is provided, this file is required.

Field Name	Type	Required	Description
`feed_publisher_name`	Text	Required	Full name of the organization that publishes the dataset. This might be the same as one of the `agency.agency_name` values.
`feed_publisher_url`	URL	Required	URL of the dataset publishing organization's website. This may be the same as one of the `agency.agency_url` values.
`feed_lang`	Language code	Required	Default language for the text in this dataset. This setting helps GTFS consumers choose capitalization rules and other language-specific settings for the dataset. To define another language, use the `language` field in `translations.txt`. Multilingual datasets might be the default language with the original text in multiple languages. In such cases, use the ISO 639-2 language code `mul` in the `feed_lang` field. Provide a translation for each of the languages used in the dataset in `translations.txt`. If all of the original text in the dataset is in the same language, don't use `mul`. For example, a dataset in Switzerland might set the original `stops.stop_name` field populated with stop names in different languages. Each stop name is written in accordance with the dominant language in that stop's geographic location. Stop names include Genève for the French-speaking city of Geneva, Zürich for the German-speaking city of Zurich, and Biel/Bienne for the bilingual city of Biel/Bienne. Set `feed_lang=mul` and provide the following translations in `translations.txt`:

Field Name	Type	Required	Description
			• German: "Genf," "Zürich," and "Biel" • French: "Genève," "Zurich," and "Bienne" • Italian: "Ginevra," "Zurigo," and "Bienna" • English: "Geneva," "Zurich," and "Biel/Bienne"
`default_lang`	Language code	Optional	Defines the language used when the data consumer doesn't know the language of the rider. It's often defined as `en`, English.
`feed_start_date`	Date	Optional	The dataset provides complete and reliable schedule information for service in the period from the beginning of the `feed_start_date` day to the end of the `feed_end_date` day. Both days can be left empty if unavailable. The `feed_end_date` date must not precede the `feed_start_date` date if both are given. Dataset providers are encouraged to give schedule data outside this period to advise of likely future service, but dataset consumers should treat it mindful of its non-authoritative status. If `feed_start_date` or `feed_end_date` extend beyond the active calendar dates defined in `calendar.txt` and `calendar_dates.txt`, the dataset is making an explicit assertion that there is no service for dates within the `feed_start_date` to `feed_end_date` range but not included in the active calendar dates.
`feed_end_date`	Date	Optional	Refer to the `feed_start_date` row in this table.
`feed_version`	Text	Optional	String that indicates the current version of their GTFS dataset. GTFS-consuming applications can display this value to help dataset publishers determine whether the latest dataset has been incorporated.
`feed_contact_email`	Email	Optional	Email address for communication regarding the GTFS dataset and data publishing practices. `feed_contact_email` is a technical contact for GTFS-consuming applications. Provide customer service contact information through `agency.txt`.
`feed_contact_url`	URL	Optional	URL for contact information, a web-form, support desk, or other tools for communication regarding the GTFS dataset and data publishing practices. `feed_contact_url` is a technical contact for GTFS-consuming applications. Provide customer service contact information through `agency.txt`.

translations.txt

File: **Optional**

Field Name	Type	Required	Description
`table_name`	Enum	Required	Defines the dataset table that contains the field to be translated. The following values are allowed: • `agency` • `stops` • `routes` • `trips` • `stop_times` • `feed_info` • `pathways` • `levels` • `attributions` ★ Note: Don't include the **`.txt`** file extension after the table's name.
`field_name`	Text	Required	Provides the name of the field to be translated. Fields with the type "Text" can be translated, while fields with the types "URL," "Email," and "Phone number" can be included here to provide those resources in the correct language. ★ Note: Fields with other types are ignored and will not be translated.
`language`	Language code	Required	Provides the language of translation. If this language is the same as the one from `feed_lang` in `feed_info.txt`, the original value of the field is the default value used in languages without specific translations. For example, in Switzerland, a city in a bilingual canton is officially called "Biel/Bienne," but it would simply be called "Bienne" in French and "Biel" in German.
`translation`	Text, URL, Email, or Phone number	Required	Provides the translated value for the specified `field_name`.

`record_id`	ID	Conditionally Required	Defines the record that corresponds to the field to be translated. The value in `record_id` needs to be a main ID from a dataset table, as defined in the following table:
			`table_name` → `record_id`
			`agency` → `agency_id`
			`stops` → `stop_id`
			`routes` → `route_id`
			`trips` → `trip_id`
			`stop_times` → `trip_id`

<table>
<tr><th>Field Name</th><th>Type</th><th>Required</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td>pathways</td><td>pathway_id</td></tr>
<tr><td>levels</td><td>level_id</td></tr>
<tr><td>attributions</td><td>attribution_id</td></tr>
</table>
The following conditions determine how this field can be used:

Forbidden if table_name equals feed_info.
Forbidden if field_value is defined.
Required if field_value is empty.

</td></tr>
<tr><td>record_sub_id</td><td>ID</td><td>Conditionally Required</td><td>
Helps to translate the record that contains the field when the table referenced in record_id doesn’t have a unique ID. The value in record_sub_id is the secondary ID of a dataset table, as defined in the following table:
<table>
<tr><th>table_name</th><th>record_sub_id</th></tr>
<tr><td>agency</td><td>NONE</td></tr>
<tr><td>stops</td><td>NONE</td></tr>
<tr><td>routes</td><td>NONE</td></tr>
<tr><td>trips</td><td>NONE</td></tr>
<tr><td>stop_times</td><td>stop_sequence</td></tr>
<tr><td>pathways</td><td>NONE</td></tr>
<tr><td>levels</td><td>NONE</td></tr>
<tr><td>attributions</td><td>NONE</td></tr>
</table>
The following conditions determine how this field can be used:

Forbidden if table_name equals feed_info.
Forbidden if field_value is defined.
Required if table_name equals stop_times and record_id is defined.

</td></tr>
<tr><td>field_value</td><td>Text, URL, Email, or Phone number</td><td>Conditionally Required</td><td>
Instead of using record_id and record_sub_id to define which record needs to be translated, field_value can be used to define the value for translation. When used, the translation is applied when the field identified by table_name and field_name contains the exact same value defined in field_value.
The field must exactly match the value defined in field_value. If only a subset of the value matches field_value, the translation isn't applied.
If two translation rules match the same record, one with field_value and the other one with record_id, then the rule with record_id is the one that needs to be used.
The following conditions determine how this field can be used:

Forbidden if table_name equals feed_info.
Forbidden if record_id is defined.

</td></tr>
</table>

Field Name	Type	Required	Description
			• Required if `record_id` is empty.

attributions.txt

File: **Optional**

Field Name	Type	Required	Description
`attribution_id`	ID	Optional	Identifies an attribution for the dataset, or a subset of it. This field is useful for translations.
`agency_id`	ID referencing	Optional	The agency to which the attribution applies. If one `agency_id`, `route_id`, or `trip_id` attribution is defined, the other fields must be empty. If none are specified, the attribution applies to the whole dataset.
`route_id`	ID referencing	Optional	This field functions in the same way as `agency_id`, except the attribution applies to a route. Multiple attributions can apply to the same route.
`trip_id`	ID referencing	Optional	This field functions in the same way as `agency_id`, except the attribution applies to a trip. Multiple attributions can apply to the same trip.
`organization_name`	Text	Required	The name of the organization that the dataset is attributed to.
`is_producer`	Enum	Optional	The role of the organization is producer. Allowed values include the following: • `0` or empty: Organization doesn't have this role. • `1`: Organization does have this role. At least one of the fields, either `is_producer`, `is_operator`, or `is_authority`, must be set at `1`.
`is_operator`	Enum	Optional	Functions in the same way as `is_producer`, except the role of the organization is operator.
`is_authority`	Enum	Optional	Functions in the same way as `is_producer`, except the role of the organization is authority.
`attribution_url`	URL	Optional	The URL of the organization.
`attribution_email`	Email	Optional	The email of the organization.
`attribution_phone`	Phone number	Optional	The phone number of the organization.

Transit Partners Help Center

Public Feeds

Participate

Participate in Google Transit

Connect

GitHub

Google Lat Long Blog

Google Developers

Android Chrome Firebase Google Cloud Platform All products

Terms | Privacy

Sign up for the Google Developers newsletter

Subscribe

Language