General Transit Feed Specification (GTFS) – Timetable and Realtime Feed for PTMS - Onboarding Sydney Metro

Consumer Guide

Contents

Introduction ... 3
1.1 Overview ... 3
1.2 Purpose of PTMS GTFS Feed ... 3
1.3 Definition of Terms Used ... 4
1.4 Lifecycle of PTMS GTFS Timetable Feed ... 4
1.4.1 Variations ... 5
1.4.2 Extensions ... 5
1.5 Publication Cycle & Scope ... 5
1.5.1 One Bundle per Operator ... 5
1.5.2 Validity Period ... 5
1.5.3 Generation Triggers ... 6
1.5.4 Modes / Geography ... 6
1.6 Standards Applied ... 6
1.6.1 Identifiers ... 6
1.6.2 Value Quoting ... 6
1.7 File Structure ... 7
1.7.1 Single Contents: ... 7
1.8 File Size ... 7
1.9 Overview ... 8
1.10 agency.txt ... 10
1.11 Routes.txt ... 12
1.12 Trips.txt ... 15
1.13 Calendar.txt ... 18
1.14 Calendar_dates.txt ... 21
1.15 Shapes.txt ... 23
1.16 stops.txt ... 26
1.17 Stop_times.txt ... 31
1.18 notes.txt ... 37

GTFS Timetable Feed Usage Notes ... 38
1.19 Stop Arrival and Departure Times ... 38
1.20 GTFS Bundle and Operators ... 38
1.21 GTFS Realtime ... 39
1.22 Trip Updates Message ... 41
1.22.1 Trip Updates Message ... 45
1.23 Vehicle Position Updates Message ... 58
1.23.1 Vehicle Position Updates Message ... 60
1.24 Occupancy Status ... 72
1.25 Carriage Descriptor ... 73
1.26 Transit_realtime.tfnsw_vehicle_descriptor ... 74
1.27 Alert Update Message ... 76
1.29 Alert ... 79
1.29.1 Time Range ... 80
1.29.2 Entity Selector ... 80
1.29.3 Trip Descriptor ... 81
1.29.4 Cause ... 85
1.29.5 Effect ... 87
1.29.6 Translated String ... 88
1.29.7 Translation ... 88
1.30 FAQs for the V2 - Toustone Feed: ... 89

Change Record and Documentation Control

Version	Date	Change Reference
0.1	09/08/2023	Draft based on CSELR Release Notes
0.2	10/08/2023	Updated the information relevant to Sydney Metro and reviewed with internal project team
0.3	21/08/2023	Final review with project team
0.4	07/12/2023	Added FAQs from Endorsed Internal Consumers

Related Documents

Ref	Document Name	Version / Date	Location
01	GTFS & GTFS R – TfNSW Implementation Specification	V1.0.3 12/12/2020	PDF Adobe Acrobat Document
02	GTFS & GTFS R – TfNSW Implementation Specification	V1.0 25/06/2019	PDF Adobe Acrobat Document

Introduction

1.1 Overview

The purpose of this document is to describe the structure and contents of the PTMS GTFS S Timetable data feed supplied by NSW Transport for consumption and use by Transport for NSW systems, business units and application developers.

The PTMS Timetable data feed is in the form of a General Transit Feed Specification, referred as 'GTFS'.

The intended audience of this document is application developers.

While GTFS is well documented, it provides several optional fields and some flexibility in how to populate some fields. This document provides some general information regarding the specific contents and structure of the GTFS Timetable Feed generated through PTMS.

1.2 Purpose of PTMS GTFS Feed

The purpose of the PTMS GTFS Timetable Feed is to publish in advance the schedules and route information of services operated under Sydney Metro.

Consumers of the PTMS GTFS Timetable Feed are expected to include:

- TfNSW Transport Info (http://www.transportnsw.info);
- Realtime Transport App providers (http://www.transportnsw.info/en/travelling-with-us/keep-updated/apps/real-time-transport.page);
- Open Data users (https://opendata.transport.nsw.gov.au)
- Opal Travel App, Facebook Messenger, Google Assistant, TripView, NextThere and AnyTrip

1.3 Definition of Terms Used

GTFS	The General Transit Feed Specification (GTFS) defines a common format for public transportation schedules and associated geographic information.
GTFS-R	GTFS-realtime is a feed specification that allows public transportation agencies to provide realtime updates about their fleet to application developers. It is an extension to GTFS.
SIRI	Standard Interface for Real-time Interface - XML protocol to allow distributed computers to exchange real-time information about public transport services and vehicles.
SIRI ET	SIRI Estimated Timetable Message – is a message sent from the AVLS (at regular intervals) containing all the necessary information that is required for providing trip updates in relation to the scheduled timetable
SIRI VM	SIRI Vehicle Movement – is a message sent from the AVLS (at regular intervals) containing all the necessary information that is required for providing the vehicle position
Traffic Studio	Traffic Studio is a system to upload TXC files that contain the scheduled timetable information, the routes and journey patterns that the operator services will follow and trips information for the period of operation
Sydney Metro City and Southwest (SMCSW)	Sydney Metro City and Southwest will extend from the end of Sydney Metro Northwest at Chatswood under Sydney Harbour, through new CBD stations and south west to Bankstown.
Operator	Sydney Metro
TXC	TransXchange Specification – is a collection of information about the timetable of scheduled operator services, the spatial path of the routes the operator services follow, and the trips scheduled for an operating day
Trip ID	A globally unique identifier for operator trips derived as part of the SIRI Broker transformation process. Provides the unique 'Trip ID' used in the GTFS Feed and provides the link to GTFS Real-time data.
PTMS	Public Transport Messaging Services is a suite of TXC and SIRI messages implemented to provide Scheduled timetable and real-time information through the SIRI Broker

1.4 Lifecycle of PTMS GTFS Timetable Feed

A brief overview of the process to generate the operator GTFS Timetable feed.

- Operators of contracted services submit Timetable information a minimum of 2 weeks prior to the 'activation date' of timetable data.
- The TfNSW system Traffic Studio is the application where the operator will upload, verify and accept the scheduled timetable data and queue the data for a global import

  As part of this process, the file is analysed for data integrity between Stop Points, Route Sections, Routes, Journey Pattern Sections, Services and Vehicle Journeys for a given Operator.
- Nightly on the date of timetable upload @ 11.00 pm a global import is performed in SIRI Broker to upload the timetable information on the database and generate a GTFS S bundle which should have at least a minimum of 100 days of operating period.
- The generated GTFS S bundle is then pushed to the GTFS Server

1.4.1 Variations

The GTFS S bundle conforms to the GTFS & GTFS R – TfNSW Implementation Specification V1.0

1.4.2 Extensions

The GTFS S bundle conforms to the GTFS & GTFS R – TfNSW Implementation Specification V1.0

1.5 Publication Cycle & Scope

1.5.1 One Bundle per Operator

In order to optimise the GTFS S Bundle file sizes, and to align generation with actual updates received from Operators, the full GTFS S timetable feed is generated once a TXC file is uploaded, verified and accepted in Traffic Studio through the scheduled process of global imports.

- Each Operators Contract will be represented in a single GTFS S .zip file.
  For example, there will be a GTFS S bundle for Sydney Metro, CSELR, one for Ferries and so-on.
- Under PTMS there is a concept of Long Term and Short-Term Timetable introduced. Operator can upload two TXC files on Traffic Studio. One file can contain the schedule for a minimum of 100 days and a maximum of 180 days which is termed as the Long-Term timetable and the other TXC file can be for a short period (usually for a single day) which overwrites the single day timetable in the Long-Term timetable TXC file. This short-term timetable can be the planned timetable for a special event for example a sporting event.

1.5.2 Validity Period

For each Operator the GTFS S bundle will contain:

- timetables commencing the day of generation.
- a minimum of 100 days of timetable
  - Schedule information provided by Operator in full.
    This may be with a minimum of 100 days up to a maximum of 180 days.
- Days of special operations, for example School days schedule, special events like sporting events at SCG etc.
- Days of non-operation, for example weekends

1.5.3 Generation Triggers

GTFS S are generated:

- Every time a TXC file is uploaded to Traffic Studio and has been globally imported on the night, a GTFS S bundle will be generated and pushed to the GTFS Server
- On a failover of the SIRI Broker from primary region to a secondary region
- On a server upgrade (Code/Windows)

1.5.4 Modes / Geography

The GTFS Timetable specified in this document is capable to support data for:

- All services contracted to TfNSW

Currently the services that are on-boarded on the PTMS platform are detailed in **Section 1.20 below**.

The PTMS platform can support all operator GTFS S bundles who operate in the region of NSW.

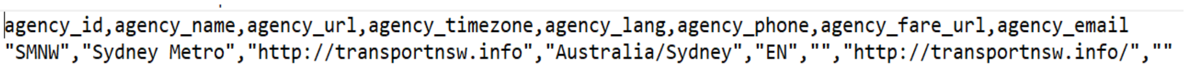
1.6 Standards Applied

1.6.1 Identifiers

Many of the identifiers used within the feed include underscore (_) to join elements. Care may need to be taken to consider this when using the data or developing applications.

1.6.2 Value Quoting

All values in the text files are double-quoted. This includes numeric values and empty values. For example (from agency.txt):



```
agency_id,agency_name,agency_url,agency_timezone,agency_lang,agency_phone,agency_fare_url,agency_email
"SMNW","Sydney Metro","http://transportnsw.info","Australia/Sydney","EN","","http://transportnsw.info/",""
```

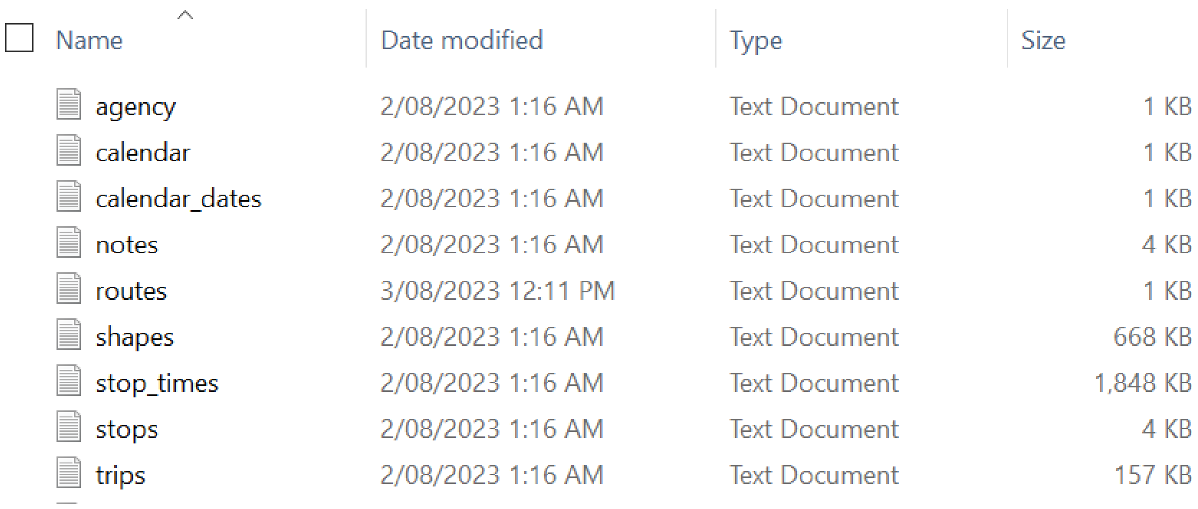
1.7 File Structure

1.7.1 Single Contents:

Each GTFS Bundle will be a ‘ZIP’ format compressed .zip file containing 9 text files.

Each file within the .zip file is a comma delimited / comma separated and double quoted format file with the “.txt” extension.

For example:



□ Name	Date modified	Type	Size
agency	2/08/2023 1:16 AM	Text Document	1 KB
calendar	2/08/2023 1:16 AM	Text Document	1 KB
calendar_dates	2/08/2023 1:16 AM	Text Document	1 KB
notes	2/08/2023 1:16 AM	Text Document	4 KB
routes	3/08/2023 12:11 PM	Text Document	1 KB
shapes	2/08/2023 1:16 AM	Text Document	668 KB
stop_times	2/08/2023 1:16 AM	Text Document	1,848 KB
stops	2/08/2023 1:16 AM	Text Document	4 KB
trips	2/08/2023 1:16 AM	Text Document	157 KB

1.8 File Size

Compressed (.zip) files range in size from 300KB to 2MB.

Files are unlikely to be larger than 10MB

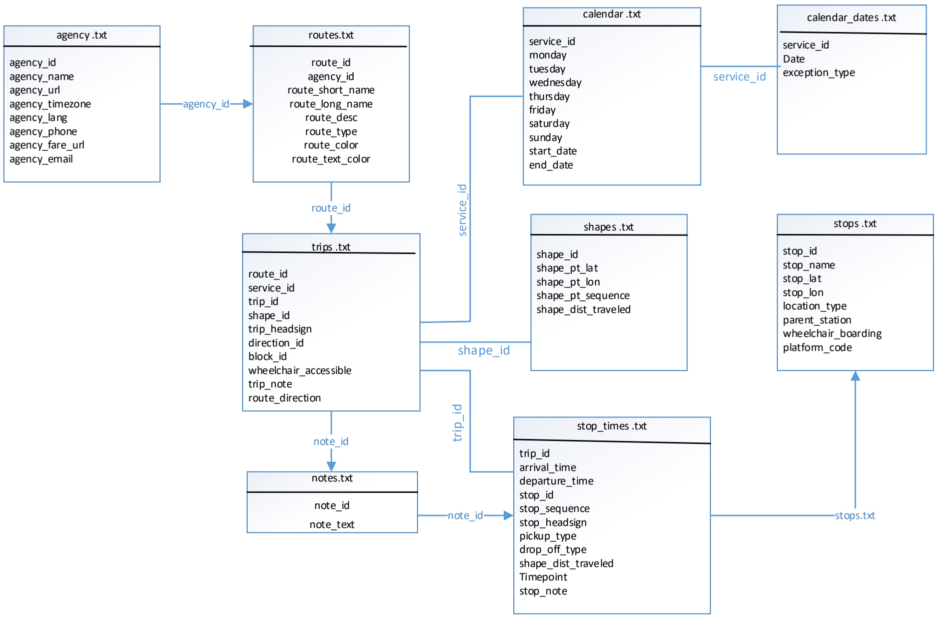
GTFS S Timetable feed for PTMS

1.9 Overview

The following table will provide the list of files applicable for TfNSW GTFS-S bundle.

File Name	Required by TfNSW	Defines
agency.txt	**Yes**	One or more transit agencies that provide the data in this feed.
stops.txt	**Yes**	Individual locations where vehicles pick up or drop off passengers.
routes.txt	**Yes**	Transit routes. A route is a group of trips that are displayed to riders as a single service.
trips.txt	**Yes**	Trips for each route. A trip is a sequence of two or more stops that occurs at specific time.
stop_times.txt	**Yes**	Times that a vehicle arrives at and departs from individual stops for each trip.
calendar.txt	**Yes**	Dates for service IDs using a weekly schedule. Specify when service starts and ends, as well as days of the week where service is available.
calendar_dates.txt	**Yes**	Exceptions for the service IDs defined in the calendar.txt file. If calendar_dates.txt includes ALL dates of service, this file may be specified instead of calendar.txt.
shapes.txt	**Yes**	Rules for drawing lines on a map to represent a transit organization's routes.
notes.txt	**Yes**	TfNSW Extension to the GTFS standard used for providing descriptive information about trips and stops.
fare_attributes.txt	**No**	Fare information for a transit organization's routes.
fare_rules.txt	**No**	Rules for applying fare information for a transit organization's routes.
frequencies.txt	**No**	Headway (time between trips) for routes with variable frequency of service.
transfers.txt	**No**	Rules for making connections at transfer points between routes.
feed_info.txt	**No**	Additional information about the feed itself, including publisher, version, and expiration information.

The following diagram provides a physical data model view of the relationship between the TfNSW GTFS S Timetable .txt files and fields.



Understanding this diagram:

- This is a representation of the relationships between the .txt files of the GTFS Timetable as implemented by TfNSW.
- Each block represents a .txt file and within the block for the .txt file are the fields which have been implemented in the GTFS S bundle generated via PTMS
- GTFS files and columns not implemented by TfNSW are not shown.

1.10 agency.txt

The Agency file identifies all providers whose timetable data has been included in the GTFS file-set generated.

Field specific requirements required by TfNSW in conjunction with the published GTFS standards for the generation of an **agency.txt** file are defined below:

Field	Required by TfNSW	Published GTFS Field Description	TfNSW specific requirements, additional business rules and sample data
agency_id	Yes	The agency_id field is an ID that uniquely identifies a transit agency. A transit feed may represent data from more than one agency. The agency_id is dataset unique. This field is mandatory and must be populated for transit feeds	- This is the operator's agency id, allocated by TfNSW. - For Example: "SMNW". Refer to Appendix for list of Assigned Agencies.
agency_name	Yes	The agency_name field contains the full name of the transit agency. Google Maps will display this name.	-The Operator's 'Customer Facing Name'. Note that this is not necessarily the same as the legal entity name. Refer to the Appendix for approved agency names. - Must be formatted in TITLE CAPS. - For Example: "Sydney Metro". Refer to Appendix for list of Assigned Agencies.
agency_url	Yes	The agency_url field contains the URL of the transit agency. The value must be a fully qualified URL that includes http:// or https://, and any special characters in the URL must be correctly escaped. Seehttp://www.w3.org/Addressing/URL/4_URI_Recommentations.html for a description of how to create fully qualified URL values.	- Must always be a fixed static value of "http://transportnsw.info" - Must be formatted in LOWERCASE
agency_timezone	Yes	The agency_timezone field contains the timezone where the transit agency is located. Timezone names never contain the space character but may contain an underscore. Please refer tohttp://en.wikipedia.org/wiki/List_of_tz_zones for a list of valid values. If multiple agencies are specified in the feed, each must have the same agency_timezone.	- Must always be a fixed static value of "Australia/Sydney" - Must be formatted in TITLE CAPS.

agency_lang	Yes	The agency_lang field contains a two-letter ISO 639-1 code for the primary language used by this transit agency. The language code is case-insensitive (both en and EN are accepted). This setting defines capitalization rules and other language-specific settings for all text contained in this transit agency's feed. Please refer to http://www.loc.gov/standards/iso639-2/php/code_list.php for a list of valid values.	- Must always be a fixed static value of "EN" - Must be formatted in TITLE CAPS.
agency_phone	Yes	The agency_phone field contains a single 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.	- Blank
agency_fare_url	No	The agency_fare_url specifies the URL of a web page that allows a rider to purchase tickets or other fare instruments for that agency online. The value must be a fully qualified URL that includes http:// orhttps://, and any special characters in the URL must be correctly escaped. Seehttp://www.w3.org/Addressing/URL/4_URI_Recommentations.html for a description of how to create fully qualified URL values.	- Feed "http://transportnsw.info/"
agency_email	No	The agency_email field contains a single valid email address actively monitored by the agency’s customer service department. This email address will be considered a direct contact point where transit riders can reach a customer service representative at the agency.	- Blank

1.11 Routes.txt

The Routes file identifies all the routes for a single contract included in the GTFS file-set generated.

The data provided must reflect a customer view of the timetable, with the grouping of trips in logical routes. It must reflect a customer journey and not split trips at non-revenue stops or city circle loop.

Field specific requirements required by TfNSW in conjunction with the published GTFS standards for the generation of a **routes.txt** file are defined below:

Field	Required by TfNSW	Published GTFS Field Description	TfNSW specific requirements, additional business rules and sample data
route_id	Yes	The route_id field contains an ID that uniquely identifies a route. The route_id is dataset unique.	- Unique identifier for a route independently generated by the data provider - The route_id is to be unique within the GTFS file set provided. - For Example: "SMNW_M"
agency_id	Yes	The agency_id field defines an agency for the specified route. This value is referenced from the agency.txt file. Use this field when you are providing data for routes from more than one agency.	- This is the operator's agency id, allocated by TfNSW. - For Example: "SMNW"
route_short_name	Yes	The route_short_name contains the 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. At least one of route_short_name or route_long_name must be specified, or potentially both if appropriate. If the route does not have a short name, please specify a route_long_name and use an empty string as the value for this field.	- The short code identifying the Route to the public. Refer to Appendix for approved route short names. - This field is CASE SENSITIVE - For Example: "M" - Should be no longer than 4-characters - Must be registered in Route Services - Refer to TfNSW Wayfinding Route Numbering & Naming Standard

route_long_name	Yes	The route_long_name contains the full name of a route. This name is generally more descriptive than the route_short_name and will often include the route's destination or stop. At least one of route_short_name or route_long_name must be specified, or potentially both if appropriate. If the route does not have a long name, please specify a route_short_name and use an empty string as the value for this field.	- The long name identifying the Route to the public. Refer to Appendix for approved route long names. - This field is CASE SENSITIVE - For Example: "Metro North West Line" - Must be registered in Route Services - Refer to TfNSW Wayfinding Route Numbering & Naming Standard
route_desc	Yes	The route_desc field contains a description of a route. Please provide useful, quality information. Do not simply duplicate the name of the route. For 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)."	- Indicates the Network that the Route belongs to as defined by TfNSW Wayfinding. Refer to Appendix for approved network names. - For Example: "Sydney Metro Network" - Must be registered in TfNSW Route Services
route_type	Yes	The route_type field describes the type of transportation used on a route. Valid values for this field 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 car. Used for street-level cable cars where the cable runs beneath the car. 6 - Gondola, Suspended cable car. Typically used for aerial cable cars where the car is suspended from the cable. 7 - Funicular. Any rail system designed for steep inclines.	- Route type is defined by TfNSW Wayfinding. - Refer to Appendix for approved route types. - Route Type is fetched from Services.csv Master Data - For Example: "401"

route_colour	Yes	In systems that have colors assigned to routes, the route_color field defines a color that corresponds to a route. The color must be provided as a six-character hexadecimal number, for example, 00FFFF. If no color is specified, the default route color is white (FFFFFF). The color difference between route_color and route_text_color should provide sufficient contrast when viewed on a black and white screen. The W3C Techniques for Accessibility Evaluation And Repair Tools document offers a useful algorithm for evaluating color contrast. There are also helpful online tools for choosing contrasting colors, including the snook.ca Color Contrast Check application.	- Must use the TfNSW approved colours see Appendix for TfNSW approved colours - For Example: "168388"
route_text_color	Yes	The route_text_color field can be used to specify a legible color to use for text drawn against a background of route_color. The color must be provided as a six-character hexadecimal number, for example, FFD700. If no color is specified, the default text color is black (000000). The color difference between route_color and route_text_color should provide sufficient contrast when viewed on a black and white screen.	- Must always be a fixed static value of "FFFFFF"
route_url	No	The route_url field contains the URL of a web page about that particular route. This should be different from the agency_url. The value must be a fully qualified URL that includes http:// or https://, and any special characters in the URL must be correctly escaped. Seehttp://www.w3.org/Addressing/URL/4_URI_Recommentations.html for a description of how to create fully qualified URL values.	- Where provided, this should refer to the TfNSW website and/or designated /URL and not that of the operator

1.12 Trips.txt

The Trips file provides information on all the trips for a contract whose timetable data has been included in the GTFS file-set generated.

Field specific requirements required by TfNSW in conjunction with the published GTFS standards for the generation of a **trips.txt** file are defined below:

Field	Required by TfNSW	Published GTFS Field Description	TfNSW specific requirements, additional business rules and sample data
route_id	Yes	The route_id field contains an ID that uniquely identifies a route. This value is referenced from the routes.txt file.	Unique identifier for a route independently generated by Data Provider. - For Example: "SMNW_M"
service_id	Yes	The service_id contains an ID that uniquely identifies a set of dates when service is available for one or more routes. This value is referenced from the calendar.txt or calendar_dates.txt file.	Unique identifier for a route independently generated by Data Provider that identifies a set of dates when a service is available for one or more routes. - For Example: "1610442"
trip_id	Yes	The trip_id field contains an ID that identifies a trip. The trip_id is dataset unique.	- Unique identifier for a trip independently generated by Data Provider. - The trip_id is to be unique within the GTFS file set provided. - trip_id is the primary reference identifier for matching static timetables with real time data feeds (GTFS-R) - For Example: "M-I-CUD-CHW-1-0435-3104:1000"
trip_headsign	Yes	The trip_headsign field contains the text that appears on a sign that identifies the trip's destination to passengers. Use this field to distinguish between different patterns of service in the same route. If the headsign changes during a trip, you can override the trip_headsign by specifying values for the stop_headsign field in stop_times.txt.	- The trip_headsign field is to contain the text that appears on a sign that identifies the trip's destination to passengers. - The destination stop for the strip is to be applied - For Example: "Chatswood"

trip_short_name	No	The trip_short_name field contains the text that appears in schedules and sign boards to identify the trip to passengers, for example, to identify train numbers for commuter rail trips. If riders do not commonly rely on trip names, please leave this field blank. A trip_short_name value, if provided, should uniquely identify a trip within a service day; it should not be used for destination names or limited/express designations.	- The trip_short_name is the service number - Blank
direction_id	Yes	The direction_id field contains a binary value that indicates the direction of travel for a trip. Use this field to distinguish between bi-directional trips with the same route_id. This field is not used in routing; it provides a way to separate trips by direction when publishing time tables. You can specify names for each direction with the trip_headsign field. - 0 - travel in one direction (e.g. outbound travel) - 1 - travel in the opposite direction (e.g. inbound travel) For example, you could use the trip_headsign and direction_id fields together to assign a name to travel in each direction for a set of trips. A trips.txt file could contain these rows for use in time tables: trip_id,...,trip_headsign,direction_id 1234,...,to Airport,0 1505,...,to Downtown,1	- Indicates the direction (outgoing, inbound) of the Trip. - 'Loop' services will be indicated as '1 – Inbound' so the possible values will be: - 1 – Inbound - 0 – Outbound - 1 – Loop (becomes Inbound) - Trip stopping pattern direction Id should be different to the stopping pattern in the opposite direction - For Example: "1"

block_id	Yes	The block_id field identifies the block to which the trip belongs. A block consists of two or more sequential trips made using the same vehicle, where a passenger can transfer from one trip to the next just by staying in the vehicle. The block_id must be referenced by two or more trips in trips.txt.	Unique identifier for a block independently generated by data provider - Blank
shape_id	Yes	The shape_id field contains an ID that defines a shape for the trip. This value is referenced from the shapes.txt file. The shapes.txt file allows you to define how a line should be drawn on the map to represent a trip.	- Unique identifier for a shape independently generated by data provider - For Example: "2157"
wheelchair_accessible	Yes	1 - indicates that the vehicle being used on this particular trip can accommodate at least one rider in a wheelchair 2 - indicates that no riders in wheelchairs can be accommodated on this trip	A value of either "1" or "2" must be provided
bikes_allowed	Yes	0 (or empty) - indicates that there is no bike information for the trip 1 - indicates that the vehicle being used on this particular trip can accommodate at least one bicycle 2 - indicates that no bicycles are allowed on this trip	A value of either "1" or "2" must be provided
trip_note	Yes	TfNSW Extension - Trip notes are provided by operators for each trip to provide further information to customers.	- Reference key to the notes.txt file (a TfNSW extension to the standard GTFS – refer to Section 3.9 Notes.txt) where trip notes can be defined. - For trips without a trip note this field can be left blank e.g., ""

route_direction	Yes	Route directions are provided by Operators for each trip and are to be made available to Customers.	- The customer facing description of the route direction as approved by TfNSW (refer to Section 4.1.4 Service Direction Name). - For Example: “Tallawong to Chatswood”

1.13 Calendar.txt

The Calendar file provides a set of dates when service is available for one or more routes.

TfNSW require that timetable data provided is at least 100 days in advance. This means:

- The **start_date** should be <today> i.e. the day the GTFS file set was generated. This represents the effective start date of the GTFS file-set bundle.
- The **end_date** should be **100 days** from <today> and represents the effective end date of the GTFS file-set bundle provided.

Field specific requirements required by TfNSW in conjunction with the published GTFS standards for the generation of a **calendar.txt** file are defined below:

Field	Required by TfNSW	Published GTFS Field Description	TfNSW specific requirements, additional business rules and sample data
service_id	Yes	The service_id contains an ID that 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. This value is dataset unique. It is referenced by the trips.txt file.	Unique identifier for a route independently generated by Data Provider that identifies a set of dates when a service is available for one or more routes. - For Example: "1610442"

monday	Yes	The monday field contains a binary value that indicates whether the service is valid for all Mondays. - A value of 1 indicates that service is available for all Mondays in the date range. (The date range is specified using the start_date and end_date fields.) - A value of 0 indicates that service is not available on Mondays in the date range. Note: You may list exceptions for particular dates, such as holidays, in the calendar_dates.txt file.	For Example: "1"
tuesday	Yes	The tuesday field contains a binary value that indicates whether the service is valid for all Tuesdays. - A value of 1 indicates that service is available for all Tuesdays in the date range. (The date range is specified using the start_date and end_date fields.) - A value of 0 indicates that service is not available on Tuesdays in the date range. Note: You may list exceptions for particular dates, such as holidays, in the calendar_dates.txt file.	For Example: "0"
wednesday	Yes	The wednesday field contains a binary value that indicates whether the service is valid for all wednesdays. - A value of 1 indicates that service is available for all Wednesdays in the date range. (The date range is specified using the start_date and end_date fields.) - A value of 0 indicates that service is not available on Wednesdays in the date range. Note: You may list exceptions for particular dates, such as holidays, in the calendar_dates.txt file.	For Example: "1"

thursday	Yes	The thursday field contains a binary value that indicates whether the service is valid for all thursdays. - A value of 1 indicates that service is available for all Thursday in the date range. (The date range is specified using the start_date and end_date fields.) - A value of 0 indicates that service is not available on Thursdays in the date range. Note: You may list exceptions for particular dates, such as holidays, in the calendar_dates.txt file.	For Example: "0"
friday	Yes	The friday field contains a binary value that indicates whether the service is valid for all fridays. - A value of 1 indicates that service is available for all Friday in the date range. (The date range is specified using the start_date and end_date fields.) - A value of 0 indicates that service is not available on Fridays in the date range. Note: You may list exceptions for particular dates, such as holidays, in the calendar_dates.txt file.	For Example: "1"
saturday	Yes	The saturday field contains a binary value that indicates whether the service is valid for all saturdays. - A value of 1 indicates that service is available for all Saturday in the date range. (The date range is specified using the start_date and end_date fields.) - A value of 0 indicates that service is not available on Saturdays in the date range. Note: You may list exceptions for particular dates, such as holidays, in the calendar_dates.txt file.	For Example: "0"

sunday	Yes	The sunday field contains a binary value that indicates whether the service is valid for all sundays. - A value of 1 indicates that service is available for all Sunday in the date range. (The date range is specified using the start_date and end_date fields.) - A value of 0 indicates that service is not available on Sunday in the date range. Note: You may list exceptions for particular dates, such as holidays, in the calendar_dates.txt file.	For Example: "0"
start_date	Yes	The start_date field contains the start date for the service. The start_date field's value should be in YYYYMMDD format.	For Example: "20230915"
end_date	Yes	The end_date field contains the end date for the service. This date is included in the service interval. The end_date field's value should be in YYYYMMDD format.	For Example: "20230915"

1.14 Calendar_dates.txt

The Calendar dates file provides the ability to define service exceptions for one or more of the provided routes. This mechanism is commonly used to override standard timetables defined in the Calendar.txt file e.g. when a particular service will not be running on a specific day for a limited period of time.

Field specific requirements required by TfNSW in conjunction with the published GTFS standards for the generation of a **calendar_dates.txt** file are defined below:

Field	Required by TfNSW	Published GTFS Field Description	TfNSW specific requirements, additional business rules and sample data
service_id	Yes	The service_id contains an ID that uniquely identifies a set of dates when a service exception is available for one or more routes. Each (service_id, date) pair can only appear once in calendar_dates.txt. If the service_id value appears in both the calendar.txt and calendar_dates.txt files, the information in calendar_dates.txt modifies the service information specified in calendar.txt. This field is referenced by the trips.txt file.	Unique identifier for a route independently generated by the data provider that identifies a set of dates when a service is available for one or more routes. For Example: "1610442"
date	Yes	The date field specifies a particular date when service availability is different than the norm. You can use the exception_type field to indicate whether service is available on the specified date. The date field's value should be in YYYYMMDD format.	For Example: "20230707"
exception_type	Yes	The exception_type indicates whether service is available on the date specified in the date field. - A value of 1 indicates that service has been added for the specified date. - A value of 2 indicates that service has been removed for the specified date. For example, suppose a route has one set of trips available on holidays and another set of trips available on all other days. You could have one service_id that corresponds to the regular service schedule and another service_id that corresponds to the holiday schedule. For a particular holiday, you would use the calendar_dates.txt file to add the holiday to the holiday service_id and to remove the holiday from the regular service_id schedule.	For Example: "1"

1.15 Shapes.txt

The Shapes file contains data to enable visual representation of a trip's path based on the route paths points.

Field specific requirements required by TfNSW in conjunction with the published GTFS standards for the generation of a **shapes.txt** file are defined below:

Field	Required by TfNSW	Published GTFS Field Description	TfNSW specific requirements, additional business rules and sample data
shape_id	Yes	The shape_id field contains an ID that uniquely identifies a shape.	- Unique identifier for a shape independently generated by Data Provider. - Unique identifier generated by the Data Provider timetable system that identifies the shape in question. - For Example: “2157”
shape_pt_lat	Yes	The shape_pt_lat field associates a shape point's latitude with a shape ID. The field value must be a valid WGS 84 latitude. Each row in shapes.txt represents a shape point in your shape definition. For example, if the shape "A_shp" has three points in its definition, the shapes.txt file might contain these rows to define the shape: A_shp,37.61956,-122.48161,0 A_shp,37.64430,-122.41070,6 A_shp,37.65863,-122.30839,11	- For Example: "-33.691543" - Formatted to at least 6 decimal places

shape_pt_lon	Yes	The shape_pt_lon field associates a shape point's longitude with a shape ID. The field value must be a valid WGS 84 longitude value from -180 to 180. Each row in shapes.txt represents a shape point in your shape definition. For example, if the shape "A_shp" has three points in its definition, the shapes.txt file might contain these rows to define the shape: A_shp,37.61956,-122.48161,0 A_shp,37.64430,-122.41070,6 A_shp,37.65863,-122.30839,11	- For Example: “150.906115" - Formatted to at least 6 decimal places
shape_pt_sequence	Yes	The shape_pt_sequence field associates the latitude and longitude of a shape point with its sequence order along the shape. The values for shape_pt_sequence must be non-negative integers, and they must increase along the trip. For example, if the shape "A_shp" has three points in its definition, the shapes.txt file might contain these rows to define the shape: A_shp,37.61956,-122.48161,0 A_shp,37.64430,-122.41070,6 A_shp,37.65863,-122.30839,11	For Example: "3"

shape_dist_traveled	Yes	When used in the shapes.txt file, the shape_dist_traveled field positions a shape point as a distance traveled along a shape from the first shape point. The shape_dist_traveled field represents a real distance traveled along the route in units such as feet or kilometers. This information allows the trip planner to determine how much of the shape to draw when showing part of a trip on the map. The values used for shape_dist_traveled must increase along with shape_pt_sequence: they cannot be used to show reverse travel along a route. The units used for shape_dist_traveled in the shapes.txt file must match the units that are used for this field in the stop_times.txt file. For example, if a bus travels along the three points defined above for A_shp, the additional shape_dist_traveled values (shown here in kilometers) would look like this: 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	For Example: "20"

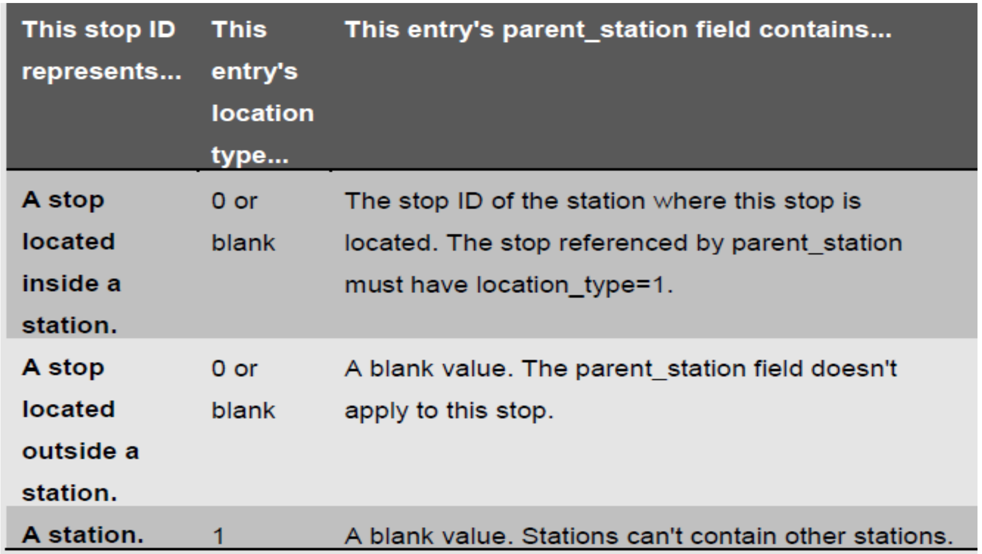
1.16 stops.txt

The Stops.txt file contains the stops that can be referenced by one or more trips within the GTFS file-set bundle.

Field specific requirements required by TfNSW in conjunction with the published GTFS standards for the generation of a **stops.txt** file are defined below:

Field	Required by TfNSW	Published GTFS Field Description	TfNSW specific requirements, additional business rules and sample data
stop_id	Yes	The stop_id field contains an ID that uniquely identifies a stop or station. Multiple routes may use the same stop. The stop_id is dataset unique.	Must be the TfNSW registered stop numbers from the TfNSW Transit Stop Management (TSM) data store, the source of truth for all stops in NSW. The stop numbers must follow the TfNSW stop numbering and Naming standard. For Example: “2154262”
stop_code	No	The stop_code field contains short text or a number that uniquely identifies the stop for passengers. Stop codes are often used in phone-based transit information systems or printed on stop signage to make it easier for riders to get a stop schedule or real-time arrival information for a particular stop. The stop_code field should only be used for stop codes that are displayed to passengers. For internal codes, use stop_id. This field should be left blank for stops without a code.	NA

stop_name	Yes	The stop_name field contains the name of a stop or station. Please use a name that people will understand in the local and tourist vernacular.	The correct “customer facing” name of the Stop. Stop names must be sourced from the TfNSW Transit Stop Management (TSM) data store, the source of truth of stop names in NSW. The stop names must follow the TfNSW stop numbering and Naming standard. For Example: "Castle Hill Station, Platform 1"
stop_desc	No	The stop_desc field contains a description of a stop. Please provide useful, quality information. Do not simply duplicate the name of the stop.	NA
stop_lat	Yes	The stop_lat field contains the latitude of a stop or station. The field value must be a valid WGS 84 latitude.	For Example: “-33.731481" Formatted to at least 6 decimal places
stop_lon	Yes	The stop_lon field contains the longitude of a stop or station. The field value must be a valid WGS 84 longitude value from -180 to 180.	For Example: "151.007701" Formatted to at least 6 decimal places
zone_id	No	The zone_id field defines the fare zone for a stop ID. Zone IDs are required if you want to provide fare information using fare_rules.txt. If this stop ID represents a station, the zone ID is ignored.	NA
stop_url	No	The stop_url field contains the URL of a web page about a particular stop. This should be different from the agency_url and the route_url fields. The value must be a fully qualified URL that includes http:// or https://, and any special characters in the URL must be correctly escaped. Seehttp://www.w3.org/Addressing/URL/4_URI_Recommentations.html for a description of how to create fully qualified URL values	NA

<table>
<tr>
<td>location_type</td>
<td>Yes</td>
<td>The location_type field identifies whether this stop ID represents a stop or station. If no location type is specified, or the location_type is blank, stop IDs are treated as stops. Stations may have different properties from stops when they are represented on a map or used in trip planning.
The location type field can have the following values:
- 0 or blank - Stop. A location where passengers board or disembark from a transit vehicle.
- 1 - Station. A physical structure or area that contains one or more stop.</td>
<td>Indicates that the stop is a ‘Stop’ (stopping point) as opposed to a ‘Station’ (stop group). For Example: "0"</td>
</tr>
<tr>
<td>parent_station</td>
<td>Yes</td>
<td>For stops that are physically located inside stations, the parent_station field identifies the station associated with the stop. To use this field, stops.txt must also contain a row where this stop ID is assigned location type=1.

<table>
<tr><th>This stop ID represents...</th><th>This entry's location type...</th><th>This entry's parent_station field contains...</th></tr>
<tr><td><b>A stop located inside a station.</b></td><td>0 or blank</td><td>The stop ID of the station where this stop is located. The stop referenced by parent_station must have location_type=1.</td></tr>
<tr><td><b>A stop located outside a station.</b></td><td>0 or blank</td><td>A blank value. The parent_station field doesn't apply to this stop.</td></tr>
<tr><td><b>A station.</b></td><td>1</td><td>A blank value. Stations can't contain other stations.</td></tr>
</table>
</td>
<td>Parent stations as supplied by TfNSW from the TfNSW Transit Stop Management (TSM) data store. For Example: Castle Hill Station, Platform 1 ‘s “parent_station” is “2154391” for Castle Hill station</td>
</tr>
</table>

stop_timezone	No	The stop_timezone field contains the timezone in which this stop or station is located. Please refer to Wikipedia List of Timezones for a list of valid values. If omitted, the stop should be assumed to be located in the timezone specified by agency_timezone in agency.txt. When a stop has a parent station, the stop is considered to be in the timezone specified by the parent station's stop_timezone value. If the parent has no stop_timezone value, the stops that belong to that station are assumed to be in the timezone specified by agency_timezone, even if the stops have their own stop_timezone values. In other words, if a given stop has a parent_station value, any stop_timezone value specified for that stop must be ignored. Even if stop_timezone values are provided in stops.txt, the times in stop_times.txt should continue to be specified as time since midnight in the timezone specified by agency_timezone in agency.txt. This ensures that the time values in a trip always increase over the course of a trip, regardless of which timezones the trip crosses.	NA

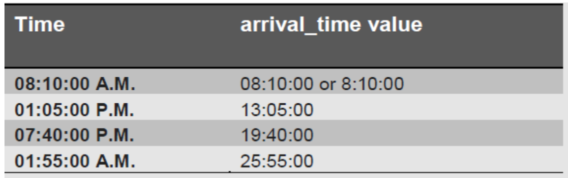
wheelchair_boarding	Yes	The wheelchair_boarding field identifies whether wheelchair boardings are possible from the specified stop or station. The field can have the following values: -0 (or empty) - indicates that there is no accessibility information for the stop -1 - indicates that at least some vehicles at this stop can be boarded by a rider in a wheelchair -2 - wheelchair boarding is not possible at this stop When a stop is part of a larger station complex, as indicated by a stop with a parent_station value, the stop's wheelchair_boarding field has the following additional semantics: - 0 (or empty) - the stop will inherit its wheelchair_boarding value 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	Indicates that a stop is wheelchair accessible. -For Example: “1”
platform_code	Yes	TfNSW Extension	Platform/stop number as detailed in the stop name For Example: “1” for Platform 1

1.17 Stop_times.txt

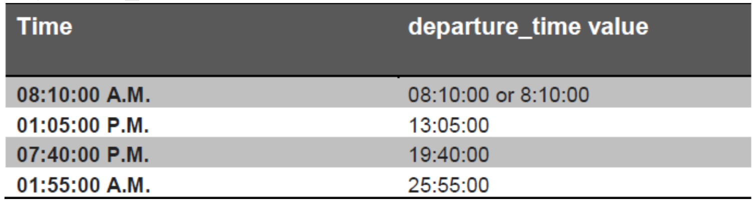
The Stops times file provides stop time information for all stops included in trips defined within the GTFS file-set bundle.

Field specific requirements required by TfNSW in conjunction with the published GTFS standards for the generation of a **stops_times.txt** file are defined below:

Field	Required by TfNSW	Published GTFS Field Description	TfNSW specific requirements, additional business rules and sample data
trip_id	Yes	The trip_id field contains an ID that identifies a trip. This value is referenced from the trips.txt file.	-For Example: “M-I-CUD-CHW-1-0435-3007:1000”

<table>
<tr>
<td>arrival_time</td>
<td>Yes</td>
<td>The arrival_time specifies the arrival time at a specific stop for a specific trip on a route. The time is measured from "noon minus 12h" (effectively midnight, except for days on which daylight savings time changes occur) at the beginning of the service date. For times occurring after midnight on the service date, 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 you don't have separate times for arrival and departure at a stop, enter the same value for arrival_time and departure_time.
If this stop isn't a time point, use an empty string value for the arrival_time and departure_time fields. Stops without arrival times will be scheduled based on the nearest preceding timed stop. To ensure accurate routing, please provide arrival and departure times for all stops that are time points. Do not interpolate stops.
You must specify arrival and departure times for the first and last stops in a trip. Times must be eight digits in HH:MM:SS format (H:MM:SS is also accepted, if the hour begins with 0). Do not pad times with spaces. The following columns list stop times for a trip and the proper way to express those times in the arrival_time field:

<table>
<tr><th>Time</th><th>arrival_time value</th></tr>
<tr><td>08:10:00 A.M.</td><td>08:10:00 or 8:10:00</td></tr>
<tr><td>01:05:00 P.M.</td><td>13:05:00</td></tr>
<tr><td>07:40:00 P.M.</td><td>19:40:00</td></tr>
<tr><td>01:55:00 A.M.</td><td>25:55:00</td></tr>
</table>

Note: Trips that span multiple dates will have stop times greater than 24:00:00. For example, if a trip begins at 10:30:00 p.m. and ends at 2:15:00 a.m. on the following day, the stop times would be 22:30:00 and 26:15:00. Entering those stop times as 22:30:00 and 02:15:00 would not produce the desired results.</td>
<td>The arrival time at a specific stop for a specific trip on a route. Times are to be provided to seconds (level of accuracy) Times for trips starting before 04:00 am will be expressed in ‘36 hour format’. For Example: “"04:52:49"”</td>
</tr>
</table>

<table>
<tr><td>departure_time</td><td>Yes</td><td>The departure_time specifies the departure time from a specific stop for a specific trip on a route. The time is measured from "noon minus 12h" (effectively midnight, except for days on which daylight savings time changes occur) at the beginning of the service date. For times occurring after midnight on the service date, 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 you don't have separate times for arrival and departure at a stop, enter the same value for arrival_time and departure_time.
If this stop isn't a time point, use an empty string value for the arrival_time and departure_time fields. Stops without arrival times will be scheduled based on the nearest preceding timed stop. To ensure accurate routing, please provide arrival and departure times for all stops that are time points. Do not interpolate stops.
You must specify arrival and departure times for the first and last stops in a trip.
Times must be eight digits in HH:MM:SS format (H:MM:SS is also accepted, if the hour begins with 0). Do not pad times with spaces. The following columns list stop times for a trip and the proper way to express those times in the departure_time field:
<table>
<tr><th>Time</th><th>departure_time value</th></tr>
<tr><td><b>08:10:00 A.M.</b></td><td>08:10:00 or 8:10:00</td></tr>
<tr><td><b>01:05:00 P.M.</b></td><td>13:05:00</td></tr>
<tr><td><b>07:40:00 P.M.</b></td><td>19:40:00</td></tr>
<tr><td><b>01:55:00 A.M.</b></td><td>25:55:00</td></tr>
</table>
Note: Trips that span multiple dates will have stop times greater than 24:00:00. For example, if a trip begins at 10:30:00 p.m. and ends at 2:15:00 a.m. on the following day, the stop times would be 22:30:00 and 26:15:00. Entering those stop times as 22:30:00 and 02:15:00 would not produce the desired results.</td><td>The departure time from a specific stop for a specific trip on a route.
Times are to be provided to seconds (level of accuracy)
Times for trips starting before 04:00 am will be expressed in ‘36 hour format’.
For Example: "04:53:18"</td></tr>
<tr><td>stop_id</td><td>Yes</td><td>The stop_id field contains an ID that uniquely identifies a stop. Multiple routes may use the same stop. The stop_id is referenced from the stops.txt file. If location_type is used in stops.txt, all stops referenced in stop_times.txt must</td><td>Must be a TfNSW registered stop number from the TfNSW Transit Stop Management</td></tr>
</table>

		have location_type of 0. Where possible, stop_id values should remain consistent between feed updates. In other words, stop A with stop_id 1 should have stop_id 1 in all subsequent data updates. If a stop is not a time point, enter blank values for arrival_time and departure_time.	('TSM) data store the source of truth for all stops in NSW. The stop numbers must follow the TfNSW stop numbering and Naming standard. For Example: “2126159”
stop_sequence	Yes	The stop_sequence field identifies the order of the stops for a particular trip. The values for stop_sequence must be non-negative integers, and they must increase along the trip. For example, the first stop on the trip could have a stop_sequence of 1, the second stop on the trip could have a stop_sequence of 23, and the third stop could have a stop_sequence of 40, and so on.	The sequence of the Stop within the Trip. 'For Example: “8” (indicating that the stop is the 8th in the Trip)
stop_headsign	Yes	The stop_headsign field contains the text that appears on a sign that identifies the trip's destination to passengers. Use this field to override the default trip_headsign when the headsign changes between stops. If this headsign is associated with an entire trip, use trip_headsign instead.	The stop headsign must applied from the TfNSW registered stop numbers from the TfNSW Transit Stop Management (TSM) data store the source of truth for all stops in NSW. Must not be greater than 15 characters (the current limit) and follow the TfNSW naming standard For Example: “Chatswood”
pickup_type	Yes	The pickup_type field indicates whether passengers are picked up at a stop as part of the normal schedule or whether a pickup at the stop is not available. This field also allows the transit agency to indicate that passengers must call the agency or notify the driver to arrange a pickup at a particular stop. Valid values for this field are: - 0 - Regularly scheduled pickup - 1 - No pickup available - 2 - Must phone agency to arrange pickup - 3 - Must coordinate with driver to arrange pickup The default value for this field is 0.	For Example: “0” (indicating that regularly scheduled pickup occurs at the Stop for the Trip)

drop_off_type	Yes	The drop_off_type field indicates whether passengers are dropped off at a stop as part of the normal schedule or whether a drop off at the stop is not available. This field also allows the transit agency to indicate that passengers must call the agency or notify the driver to arrange a drop off at a particular stop. Valid values for this field are: - 0 - 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 The default value for this field is 0.	For Example: “0” (indicating that no drop off available for the Stop for the Trip)
shape_dist_traveled	Yes	When used in the stop_times.txt file, the shape_dist_traveled field positions a stop as a distance from the first shape point. The shape_dist_traveled field represents a real distance traveled along the route in units such as feet or kilometers. For example, if a bus travels a distance of 5.25 kilometers from the start of the shape to the stop, the shape_dist_traveled for the stop ID would be entered as "5.25". This information allows the trip planner to determine how much of the shape to draw when showing part of a trip on the map. The values used for shape_dist_traveled must increase along with stop_sequence: they cannot be used to show reverse travel along a route. The units used for shape_dist_traveled in the stop_times.txt file must match the units that are used for this field in the shapes.txt file.	Where possible shape distance travelled should be provided For Example: “15760”

timepoint	Yes	The timepoint field can be used to indicate if the specified arrival and departure times for a stop are strictly adhered to by the transit vehicle or if they are instead approximate and/or interpolated times. The field allows a GTFS producer to provide interpolated stop times that potentially incorporate local knowledge, but still indicate if the times are approximate. For stop-time entries with specified arrival and departure times, valid values for this field are: - empty - Times are considered exact. - 0 - Times are considered approximate. - 1 - Times are considered exact. For stop-time entries without specified arrival and departure times, feed consumers must interpolate arrival and departure times. Feed producers may optionally indicate that such an entry is not a timepoint (value=0) but it is an error to mark a entry as a timepoint (value=1) without specifying arrival and departure times.	Indicates whether the Stop is a timing point for the Trip. All stops for Train, Light, Ferry, and Coach are timing points. For bus services only the first and last stop and key locations such as Points of Interest, interchanges along a route are timing points. For Example: “1”
stop_note	Yes	TfNSW Extension Stop notes are provided by Operators for each stop and are to be made available to Customers.	Reference key to the notes.txt file (a TfNSW extension to the standard GTFS – refer to Section 3.9 Notes.txt) where stop notes can be defined. Unique identifier for a note For stops without a stop note this field can be left blank e.g. “” For Example: no example – system generated

1.18 notes.txt

The Notes file is a **TfNSW defined extension** to the standard GTFS file-set. It is used to provide additional informed such as irregularities of a trip or special conditions on a stop to complement standard timetable information. This file contains the actual note text and is reference by the **Trips.txt** file (via the trip_note field) and or the **Stop_times.txt** file (via the stop_note field).

Field specific requirements required by TfNSW in conjunction with the published GTFS standards for the generation of a **notes.txt** file are defined below:

Field	Required by TfNSW	Published GTFS Field Description	TfNSW specific requirements, additional business rules and sample data
note_id	Yes	The note_id field contains an ID that uniquely identifies a note. This ID is referenced in the stop_notes.txt and trips.txt files.	Unique identifier for a note independently generated by Data Provider. For Example: "00001:7808:1"
note_text	Yes	The note_text field contains the note text that will be made available to customers.	Leading and trailing spaces should be trimmed For Example: “Gates close two minutes before scheduled departure time.” Notes must follow the TfNSW guidelines notes and must not be greater than (to be confirmed) characters. For Example: "Train on [Platform 3] departing for [Tallawong]"

GTFS Timetable Feed Usage Notes

1.19 Stop Arrival and Departure Times

End users should display arrival_time and departure_time as supplied in stop_times.txt. Users should not interpolate values between timepoints.

1.20 GTFS Bundle and Operators

The current agency ID of GTFS Bundle is "SMNW"

1.21 GTFS Realtime

"GTFS-realtime is a feed specification that allows public transportation agencies to provide realtime updates about their trips, vehicles and service alerts to application developers. It is an extension to GTFS (General Transit Feed Specification), an open data format for public transportation schedules, trips updates, vehicle movements, service alerts and associated geographic information. GTFS-realtime was designed around ease of implementation, good GTFS interoperability and a focus on passenger information."

- 'What is GTFS-realtime', Google Realtime Transit Overview (https://developers.google.com/transit/gtfs-realtime/)

The TfNSW PTMS GTFS Realtime feed is sourced through AVLS as part of the live vehicle tracking system.

A GTFS-realtime feed lets transit agencies provide consumers with realtime information about disruptions to their service (stations closed, lines not operating, important delays, etc.) location of their vehicles, and expected arrival/departure times.

The specification currently supports the following types of information:

- Trip updates – early/delays, cancellations, add trips and changed routes
- Service alerts – early/delays, add trips, cancellations, change stops, long delays and unforeseen events affecting a station, route or the entire network
- Vehicle positions - information about the vehicles including location, occupancy and congestion level

PTMS TfNSW Realtime updates including Service Alerts also supports both Trip updates and Vehicle Positions. Trip Update and Vehicle Position both use the proto buff format.

Following TfNSW Rules are applicable to the feeds:

1. **Configurable capacity** – Feeds are configured to contain all the trips that are currently in progress and started up to 30 minutes in the past as well as trips that will commence within next 60 minutes. This capacity is configurable in the SIRI Broker and can be extended/reduced depending on future needs.
2. **Operator Exclusion List** – Feeds currently work for all revenue trips i.e., non-commercial trips serviced by the operators will not be available in the feed.
3. **Change Path List** – Feeds currently work to support change path which include change in platform and cancelled stops.
4. **Prediction management** - PTMS does not have any prediction engines and make use of the predictions from the AVLS system.
5. **Next trip early/delay calculation** – PTMS does not have any built in early/delay calculation engine. Early/delay calculations are part of the AVLS system.
6. **Additional Vehicle information** – Vehicle position updates feed is extended to include following information about each vehicle that is in service:
   a. Vehicle Occupancy model
   b. Vehicle information supported under the TfNSW GTFS S & GTFS R implementation specification
7. **Publishing cancelled trips** – Trip updates feed supports the publishing of trips that are cancelled (commenced their Journey or not) by the operators via PTMS. Reference to all the vehicles that are associated with a cancelled trip will be removed from the vehicle position updates feed.
8. **GTFS-R Update frequency** –

a. Trip updates will be sent by the AVLS system every time there is a change in early/delay predictions that exceed the Hysteresis. Currently it has been recommended and set to 20 seconds.
b. Trip updates commence 60 minutes prior to the trip commencement
c. Trip updates continue for 30 minutes post the trip completion
d. Vehicle Positions are received at least every 15 seconds from the Vehicle to the AVLS backend system and then delivered to GTFS Server via the SIRI Broker
e. Vehicle Position updates commences when the vehicle status is recorded as “atOrigin”. While the vehicle is performing the journey the status reported is “inProgress and when the vehicle completes the trip, status reported is “completed”
f. Service Alerts automatically generated on Long Delays, Add Trips, Cancellation of trips and Change Paths.
g. Service Alerts are manually created and completed on an external Incident Management system.

1.22 Trip Updates Message

Message Structure	Message Example

{
"header": {
"gtfs_realtime_version": ,
"incrementality": ,
"incrementalitySpecified": ,
"timestamp": ,
"timestampSpecified":
},
"entity": [
{
"id": ,
"is_deletedSpecified": ,
"trip_update": {
"trip": {
"trip_id": ,
"trip_idSpecified": ,
"route_id": ,
"route_idSpecified": ,
"direction_id": ,
"direction_idSpecified": ,
"start_time": ,
"start_timeSpecified": ,
"start_date": ,
"start_dateSpecified": ,
"schedule_relationship": ,
"schedule_relationshipSpecified":
},
"vehicle": {
"id": ,
"idSpecified": ,
"label": ,
"labelSpecified": ,
"license_plateSpecified": ,
"tfnsw_vehicle_descriptor":
},
"stop_time_update": [
{
"stop_sequence": ,
"stop_sequenceSpecified": ,
"stop_id": ,
"stop_idSpecified": ,
"arrival": ,
"departure": {
"delay": ,
"delaySpecified": ,
"time": ,
"timeSpecified": ,
"uncertaintySpecified":
	entity {
	id: "28/2023-07-20T14:00:35+10:00/M-I-CUD-CHW-1-1501-3116"
	trip_update {
	trip {
	trip_id: "M-I-CUD-CHW-1-1501-3116:1000"
	route_id: "SMNW_M"
	direction_id: 1
	start_time: "15:01:00"
	start_date: "20230720"
	schedule_relationship: SCHEDULED
	}
	vehicle {
	id: "Unassigned"
	label: "Unassigned"
	}
	stop_time_update {
	stop_sequence: 1
	stop_id: "2155270"
	departure {
	delay: 0
	time: 1689829260
	}
	schedule_relationship: SCHEDULED
	}
	stop_time_update {
	stop_sequence: 2
	stop_id: "2155267"
	arrival {
	delay: 0
	time: 1689829385
	}
	departure {
	delay: 0
	time: 1689829414
	}
	schedule_relationship: SCHEDULED
	}
	stop_time_update {
	stop_sequence: 3
	stop_id: "2155265"
	arrival {
	delay: 0
	time: 1689829551

},
"schedule_relationship": ,
"schedule_relationshipSpecified":
},
],
"timestamp": ,
"timestampSpecified": ,
"delaySpecified":
},
"vehicle": ,
"alert":
},
}
]
}
	}
	departure {
	delay: 0
	time: 1689829580
	}
	schedule_relationship: SCHEDULED
	}
	stop_time_update {
	stop_sequence: 4
	stop_id: "2153402"
	arrival {
	delay: 0
	time: 1689829699
	}
	departure {
	delay: 0
	time: 1689829728
	}
	schedule_relationship: SCHEDULED
	}
	stop_time_update {
	stop_sequence: 5
	stop_id: "2153404"
	arrival {
	delay: 0
	time: 1689829854
	}
	departure {
	delay: 0
	time: 1689829883
	}
	schedule_relationship: SCHEDULED
	}
	stop_time_update {
	stop_sequence: 6
	stop_id: "2154264"
	arrival {
	delay: 0
	time: 1689830014
	}
	departure {
	delay: 0
	time: 1689830043
	}
	schedule_relationship: SCHEDULED
	}
	stop_time_update {
	stop_sequence: 7
	stop_id: "2154262"

```
  arrival {
    delay: 0
    time: 1689830164
  }
  departure {
    delay: 0
    time: 1689830193
  }
  schedule_relationship: SCHEDULED
}
stop_time_update {
  stop_sequence: 8
  stop_id: "2126159"
  arrival {
    delay: 0
    time: 1689830329
  }
  departure {
    delay: 0
    time: 1689830358
  }
  schedule_relationship: SCHEDULED
}
stop_time_update {
  stop_sequence: 9
  stop_id: "2121225"
  arrival {
    delay: 0
    time: 1689830668
  }
  departure {
    delay: 0
    time: 1689830697
  }
  schedule_relationship: SCHEDULED
}
stop_time_update {
  stop_sequence: 10
  stop_id: "2113351"
  arrival {
    delay: 0
    time: 1689830888
  }
  departure {
    delay: 0
    time: 1689830917
  }
  schedule_relationship: SCHEDULED
}
```

```
    stop_time_update {
      stop_sequence: 11
      stop_id: "2113341"
      arrival {
        delay: 0
        time: 1689831009
      }
      departure {
        delay: 0
        time: 1689831038
      }
      schedule_relationship: SCHEDULED
    }
    stop_time_update {
      stop_sequence: 12
      stop_id: "2113361"
      arrival {
        delay: 0
        time: 1689831134
      }
      departure {
        delay: 0
        time: 1689831163
      }
      schedule_relationship: SCHEDULED
    }
    stop_time_update {
      stop_sequence: 13
      stop_id: "2067142"
      arrival {
        delay: 0
        time: 1689831480
      }
      schedule_relationship: SCHEDULED
    }
    timestamp: 1689825635
  }
}
```

1.22.1 Trip Updates Message

Field Name	*Type*	*Required*	*Cardinality*	*Description*	*TfNSW Rules*
Id	string	Required	One		The id should be traceable back to the incoming source SIRI message
trip	TripDescriptor	Required	One	The Trip that this message applies to. There can be at most one TripUpdate entity for each actual trip instance. If there is none, that means there is no prediction information available. It does *not* mean that the trip is progressing according to schedule.	All the trips that should be in progress as per the schedule are listed up to the configured time period of 120 minutes (configurable) in the past. All the trips that will be running in the future (up to the configured time period of 60 minutes (configurable), as per the schedule are listed All unscheduled trips that are currently in progress will contain a valid trip_id and will be are listed.
vehicle	VehicleDescriptor	Optional	One	Additional information on the vehicle that is serving this trip.	*TfNSW to revisit at a later date*
stop_time_update	StopTimeUpdate	Conditionally required	Many	Updates to StopTimes for the trip (both future, i.e., predictions, and in some cases, past ones, i.e., those that already happened). The updates must be sorted by stop_sequence, and apply for all the	REPLACEMENT - only future TSNs will contain stop time updates, except for REPLACEMENT services where the entire stop sequence is provided.

				following stops of the trip up to the next specified stop_time_update. At least one stop_time_update must be provided for the trip unless the trip.schedule_relationship is CANCELED - if the trip is canceled, no stop_time_updates need to be provided.	CANCELED –stop time information is not required for CANCELED services
timestamp	uint64	Optional	One	Moment at which the vehicle's real-time progress was measured. In POSIX time (i.e., the number of seconds since January 1st 1970 00:00:00 UTC).	For each trip that is in progress and being serviced by a vehicle, this field will be populated with the driver console's time from the most recent message from that vehicle else this field will not be included in the feed.
delay	int32	Optional	One	The current schedule deviation for the trip. Delay should only be specified when the prediction is given relative to some existing schedule in GTFS. Delay (in seconds) can be positive (meaning that the vehicle is late) or negative (meaning that the vehicle is ahead of schedule). Delay of 0 means that the vehicle is exactly on time. Delay information in StopTimeUpdates take precedent of trip-level delay information, such that trip-level delay is only propagated until the next stop along the trip with a StopTimeUpdate delay value specified. Feed providers are strongly encouraged to provide a TripUpdate.timestamp value indicating when the delay value was last updated, in order to evaluate the freshness of the data. Caution: this field is still experimental, and subject to change. It may be formally adopted in the future.	N/A - this field is not populated High frequency services such as Metro and light rail run to an operational timetable as per the relevant GTFS bundles, however they will adjust to headway in response to operational requirements throughout the day. Therefore, we recommend ignoring the delay information passed on in GTFS-R feeds for these routes, and only showing real-time arrival/departure times to customers.

1.22.1.1 Trip Descriptor

Field Name	***Type***	***Required***	***Cardinality***	***Description***	***TfNSW Rules***
trip_id	string	Conditionally required	One	The trip_id from the GTFS feed that this selector refers to. For non frequency-based trips (trips not defined in GTFS frequencies.txt), this field is enough to uniquely identify the trip. For frequency-based trips defined in GTFS frequencies.txt, trip_id, start_time, and start_date are all required. For scheduled-based trips (trips not defined in GTFS frequencies.txt), trip_id can only be omitted if the trip can be uniquely identified by a combination of route_id, direction_id, start_time, and start_date, and all those fields are provided.	Required by TfNSW trip_id must be unique within a GTFS-R feed For GTFS-R messages that have duplicate trip_ids these trips will be set back to scheduled time. All real time information will be ignored by the consuming system. In GTFS-R used elements must be consistent with values coming from GTFS. The trip_id in GTFS_R must match in GTFS, except for trips which have been ADDED. This field is to be mapped to the Vehicle Journey ID in the input feed
route_id	string	Conditionally required	One	The route_id from the GTFS that this selector refers to. If trip_id is omitted, route_id must be provided.	Required by TfNSW In GTFS-R used element route_id must be consistent with values coming from GTFS. For inconsistent route_id´s, the trip is mapped back to the GTFS using the stopping sequence in order to derive a route id.

direction_id	uint32	Conditionally required	One	The direction_id from the GTFS feed trips.txt file, indicating the direction of travel for trips this selector refers to. If trip_id is omitted, direction_id must be provided. Caution: this field is still experimental, and subject to change. It may be formally adopted in the future.	For trip_update:: SCHEDULED where direction_id is not provided, then direction is taken from GTFS on basis of consistent trip_id´s. For trip_update::ADDED, direction is required

start_time	string	Conditionally required	One	The initially scheduled start time of this trip instance. When the trip_id corresponds to a nonfrequency-based trip, this field should either be omitted or be equal to the value in the GTFS feed. When the trip_id correponds to a frequency-based trip defined in GTFS frequencies.txt, start_time is required and must be specified for trip updates and vehicle positions. If the trip corresponds to exact_times=1 GTFS record, then start_time must be some multiple (including zero) of headway_secs later than frequencies.txt start_time for the corresponding time period. If the trip corresponds to exact_times=0, then its start_time may be arbitrary, and is initially expected to be the first departure of the trip. Once established, the start_time of this frequency-based exact_times=0 trip should be considered immutable, even if the first departure time changes -- that time change may instead be reflected in a StopTimeUpdate. If trip_id is omitted, start_time must be provided. Format and semantics of the field is same as that of GTFS/frequencies.txt/start_time, e.g., 11:15:35 or 25:15:35.	Required by TfNSW This is to be equivalent of the trip start time associated with the trip_id in the schedule.

start_date	string	Conditionally required	One	The start date of this trip instance in YYYYMMDD format. For scheduled trips (trips not defined in GTFS frequencies.txt), this field must be provided to disambiguate trips that are so late as to collide with a scheduled trip on a next day. For example, for a train that departs 8:00 and 20:00 every day, and is 12 hours late, there would be two distinct trips on the same time. This field can be provided but is not mandatory for schedules in which such collisions are impossible - for example, a service running on hourly schedule where a vehicle that is one hour late is not considered to be related to schedule anymore. This field is required for frequency-based trips defined in GTFS frequencies.txt. If trip_id is omitted, start_date must be provided.	Required by TfNSW Where we have repeated trip_id on subsequent days (everyday trips) a start_date is required This is to be equivalent of the trip start date associated with the trip ID in the schedule. Where a trip id is not available and route id (and route variant number) cannot be mapped to the trips in the schedule then this field will contain the present date.
schedule_relationship	ScheduleRelationship	Optional	One		Required by TfNSW

1.22.1.1.1 Schedule Relationship

Value	*Comment*	*TfNSW Rules*
SCHEDULED	Trip that is running in accordance with its GTFS schedule, or is close enough to the scheduled trip to be associated with it.	GTFS-R trip_id must exist and is unique in corresponding GTFS data When no other trip with same trip_id is in in Progress when the trip entity is created then the schedule relationship for the trip will be SCHEDULED

ADDED	An extra trip that was added in addition to a running schedule, for example, to replace a broken vehicle or to respond to sudden passenger load.	GTFS-R trip_id must not exist in corresponding GTFS data and must be unique across all GTFS/GTFS-R trip_ids The trip must be provided with full stop sequence. Trip is consistent and persistent across messages. trip_id of CANCELED trip must not be reused for a new ADDED trip The route_id of an added trip must refer to a route_id in GTFS. direction_id is required for ADDED trips When another trip with same trip_id is in progress when the trip entity is created then the schedule relationship for the trip will be 'ADDED'
UNSCHEDULED	A trip that is running with no schedule associated to it - this value is used to identify trips defined in GTFS frequencies.txt with exact_times = 0. It should not be used to describe trips not defined in GTFS frequencies.txt, or trips in GTFS frequencies.txt with exact_times = 1.	These are additional trips, that do not go on the same stop sequence or do have a different timing than an existing GTFS scheduled trip. GTFS-R trip_id must not exist in corresponding GTFS data and must be unique across all GTFS/GTFS-R trip_ids UNSCHEDULED trips do not have a trip_id but they have route_id. Real-time information is provided only for future stops. There must never more than one UNSCHEDULED trip per route at the same time.
		UNSCHEDULED trips are provided with arrival_delay and departure_delay are set to 0. Arrival time and departure time should be populated for all UNSCHEDULED service stops
CANCELED	A trip that existed in the schedule but was removed.	The trip_id must be either a SCHEDULED trip in GTFS or the trip_id must be an ADDED trip which does not exist in GTFS data.

REPLACEMENT	A replacement service for a scheduled or added trip	REPLACMENT was compliant with an older GTFS-R standard, but is not compliant with the latest version of GTFS-R. route_id and trip_id in trip_update::REPLACEMENT references to GTFS data Where Stop sequence GTFS-R = GTFS REPLACEMENT trips are handled as trip_update::SCHEDULED Stop sequence GTFS-R != GTFS REPLACEMENT trips are handled as if the existing scheduled trip is rerouted. Full Stop Sequence and Full Stop List must be provided, including passed stops if a trip is in progress. Passing Stops stops must have a schedule_relationship of SKIPPED The message must continually send the full stop list and stop sequence for all updates until the trip is completed. Both the stop_time and stop_date are required. Both arrival_time and departure_time are required for each stop. Delay for each stop is also required in order for consumers to calculate the intended scheduled time of a replacement trip

1.22.1.2 Vehicle Descriptor

Field Name	***Type***	***Required***	***Cardinality***	***Description***	***TfNSW Rules***
id	string	Optional	One	Internal system identification of the vehicle. Should be unique per vehicle, and is used for tracking the vehicle as it proceeds through the system. This id should not be made visible to the enduser; for that purpose use the label field	Required by TfNSW This should be mapped to the SIRI Vehicle Descriptor.
label	string	Optional	One	User visible label, i.e., something that must be shown to the passenger to help identify the correct vehicle.	NA
license_plate	string	Optional	One	The license plate of the vehicle.	NA

1.22.1.3 StopTimeUpdate

Field Name	***Type***	***Required***	***Cardinality***	***Description***	***TfNSW Rules***
stop_sequence	uint32	Conditionally required	One	Must be the same as in stop_times.txt in the corresponding GTFS feed. Either stop_sequence or stop_id must be provided within a StopTimeUpdate - both fields cannot be empty. stop_sequence is required for trips that visit the same stop_id more than once (e.g., a loop) to disambiguate which stop the prediction is for.	Required by TfNSW
stop_id	string	Conditionall	One	Must be the same as in stops.txt in the corresponding	Required by TfNSW

		y required		GTFS feed. Either stop_sequence or stop_id must be provided within a StopTimeUpdate - both fields cannot be empty.	In GTFS-R used elements must be consistent with values coming from GTFS. stop_id and stop_sequence_id in GTFS_R must match with GTFS . For each trip listed as per the schedule, all the related stop ID's will be listed For all the 'unscheduled' trips listed all the related stop ID's will be listed
arrival	StopTimeEvent	Conditionally required	One	If schedule_relationship is empty or SCHEDULED, either arrival or departure must be provided within a StopTimeUpdate - both fields cannot be empty. arrival and departure may both be empty when schedule_relationship is SKIPPED. If schedule_relationship is NO_DATA, arrival and departure must be empty.	Required by TfNSW
departure	StopTimeEvent	Conditionally required	One	If schedule_relationship is empty or SCHEDULED, either arrival or departure must be provided within a StopTimeUpdate - both fields cannot be empty. arrival and departure may both be empty when schedule_relationship is SKIPPED. If schedule_relationship is NO_DATA, arrival and departure must be empty.	Required by TfNSW
schedule_relationship	ScheduleRelationship	Optional	One	The default relationship is SCHEDULED.	

1.22.1.3.1 StopTimeEvent

Field Name	Type	Required	Cardinality	Description	TfNSW Rules
delay	int32	Conditionally required	One	Delay (in seconds) can be positive (meaning that the vehicle is late) or negative (meaning that the vehicle is ahead of schedule). Delay of 0 means that the vehicle is exactly on time. Either delay or time must be provided within a StopTimeEvent - both fields cannot be empty.	stop_time_update::delay for all trips that have trip_update:: = SCHEDULED is mandatory, for stops in the sequence. When the arrival field is populated in the 'StopTimeUpdate' entity then one of the following conditions will apply: For a particular 'TSN', compare the predicted arrival time for that 'TSN' with the scheduled arrival time and if there is delay then convert it in seconds and populate this field with it. For a particular 'TSN', compare the predicted arrival time for that 'TSN' with the scheduled arrival time and if it is early then convert it in seconds, prefix it with '-' symbol and populate this field with it. For a particular 'TSN', compare the predicted arrival time for that 'TSN' with the scheduled arrival time and if there no difference then populate this field with 0. For a particular 'TSN', if a prediction is not available then populate this field with 0. When the departure field is populated in the StopTimeUpdate' entity then one of the following conditions will apply: For a particular 'TSN', compare the predicted departure time for that 'TSN' with the scheduled departure time and if there is delay then convert it in seconds and populate this field with it. For a particular 'TSN', compare the predicted departure time for that 'TSN' with the scheduled departure time and if it is early then convert it in seconds, prefix it with '-' symbol and populate this field with it. For a particular 'TSN', compare the predicted departure time for that 'TSN' with the scheduled departure time and if there no

					difference then populate this field with 0. d. For a particular 'TSN', if a prediction is not available then populate this field with 0.
time	int64	Conditionally required	One	Event as absolute time. In POSIX time (i.e., number of seconds since January 1st 1970 00:00:00 UTC). Either delay or time must be provided within a StopTimeEvent - both fields cannot be empty.	For all trips that have trip_update:: = SCHEDULED, actual time will be calculated by adding stop_time_update::delay to scheduled time taken from GTFS. This is true for arrival and departure. Therefore trip_update::time is optional. When the arrival field is populated in the 'StopTimeUpdate' entity then this field contains the predicted arrival time when the vehicle will be arriving at the TSN.
uncertainty	int32	Optional	One	If uncertainty is omitted, it is interpreted as unknown. To specify a completely certain prediction, set its uncertainty to 0.	Uncertainty applies to both the time and the delay value of a StopTimeUpdate. The uncertainty specifies the expected error in true delay as an integer in seconds. For example, a service with an estimated delay of 60-seconds arriving to its next stop within a 30-second window of error (+/- 15 seconds) will have an Uncertainty value of 30.

1.22.1.3.2 Schedule Relationship

Value	*Comment*	*TfNSW Rules*
SCHEDULED	The vehicle is proceeding in accordance with its static schedule of stops, although not necessarily according to the times of the schedule. This is the default behaviour. At least one of arrival and departure must be provided. If the schedule for this stop contains both arrival and departure times then so must this update.	Stop must have same stop_sequence in GTFS and GTFS-R For each trip listed as per the schedule, if a vehicle is associated then this option will be selected for that trip For each 'unscheduled' trip listed this option will be selected for that trip
SKIPPED	The stop is skipped, i.e., the vehicle will not stop at this stop. Arrival and departure are optional.	Skipped stop must have the same stop_sequence in GTFS and GTFS-R. All stops before and after the skipped stop do not change the stop_sequence When the scheduled arrival time at a TSN is less than the feed generation time then the 'schedule_relationship' will be populated with a value of 'SKIPPED'
NO_DATA	No data is given for this stop. It indicates that there is no real-time information available. When set NO_DATA is propagated through subsequent stops so this is the recommended way of specifying from which stop you do not have real-time information. When NO_DATA is set neither arrival nor departure should be supplied.	stop must have same stop_sequence in GTFS and GTFS-R NO_DATA results in real-time not being available for the full trip

1.23 Vehicle Position Updates Message

Message Structure	Message Example

{
"header": {
"gtfs_realtime_version": ,
"incrementality": ,
"incrementalitySpecified": ,
"timestamp": ,
"timestampSpecified":
},
"entity": [
{
"id": ,
"is_deletedSpecified": ,
"trip_update": ,
"vehicle": {
"trip": {
"trip_id": ,
"trip_idSpecified": ,
"route_id": ,
"route_idSpecified": ,
"direction_id": ,
"direction_idSpecified": ,
"start_time": ,
"start_timeSpecified": ,
"start_date": ,
"start_dateSpecified": ,
"schedule_relationship": ,
"schedule_relationshipSpecified":
},
"vehicle": {
"id": ,
"idSpecified": ,
"label":,
"labelSpecified": ,
"license_plateSpecified": ,
"tfnsw_vehicle_descriptor": {
"air_conditioned": ,
"air_conditionedSpecified": ,
"wheelchair_accessible": ,
"wheelchair_accessibleSpecified": ,
"vehicle_model": ,
"vehicle_modelSpecified": ,
"performing_prior_tripSpecified": ,
"special_vehicle_attributes": ,
"special_vehicle_attributesSpecified":
}
},
"position": {
"latitude": ,
	header {
	gtfs_realtime_version: "1.0"
	incrementality: FULL_DATASET
	timestamp: 1689829362
	}
	entity {
	id: "0/2023-07-20T05:02:37Z/RS001"
	vehicle {
	trip {
	trip_id: "M-I-CUD-CHW-2-1505-3128:1000"
	route_id: "SMNW_M"
	direction_id: 1
	start_time: "15:05:00"
	start_date: "20230720"
	schedule_relationship: SCHEDULED
	}
	vehicle {
	id: "RS001"
	label: "RS001"
	license_plate: "RS001"
	tfnsw_vehicle_descriptor {
	air_conditioned: true
	wheelchair_accessible: 1
	vehicle_model: "Alstom Metropolis"
	special_vehicle_attributes: 0
	}
	}
	position {
	latitude: -33.6913567
	longitude: 150.906723
	bearing: 70
	speed: 0
	}
	current_stop_sequence: 1
	stop_id: "2155269"
	current_status: STOPPED_AT
	timestamp: 1689829357
	congestion_level: SEVERE_CONGESTION
	occupancy_status: MANY_SEATS_AVAILABLE
	consist {
	name: "DTC1"
	position_in_consist: 0
	occupancy_status: MANY_SEATS_AVAILABLE
	quiet_carriage: false

```
     "longitude": ,
     "bearing": ,
     "bearingSpecified": ,
     "odometer": ,
     "odometerSpecified": ,
     "speed": ,
     "speedSpecified":
    },
    "current_stop_sequence": ,
    "current_stop_sequenceSpecified": ,
    "stop_id": ,
    "stop_idSpecified": ,
    "current_status": ,
    "current_statusSpecified": ,
    "timestamp": ,
    "timestampSpecified": ,
    "congestion_level": ,
    "congestion_levelSpecified": ,
    "occupancy_statusSpecified": ,
    "consist": []
   },
   "alert":
  },
  }
 ]
}
```

```
      toilet: NONE
      luggage_rack: false
    }
    consist {
      name: "MPC1"
      position_in_consist: 0
      occupancy_status: MANY_SEATS_AVAILABLE
      quiet_carriage: true
      toilet: NONE
      luggage_rack: false
    }
    consist {
      name: "MC1"
      position_in_consist: 0
      occupancy_status: MANY_SEATS_AVAILABLE
      quiet_carriage: false
      toilet: NORMAL
      luggage_rack: false
    }
    consist {
      name: "MC2"
      position_in_consist: 0
      occupancy_status: MANY_SEATS_AVAILABLE
      quiet_carriage: false
      toilet: NONE
      luggage_rack: true
    }
    consist {
      name: "MPC2"
      position_in_consist: 0
      occupancy_status: MANY_SEATS_AVAILABLE
      quiet_carriage: true
      toilet: NORMAL
      luggage_rack: true
    }
    consist {
      name: "DTC2"
      position_in_consist: 0
      occupancy_status: MANY_SEATS_AVAILABLE
      quiet_carriage: true
      toilet: NONE
      luggage_rack: true
    }
  }
}
```

1.23.1 Vehicle Position Updates Message

Field Name	*Type*	*Required*	*Cardinality*	*Description*	*TfNSW Rules*
Id	string	required	One		The id should be traceable back to the source providing system
trip	TripDescriptor	Optional	One	The Trip that this vehicle is serving. Can be empty or partial if the vehicle can not be identified with a given trip instance.	Required by TfNSW All the trips that should be in progress as per the schedule are listed up to the configured time period of 120 minutes (configurable) the past. All the trips that will be running in the future up to the configured time period of 60 minutes (configurable) as per the schedule are listed All unscheduled trips that are currently in progress will contain a valid trip_id and will be listed.
vehicle	VehicleDescriptor	Optional	One	Additional information on the vehicle that is serving this trip. Each entry should have a unique vehicle id.	Required by TfNSW
position	Position	Optional	One	Current position of this vehicle.	Required by TfNSW

current_stop_sequence	uint32	Optional	One	The stop sequence index of the current stop. The meaning of current_stop_sequence (i.e., the stop that it refers to) is determined by current_status. If current_status is missing IN_TRANSIT_TO is assumed.	Required by TfNSW If the vehicle is in transit to the first stop then the current_stop_sequence will be “1”.
stop_id	string	Optional	One	Identifies the current stop. The value must be the same as in stops.txt in the corresponding GTFS feed.	Required by TfNSW If the vehicle is in transit to the first stop then the stop_id will be that of the first TSN. This field is equivalent of 'TSN' sent in the message from the vehicle
current_status	VehicleStopStatus	Optional	One	The exact status of the vehicle with respect to the current stop. Ignored if current_stop_sequence is missing.	Required by TfNSW
timestamp	uint64	Optional	One	Moment at which the vehicle's position was measured. In POSIX time (i.e., number of seconds since January 1st 1970 00:00:00 UTC).	Required by TfNSW This field is equivalent of 'ConsoleTime' in the message received from the vehicle
congestion_level	CongestionLevel	Optional	One		Required by TfNSW

occupancy_status	*OccupancyStatus*	*Required*	One	The degree of passenger occupancy of the vehicle.	Required by TfNSW
Transit_realtime.tfnsw_vehicle_descriptor	User extension	*Optional*		Include following information about the bus in service: Air-conditioning is available or not and Wheelchair access is available or not	Required by TfNSW This field contains following information: Air-conditioning availability – sent as a 'True/False' value Wheelchair access availability – sent as a bit value – 0 being not available and 1 being available Vehicle Model – sent as text value Following information is sent as a bitmask in the 'special_vehicle_attributes': Wi-Fi availability – bitmask value 0001 Christmas bus or not – bitmask value 0010 Both Wi-Fi and Christmas bus -bitmask value 0011 TfNSW has been assigned extension 1007 for these attributes.
CarriageDescriptor	User extension	*Optional*	Many	To pass on information at a carriage level such as occupancy levels and facilitates available.	Required by TfNSW TfNSW has been assigned extension 1007 for these attributes.

1.23.1.1 Trip Descriptor

Field Name	***Type***	***Required***	***Cardinality***	***Description***	***TfNSW Rules***
trip_id	string	Conditionally required	One	The trip_id from the GTFS feed that this selector refers to. For non frequencybased trips (trips not defined in GTFS frequencies.txt), this field is enough to	Required by TfNSW trip_id must be unique within a GTFS-R feed For GTFS-R messages that have duplicate trip_ids these trips will be set back to scheduled time. All real time
				uniquely identify the trip. For frequency-based trips defined in GTFS frequencies.txt, trip_id, start_time, and start_date are all required. For scheduled-based trips (trips not defined in GTFS frequencies.txt), trip_id can only be omitted if the trip can be uniquely identified by a combination of route_id, direction_id, start_time, and start_date, and all those fields are provided.	information will be ignored. In GTFS-R used elements must be consistent with values coming from GTFS. The trip_id in GTFS_R must match in GTFS, except for trips which have been ADDED. This field is to be mapped to the Vehicle Journey ID in the input feed
route_id	string	Conditionally required	One	The route_id from the GTFS that this selector refers to. If trip_id is omitted, route_id must be provided.	Required by TfNSW In GTFS-R used element route_id must be consistent with values coming from GTFS. For inconsistent route_id´s, the trip is mapped back to the GTFS using the stopping sequence in order to derive a route id.

direction_id	uint32	Conditionally required	One	The direction_id from the GTFS feed trips.txt file, indicating the direction of travel for trips this selector refers to. If trip_id is omitted, direction_id must be provided. Caution: this field is still experimental, and subject to change. It may be formally adopted in the future.	For trip_update:: SCHEDULED where direction_id is not provided, then direction is taken from GTFS on basis of consistent trip_id´s. For trip_update::ADDED, direction is required
start_time	string	Conditionally required	One	The initially scheduled start time of this trip instance. When the trip_id corresponds to a non-frequency-based trip, this field should either be omitted or be equal to the value in the GTFS feed. When the trip_id correponds to a frequency-based trip defined in GTFS	Required by TfNSW This is to be equivalent of the trip start time associated with the trip ID in the schedule.

				frequencies.txt, start_time is required and must be specified for trip updates and vehicle positions. If the trip corresponds to exact_times=1 GTFS record, then start_time must be some multiple (including zero) of headway_secs later than frequencies.txt start_time for the corresponding time period. If the trip corresponds to exact_times=0, then its start_time may be arbitrary, and is initially expected to be the first departure of the trip. Once established, the start_time of this frequency-based exact_times=0 trip should be considered immutable, even if the first departure time changes -- that time change may instead be reflected in a StopTimeUpdate. If trip_id is omitted, start_time must be provided. Format and semantics of the field is same as that of GTFS/frequencies.txt/start_time, e.g., 11:15:35 or 25:15:35.	
start_date	string	Conditionally required	One	The start date of this trip instance in YYYYMMDD format. For scheduled trips (trips not defined in GTFS frequencies.txt), this field must be provided to disambiguate trips that are so late as to collide with a scheduled trip on a next day. For example, for a train that departs 8:00 and 20:00 every day, and is 12 hours late, there would be two distinct trips on the same time. This field can be provided but is not	Required by TfNSW This is to be equivalent of the trip start date associated with the trip ID in the schedule. Where we have repeated trip_id on subsequent days (everyday trips) a start_date is required Where a trip_id is not available and route_id (and route variant number) cannot be mapped to the trips in the schedule then this field will contain the present date.

				mandatory for schedules in which such collisions are impossible - for example, a service running on hourly schedule where a vehicle that is one hour late is not considered to be related to schedule anymore. This field is required for frequency-based trips defined in GTFS frequencies.txt. If trip_id is omitted, start_date must be provided.	
schedule_relationship	ScheduleRelationship	Optional	One		Required by TfNSW

1.23.1.1.1 Schedule Relationship

Value	*Comment*	*TfNSW Rules*
SCHEDULED	Trip that is running in accordance with its GTFS schedule, or is close enough to the scheduled trip to be associated with it.	GTFS-R trip_id must exist and is unique in corresponding GTFS data When no other trip with same trip_id is in in Progress when the trip entity is created then the schedule relationship for the trip will be SCHEDULED
ADDED	An extra trip that was added in addition to a running schedule, for example, to replace a broken vehicle or to respond to sudden passenger load.	GTFS-R trip_id must not exist in corresponding GTFS data and must be unique across all GTFS/GTFS-R trip_ids The trip must be provided with full stop sequence. Trip is consistent and persistent across messages. trip_id of CANCELED trip must not be reused for a new ADDED trip
		The route_id of an added trip must refer to a route_id in GTFS. direction_id is required for ADDED trips When another trip with same trip_id is in progress when the trip entity is created then the schedule relationship for the trip will be 'ADDED'

UNSCHEDULED	A trip that is running with no schedule associated to it - this value is used to identify trips defined in GTFS frequencies.txt with exact_times = 0. It should not be used to describe trips not defined in GTFS frequencies.txt, or trips in GTFS frequencies.txt with exact_times = 1.	These are additional trips, that do not go on the same stop sequence or do have a different timing than an existing GTFS scheduled trip. GTFS-R trip_id must not exist in corresponding GTFS data and must be unique across all GTFS/GTFS-R trip_ids UNSCHEDULED trips do not have a trip_id but they have route_id. Real-time information is provided only for future stops. There must never more than one UNSCHEDULED trip per route at the same time. UNSCHEDULED trips are provided with arrival_delay and departure_delay are set to 0. Arrival time and departure time should be populated for all UNSCHEDULED service stops
CANCELED	A trip that existed in the schedule but was removed.	The trip_id must be either a SCHEDULED trip in GTFS or the trip_id must be an ADDED trip which does not exist in GTFS data.
REPLACEMENT	A replacement service for a scheduled or added trip	REPLACMENT was compliant with an older GTFS-R standard, but is not compliant with the latest version of GTFS-R. route_id and trip_id in trip_update::REPLACEMENT references to GTFS data Where Stop sequence GTFS-R = GTFS REPLACEMENT trips are handled as trip_update::SCHEDULED Stop sequence GTFS-R != GTFS REPLACEMENT trips are handled as if the existing scheduled trip is rerouted.

		Full Stop Sequence and Full Stop List must be provided, including passed stops if a trip is in progress and for platform changes the TSN of the new stop needs to be included. Passing Stops stops must have a schedule_relationship of SKIPPED The message must continually send the full stop list and stop sequence for all updates until the trip is completed. Both the stop_time and stop_date are required. Both arrival_time and departure_time are required for each stop. Delay for each stop is also required in order for consumers to calculate the intended scheduled time of a replacement trip

1.23.1.2 Vehicle Descriptor

Field Name	***Type***	***Required***	***Cardinality***	***Description***	***TfNSW Rules***
id	string	Optional	One	Internal system identification of the vehicle. Should be unique per vehicle, and is used for tracking the vehicle as it proceeds through the system. This id should not be made visible to the enduser; for that purpose use the label field	Required by TfNSW This should be traceable to the incoming SIRI Vehicle Monitoring message.
label	string	Optional	One	User visible label, i.e., something that must be shown to the passenger to help identify the correct vehicle.	Examples include using the vehicle name or service name.
license_plate	string	Optional	One	The license plate of the vehicle.	

1.23.1.3 Position

Field Name	*Type*	*Required*	*Cardinality*	*Description*	*TfNSW Rules*
latitude	float	Required	One	Degrees North, in the WGS-84 coordinate system.	Required by TfNSW Formatted to at least 6 decimal places
longitude	float	Required	One	Degrees East, in the WGS-84 coordinate system.	Required by TfNSW Formatted to at least 6 decimal places
bearing	float	Optional	One	Bearing, in degrees, clockwise from True North, i.e., 0 is North and 90 is East. This can be the compass bearing, or the direction towards the next stop or intermediate location. This should not be deduced from the sequence of previous positions, which clients can compute from previous data.	Required by TfNSW Rounded to 2 decimal places
odometer	double	Optional	One	Odometer value, in meters.	
speed	float	Optional	One	Momentary speed measured by the vehicle, in meters per second.	Required by TfNSW Rounded to 2 decimal places

1.23.1.4 Vehicle Stop Status

Value	***Comment***	***TfNSW Rules***
INCOMING_AT	The vehicle is just about to arrive at the stop (on a stop display, the vehicle symbol typically flashes).	No additional requirements
STOPPED_AT	The vehicle is standing at the stop.	No additional requirements
IN_TRANSIT_TO	The vehicle has departed the previous stop and is in transit.	No additional requirements

1.23.1.5 Congestion Level

Following is the notation for the mathematical formulas below:

$\ddot{x}$ = mean average σLong = Long term

standard deviation $\ddot{x}\text{Long}$ = mean Long

term average $\ddot{x}\text{short}$ = mean Short term

average

Value	***TfNSW Rules***
UNKNOWN_CONGESTION_LEVEL	Unknown is when the short term average is not known. Ie $\ddot{x}\text{short}$ has no value
RUNNING_SMOOTHLY	$\ddot{x}\text{Long} + \sigma\text{Long} > \ddot{x}\text{short}$
STOP_AND_GO	$\ddot{x}\text{Long} + \sigma\text{Long} <= \ddot{x}\text{short} < \ddot{x}\text{Long} + 2\sigma\text{Long}$
CONGESTION	$\ddot{x}\text{Long} + 2\sigma\text{Long} <= \ddot{x}\text{short} < \ddot{x}\text{Long} + 3\sigma\text{Long}$
SEVERE_CONGESTION	$\ddot{x}\text{Long} + 3\sigma\text{Long} <= \ddot{x}\text{short}$

1.24 Occupancy Status

Value	*Comment*	*TfNSW Rules*
EMPTY	The vehicle is considered empty by most measures, and has few or no passengers onboard, but is still accepting passengers.	Not Populated
MANY_SEATS_AVAILABLE	The vehicle has a large percentage of seats available. What percentage of free seats out of the total seats available is to be considered large enough to fall into this category is determined at the discretion of the producer.	Where the count of the number of passengers received from the vehicle is 50% or less than the seating capacity of the vehicle
FEW_SEATS_AVAILABLE	The vehicle has a small percentage of seats available. What percentage of free seats out of the total seats available is to be considered small enough to fall into this category is determined at the discretion of the producer.	Where the count of the number of passengers received from the vehicle is equal to or less than the seating capacity of the vehicle
STANDING_ROOM_ONLY	The vehicle can currently accommodate only standing passengers.	Where the count of the number of passengers received from the vehicle is greater than the seating capacity of the vehicle
CRUSHED_STANDING_ROOM_ONLY	The vehicle can currently accommodate only standing passengers and has limited space for them.	Not Populated
FULL	The vehicle is considered full by most measures but may still be allowing passengers to board.	Not Populated
NOT_ACCEPTING_PASSENGERS	The vehicle cannot accept passengers.	Not Populated

Note: If the vehicle cannot provide the capability to provide Occupancy in Real-time, the Occupancy status must be omitted. Omission of "Occupancy Status" is taken to mean Occupancy_Status = "UNKNOWN".

1.25 Carriage Descriptor

Field Name	*Type*	*Required*	*Cardinality*	*Description*	*TfNSW Rules*
name	string	optional	one	This should refer to a value customers can see	No additional rules
position_in_consist	Int	required	one	Carriage position in the consist. The position of carriages is relative to the current leading carriage and commences with 1 for the leading carriage.	No additional rules
occupancy_status	OccupancyStatus	required	one	OccupancyStatus of the vehicle or individual carriage. For vehicles with more than one passenger carriage, the carriage level provides a more granular view. The field in VehiclePosition should be a rollup of the individual carriage occupancy metrics.	If passenger count is not available, then omit the occupany_status field.
quiet_carriage	Boolean	optional	one	Is this a quiet carriage - used to indicate to passengers they should keep noise to a minimum	No additional rules

toilet	ToiletStatus	optional	One	Does the carriage have a toilet and if so, is it an accessible toilet?
				NONE = 0; // No toilet in this carriage
				NORMAL = 1; // regular toilet for able bodied people
				ACCESSIBLE = 2;
					No additional rules

luggage_rack	Boolean	Optional	One	Does the carriage have luggage racks	

1.26 Transit_realtime.tfnsw_vehicle_descriptor

Field Name	***Type***	***Required***	***Cardinality***	***Description***	***TfNSW Rules***
air_conditioned	True/false	Optional	One		Air-conditioning availability – sent as a 'True/False' value
wheelchair_accessible	Boolean	Optional	One		Wheelchair access availability – sent as a bit value – 0 being not available and 1 being available
vehicle_model	string	Optional	One		For Bus, this field will be a combination of the following information - vehicleManufacturerName, chassisName, vehicleBodyManufacturerName and bodyName; separated by tilde (~). E.g. Mercedes~O405NH~Custom Coaches~CUSTOMCITARO For other mode it will be a single description of the model e.g., for Sydney Metro it is showing as "Alstom Metropolis"
performing_prior_trip	True/False	Optional	One		When the vehicle has not yet commenced its current trip and is performing the prior trip this will be populated as "true"

special_vehicle_attributes	Boolean	Optional	One		Following information is sent as a bitmask in the ‘special_vehicle_attributes’: Wi-Fi availability – bitmask value 0001 Christmas bus or not – bitmask value 0010 Both Wi-Fi and Christmas bus –bitmask value 0011

1.27 Alert Update Message

Message Structure	Message Example

{
"header": {
"gtfs_realtime_version": ,
"incrementality": ,
"incrementalitySpecified": ,
"timestamp": ,
"timestampSpecified":
},
"entity": [
{
"id": ,
"is_deletedSpecified": ,
"trip_update": ,
"vehicle": ,
"alert": {
"active_period": [
{
"start": ,
"startSpecified":
"end": ,
"endSpecified":
}
],
"informed_entity": [
{
"agency_idSpecified": ,
"route_id": ,
"route_idSpecified": ,
"route_type": ,
"route_typeSpecified": ,
"trip": {
"trip_id": ,
"trip_idSpecified": ,
"route_idSpecified": ,
"direction_idSpecified": ,
"start_timeSpecified": ,
"start_date": ,
"start_dateSpecified": ,
"schedule_relationshipSpecified":
},
"stop_idSpecified":
},
{
"agency_idSpecified": ,
"route_id": ,
"route_idSpecified": ,
"route_type": ,
"route_typeSpecified": ,
"trip": {
	entity {
	id: "eaf3730a-2e72-5804-aa4e-e99ce2b6f7ae"
	alert {
	active_period {
	start: 1691164800
	end: 1691337600
	}
	informed_entity {
	agency_id: "SMNW"
	route_id: "SMNW_M"
	direction_id: 1
	}
	informed_entity {
	agency_id: "SMNW"
	route_id: "SMNW_M"
	direction_id: 0
	}
	cause: MAINTENANCE
	effect: MODIFIED_SERVICE
	url {
	translation {
	text: "https://transportnsw.info/alerts/details#/ems-19585"
	language: "en"
	}
	}
	header_text {
	translation {
	text: "Buses replace metro between Tallawong and Chatswood"
	language: "en"
	}
	}
	description_text {
	translation {
	text: "Saturday 5 and Sunday 6 August\nBuses replace metro between Tallawong and Chatswood."
	language: "en"
	}
	
	}
	}

```
     "trip_id": ,
     "trip_idSpecified": ,
     "route_idSpecified": ,
     "direction_idSpecified": ,
     "start_timeSpecified": ,
     "start_date": ,
     "start_dateSpecified": ,
     "schedule_relationshipSpecified":
    },
    "stop_idSpecified":
   },
   {
    "agency_idSpecified": ,
    "route_id": ,
    "route_idSpecified": ,
    "route_type": ,
    "route_typeSpecified": ,
    "trip": ,
    "stop_idSpecified":
   },
   {
    "agency_idSpecified": ,
    "route_idSpecified": ,
    "route_typeSpecified": ,
    "trip": ,
    "stop_id":,
    "stop_idSpecified":
   },
   {
    "agency_id": ,
    "agency_idSpecified": ,
    "route_idSpecified": ,
    "route_typeSpecified": ,
    "trip": ,
    "stop_idSpecified":
   }
  ],
  "cause": ,
  "causeSpecified": ,
  "effect": ,
  "effectSpecified": ,
  "url": ,
  "header_text": {
   "translation": [
    {
     "text": ,
     "language": ,
     "languageSpecified":
    }
   ]
  },
  "description_text": {
   "translation": [
      {
       "text": ,
       "language": ,
       "languageSpecified":
      }
     ]
    }
   }
  },
 ]
}
```

1.29 Alert

Field Name	***Type***	***Required***	***Cardinality***	***Description***	***TfNSW Rules***
active_period	TimeRange	Required	Many	Time when the alert should be shown to the user. If missing, the alert will be shown as long as it appears in the feed. If multiple ranges are given, the alert will be shown during all of them.	Required by TfNSW
informed_entity	EntitySelector	Required	Many	Entities whose users we should notify of this alert. At least one informed_entity must be provided.	
cause	Cause	Optional	One		
effect	Effect	Required	One		
url	TranslatedString	Optional	One	The URL which provides additional information about the alert.	
header_text	TranslatedString	Required	One	Header for the alert. This plain-text string will be highlighted, for example in boldface.	Short description of situation no greater than (*to be confirmed*) characters
description_text	TranslatedString	Required	One	Description for the alert. This plain-text string will be formatted as the body of the alert (or shown on an explicit "expand" request by the user). The information in the description should add to the information of the header.	Description to provide call to action and details on the reason for the alert.

1.29.1 Time Range

Field Name	*Type*	*Required*	*Cardinality*	*Description*	*TfNSW Rules*
start	uint64	Conditionally required	One	Start time, in POSIX time (i.e., number of seconds since January 1st 1970 00:00:00 UTC). If missing, the interval starts at minus infinity. If a TimeRange is provided, either start or end must be provided - both fields cannot be empty.	
end	uint64	Conditionally required	One	End time, in POSIX time (i.e., number of seconds since January 1st 1970 00:00:00 UTC). If missing, the interval ends at plus infinity. If a TimeRange is provided, either start or end must be provided - both fields cannot be empty.	

1.29.2 Entity Selector

Field Name	*Type*	*Required*	*Cardinality*	*Description*	*TfNSW Rules*
agency_id	string	Conditionally required	One	At least one specifier must be given - all fields in an EntitySelector cannot be empty.	Required by TfNSW
route_id	string	Conditionally required	One	At least one specifier must be given - all fields in an EntitySelector cannot be empty.	Provide on the route, line level.

Field Name	*Type*	*Required*	*Cardinality*	*Description*	*TfNSW Rules*
route_type	int32	Conditionally required	One	At least one specifier must be given - all fields in an EntitySelector cannot be empty.	
trip	TripDescriptor	Conditionally required	One	At least one specifier must be given - all fields in an EntitySelector cannot be empty.	If there is only one trip affected
stop_id	string	Conditionally required	One	At least one specifier must be given - all fields in an EntitySelector cannot be empty.	Station, Wharf, Stop

1.29.3 Trip Descriptor

Field Name	*Type*	*Required*	*Cardinality*	*Description*	*TfNSW Rules*
trip_id	string	Conditionally required	One	The trip_id from the GTFS feed that this selector refers to. For non frequency-based trips (trips not defined in GTFS frequencies.txt), this field is enough to uniquely identify the trip. For frequencybased trips defined in GTFS frequencies.txt, trip_id, start_time, and start_date are all required. For scheduled-based trips (trips not defined in GTFS frequencies.txt), trip_id can only be omitted if the trip can be uniquely identified by a combination of route_id, direction_id, start_time, and start_date, and all those fields are provided.	

route_id	string	Conditionally required	One	The route_id from the GTFS that this selector refers to. If trip_id is omitted, route_id must be provided.	
direction_id	uint32	Conditionally required	One	The direction_id from the GTFS feed trips.txt file, indicating the direction of travel for trips this selector refers to. If trip_id is omitted, direction_id must be provided. Caution: this field is still experimental, and subject to change. It may be formally adopted in the future.	

Start_time	string	Conditionally required	One	The initially scheduled start time of this trip instance. When the trip_id corresponds to a non-frequency-based trip, this field should either be omitted or be equal to the value in the GTFS feed. When the trip_id correponds to a frequency-based trip defined in GTFS frequencies.txt, start_time is required and must be specified for trip updates and vehicle positions. If the trip corresponds to exact_times=1 GTFS record, then start_time must be some multiple (including zero) of headway_secs later than frequencies.txt start_time for the corresponding time period. If the trip corresponds to exact_times=0, then its start_time may be arbitrary, and is initially expected to be the first departure of the trip. Once established, the start_time of this frequency-based exact_times=0 trip should be considered immutable, even if the first departure time changes -- that time change may instead be reflected in a StopTimeUpdate. If trip_id is omitted, start_time must be provided. Format and semantics of the field is same as that of GTFS/frequencies.txt/start_time, e.g., 11:15:35 or 25:15:35.	

start_date	string	Conditionally required	One	The start date of this trip instance in YYYYMMDD format. For scheduled trips (trips not defined in GTFS frequencies.txt), this field must be provided to disambiguate trips that are so late as to collide with a scheduled trip on a next day. For example, for a train that departs 8:00 and 20:00 every day, and is 12 hours late, there would be two distinct trips on the same time. This field can be provided but is not mandatory for schedules in which such collisions are impossible - for example, a service running on hourly schedule where a vehicle that is one hour late is not considered to be related to schedule anymore. This field is required for frequency-based trips defined in GTFS frequencies.txt. If trip_id is omitted, start_date must be provided.	
schedule_relationship	ScheduleRelationship	Optional	One		

1.29.3.1 Schedule Relationship

Value	*Comment*	*TfNSW Rules*
SCHEDULED	Trip that is running in accordance with its GTFS schedule, or is close enough to the scheduled trip to be associated with it.	No additional rules
ADDED	An extra trip that was added in addition to a running schedule, for example, to replace a broken vehicle or to respond to sudden passenger load.	No additional rules

UNSCHEDULED	A trip that is running with no schedule associated to it - this value is used to identify trips defined in GTFS frequencies.txt with exact_times = 0. It should not be used to describe trips not defined in GTFS frequencies.txt, or trips in GTFS frequencies.txt with exact_times = 1.	No additional rules
CANCELED	A trip that existed in the schedule but was removed.	No additional rules
REPLACEMENT	The trip identified in the GTFS timetable has been modified.	No additional rules
SKIPPED	The stop is skipped, i.e., the vehicle will not stop at this stop. Arrival and departure are optional.	No additional rules

1.29.4 Cause

Value	*TfNSW Rules*
UNKNOWN_CAUSE	No additional rules
OTHER_CAUSE	No additional rules
TECHNICAL_PROBLEM	No additional rules
STRIKE	No additional rules
DEMONSTRATION	No additional rules
ACCIDENT	No additional rules

HOLIDAY	No additional rules
WEATHER	No additional rules
MAINTENANCE	No additional rules
CONSTRUCTION	No additional rules
POLICE_ACTIVITY	No additional rules
MEDICAL_EMERGENCY	No additional rules

1.29.5 Effect

Value	*TfNSW Rules*
NO_SERVICE	No additional rules
REDUCED_SERVICE	No additional rules
SIGNIFICANT_DELAYS	No additional rules
DETOUR	No additional rules
ADDITIONAL_SERVICE	No additional rules
MODIFIED_SERVICE	No additional rules
OTHER_EFFECT	No additional rules
UNKNOWN_EFFECT	No additional rules
STOP_MOVED	No additional rules

1.29.6 Translated String

Field Name	*Type*	*Required*	*Cardinality*	*Description*	*TfNSW Rules*
translation	Translation	Required	Many	At least one translation must be provided.	

1.29.7 Translation

Field Name	*Type*	*Required*	*Cardinality*	*Description*	*TfNSW Rules*
text	string	Required	One	A UTF-8 string containing the message.	
language	string	Conditionally required	One	BCP-47 language code. Can be omitted if the language is unknown or if no internationalization is done at all for the feed. At most one translation is allowed to have an unspecified language tag - if there is more than one translation, the language must be provided.	

1.30 FAQs for the V2 - Toustone Feed:

Field Name	*Third Party Endorsed Apps Feedback*	*Response from Operator and TfNSW*
occupancy_status	Could you please explain if real-time occupancy is actually real-time or estimated?	It is calculated based on real time.
Position_in_consist	position_in_consit: 0: Is anyone clear on current ordering of the consist items? Observations have been made that the first consist item in the array is always the Chatswood end, regardless of whether the train is headed to Tallawong or Chatswood. So, the ordering of the carriages is fixed and not based on the direction of travel. Would this be a fair assumption?	Yes, the consists are fixed in direction and they cannot be turned around in the network. N.B. The only time these will ever change is if there was a catastrophic/major incident which damaged 2 sets of train carriages, and then the carriages may need to be swapped to make 1 set usable again. Tentative Timeframe to get this defect fixed is: end of Q1 2024
Block_ID	I would be interested in some guidance on a problem i have long had with the metro GTFSR: I'm at chatswood platform 3, I'm waiting for the next metro, i can see a time prediction for it but not the vehicle position. how do we figure out which vehicle to show for a metro is heading into chatswood the other direction (or waiting in the siding) that hasn't formed yet?	This information is currently not provided, and Metro does not provide block messages.

“performing_prior_trip”	if you link up the run numbers (e.g., 118-13 forms 118-14, 118-14 forms 118-15) together it does not provide reliable results. could they dynamically provide formation link IDs? or duplicate the vehicle and use performing_prior_trip like PTIPS do?	This feed does not provide details for performing_prior_trip for the vehicle.