

## 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
<pre>{   "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": ,             }           }         ]       }     }   ] }</pre>	<pre>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       }     }   } }</pre>

<pre>         },         "schedule_relationship": ,         "schedule_relationshipSpecified": ,       ],       "timestamp": ,       "timestampSpecified": ,       "delaySpecified": ,     },     "vehicle": ,     "alert": ,   }, } ] } </pre>	<pre>         }         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"       }     }   } } </pre>
--	--

```
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

<b>Field Name</b>	<b>Type</b>	<b>Required</b>	<b>Cardinality</b>	<b>Description</b>	<b>TfNSW Rules</b>
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 <i>not</i> mean that the trip is progressing according to schedule.	<p>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.</p> <p>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)</p> <p>All unscheduled trips that are currently in progress will contain a valid trip_id and will be listed.</p>
vehicle	VehicleDescriptor	Optional	One	Additional information on the vehicle that is serving this trip.	<i>TfNSW to revisit at a later date</i>
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.

				<p>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.</p>	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	<p>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.</p> <p>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.</p> <p>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.</p> <p>Caution: this field is still experimental, and subject to change. It may be formally adopted in the future.</p>	<p>N/A - this field is not populated</p> <p>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.</p>

### 1.22.1.1 Trip Descriptor

<i>Field Name</i>	<i>Type</i>	<i>Required</i>	<i>Cardinality</i>	<i>Description</i>	<i>TfNSW Rules</i>
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.	<p>Required by TfNSW</p> <p>trip_id must be unique within a GTFS-R feed</p> <p>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.</p> <p>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.</p> <p>This field is to be mapped to the Vehicle Journey ID in the input feed</p>
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.	<p>Required by TfNSW</p> <p>In GTFS-R used element route_id must be consistent with values coming from GTFS.</p> <p>For inconsistent route_id's, the trip is mapped back to the GTFS using the stopping sequence in order to derive a route id.</p>

direction_id	uint32	Conditionally required	One	<p>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.</p> <p>Caution: this field is still experimental, and subject to change. It may be formally adopted in the future.</p>	<p>For trip_update:: SCHEDULED where direction_id is not provided, then direction is taken from GTFS on basis of consistent trip_id's.</p> <p>For trip_update::ADDED, direction is required</p>
--------------	--------	------------------------	-----	---	---

start_time	string	Conditionally required	One	<p>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 corresponds 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.</p>	<p>Required by TfNSW</p> <p>This is to be equivalent of the trip start time associated with the trip_id in the schedule.</p>
------------	--------	------------------------	-----	---	--

start_date	string	Conditionally required	One	<p>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.</p>	<p>Required by TfNSW</p> <p>Where we have repeated trip_id on subsequent days (everyday trips) a start_date is required</p> <p>This is to be equivalent of the trip start date associated with the trip ID in the schedule.</p> <p>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.</p>
schedule_relationship	ScheduleRelationship	Optional	One		Required by TfNSW

### 1.22.1.1 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.	<p>GTFS-R trip_id must exist and is unique in corresponding GTFS data</p> <p>When no other trip with same trip_id is in Progress when the trip entity is created then the schedule relationship for the trip will be SCHEDULED</p>

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.	<p>GTFS-R trip_id must not exist in corresponding GTFS data and must be unique across all GTFS/GTFS-R trip_ids</p> <p>The trip must be provided with full stop sequence. Trip is consistent and persistent across messages.</p> <p>trip_id of CANCELED trip must not be reused for a new ADDED trip</p> <p>The route_id of an added trip must refer to a route_id in GTFS.</p> <p>direction_id is required for ADDED trips</p> <p>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'</p>
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.	<p>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.</p> <p>GTFS-R trip_id must not exist in corresponding GTFS data and must be unique across all GTFS/GTFS-R trip_ids</p> <p>UNSCHEDULED trips do not have a trip_id but they have route_id.</p> <p>Real-time information is provided only for future stops.</p> <p>There must never more than one UNSCHEDULED trip per route at the same time.</p>
		<p>UNSCHEUDLED trips are provided with arrival_delay and departure_delay are set to 0.</p> <p>Arrival time and departure time should be populated for all UNSCHEDULED service stops</p>
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	<p>REPLACEMENT was compliant with an older GTFS-R standard, but is not compliant with the latest version of GTFS-R.</p> <p>route_id and trip_id in trip_update::REPLACEMENT references to GTFS data</p> <p>Where Stop sequence GTFS-R = GTFS REPLACEMENT trips are handled as trip_update::SCHEDULED</p> <p>Stop sequence GTFS-R != GTFS REPLACEMENT trips are handled as if the existing scheduled trip is rerouted.</p> <p>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.</p> <p>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</p>
-------------	---	--

### 1.22.1.2 Vehicle Descriptor

<i>Field Name</i>	<i>Type</i>	<i>Required</i>	<i>Cardinality</i>	<i>Description</i>	<i>TfNSW Rules</i>
id	<a href="#">string</a>	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	<a href="#">string</a>	Optional	One	User visible label, i.e., something that must be shown to the passenger to help identify the correct vehicle.	NA
license_plate	<a href="#">string</a>	Optional	One	The license plate of the vehicle.	NA

### 1.22.1.3 StopTimeUpdate

<i>Field Name</i>	<i>Type</i>	<i>Required</i>	<i>Cardinality</i>	<i>Description</i>	<i>TfNSW Rules</i>
stop_sequence	<a href="#">uint32</a>	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	<a href="#">string</a>	Conditionally required	One	Must be the same as in stops.txt in the corresponding	Required by TfNSW

		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	<a href="#">StopTimeEvent</a>	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	<a href="#">StopTimeEvent</a>	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	<a href="#">ScheduleRelationship</a>	Optional	One	The default relationship is SCHEDULED.	

### 1.22.1.3.1 StopTimeEvent

<i>Field Name</i>	<i>Type</i>	<i>Required</i>	<i>Cardinality</i>	<i>Description</i>	<i>TfNSW Rules</i>
delay	int32	Conditionally required	One	<p>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.</p> <p>Either delay or time must be provided within a StopTimeEvent - both fields cannot be empty.</p>	<p>stop_time_update::delay for all trips that have trip_update:: = SCHEDULED is mandatory, for stops in the sequence.</p> <p>When the arrival field is populated in the 'StopTimeUpdate' entity then one of the following conditions will apply:</p> <ul style="list-style-type: none"> <li>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.</li> <li>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.</li> <li>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.</li> <li>For a particular 'TSN', if a prediction is not available then populate this field with 0.</li> </ul> <p>When the departure field is populated in the StopTimeUpdate' entity then one of the following conditions will apply:</p> <ul style="list-style-type: none"> <li>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.</li> <li>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.</li> <li>For a particular 'TSN', compare the predicted departure time for that 'TSN' with the scheduled departure time and if there no</li> </ul>

					<p>difference then populate this field with 0.</p> <p>d. For a particular 'TSN', if a prediction is not available then populate this field with 0.</p>
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.	<p>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.</p> <p>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.</p>
uncertainty	int32	Optional	One	If uncertainty is omitted, it is interpreted as unknown. To specify a completely certain prediction, set its uncertainty to 0.	<p>Uncertainty applies to both the time and the delay value of a StopTimeUpdate.</p> <p>The uncertainty specifies the expected error in true delay as an integer in seconds.</p> <p>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.</p>

### 1.22.1.3.2 Schedule Relationship

Value	Comment	TfNSW Rules
SCHEDULED	<p>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.</p>	<p>Stop must have same stop_sequence in GTFS and GTFS-R</p> <p>For each trip listed as per the schedule, if a vehicle is associated then this option will be selected for that trip</p> <p>For each 'unscheduled' trip listed this option will be selected for that trip</p>
SKIPPED	<p>The stop is skipped, i.e., the vehicle will not stop at this stop. Arrival and departure are optional.</p>	<p>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</p> <p>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'</p>
NO_DATA	<p>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.</p>	<p>stop must have same stop_sequence in GTFS and GTFS-R</p> <p>NO_DATA results in real-time not being available for the full trip</p>

## 1.23 Vehicle Position Updates Message

Message Structure	Message Example
<pre>{   "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_plate": ,           "tfnsw_vehicle_descriptor": {             "air_conditioned": ,             "air_conditionedSpecified": ,             "wheelchair_accessible": ,             "wheelchair_accessibleSpecified": ,             "vehicle_model": ,             "vehicle_modelSpecified": ,             "performing_prior_trip": ,             "special_vehicle_attributes": ,             "special_vehicle_attributesSpecified": ,           }         },         "position": {           "latitude": ,           "longitude": ,           "bearing": ,           "speed": ,         }       }     }   ] }</pre>	<pre>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     }   } }</pre>

<pre>         "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": ,   } ] } </pre>	<pre>       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     }   } } </pre>
---	---

### 1.23.1 Vehicle Position Updates Message

<b>Field Name</b>	<b>Type</b>	<b>Required</b>	<b>Cardinality</b>	<b>Description</b>	<b>TfNSW Rules</b>
Id	string	required	One		The id should be traceable back to the source providing system
trip	<a href="#">TripDescriptor</a>	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.	<p>Required by TfNSW</p> <p>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.</p> <p>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</p> <p>All unscheduled trips that are currently in progress will contain a valid trip_id and will be listed.</p>
vehicle	<a href="#">VehicleDescriptor</a> or	Optional	One	Additional information on the vehicle that is serving this trip. Each entry should have a unique vehicle id.	Required by TfNSW
position	<a href="#">Position</a>	Optional	One	Current position of this vehicle.	Required by TfNSW

current_stop_sequence	<a href="#">uint32</a>	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	<a href="#">string</a>	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	<a href="#">VehicleStopStatus</a>	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	<a href="#">uint64</a>	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	<a href="#">CongestionLevel</a>	Optional	One		Required by TfNSW

occupancy_status	<a href="#"><u>OccupancyStatus</u></a>	Required	One	The degree of passenger occupancy of the vehicle.	Required by TfNSW
Transit_realtim e.tfnsw_vehicle _descriptor	User extension	<i>Optional</i>		Include following information about the bus in service: Air-conditioning is available or not and Wheelchair access is available or not	<p>Required by TfNSW</p> <p>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</p> <p>TfNSW has been assigned extension 1007 for these attributes.</p>
CarriageDescriptor	User extension	<i>Optional</i>	Many	To pass on information at a carriage level such as occupancy levels and facilitates available.	<p>Required by TfNSW</p> <p>TfNSW has been assigned extension 1007 for these attributes.</p>

### 1.23.1.1 Trip Descriptor

<b>Field Name</b>	<b>Type</b>	<b>Required</b>	<b>Cardinality</b>	<b>Description</b>	<b>TfNSW Rules</b>
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	<p>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.</p> <p>Caution: this field is still experimental, and subject to change. It may be formally adopted in the future.</p>	<p>For trip_update:: SCHEDULED where direction_id is not provided, then direction is taken from GTFS on basis of consistent trip_id's.</p> <p>For trip_update::ADDED, direction is required</p>
start_time	string	Conditionally required	One	<p>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 corresponds to a frequency-based trip defined in GTFS</p>	<p>Required by TfNSW</p> <p>This is to be equivalent of the trip start time associated with the trip ID in the schedule.</p>

				<p>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.</p>	
start_date	string	Conditionally required	One	<p>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</p>	<p>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.</p>

				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	<a href="#">ScheduleRelationship</a>	Optional	One		Required by TfNSW

### 1.23.1.1.1 Schedule Relationship

<i>Value</i>	<i>Comment</i>	<i>TfNSW Rules</i>
SCHEDULED	Trip that is running in accordance with its GTFS schedule, or is close enough to the scheduled trip to be associated with it.	<p>GTFS-R trip_id must exist and is unique in corresponding GTFS data</p> <p>When no other trip with same trip_id is in Progress when the trip entity is created then the schedule relationship for the trip will be SCHEDULED</p>
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.	<p>GTFS-R trip_id must not exist in corresponding GTFS data and must be unique across all GTFS/GTFS-R trip_ids</p> <p>The trip must be provided with full stop sequence. Trip is consistent and persistent across messages.</p> <p>trip_id of CANCELED trip must not be reused for a new ADDED trip</p>
		<p>The route_id of an added trip must refer to a route_id in GTFS.</p> <p>direction_id is required for ADDED trips</p> <p>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'</p>

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.	<p>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.</p> <p>GTFS-R trip_id must not exist in corresponding GTFS data and must be unique across all GTFS/GTFS-R trip_ids</p> <p>UNSCHEDULED trips do not have a trip_id but they have route_id.</p> <p>Real-time information is provided only for future stops.</p> <p>There must never more than one UNSCHEDULED trip per route at the same time.</p> <p>UNSCHEUDLED trips are provided with arrival_delay and departure_delay are set to 0.</p> <p>Arrival time and departure time should be populated for all UNSCHEDULED service stops</p>
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	<p>REPLACEMENT was compliant with an older GTFS-R standard, but is not compliant with the latest version of GTFS-R.</p> <p>route_id and trip_id in trip_update::REPLACEMENT references to GTFS data</p> <p>Where Stop sequence GTFS-R = GTFS</p> <p>REPLACEMENT trips are handled as trip_update::SCHEDULED</p> <p>Stop sequence GTFS-R != GTFS</p> <p>REPLACEMENT trips are handled as if the existing scheduled trip is rerouted.</p>

		<p>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.</p> <p>Passing Stops stops must have a schedule_relationship of SKIPPED</p> <p>The message must continually send the full stop list and stop sequence for all updates until the trip is completed.</p> <p>Both the stop_time and stop_date are required.</p> <p>Both arrival_time and departure_time are required for each stop.</p> <p>Delay for each stop is also required in order for consumers to calculate the intended scheduled time of a replacement trip</p>
--	--	--

### 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

<i>Field Name</i>	<i>Type</i>	<i>Required</i>	<i>Cardinality</i>	<i>Description</i>	<i>TfNSW Rules</i>
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:

$\bar{x}$  = mean average  $\sigma_{Long}$  = Long term

standard deviation  $\bar{x}_{Long}$  = mean Long

term average  $\bar{x}_{short}$  = mean Short term

average

Value	TfNSW Rules
UNKNOWN_CONGESTION_LEVEL	Unknown is when the short term average is not known. ie $\bar{x}_{short}$ has no value
RUNNING_SMOOTHLY	$\bar{x}_{Long} + \sigma_{Long} > \bar{x}_{short}$
STOP_AND_GO	$\bar{x}_{Long} + \sigma_{Long} \leq \bar{x}_{short} < \bar{x}_{Long} + 2\sigma_{Long}$
CONGESTION	$\bar{x}_{Long} + 2\sigma_{Long} \leq \bar{x}_{short} < \bar{x}_{Long} + 3\sigma_{Long}$
SEVERE_CONGESTION	$\bar{x}_{Long} + 3\sigma_{Long} \leq \bar{x}_{short}$

## 1.24 Occupancy Status

<b>Value</b>	<b>Comment</b>	<b>TfNSW Rules</b>
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

: 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

<i>Field Name</i>	<i>Type</i>	<i>Required</i>	<i>Cardinality</i>	<i>Description</i>	<i>TfNSW Rules</i>
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 occupancy_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		<p>For Bus, this field will be a combination of the following information - vehicleManufacturerName, chassisName, vehicleBodyManufacturerName and bodyName; separated by tilde (~).</p> <p>E.g. Mercedes~O405NH~Custom Coaches~CUSTOMCITARO</p> <p>For other mode it will be a single description of the model e.g., for Sydney Metro it is showing as “Alstom Metropolis”</p>
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
----------------------------	---------	----------	-----	--	--