NSW Trains 4Trak GTFS & GTFS-R Technical Documentation

Contents

1. Overview

The GTFS(general transit feed specification) feed provides static timetabling information about a transit network. Including stops, routes, trip schedules, trip geometry and information about the transit agency's running the trips. This data is suitable for planning trips as well as presenting trip information to a customer via a graphical user interface. There is also a real time extension to this which shares some information but is a separate feed. This real time feed specifies which trips are currently running on the network, the position of each of the vehicles completing the trips, the difference between the scheduled and actual departure times as well as the difference between the actual and scheduled arrival times.

The GTFS static feed is composed of ten CSV files which are all contained within a single zip file. Each of the files contain scheduled information about a different aspect of the transit network. For example stops_times.txt contains information about the scheduled time and order of stops for different trips.

The GTFS real time feed is composed of two separate protocol buffer files. One contains vehicle position information, the other contains a trip update of the historical and projected arrival and departure times. Protocol buffers are a mechanism developed by google for serialising data. Both protocol buffers contain trip descriptors which specify what trip the information is relevant to.

This document specifies the details of the fields and meaning of each of the files which are included in the GTFS bundle feed as well as the GTFS real-time feed. As well as information about how to configure the feed.

The bundle and real time components of the feed both conform to the GTFS. There are some fields which are not requirements of the specification but are permissible extensions.

This documentation details of the GTFS static and GTFS real time feeds as of 4Trak release 4.11.0.

2. Data Feed Access

The GTFS feed data can be accessed by sending a HTTP get request to the appropriate URL. The appropriate URL depends on which of GTFS files need to be accessed. These URLs and associated accounts can be provided via registration to the Transport for NSW Open Data Program.

The registration for the open data program can be found at this site.

https://opendata.transport.nsw.gov.au/site/en_us/home.html

3. General Transit Feed Specification

3.1.1 GTFS Compliance

The GTFS static feed is compliant with the specification reference published by google on February 3, 2016. The GTFS real time feed is also compliant with the GTFS reference published by google on February 26, 2015. The references for both feed components specifications can be found at the following URL's.

> GTFS Bundle reference: https://developers.google.com/transit/gtfs/reference
> GTFS Real time reference: https://developers.google.com/transit/gtfs-realtime/reference

3.1.2 Publication Schedule

The GTFS bundle and GTFS real time have two independent publication schedules. A new GTFS bundle is published at 1:00 am Australian Eastern standard time if it is the upload day. Or there have been changes to either the templates, stops, agencies or routes data which would affect the contents of the bundle. The upload day can be any day of the week and is a configurable setting.

Both the trip update and vehicle position files in the GTFS real time are published every 30 seconds simultaneously.

3.1.3 Validity Period

The data provided in the GTFS bundle will be valid for a hundred days after the bundle is initially published unless changes are made to the timetable since the bundle was uploaded.

3.1.4 Value Quoting

All fields with in the GTFS bundle are enclosed in double quotes. This is not the case for the real time feed fields within the vehicle position or trip update files.

3.1.5 Feed Size

The size of the GTFS bundle will vary depending on how many trips are scheduled over the next 100 days and on how many stops have been scheduled to be part of the trips. The bundle will typically be in the order of 20 MB but this is subject to variability.

The protocol buffers which make up the real time feed vary much more in size than the bundle. The size of the vehicle position files depend on the number of trips currently running. Its size is typically 2-20 KB.

The size of the trip update is dependent on the number of currently running trips, the size of the currently running trips and how much progress the running trips have made. The trip update file size is typically 10-60KB. Both real time files are substantially smaller in the early morning.

3.1.6 Template Download

Every time the GTFS bundle runs it will download a new set of templates. The templates which will be downloaded are relevant for a hundred and one day range. The upload day in the settings defines where this range starts and ends. The templates will be cover one day before the upload day and a hundred days after.

3.2 Agency.txt

The agency.txt file specifies values relevant to the agencies whose transit information is included in the feed.

The following fields are populated within the agency.txt file:

Field Name	Description	Example	Referenced in
agency_id	The agency_id is an ID which uniquely identifies a transport agency in the feed. It links different routes to their agency.	711	trips.txt, routes.txt
agency_name	The agency name is the full name of the transit agency. This	NSW TrainLink Train for Regional Trains	

Field Name	Description	Example	Referenced in
	is an identifying value which can be displayed to the customer.		
agency_url	This value is the URL of the transit agency.	http://transportnsw.info	
agency_timezone	This value is the time zone the transit agency is located in.	Australia/Sydney	
agency_lang	This field contains the ISO 639-1 code for the language used by the transit agency.	EN	
agency_phone	This field contains a single telephone number for the transit agency.	131500	

3.3 Calendar.txt

The calendar.txt file specifies two dates between which the service is valid for as well as the days of the week that the service will run on.

The following fields are populated within the calendar.txt file:

Field Name	Description	Example	Referenced in
service_id	The service_ID is a value which uniquely identifies a service with in the feed. Service ID for **Coaches** contains three fields:. The first field is the run number of the service. The second value is the start date of the trips validity. The date is in ddMMyy format. The third field represents the days of the week the service will run on. It is a seven digit binary number converted into a decimal number which indicates the days of the week the service runs. Each of the digits in the binary number represents a day of the week. With the left most value indicating a Monday and the right most value indicating a Sunday. For example 4 is 0000100 as a 7	**Coach** 575.031217.64 **Train**: N158.959.071220.124	Trips.txt

Field Name	Description	Example	Referenced in
	digit binary number. This value indicates the service will run on a Friday only. Service ID for **Trains** contains four fields: The first field is the run number of the service. The second value is the tps timetable_id The third value is the start date of the trips validity. The date is in ddMMyy format. The fourth field represents the days of the week the service will run on. It is a seven digit binary number converted into a decimal number which indicates the days of the week the service runs. Each of the digits in the binary number represents a day of the week. With the left most value indicating a Monday and the right most value indicating a Sunday. For example 4 is 0000100 as a 7 digit binary number. This value indicates the service will run on a Friday only.		
monday	This field is 1 if the service runs on Monday. The field is 0 if it does not.	1	
tuesday	This field is 1 if the service runs on Tuesday. The field is 0 if it does not.	0	
wednesday	This field is 1 if the service runs on Wednesday. The field is 0 if it does not.	1	
thursday	This field is 1 if the service runs on Thursday. The field is 0 if it does not.	0	
friday	This field is 1 if the service runs on Friday. The field is 0 if it does not.	1	

Field Name	Description	Example	Referenced in
saturday	This field is 1 if the service runs on Saturday. The field is 0 if it does not.	0	
sunday	This field is 1 if the service runs on Sunday. The field is 0 if it does not.	1	
start_date	The start date specifies the day that the validity of this service begins. The date is in YYYYMMDD format.	20160606	
end_date	The end date specifies the day the validity of this service ends. The date is in YYYYMMDD format.	20160612	

3.4 Calendar_dates.txt

The calendar_dates.txt file can be used to specify any exceptions to the normal running of a service. This file is not populated in GTFS bundle release 3.21.1 but this may change in future releases.

3.5 Routes.txt

The routes.txt file contains all the transit routes relevant to the trips in the feed.

The following fields are populated within the routes.txt file:

Field Name	Description	Example	Referenced in
route_id	The route_id is a unique identifier for a particular route. The ID for Diesel trains and coaches is composed of three separate fields. The first field indicates it is a unique ID created by 4Tel. The second field is a T or C to indicate if it is a coach route or train route. The final field is the run number or region abbreviation which corresponds to this routes region. Electric train route ID is a single field description.	4T.T.NT31	trips.txt, real time trip descriptors
agency_id	The agency is a value which uniquely identifies a transport agency. In the context of routes.txt it indicates that the route is associated with the specified agency.	711	agency.txt
route_short_name	This field contains a short abstract identifier for a route. It is often the run number of the trips on the route.	31	

Field Name	Description	Example	Referenced in
route_long_name	This field contains the full name of a particular route. This will often contain the days that the routes trips will run on as well as the starting location and terminating location if it is a coach. And the name of the train line if it is a train. This is the field most appropriate for customer facing displays.	North Coast NSW Line (Service no. 031)	
route_desc	This field specifies which transit network a particular route is on.	Regional Trains and Coaches Network	
route_type	There are three types of routes specified in this GTFS feed. 100 → Railway Service 106 → Regional Rail Service 204 → Regional Coach Service	106	
route_color	The route_color field defines a colour that corresponds to a route. It is the colour that should be used to render the routes background visually in apps that consume the GTFS data. The colour is provided as a six-character hexadecimal number.	F6891F	
route_text_color	The route_text_color field is used to specify a colour for text drawn against a background of route_color. This is provided as a six-character hexadecimal number,	FFFFFF	
booking_required	The booking_required field is used to determine if you are required to book the trip to board. 1 for true 0 for false	1	

3.6 Trips.txt

The trips.txt file is a list of all the trips which will run over validity period. It links data from different files together into a particular trip.

The trips.txt file is composed of the fields set out in the table below:

Field Name	Description	Example	Referenced in
route_id	This field associates a trip with its route. The route specifies what service the customer will see this trip as.	4T.C.135	route.txt

8

Field Name	Description	Example	Referenced in
service_id	The service_ID is a value which uniquely identifies a service with in the feed. Service ID for **Coaches** contains three fields:. The first field is the run number of the service. The second value is the start date of the trips validity. The date is in ddMMyy format. The third field represents the days of the week the service will run on. It is a seven digit binary number converted into a decimal number which indicates the days of the week the service runs. Each of the digits in the binary number represents a day of the week. With the left most value indicating a Monday and the right most value indicating a Sunday. For example 4 is 0000100 as a 7 digit binary number. This value indicates the service will run on a Friday only. Service ID for **Trains** contains four fields: The first field is the run number of the service. The second value is the tps timetable_id The third value is the start date of the trips validity. The date is in ddMMyy format. The fourth field represents the days of the week the service will run on. It is a seven digit binary number converted into a decimal number which indicates the days of the week the service runs. Each of the digits in the binary number represents a day of the week. With the left most value indicating a Monday and the right most value indicating a Sunday. For example 4 is 0000100 as a 7 digit binary number. This value indicates the service will run on a Friday only.	**Coach** 575.031217.64 **Train**: N158.959.071220.124	calendar.txt

Field Name	Description	Example	Referenced in
trip_id	The trip ID identifies the trip and is unique in the trips.txt file. **Coaches**: The trip id has all the components of the service id as well the time of day the trip will start running in 24 hour time. This is the fourth field in the id. [run number].[start date of trip validity].[days of operation for timetable].[time for start of trip] **Trains**: The trip id has all the components of the service id as well as the consist type, number of cars and the tps trip header ID. [tps trip name (aka run number)].[tps timetable id]].[start date of trip validity].[days of operation for timetable].[consist type].[number of cars].[tps trip header id] Consist types: A = A Set B = B Set C = C Set D = NIF H = H Set J = Hunter Railcar K = K Set M = M Set N = Endeavour Railcar P = eXplorer T = T Set V = V Set X = XPT	**Coach:** 328.021217.127.0540 **Train:** N158.959.071220.124.V.8.64012548	stop_times.txt, real time trip descriptors
trip_headsign	The trip headsign is a station name used to indicate the final destination of the trip. This may also include via information as it applies to the trip	Taree	
trip_short_name	This field contains the run number of the trip.	135	
direction_id	This field indicates whether a trip is inbound or outbound relative to Central station. A value of 0 indicates an outbound trip and a value of 1 indicates an inbound trip.	0	
block_id	Currently not in use by NSW Trains		

Field Name	Description	Example	Referenced in
shape_id	This value is a unique identifier for a shape from the shapes.txt file and means this trip is expected to move through the geometry defined by that shape.	4T.C.135.1.H	
wheelchair_accessible	This field is always populated with "1" which means the vehicle can accommodate at least one wheel chair.		
bikes_allowed	This field is not populated in the current release.		
trip_note	A trip note provides additional information about a trip which does not fit into the regular fields. This can be populated with an ID which will correspond to a notes.txt entry. The note.txt entry provides additional information about the trip. Not all trips will have an accompanying note.	70005	
route_direction	This field uses the first and last stop names in the trip separated by "to" which indicates the directionality of the trip.	Broadmeadow to Taree	
vehicle_category	This field describes the vehicle category and matches the associated vehicle_category_id in the vehicle_categories.txt file	XPL3R	vehicle_categories.txt and vehicle_couplings.txt

3.7 Stops.txt

The purpose of the stops.txt file is to contain details about all the stops in the network. These details include the geographical location and the stop ID which is used to reference stops in stop_times.txt. A stop in this file can either be a sub-location or a parent location. A parent location is a location that contains other locations for example a station. A sub location is a location inside another location like a platform.

The stops.txt file is composed of the fields set out in the table below:

Field Name	Description	Example	Referenced in
stop_id	This field contains an ID that uniquely identifies a stop.	233621	stop_times.txt, trip update and vehicle position real time files.
Stop_code	This field contains the same information as the stop_id	233621	

stop_name	This field contains the full name of a particular stop or station. This value is an identifier that can be easily recognised by customers.	Aberdeen Station Platform 1	
stop_lat	The stop_lat field contains the latitude of a stop in decimal degree.	-32.166969	
stop_lon	The stop_lon field contains the longitude of a stop in decimal degrees.	150.892016	
location_type	The location type indicates whether a stop is a parent location or sub location. If the value is 0 it indicates a single stop. A value of 1 indicates it is a parent station which contains another stop or stops.	0	
parent_station	If the stop is a parent location or a coach stop then this field is blank. Otherwise it contains the stop_id of the location which contains the stop.	233610	
wheelchair_boarding	This field is 0 by default unless the stop is a coach stop and is specifically set to wheelchair accessible in 4Trip. A value of "1" means at least one wheel chair can board the trip. A value of "0" means there is not sufficient information on wheel chair boarding for this stop.	0	
platform_code	The platform code indicates which platform an entries stop corresponds to. If the stop is a parent station or coach stop this value is blank.	1	
stop_timezone	The stop_timezone field contains the time zone in which this stop is located. The names of the time zones can be found on the Wikipedia List of Timezones.		

3.8 Stop_times.txt

The stops_times.txt file contains a list of all the stops completed by all the trips and the times that the stops happen.

The stop_times.txt file is composed of the fields set out in the table below:

Field Name	Description	Example	Referenced in
trip_id	The trip ID identifies the trip and is unique in the trips.txt file. **Coaches**: The trip id has all the components of the service id as well the time of day the trip will start running in 24 hour time. This is the fourth field in the id. [run number].[start date of trip validity].[days of operation for timetable].[time for start of trip] **Trains**: The trip id has all the components of the service id as well as the consist type, number of cars and the tps trip header ID. [tps trip name (aka run number)].[tps timetable id]].[start date of trip validity].[days of operation for timetable].[consist type].[number of cars].[tps trip header id] Consist types: A = A Set B = B Set C = C Set D = NIF H = H Set J = Hunter Railcar K = K Set M = M Set N = Endeavour Railcar P = eXplorer T = T Set V = V Set X = XPT	**Coach:** 328.021217.127.0540 **Train:** N158.959.071220.124.V.8.64012548	Trips.txt
arrival_time	The arrival_time field specifies the time a trip arrives at a particular stop. If a trip runs over more than one day a value greater than the maximum usually allowed in 24 hour time will be used. For example if a trip runs from 11:00 pm to 01:30 am the arrival time for the final stop will be 25:30:00.	18:00:00	

Field Name	Description	Example	Referenced in
	Times in stop_times.txt are six digit values in HH:MM:SS format. It is possible to set the times for the stops to be in the local time zone of the stop. If the FourTripTimeZone is set in the system config. Then times will be adjusted to be in the time zone of the stop.		
departure_time	The departure_time field specifies the time a trip departs from a particular stop. If a trip runs over more than one day a value greater than 24 hour time can be used. For example if a trip departs from its second last stop at 01:30 pm the departure time for the second last stop will be 25:30:00. Times are six digits in HH:MM:SS format. It is possible to set the times for the stops to be in the local time zone of the stop. If the FourTripTimeZone is set in the system config. Then times will be adjusted to be in the time zone of the stop. It is also possible to set the departure time to be equal to the arrival time. By altering the system config.	25:30:00	
stop_id	This field contains the unique ID for this entries stop indicating that the trip stops at this location at the specified time.	229268	stops.txt
stop_sequence	The stop_sequence field identifies the order a stop occurs in the entries corresponding trip.	1	
stop_headsign	The stop_headsign is used when required to override the trip_headsign **Note:** TfNSW standard specifies this is to be no longer than 15 characters, however NSW Trains specifies this is consistent with RTTA output which may be longer than 15	Kiama via Hurstville	

Field Name	Description	Example	Referenced in
	characters as it includes "via" information		

pickup_type	This field specifies whether a trip will include pick up at a particular stop, if it will stop at all or if special instruction will need to be given for pick up at this stop.
	  0-Regular pick up
	  1-No pick up
	  3-Must coordinate with driver for pick up.
		0	

drop_off_type	This field specifies whether a trip will drop passengers at a particular stop, if there will be any drop off or if special instruction will need to be given for passengers to be dropped off at this stop.
	  0-regular pick up
	  1-no pick up
	  3-Must coordinate with driver for drop off.
		0	

timepoint	This field indicates whether a particular stop time should be considered approximate or exact. Currently this will always be populated with a value of 1. 0-indicates the time is approximate 1-indicates the time is exact	0	
stop_note	This field contains an ID that associates the stop with a note in notes.txt. Stop notes contain additional information about a stop which is not included in the stop entries normal fields.	70002	Notes.txt
shape_dist_travelled	This field specifies how far along a trip a particular stop is in meters. This value is taken as the distance travelled value in the shape entry which is closest to the stop. It is measured in meters.	24252.3918665822	Shapes.txt

3.9 Shapes.txt

The shapes.txt file contains a list of all the shapes relevant to the trips. A shape is a sequence of latitudes and longitudes combined with a sequence value which indicates the entries order in the shape. A shape defines the geometry of its corresponding trip.

The shapes.txt file is composed of the fields set out in the table below:

Field Name	Description	Example	Referenced in
shape_id	This field contains an ID which uniquely identifies a shape.	4T.T.SP41.2.H	trips.txt
shape_pt_lat	This field specifies the latitude of a single coordinate in a shape in decimal degrees.	-34.7582496747171	
shape_pt_lon	This field specifies the longitude of a single coordinate in a shape in decimal degrees.	149.719581986594	
shape_pt_sequence	This field specifies the order a particular coordinate is in within the shape.	1	
shape_dist_traveled	This field specifies the distance between a given point and the first point in the shape if you travel along the shape.	0	stop_times.txt

3.10 vehicle_categories.txt

The vehicle categories file specifies information about the vehicles used in the transit network.

Field Name	Description	Example	Referenced in
vehicle_category_id	Id for the vehicle category	A8	
vehicle_category_name	Full name of the vehicle category	8 Car Waratah	

3.11 vehicle_couplings.txt

The vehicle couplings file specifies information about the vehicle car information in a consist for the transit network.

Field Name	Description	Example	Referenced in
parent_id	Matches the vehicle_category_id	A8	Vehicle Categories
child_id	Description of car type	ACar	
child_sequence	Sequence in consist	1	
child_label	Label of sequence in consist	1	

3.12 vehicle_boardings.txt

The vehicle boardings file specifies information about the feed rather than the transit network.

Field Name	Description	Example	Referenced in
vehicle_category_id	Matches the vehicle_category_id	A8	Vehicle Categories
child_sequence	Sequence in consist that can board at the specified location	1	
grandchild_sequence	Not currently used		
boarding_area_id	Location that the boarding can occur	233621	Stops.txt

3.13 Notes.txt

The notes.txt file is used to add written detail to specific stops and trips.

Field Name	Description	Example	Referenced in
note_id	This is a unique id which links a trip or stop to a particular note.	70001	trips.txt, stop_times.txt, stops.txt
note_text	The note_text contains written detail about the relevant trip or stop.	Stops to pick up and set down booked customers only when required.	

3.14 Feed_info

The feed info file specifies information about the feed rather than the transit network.

Field Name	Description	Example	Referenced in
feed_publisher_name	This field specifies the organisation which is responsible for the feed.	NSW TrainLink	
feed_publisher_url	This field contains the full url of the publisher.	http://www.nswtrainlink.info/	
feed_lang	This field contains the ISO 639-1 code for the language used by the transit agency.	en	
feed_version	This field contains the time and date the bundle was	08122017-113649	

Field Name	Description	Example	Referenced in
	produced in ddMMyyyy-HHmmss format.		

4. General Transit Feed Specification – Real time

4.1 Real Time

All the trips in the GTFS bundle will eventually be in the real time feed. They will enter the real time feed when they start running or are matched and will be removed from the Real Time feed when the trip ends.

The GTFS real time feed is composed of two protocol buffer files. A trip update file and a vehicle position file. The trip update and vehicle position files have common fields. One set of common fields is the header and the other is the trip descriptor.

4.1.1 Header

```
header {
   gtfs_realtime_version: "1.0"
   incrementality: FULL_DATASET
   timestamp: 1515365442
}
```

The following table describes the fields in the header.

Field Name	Description	Example	Referenced in
gtfs_realtime_version	This field contains the version or the GTFS real time.	1.0	
timestamp	This timestamp indicates when the file was created.	1465534901	
Incrementality	This field will always be populated with the full data set value. This indicates that the real time files contain a full snap shot of all the running trips not a subset of the running trips.	FULL_DATASET	

4.1.2 Trip Descriptor

A trip descriptor accompanies every real time entity. Its purpose is to indicate which trip the entity is associated with.

```
trip {
   trip_id: "312.010118.125.0630"
```

```
    start_time: "06:30:00"
    start_date: " 20180101"
    schedule_relationship: SCHEDULED
    route_id: "4T.C.312"
}
```

The following table describes the fields in the trip descriptors.

Field Name	Description	Example	Referenced in
trip_id	This value indicates which trip in the trips.txt file a real time entity is associated with. Details of the ID can be found in the trips.txt table.	312.010118.125.06 30	trips.txt
start_time	This value indicates the start time of the trip	06:30:00	
start_date	This value indicates the start date of the trip	20180101	
schedule_relationship	This value indicates whether the trip was scheduled. All trips in the feed will be scheduled.	SCHEDULED	
route_id	This value indicates which route the vehicle is traveling on. This value corresponds to a route in routes.txt.	4T.C.725	routes.txt, trips.txt

4.2 Trip Update Entity

Each trip update file contains a single header as well as a feed entity for each of the currently running trips. Each feed entity contain a single trip descriptor and a stop time update for each of the stops that the vehicle completing the trip has arrived at or has a non-zero projected delay for. The delay values are in seconds. A positive value indicates the vehicle is early and a negative value indicates that the vehicle is late. The feed will never return a positive value for a depart delay. If the departure delay is calculated to be greater than 0 it will be set to 0 in the feeds output.

If the system config is set to make arrive the same as depart the calculation for the departure delay changes.

Instead of being:

stop.getPlannedDepart() - stop.getActualDepart()

It will be:

departureDelta = stop.getPlannedArrive() - stop.getActualDepart();

If consecutive stops have the same delay values the entries which would have repeating data maybe removed from the feed. This depends on the settings.

An example of a trip update entity.

```
id: "1"
trip_update {
    trip {
        trip_id: "312.010118.125.0630"
        start_time: "06:30:00"
        start_date: " 20180101"
        schedule_relationship: SCHEDULED
        route_id: "4T.C.312"
    }
    stop_time_update {
        stop_sequence: 7
        arrival {
            delay: -157
        }
        departure {
            delay: 295
        }
        stop_id: "23472"
    }
}
```

The following table explains the fields in the trip update entities.

Field Name	Description	Example	Referenced in
id	This is a single unique number in the file used to differentiate the different entities.	1	
stop_sequence	This value specifies the order a stop is in the trip.	4	
arrival { delay:	This value specifies the difference between the expected arrival time and the actual arrival time in seconds.	284	
time: }	Time when the vehicle has arrived at the stop Only populated when the vehicle has arrived at the stop, or it is an ADDED, otherwise not present	1562283125	
departure { delay:	This value specifies the difference between the expected departure time and the actual departure time in seconds. With the the possible exceptions described above.	332	
time: }	Time when the vehicle has left the stop Only populated when the vehicle has departed at the stop, or it is an ADDED, otherwise not present	1568932349	
stop_id:	This value is the id of the stop the delays are relevant to.	24002	stops.txt

Field Name	Description	Example	Referenced in
schedule_relationship:	This value indicates whether the trip was scheduled. All trips in the feed will be scheduled	SCHEDULED	
timestamp	The time the entity was updated	1568932349	

4.3 Vehicle Position Entity

Each Vehicle position file contains a single header and a feed entity for each of the currently running or matched trips.

This is an example of a vehicle position feed entity.

```
vehicle {
   trip {
      trip_id: "512.010118.53.0900"
      schedule_relationship: SCHEDULED
      route_id: "4T.C.512"
   }
   position {
      latitude: -30.634272
      longitude: 146.38165
      bearing: 144.0
      speed: 0.0
   }
   current_stop_sequence: 2
   current_status: IN_TRANSIT_TO
   timestamp: 1515365295
   congestion_level: UNKNOWN_CONGESTION_LEVEL
   stop_id: "28401"
   vehicle {
      id: "512"
      label: "09:00am (512) Bourke - Dubbo"
   }
}
```

The following table contains an explanation of each of the fields.

Field Name	Description	Example	Referenced in
id	This field contains a single unique number used to differentiate the entities in the file.	1	
latitude	This field specifies the latitude of the vehicle in decimal degrees.	-29.76868	
longitude	This field specifies the longitude of	151.49377	

Field Name	Description	Example	Referenced in
	the vehicle in decimal degrees.		
bearing	This field specifies the bearing of the vehicle measured in degrees from the horizontal.	59.0	
speed	Speed of position as reported (this is not an average speed). This may not be provided via position feed and if so is set to 0.0	0.0	
current_stop_sequence	The current stop sequence	2	
current_status	Status of the vehicle as it travels through the trip. Values can be: INCOMING_AT STOPPED_AT IN_TRANSIT_TO	IN_TRANSIT_TO	
timestamp	This timestamp is the time the latitude, longitude and bearing are relevant to.	1465532033	
congestion_level	This value is always populated with a default value which indicates an unknown level of congestion.	UNKNOWN_CONGESTION_LEVEL	
stop_id	The stop_id value indicates the last location the vehicle stopped at. The ID corresponds to a locations in stops.txt	23604	stops.txt

vehicle {
    id:
}
	The Vehicle ID is the run number of the trip the vehicle is currently completing.	142	

vehicle {
    label:
}
	The vehicle label is a value which customers can	11:05am (142) Moree Town - Grafton City	

Field Name	Description	Example	Referenced in
	use to identify the trip.		

5. Settings

There are multiple settings which are configurable for both the GTFS bundle and the GTFS real time feeds.

The GTFS feed is accompanied by a system configuration XML file which can be used to specify values for different GTFS feed settings. This file is named system_config.xml and is in the D:\Apps directory.

The feed also uses configuration variables from 4Trak. These config variables can be added and edited using the 4Trak admin tool.

The system config file and 4Trak config variables are both read into the static and real time feeds when they begin execution.

Because the real time feed runs continually and settings are only downloaded when execution begins if there are changes to settings that need to propagate into the real time feed it must be restarted.

5.1 GTFS Static Settings

There two following tables explain the different ways in which the GTFS static feed can be configured.

5.1.1 Config Variables

Variable Name	Default	Description
GTFS_CREATE_DEBUG_FILES	true	Sets whether or not the feed will save copies of the bundle to the output folder. To allow for normal functioning this should be set to true. If it is set to false there will be no bundles stored which is critical to normal function. The bundle feed usually does a comparison with the last bundle to determine if there is a need for an upload. True and false are the only valid values for this setting.
GTFS_DELETE_DEBUG_DAYS	8	This sets how many days the debug files will be kept for before they will be deleted. This value should be set to a positive integer.
GTFS_WS_RETRY_COUNT	3	This sets how many times the feed will retry calls to the 4Trip template service before hard failing. This setting is relevant to all web service calls including the template, location, agencies and routes. This value should be set to a positive integer.
GTFS_WS_RETRY_WAIT	30	This sets how long in seconds the feed will between web service call retries. This applies to all web service calls. This value should be set to a positive integer.
GTFS_BUNDLE_UPLOAD_TIMEOUT	120	This sets how long the feed will wait on a bundle upload without data transfer before forcing a cancellation. This value should be set to a positive integer.

Variable Name	Default	Description
GTFS_UPLOAD_DAY	1	This sets which day the bundle will upload. If there has been no changes to the bundle it will not upload unless it is the set upload day. Number 1 means the upload day is set to Sunday and 7 is Saturday. The upload day is also the day the to and from values for the template web service call are based on. For the purposes of calculating these to and from values the relevant day is always assumed to be either the current day or before the current day. For example if the current day is a Tuesday and the upload day is set to Thursday the to and from values will be calculated based around last Thursday as the Thursday of the current week is past the current day. The from value will be the start of the day before last Thursday and the to value will be 100 days after. This value should be set to a positive integer.

5.1.2 System Config

Element Name	Attribute	Default	Description
gtfsUploadServer	operator	NSW TrainLink	The operator value is used to determine which stop codes will be used to reference different locations. As different stop codes are used by different operators.
gtfsUploadServer	doUpload	true	This value can be set to true or false and determines whether the feed will try to upload or not. There are also other conditions that will determine whether or not there will be an upload. This attribute should be set to true or false.
makeArriveAndDepartTheSame	same	true	The depart values in the stop_times.txt can be set to be the same as the arrive times. If it is set to true the depart values will be the set to the arrival times and if it is set to false the depart values will be set to be what they exist as in the templates. The times may also be adjusted by their time zone. This attribute should be set to true or false.
fourTripTimeZone	timeZone	null	The time zone value is used to calculate the arrive and departure values of the stop_times.txt entries. If a value is not set or it is set to null no modifications will be made to the arrive and depart times. But if a time zone is set it will be used to set the arrive and depart times to the local time of the stops. Please refer to Wikipedia List of Timezones for a list of valid values Wikipedia List of Timezones..
gtfsServrCredentials	url		Any number of urls can be specified by using any number of gtfsServrCredentials elements. The url is the address that will be used to post the GTFS bundle.

Element Name	Attribute	Default	Description
gtfsServrCredentials	username		Any number of username values can be specified by using any number of gtfsServrCredentials elements. The username is part of the credentials that will be used to post the GTFS bundle.
gtfsServrCredentials	password		Any number of password values can be specified by using any number of gtfsServrCredentials elements. The password is part of the credentials that will be used to post the GTFS bundle.
gtfsValidRunNumbers→gtfsValidRunNumber	value	KN, CN, NP, NT, SN,SP, ST, V, WN, WP, WT, 1, 3,5, 7,8	Any number of valid run numbers can be specified. For a trip to be included in the GTFS bundle its run number must have a prefix which matches to one of the specified valid run number prefixes.

5.2 GTFS Real Time Settings

There two following tables explain the different settings in which are used to configure the GTFS real time feed.

5.2.1 Config Variables

Variable Name	Default	Description
GTFS_CREATE_DEBUG_FILES	true	Sets whether or not the feed will save copies of the protocol buffers to the output folder.
GTFS_DELETE_DEBUG_DAYS	8	This is how many days the debug files will be kept for before the clean-up operation will delete them.
GTFS_WS_RETRY_COUNT	3	This sets how many times the feed will retry the 4Trip template service before hard failing.
GTFS_WS_RETRY_WAIT	30	How long in seconds the feed will wait before retrying a web service download.
GTFS_REALTIME_UPLOAD_TIMEOUT	120	How long the feed will wait on a protocol buffer upload without data transfer before forcing a cancellation.
REMOVE_REDUNDANT_TRIP_ENTRIES	true	This settings determines whether trip update entries which repeat the delays of the previous entry will be removed or not.

5.2.2 System Config

Element Name	Attribute	Default	Description
gtfsUploadServer	operator	NSW TrainLink	The operator value is used to determine which stop codes will be used to reference locations. As different stop codes are used by different operators.

Element Name	Attribute	Default	Description
gtfsUploadServer	doUpload	true	This value can be set to true or false and determines whether the feed will try to upload or not.
makeArriveAndDepartTheSame	same	true	When the delay values are calculated for the trip updates if this value is set to true departureDelay = stop.getPlannedArrive() - stop.getActualDepart(). Otherwise departureDelay = stop.getPlannedDepart() - stop.getActualDepart(). True and false are the only valid settings.
tripUpdateUploadServerCredentials	url		Any number of urls can be specified by using any number of tripUpdateUploadServerCredentials elements. Each set of credentials is another server the file will be uploaded to. The url is the address that will be used to post the GTFS real time trip update protocol buffers.
tripUpdateUploadServerCredentials	username		Any number of username values can be specified by using any number of tripUpdateUploadServerCredentials elements. The username is part of the credentials that will be used to post the GTFS real time trip update protocol buffers.
tripUpdateUploadServerCredentials	password		Any number of password values can be specified by using any number of tripUpdateUploadServerCredentials elements. The password is part of the credentials that will be used to post the GTFS real time trip update protocol buffers.
vehiclePositionServerCredentials	url		Any number of vehicle position upload credentials can be set in the system config. Each set of credentials is another server the file will be uploaded to. The url value is the address that the GTFS vehicle position protocol buffers will be posted to.
vehiclePositionServerCredentials	username		Any number of vehicle position upload credentials can be set in the system config. Each set of credentials are another server the file will be uploaded to. The username value is the username of the credentials that will be used to upload.
vehiclePositionServerCredentials	password		Any number of vehicle position upload credentials can be set in the system config. Each set of credentials are another server the file will be uploaded to. The password value is the password of the credentials that will be used to upload.
gtfsValidRunNumbers→gtfsValidRunNumber	value	KN, CN, NP, NT, SN,SP, ST, V,	Any number of valid run numbers can be specified. For a trip to be included in the

Element Name	Attribute	Default	Description
		WN, WP, WT, 1, 3,5, 7,8	GTFS real time its run number must have a prefix which matches to one of the specified valid run number prefixes.
gtfsServiceAreas→gtfsServiceArea	name	Blue Mountains, Hunter, Griffith,Southern Highlands, North Coast, North Coast Coaches, South Coaches, North West, North West Coaches, Canberra, Melbourne, West, West Coaches	Any number of valid service areas can be specified. For a trip to be included in the GTFS real time feed its service area must match one the service areas in the system config.

6. Document Version Tracking

Ver.	Date	Author	Description of change
5.0	07/10/2022	M Wood	Add stop_headsign to stop_times.txt and update block_id description in stops.txt
5.1	20/02/2023	L Farrawell	Update the description of block_id in stops.txt
5.2	29/10/2024	L Farrawell	Added vehicle_categories.txt, vehicle_couplings.txt and vehicle_boardings.txt files and vehicle_category field to trips.txt for the new RES program.