Sydney Trains Realtime
GTFS & GTFS- R Technical Document

Contents

Document Control

Revision History

Version	Author	Issue Date	Changes
1	Sydney Trains	12/2/2013	Initial version
2	Sydney Trains	06/09/2014	Updated for TripUpdates GTFS Static Bundle Changes
2.1	TfNSW	18/09/2014	Formatted into standard report template. Consolidated access and tech document into one primary document
2.2	Sydney Trains	11/12/2014	Updated GTFS reference and added headsign information.
2.3	Sydney Trains	04/03/2016	Included charter services (section 2.2.1 and 2.2.4)
2.4	TfNSW	16/03/2016	Revised for Open Data
2.5	Sydney Trains	10/07/2018	Updated the Train Set type to support changes for Waratah Series 2 services and changes to Charter Trip ID range.
2.6	TfNSW	23/07/2018	Formatting and Waratah 2 shorthand added
2.7	Sydney Trains	05/09/2019	Updated Charter Trip_ID range
2.8	Sydney Trains	14/01/2019	Updated set types for NIF
2.9	Sydney Trains	10/01/2020	Updated Section 2.2.2. Train Set Types
3.0	TfNSW	27/07/2020	Updated Section 3.1 Coverage
3.1	TfNSW	23/10/2020	Additional Charter trip_IDs added, Indian Pacific set type removed
3.2	TfNSW	10/12/2020	Added Vehicles Extension and the corresponding Appendix B
3.3	TfNSW	7/10/2021	Update to include GTFS-R Protobuf Version 2

3.4	TfNSW	12/10/2021	Added occupancies.txt extension for Short-Term Train-load prediction feature.
3.5	TfNSW	20/08/2022	Added Train and Carriage load Prediction feature Updated GTFS_R Protobuf

References

Document Name	Network Location or Documentation Link
GTFS Reference , October 15, 2012 revision.	https://developers.google.com/transit/gtfs/changes#october-15-2012
GTFS-realtime, v1.0 October 12,2013 revision.	https://developers.google.com/transit/gtfs-realtime/changes#oct-12-2012
GTFS-Vehicles extensions (full proposal)	http://bit.ly/GTFS-Vehicles
GTFS-Occupancies extension (full proposal)	bit.ly/gtfs-occupancies

1 Data Feed Access

Open Data API Gateway endpoints are as below and are accessible using a valid API key.

GTFS Bundle

https://api.transport.nsw.gov.au/v1/gtfs/schedule/sydneytrains

GTFS-R Vehicle Positions

https://api.transport.nsw.gov.au/v2/gtfs/vehiclepos/sydneytrains

GTFS-R Trip Updates

https://api.transport.nsw.gov.au/v2/gtfs/realtime/sydneytrains

GTFS-R Service Alerts

https://api.transport.nsw.gov.au/v2/gtfs/alerts/sydneytrains

2 GTFS Bundle Notes

2.1 agency.txt

Trains run by both Sydney Trains & NSW Trains will be contained within the bundle.

Times for NSW Trains running beyond the intercity network area (bordered by Goulburn, Bathurst, Scone, Dugong and Nowra) will not be accurate beyond these stations.

2.2 trips.txt

2.2.1 trip_id

The trip_id used to uniquely identify trips has a semantic content that could be used to provide additional information about the timetabled train. The format is as follows:

<trip_name>.<timetable_id>.<timetable_version_id>.<dop_ref>.<set_type>.<number_of_cars>.<trip_instance>

e.g. '159B.1697.101.32.A.8.68334035'

The bundle contains trips scheduled for operational and scheduling purposes. Trip_names reserved for Charter services should not be displayed to customers (e.g NH01). Hardcoding this rule is not recommended.

The following series for trip_name are reserved for Charter services:

Reserved Charter Run Numbers	Area	Description
880[A-Z] – 899[A-Z] e.g. 890A	Suburban	Reserved for Charter Services
HH01 - HH99	Intercity (Metro)	Trains operating between Metropolitan area locations - private Hire
NH01 - NH99	Intercity North	Additional trains between Sydney and Newcastle Interchange – Private Hire
WH01 - WH99	Intercity West	Additional trains between Sydney and Lithgow – private hire
CH01 - CH99	Intercity Illawarra	Additional trains between Sydney and Pt Kembla/Kiama- private hire

Trips with route RTTA_REV and RTTA_DEF should also not be displayed to customers. They are non-revenue services and trips that are not matched to a valid route.

<timetable_id>.<timetable_version_id>.<dop_ref> represent the calendar, so could provide indications the bundle is out of date if values in real time feeds do not match bundle calendar values.

The fields *<timetable_id>.<timetable_version_id>.<dop_ref> within the trip_id is not recommended for use. This is reserved to keep the trip id unique.*

2.2.2 Train Set Types

The following are the possible values that could be found in the *<set_type>* field of the trip_id for passenger trains.

Value	Train Set Type
A	Waratah
B	Waratah Series 2*
C	C Set
D	Mariyung (New Intercity Fleet)
H	Oscar
J	Hunter
K	K Set
M	Millennium
N	Endeavour
P	Xplorer
S	S Set
T	Tangara
V	V Set (Intercity)
X	XPT
Z	Heritage & Private Passenger Operator

* “Waratah 2” is the approved shorthand to display in apps with limited real estate. Note that the preference is to refer to the full name “Waratah Series 2” whenever possible.

Other codes that could be encountered are:

Value	Train Set Type
G	Freight
I	Track Inspection
L	Lt Locomotive
O	Other
Q	Maintenance Track Machine
U	Bus
W	Fast Freight
Y	Other

2.2.3 block_id

The trips recorded in the GTFS bundle timetable include both trips running ‘in service’, as well as those running ‘out of service’. A single trip could have both in service & out of service sections in its journey (Out of service sections must be at start and/or end of a trip).

Trips which are out of service for their entire journey can be identified as follows:

- ❖ All stops will have pickup ×, drop off ×

Trips which are out of service for only part of their journey, can have their out of service sections identified as follows:

- ❖ Start – All stops from start of trip with pickup ×, drop off ×, till a stop with pickup ✓ is encountered.
- ❖ End – All stops working forwards from end of trip with pickup ×, drop off ×, till a stop with drop off ✓ is encountered

The Trip’s block_id has been used to identify sequences of trips for which a passenger can remain on the train in continuous travel (both continuing in the same direction, or a turn-around service) as per the GTFS specification.

The following diagram seeks to show several of the scenarios around blocking, in service & out of service trips as well as the pickup drop off flags that may occur and what to expect. Shown is a single ‘roster’, representing a series of trips made by a single train in a day.

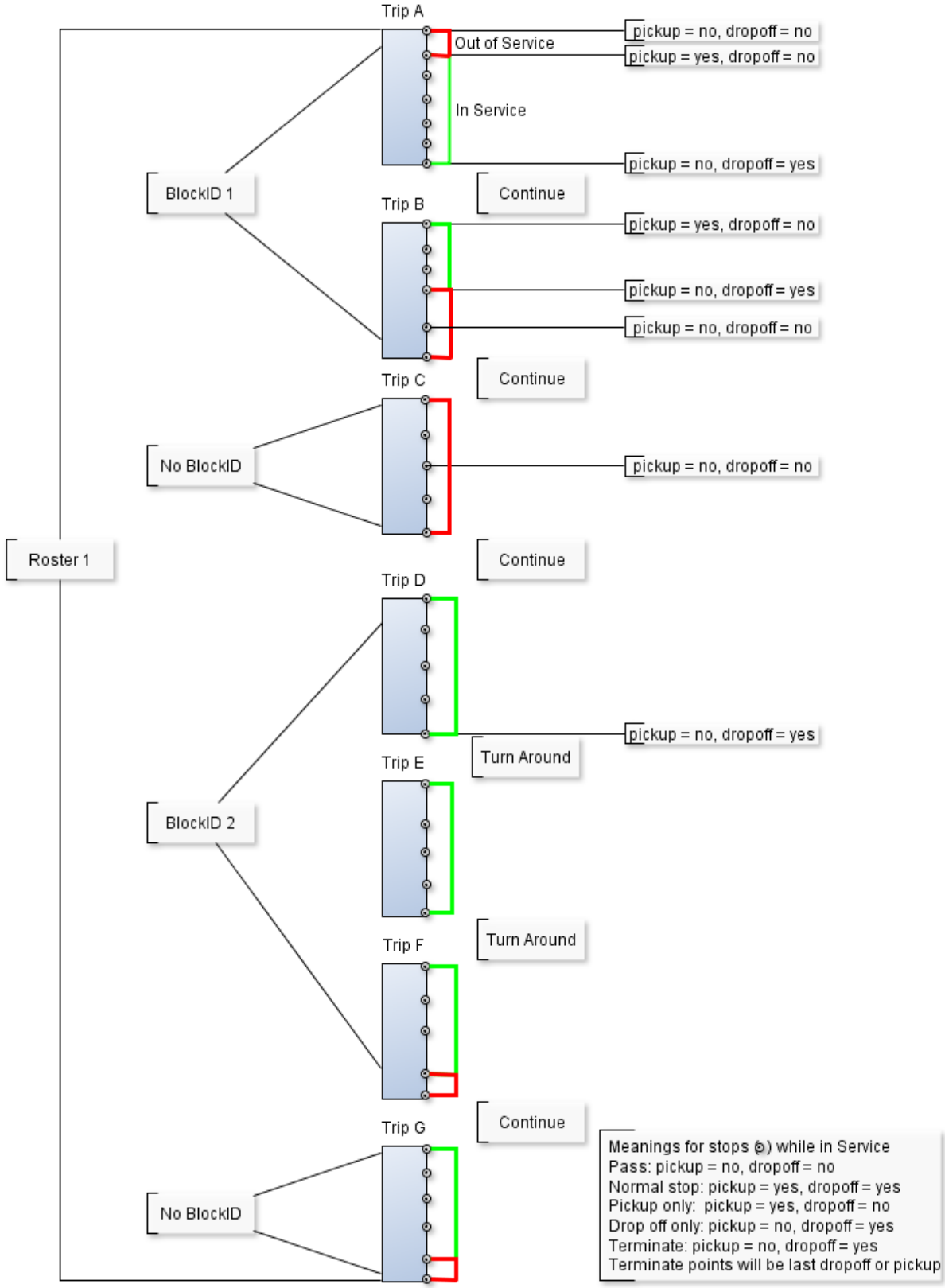

Figure 2-1 Blocking and Trip Example

2.2.4 trip_headsign

The trip_headsign is populated with destination for the passenger journey and is in the form *<Destination Station name>.*

The trip_headsign for a charter service is set to ‘Charter’.

If a via station information exists then this is provided in the trip_headsign as *<Destination Station name>* via *<Via Station name>*.

If the headsign changes during a trip, an override is specified in stop_headsign field in stop_times.txt.

2.3 vehicle-categories.txt

Whilst it has not been officially adopted as GTFS standard yet, Transport for NSW has implemented the **`GTFS-VehicleCategories`** extension to improve public transport information for NSW. At this time, it is used for both suburban and intercity trains only.

This adds the file **`vehicle-categories.txt`** to the GTFS bundle which provides information about the vehicle categories. It contains two data elements, and these are:

`vehicle_category_id`	(ID, Required) It defines an ID for a vehicle category. If used along with the **`GTFS-VehicleCouplings`** extension, this field can be either a parent vehicle defined in **`parent_id`** or a child vehicle defined in **`child_id`**.
`vehicle_category_name`	(Text, Optional) It defines the name of the vehicle category.

The **`GTFS-VehicleCategories`** describes the vehicles themselves. It also adds a new data field in the following core GTFS files:

- **`routes.txt`**
- **`trips.txt`**
- **`stop_times.txt`**

The new data field is the optional **`vehicle_category_id`** which defines a default vehicle category in **`routes.txt`** for all trips belonging to the route. This value is referenced from the **`vehicle_categories.txt`** file.

The optional **`vehicle_category_id`** defines a default vehicle category in **`trips.txt`** for the trip. It likewise defines a default vehicles category in **`stop_times.txt`** for the stop time.

The value of **`vehicle_category_id`** in either **`routes.txt`** or **`stop_times.txt`** can be overridden by the value of **`vehicle_category_id`** in **`trips.txt`**.

Below is an example of vehicle categories. For the complete list of vehicle categories, please refer to Appendix B.

vehicle_category_id	vehicle_category_name
T8	8-car Tangara
T4	4-car Tangara
Tcar	An individual Tangara car

2.4 vehicle-couplings.txt

The `GTFS-VehicleCouplings` describes the arrangement of vehicles in composed vehicles such as trains. This extension requires the `GTFS-VehicleCategories` extension and is another extension that Transport for NSW has implemented for both the suburban and intercity trains.

This also adds the file `vehicle-couplings.txt` to the GTFS bundle which defines the relationship between composed vehicles (e.g. train) and individual vehicles (e.g. carriages).

The file `vehicle-couplings.txt` added to the GTFS bundle contains four data elements, and these are:

`parent_id` — (ID, Required) It defines the hierarchy between the different vehicle categories specified in `vehicle_categories.txt`. This field contains the `vehicle_category_id` of the parent vehicle.

(Note that only 3 levels of nesting are allowed. A parent vehicle is called a grandparent vehicle when its child vehicles are also defined as parent vehicles).

`child_id` — (ID, Required) It defines hierarchy between the different vehicle categories specified in `vehicle_categories.txt`. This field contains the `vehicle_category_id` of the child vehicle. Several child vehicles can be defined per parent vehicle.

(Note that only 3 levels of nesting are allowed. A child vehicle is called a grandchild vehicle when its parent vehicle is also defined as child vehicle).

`child_sequence` — (Non-negative Integer, Required) I defines the location of the child vehicle in respect to the other child vehicles composing the parent vehicle. The order number must increase along the parent vehicle, from its head to its tail.

`child_label` — (Text, Optional) It is a short text that can be used to easily identify the child vehicle. This text may be printed or displayed on either the vehicle or the platform. If nothing is usually used to identify child vehicles, no values should be provided.

Example of Vehicle Couplings

parent_id	child_id	child_sequence	child_label
T8	Tcar	1	1
T8	Tcar	2	2
T8	Tcar	3	3
T8	Tcar	4	4
T8	Tcar	5	5
T8	Tcar	6	6
T8	Tcar	7	7
T8	Tcar	8	8

The example shown in the table above defines an 8-car Tangara (see **parent_id**). Each row corresponds to one individual car (see **child_id**) and the sequence of the cars showing which car is first, which car is second, and so on, is defined under **child_sequence.**

2.5 vehicle-boardings.txt

The **GTFS-VehicleBoardings** describes where the vehicle stops on a platform, i.e. which cars can be accessed from the platform. If the train is longer than the platform, some cars may not be accessible from the platform and this extension provides that information.

This is the third extension Transport for NSW has implemented for both suburban and intercity trains. The extension requires the **GTFS-VehicleCategories** extension, and optionally the **GTFS-VehicleCouplings** to use the **child_sequence** and **grandchild_sequence** fields.

This also adds the file **vehicle-boardings.txt** to the GTFS bundle which describes how to map the vehicles with the boarding areas of the platform.

The file **vehicle-boardings.txt** added to the GTFS bundle contains four data elements, and these are:

vehicle_category_id (ID, Required) It identifies the **vehicle_category_id** which will stop in front of this boarding area.

If using **vehicle_couplings.txt**, this field must match the **vehicle_category_id** of the grandparent vehicle, or the one of the parent vehicle if no grandparent vehicles are specified.

child_sequence (ID, Conditionally required) It contains the **child_sequence** of a child vehicle. This field is useful when the same child vehicle appears multiple times in its parent vehicle.

It is required if using **vehicle_couplings.txt**.

grandchild_sequence	(ID, Conditionally required) It contains a **child_sequence** of a grandchild vehicle. This field is useful when the same grandchild vehicle appears multiple times in its parent vehicle.

boarding_area_id (ID, Required) It represents the boarding area at which the vehicle will stop. This field references **stop_id** from **stops.txt**. The referenced object must have **location_type** of **4** or **5**.

Example of Vehicle Boardings

vehicle_ category_id	child_sequence	grandchild_ sequence	boarding_area_id
T8	8		2077291
T8	7		2077291
T8	6		2077291
T8	5		2077291
T8	4		2077291
T8	3		2077291

The example shown in the table above is an 8-car Tangara train stopping at Asquith Station Platform 1 (`boarding_area_id` of 2077291 is Asquith Station Platform 1). Note that only cars 3 to 8 are identified. The reason for this is that Asquith Station Platform 1 can accommodate 6 Tangara cars only. It is a short platform. Cars 1 and 2 will not have a platform. Customers will not be able to board cars 1 and 2. This is also an indication that the 8-car Tangara train aligns its rear with the platform. Cars 1 and 2 are front cars and these will not be accommodated to a platform.

2.6 occupancies.txt

The GTFS-Bundle contains the optional file “occupancies.txt” which provides the forecast or predicted Train load level for all revenue trips, two weeks ahead. This information is supplied by a machine-learning algorithm, based on previous real-time historical data.

The Forecasted Train occupancies are based on historical trends and data and may not account for unplanned major events or unplanned timetable changes.

2.6.1 Files added or extended

File name	State	Description
occupancies.txt	Added	(Optional) Expected or usual in-vehicle occupancy levels.

2.6.2 Field and description

Field name	Description

trip_id	(ID referencing from stop_times.trip_id, **Required**) Identifies a trip_id that an occupancy level is described.
stop_sequence	(ID referencing from stop_times.stop_sequence, Conditionally Forbidden) Identifies a stop_sequence along occupancies.trip_id for which an occupancy level is described. Defined values in occupancies.stop_sequence will apply to subsequent stop_times.stop_sequence that are not defined in occupancies.stop_sequence, for the same trip_id. Conditionally Forbidden: • Forbidden if occupancies.trip_id is empty. • Optional otherwise.
occupancy_status	(Enum, **Required**) Indicates the nominal degree of in-vehicle rider occupancy. This field refers to the GTFS Realtime OccupancyStatus enums. 0 - Empty. The vehicle is considered empty by most measures, and has few or no passengers onboard, but is still accepting passengers. 1 - 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. 2 - 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. 3 - Standing room only. The vehicle can currently accommodate only standing passengers. 4 - Crushed standing room only. The vehicle can currently accommodate only standing passengers and has limited space for them. 5 - Full. The vehicle is considered full by most measures, but may still be allowing passengers to board. 6 - Not accepting passengers. The vehicle cannot accept passengers.
monday	(Enum, Conditionally Forbidden) Indicates whether an occupancy level is valid for all Mondays in the date range specified by occupancies.start_date and occupancies.end_date. Valid options are:

	• 1: The occupancy level applies for all Mondays in the date range. • 0 or empty: The occupancy level does not apply for all Mondays in the date range. Conditionally Forbidden: • Forbidden if occupancies.end_date is empty. • Optional otherwise.
tuesday	(Enum, Conditionally Forbidden) Functions in the same way as occupancies.monday except applies to occupancies.tuesday.
wednesday	(Enum, Conditionally Forbidden) Functions in the same way as occupancies.monday except applies to occupancies.wednesday.
thursday	(Enum, Conditionally Forbidden) Functions in the same way as occupancies.monday except applies to occupancies.thursday.
friday	(Enum, Conditionally Forbidden) Functions in the same way as occupancies.monday except applies to occupancies.friday.
saturday	(Enum, Conditionally Forbidden) Functions in the same way as occupancies.monday except applies to occupancies.saturday.
sunday	(Enum, Conditionally Forbidden) Functions in the same way as occupancies.monday except applies to occupancies.sunday.
start_date	(Date, **Required**) Start date of the date interval that the usual or expected occupancy level is projected.
end_date	(Date, Optional) End date of the date interval that the usual or expected occupancy level is projected. If defined, occupancies.end_date must be greater than occupancies.start_date. If empty, the occupancy level will apply only for the date specified in occupancies.start_date.
exception	(Enum, Optional) Indicates if an occupancy exception should override existing occupancy descriptions for the same trip or stop time for overlapping dates. Valid options are: • 0 or empty: Does not override another occupancy description. 1: Overrides another occupancy description.

Additional information available at:

- https://github.com/google/transit/pull/240
- Google GTFS-Occupancies.tx extension

3 GTFS-R Vehicle Position

3.1 Coverage

Vehicle coverage is provided for the Sydney Metro region, as well as the Central Coast & Newcastle Line, and South Coast Line to Kiama.

3.2 Non Timetabled Trains

The vehicle position feed can contain data for non-timetabled trains in addition to those in the GTFS bundle. Non timetabled trips will have a different trip id format as follows:
NonTimetabled.<trip_name>

3.3 VehicleID

The vehicle Id field has semantic content and is formed by a dot (.) separated list of the carriage identification numbers forming a train.

e.g. 7654.7655.7656.7657.8532.8533.8534.8535

These are carriage numbers which have been masked.

3.4 Vehicle Position inclusion of Carriage occupancy (PassLoad)

The published GTFS-R specification for Vehicle Position includes optional passenger load information for Waratah trains (A and B sets). The schema excerpt below is from the published protobuf V2.0

Schema – Vehicle Position including Carriage Occupancy

```
// Realtime positioning information for a given vehicle.
message VehiclePosition {

  // The Trip that this vehicle is serving.
  // Can be empty or partial if the vehicle cannot be identified with a given
  // trip instance.
  optional TripDescriptor trip = 1;

  // Additional information on the vehicle that is serving this trip.
  optional VehicleDescriptor vehicle = 8;

  // Current position of this vehicle.
  optional Position position = 2;
```

Schema – Vehicle Position including Carriage Occupancy

```
// 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.
optional uint32 current_stop_sequence = 3;

// Identifies the current stop. The value must be the same as in stops.txt in
// the corresponding GTFS feed.
optional string stop_id = 7;


enum VehicleStopStatus {
  // The vehicle is just about to arrive at the stop (on a stop
  // display, the vehicle symbol typically flashes).
  INCOMING_AT = 0;

  // The vehicle is standing at the stop.
  STOPPED_AT = 1;

  // The vehicle has departed and is in transit to the next stop.
  IN_TRANSIT_TO = 2;
}

// The exact status of the vehicle with respect to the current stop.
// Ignored if current_stop_sequence is missing.
optional VehicleStopStatus current_status = 4 [default = IN_TRANSIT_TO];

// 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).
optional uint64 timestamp = 5;




// Congestion level that is affecting this vehicle.
enum CongestionLevel {
  UNKNOWN_CONGESTION_LEVEL = 0;
  RUNNING_SMOOTHLY = 1;
  STOP_AND_GO = 2;
  CONGESTION = 3;
  SEVERE_CONGESTION = 4; // NEW
}
optional CongestionLevel congestion_level = 6;

// The degree of passenger occupancy of the whole train
// (Sum of passenger count for every Carriages of the train).
enum OccupancyStatus {

  // The vehicle is considered empty by most measures, and has few or no
  // passengers on-board, but is still accepting passengers.
  EMPTY = 0;

  // The vehicle has a relatively large percentage of seats available.
```

Schema – Vehicle Position including Carriage Occupancy

```
    // 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.
    MANY_SEATS_AVAILABLE = 1;

    // The vehicle has a relatively 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 feed producer.
    FEW_SEATS_AVAILABLE = 2;

    // The vehicle can currently accommodate only standing passengers.
    STANDING_ROOM_ONLY = 3;

    // The vehicle can currently accommodate only standing passengers
    // and has limited space for them.
    CRUSHED_STANDING_ROOM_ONLY = 4;

    // The vehicle is considered full by most measures, but may still be
    // allowing passengers to board.
    FULL = 5;

    // The vehicle is not accepting additional passengers.
    NOT_ACCEPTING_PASSENGERS = 6;
  }
  optional OccupancyStatus occupancy_status = 9;

  // The extensions namespace allows 3rd-party developers to extend the
  // GTFS-realtime specification in order to add and evaluate new features and
  // modifications to the spec.
  extensions 1000 to 1999;
}
```

Schema – Vehicle Position including Carriage Occupancy

```
// New CarriageDescriptor extending VehiclePosition which includes Train
// Carriage ID and Carriage Occupancy Information.
message CarriageDescriptor {

  // Train Carriage ID
  optional string name = 1;

  // Train Carriage Position number from the heading/Driver car
  required int32 position_in_consist=2;

  // Occupancy status of individual Train Carriage.
  enum OccupancyStatus {
       EMPTY = 0;
       MANY_SEATS_AVAILABLE = 1;
       FEW_SEATS_AVAILABLE = 2;
       STANDING_ROOM_ONLY = 3;
       CRUSHED_STANDING_ROOM_ONLY = 4;
       FULL = 5;
  }
  optional OccupancyStatus occupancy_status = 3;

  // Flag to indicate if the carriage is a “Quite Carriage”
  optional bool quiet_carriage = 4 [default = false];

  // Flag to indicate if the toilet type on this carriage
  enum ToiletStatus {
       NONE = 0;
       NORMAL = 1;
       ACCESSIBLE = 2;
       }
  optional ToiletStatus toilet = 5;

  // Indicates the availability of Luggage racks on this carriage
  optional bool luggage_rack = 6 [default = false];

  extensions 1000 to 1999;
}

extend transit_realtime.VehiclePosition {
  repeated CarriageDescriptor consist = 1007; // NEW
}
```

The Sydney Trains GTFS-R feed uses the following values but does not adopt their exact meaning. Threshold levels may vary during times where physical distancing of passengers is required. The below customer facing messages are advised for each enumerator that appears in the feed.

Level	GTFS-R Enum	GTFS-R spec	Customer facing message
Green	ENUM 1	MANY_SEATS_AVAILABLE	Spaces Available
Amber	ENUM 3	STANDING_ROOM_ONLY	Limited Space
Red	ENUM 4	CRUSHED_STANDING_ROOM_ONLY	Service has reached capacity

3.5 Example of Vehicle Position message including Carriage Occupancy

```
entity {
  id: "19"
  vehicle {
    trip {
      trip_id: "105P.1697.101.32.A.8.68334670"
      schedule_relationship: SCHEDULED
      route_id: "WST_2c"
    }
    position {
      latitude: -33.7664
      longitude: 150.89584
    }
    timestamp: 1632981081
    congestion_level: UNKNOWN_CONGESTION_LEVEL
    stop_id: "Blacktown.BN96 Loc"
    vehicle {
      id: "5009.5374.7561.7216.9253.6686.2683.5403"
      label: "15:30 Penrith Station to Central Station "
    }
    occupancy_status: MANY_SEATS_AVAILABLE
    [transit_realtime.consist] {
      position_in_consist: 3
      occupancy_status: MANY_SEATS_AVAILABLE
    }
    [transit_realtime.consist] {
      position_in_consist: 6
      occupancy_status: MANY_SEATS_AVAILABLE
    }
    [transit_realtime.consist] {
      position_in_consist: 1
      occupancy_status: MANY_SEATS_AVAILABLE
    }
    [transit_realtime.consist] {
      position_in_consist: 5
      occupancy_status: MANY_SEATS_AVAILABLE
    }
    [transit_realtime.consist] {
      position_in_consist: 2
      occupancy_status: MANY_SEATS_AVAILABLE
    }
    [transit_realtime.consist] {
      position_in_consist: 4
      occupancy_status: MANY_SEATS_AVAILABLE
    }
    [transit_realtime.consist] {
      position_in_consist: 8
      occupancy_status: MANY_SEATS_AVAILABLE
    }
    [transit_realtime.consist] {
      position_in_consist: 7
      occupancy_status: MANY_SEATS_AVAILABLE
    }
  }
}
```

4 GTFS-R Service Alerts

4.1 Coverage

Service Alerts are provided for the following categories:

- Line status – Generic information relating to current operation on a line. e.g. delays, track work
- Station facilities information – Information regarding lifts and escalator breakdowns / maintenance
- General station information – Other general messages relating to stations
- Trip information – messages regarding current state of specific trips. E.g. delays, cancellation.

4.2 Trip Based Service Alerts

Trip based Service Alerts are used to convey information around train running for that trip. These would generally be short messages related to delay announcements and reasons.

When service levels (On time running) around the network degrade during disruption events, It is anticipated that individual trip delay style messages will reduce and be replaced by appropriate Line status messages regarding the disruption.

4.3 Examples

Line service alert information:

```
entity {
  id: "1"
  alert {
    informed_entity {
      agency_id: "SydneyTrains"
      route_id: "BL_1a"
    }
    url {
      translation {
        text: "https://transportnsw.info/alerts#/train"
        language: "en"
      }
    }
    header_text {
      translation {
```

```
        text: "Major Delays"
        language: "en"
      }
    }
    description_text {
      translation {
        text: "Signalling failure."
        language: "en"
      }
    }
  }
}
```

Trip based alert:

```
entity {
  id: "3"
  alert {
    informed_entity {
      agency_id: "SydneyTrains"
      trip {
        trip_id: "12-E.1171.105.124.T.8"
      }
    }
    url {
      translation {
        text: "https://transportnsw.info/alerts#/train"
        language: "en"
      }
    }
    header_text {
      translation {
        text: "Trip Update"
        language: "en"
      }
    }
    description_text {
      translation {
        text: " Cancelled Due to electrical repairs."
        language: "en"
      }
    }
  }
}
```

Station facilities alert:

```
entity {
  id: "5"
  alert {
    informed_entity {
      agency_id: "SydneyTrains"
```

```
      stop_id: "200060"
    }
    url {
      translation {
        text: "https://transportnsw.info/alerts#/train"
        language: "en"
      }
    }
    header_text {
      translation {
        text: "Escalator Unavailable"
        language: "en"
      }
    }
    description_text {
      translation {
        text: "Platform 24/25 and ESR Concourse"
        language: "en"
      }
    }
  }
}
```

5 GTFS-R Trip Updates

TripUpdates provide predicted arrival and departure time for stops along the trip. They also provide information for any changes done on the trip running on that day.

5.1 Coverage

TripUpdates are provided for the following categories:

- New trips – Insert Trip
- Any changes done on the trip running on that day. These are shown as a replacement. They typically include scenarios like Added Stops, Skipped Stops, Cancel Trip, Terminate Early, Change Start, Change Platform, Reroute Trip, Extend Trip, Hold Trip.
- Delays - For Time Predictions – delay in stop time updated is used to communicate arrival and departure delay in seconds to a scheduled GTFS Trip.
- Train and Carriage Load Prediction – Trains and Carriage Load Forecast are published for active trains at each stopping nodes. The current supported trains Fleet types are:

Fleet Type Value	Train Set type
A	Waratah
B	Waratah Series 2
C	C-Set
H	Oscar
K	K-Set
M	Millennium
S	S-Set
T	Tangara
V	V-Set (Intercity)

Schema Trip Update including Occupancy forecast

```
message TripUpdate {
  required TripDescriptor trip = 1;
  optional VehicleDescriptor vehicle = 3;

  message StopTimeEvent {
    optional int32 delay = 1;
    optional int64 time = 2;
    optional int32 uncertainty = 3;
    extensions 1000 to 1999;
  }

  message StopTimeUpdate {
    optional uint32 stop_sequence = 1;
    optional string stop_id = 4;
    optional StopTimeEvent arrival = 2;
    optional StopTimeEvent departure = 3;


    enum ScheduleRelationship {
      SCHEDULED = 0;
      SKIPPED = 1;
      NO_DATA = 2;
            UNSCHEDULED = 3;
    }
    optional ScheduleRelationship schedule_relationship = 5
       [default = SCHEDULED];

          enum OccupancyStatus { // NEW
                  EMPTY = 0;
                  MANY_SEATS_AVAILABLE = 1;
                  FEW_SEATS_AVAILABLE = 2;
                  STANDING_ROOM_ONLY = 3;
                  CRUSHED_STANDING_ROOM_ONLY = 4;
                  FULL = 5;
                  NOT_ACCEPTING_PASSENGERS = 6;
          }
          optional OccupancyStatus departure_occupancy_status = 6; // NEW

    extensions 1000 to 1999;
  }
  repeated StopTimeUpdate stop_time_update = 2;
  optional uint64 timestamp = 4;
  optional int32 delay = 5;

  extensions 1000 to 1999;
}
```

Also TripUpdates -> StopTimeUpdate has been extended to include Carriage_seq_Predictive_Occupancy to provide predictive occupancy at carriage level.

```
extend transit_realtime.TripUpdate.StopTimeUpdate {
  repeated CarriageDescriptor carriage_seq_predictive_occupancy = 1007; // NEW
}
```

5.2 Examples

Insert Trip:

```
entity {
  id: "5566.617.130.32.C.2.0"
  trip_update {
    trip {
      trip_id: "5566.617.130.32.C.2.0"
      schedule_relationship: ADDED
      route_id: "NSL_1"
    }
    stop_time_update {
      arrival {
        time: 1409874540
      }
      departure {
        time: 1409874540
      }
      stop_id: "2000336"
    }
    stop_time_update {
      arrival {
        time: 1409874690
      }
      departure {
        time: 1409874750
      }
      stop_id: "2000393"
    }
    stop_time_update {
      arrival {
        time: 1409874846
      }
      departure {
        time: 1409874888
      }
      stop_id: "2000404"
    }
    stop_time_update {
      arrival {
        time: 1409875086
      }
      departure {
        time: 1409875116
      }
      stop_id: "206142"
    }
    stop_time_update {
      arrival {
        time: 1409875200
      }
      departure {
        time: 1409875260
      }
      stop_id: "2060104"
    }
    stop_time_update {
      arrival {
```

```
      time: 1409875356
    }
    departure {
      time: 1409875386
    }
    stop_id: "2060112"
  }
  stop_time_update {
    arrival {
      time: 1409875488
    }
    departure {
      time: 1409875518
    }
    stop_id: "2065162"
  }
  stop_time_update {
    arrival {
      time: 1409875632
    }
    departure {
      time: 1409875662
    }
    stop_id: "2065153"
  }
  stop_time_update {
    arrival {
      time: 1409875788
    }
    departure {
      time: 1409875818
    }
    stop_id: "206452"
  }
  stop_time_update {
    arrival {
      time: 1409875932
    }
    departure {
      time: 1409875962
    }
    stop_id: "2067144"
  }
  stop_time_update {
    arrival {
      time: 1409876280
    }
    departure {
      time: 1409876310
    }
    stop_id: "207263"
  }
  stop_time_update {
    arrival {
      time: 1409876430
    }
    departure {
      time: 1409876460
```

```
        }
        stop_id: "2073162"
      }
      stop_time_update {
        arrival {
          time: 1409876598
        }
        departure {
          time: 1409876628
        }
        stop_id: "2074182"
      }
      stop_time_update {
        arrival {
          time: 1409876718
        }
        departure {
          time: 1409876748
        }
        stop_id: "2074192"
      }
      stop_time_update {
        arrival {
          time: 1409876832
        }
        departure {
          time: 1409876862
        }
        stop_id: "2076242"
      }
      stop_time_update {
        arrival {
          time: 1409876964
        }
        departure {
          time: 1409876994
        }
        stop_id: "2077312"
      }
      stop_time_update {
        arrival {
          time: 1409877114
        }
        departure {
          time: 1409877174
        }
        stop_id: "2077302"
      }
      timestamp: 1409851188
    }
  }
```

Replacement service:

```
  entity {
    id: "108B.617.130.124.T.8.0"
    trip_update {
      trip {
```

```
    trip_id: "108B.617.130.124.T.8.0"
    schedule_relationship: REPLACEMENT
    route_id: "NL_1a"
  }
  stop_time_update {
    arrival {
      time: 1409870700
    }
    departure {
      time: 1409870700
    }
    stop_id: "2000336"
  }
  stop_time_update {
    arrival {
      time: 1409870856
    }
    departure {
      time: 1409870916
    }
    stop_id: "2000393"
  }
  stop_time_update {
    arrival {
      time: 1409871012
    }
    departure {
      time: 1409871054
    }
    stop_id: "2000404"
  }
  stop_time_update {
    arrival {
      time: 1409871240
    }
    departure {
      time: 1409871270
    }
    stop_id: "206142"
  }
  stop_time_update {
    arrival {
      time: 1409871414
    }
    departure {
      time: 1409871474
    }
    stop_id: "2060104"
  }
  stop_time_update {
    arrival {
      time: 1409871570
    }
    departure {
      time: 1409871600
    }
    stop_id: "2060112"
  }
```

```
      stop_time_update {
        arrival {
          time: 1409871702
        }
        departure {
          time: 1409871732
        }
        stop_id: "2065162"
      }
      stop_time_update {
        arrival {
          time: 1409871864
        }
        departure {
          time: 1409871894
        }
        stop_id: "2065153"
      }
      stop_time_update {
        arrival {
          time: 1409872020
        }
        departure {
          time: 1409872050
        }
        stop_id: "206452"
      }
      stop_time_update {
        arrival {
          time: 1409872314
        }
        departure {
          time: 1409872315
        }
        stop_id: "2067143"
      }
      timestamp: 1409872237
    }
  }
```

Delay:

```
entity {
  id: "293E.617.130.120.H.8.0"
  trip_update {
    trip {
      trip_id: "293E.617.130.120.H.8.0"
      schedule_relationship: SCHEDULED
      route_id: "NCCL_2b"
    }
    stop_time_update {
      arrival {
        delay: 0
      }
      departure {
        delay: 42
      }
      stop_id: "2079101"
      schedule_relationship: SCHEDULED
    }
    stop_time_update {
      arrival {
        delay: 42
      }
      departure {
        delay: 42
      }
      stop_id: "2077291"
      schedule_relationship: SCHEDULED
    }
    stop_time_update {
      arrival {
        delay: 42
      }
      departure {
        delay: 0
      }
      stop_id: "2077301"
      schedule_relationship: SCHEDULED
    }
    stop_time_update {
      arrival {
        delay: 0
      }
      departure {
        delay: 0
      }
      stop_id: "2077311"
      schedule_relationship: SCHEDULED
    }
    timestamp: 1409869080
  }
```

Train and Carriage load Prediction/Forecast (new feature added 20-Aug-22):

```
entity {
  id: "127G.1785.102.8.A.8.73477865"
  trip_update {
    trip {
      trip_id: "127G.1785.102.8.A.8.73477865"
      schedule_relationship: SCHEDULED
      route_id: "NTH_1a"
    }
```

```
stop_time_update {
 arrival {
  delay: 86337
 }
 departure {
  delay: 86400
 }
 stop_id: "2077291"
 schedule_relationship: SCHEDULED
 departure_occupancy_status: MANY_SEATS_AVAILABLE
 [transit_realtime.carriage_seq_predictive_occupancy] {
  position_in_consist: 1
  departure_occupancy_status: MANY_SEATS_AVAILABLE
 }
 [transit_realtime.carriage_seq_predictive_occupancy] {
  position_in_consist: 2
  departure_occupancy_status: MANY_SEATS_AVAILABLE
 }
 [transit_realtime.carriage_seq_predictive_occupancy] {
  position_in_consist: 3
  departure_occupancy_status: MANY_SEATS_AVAILABLE
 }
 [transit_realtime.carriage_seq_predictive_occupancy] {
  position_in_consist: 4
  departure_occupancy_status: MANY_SEATS_AVAILABLE
 }
 [transit_realtime.carriage_seq_predictive_occupancy] {
  position_in_consist: 5
  departure_occupancy_status: MANY_SEATS_AVAILABLE
 }
 [transit_realtime.carriage_seq_predictive_occupancy] {
  position_in_consist: 6
  departure_occupancy_status: MANY_SEATS_AVAILABLE
 }
 [transit_realtime.carriage_seq_predictive_occupancy] {
  position_in_consist: 7
  departure_occupancy_status: MANY_SEATS_AVAILABLE
 }
 [transit_realtime.carriage_seq_predictive_occupancy] {
  position_in_consist: 8
  departure_occupancy_status: MANY_SEATS_AVAILABLE
 }
}
stop_time_update {
 arrival {
  delay: 86400
```

```
  }
  departure {
    delay: 86370
  }
  stop_id: "2077301"
  schedule_relationship: SCHEDULED
  departure_occupancy_status: MANY_SEATS_AVAILABLE
  [transit_realtime.carriage_seq_predictive_occupancy] {
    position_in_consist: 1
    departure_occupancy_status: MANY_SEATS_AVAILABLE
  }
  [transit_realtime.carriage_seq_predictive_occupancy] {
    position_in_consist: 2
    departure_occupancy_status: MANY_SEATS_AVAILABLE
  }
  [transit_realtime.carriage_seq_predictive_occupancy] {
    position_in_consist: 3
    departure_occupancy_status: MANY_SEATS_AVAILABLE
  }
  [transit_realtime.carriage_seq_predictive_occupancy] {
    position_in_consist: 4
    departure_occupancy_status: MANY_SEATS_AVAILABLE
  }
  [transit_realtime.carriage_seq_predictive_occupancy] {
    position_in_consist: 5
    departure_occupancy_status: MANY_SEATS_AVAILABLE
  }
  [transit_realtime.carriage_seq_predictive_occupancy] {
    position_in_consist: 6
    departure_occupancy_status: MANY_SEATS_AVAILABLE
  }
  [transit_realtime.carriage_seq_predictive_occupancy] {
    position_in_consist: 7
    departure_occupancy_status: MANY_SEATS_AVAILABLE
  }
  [transit_realtime.carriage_seq_predictive_occupancy] {
    position_in_consist: 8
    departure_occupancy_status: MANY_SEATS_AVAILABLE
  }
 }
 stop_time_update {
  arrival {
    delay: 86370
  }
  departure {
    delay: 86370
```

```
    }
    stop_id: "2077311"
    schedule_relationship: SCHEDULED
    departure_occupancy_status: MANY_SEATS_AVAILABLE
    [transit_realtime.carriage_seq_predictive_occupancy] {
      position_in_consist: 1
      departure_occupancy_status: MANY_SEATS_AVAILABLE
    }
    [transit_realtime.carriage_seq_predictive_occupancy] {
      position_in_consist: 2
      departure_occupancy_status: MANY_SEATS_AVAILABLE
    }
    [transit_realtime.carriage_seq_predictive_occupancy] {
      position_in_consist: 3
      departure_occupancy_status: MANY_SEATS_AVAILABLE
    }
    [transit_realtime.carriage_seq_predictive_occupancy] {
      position_in_consist: 4
      departure_occupancy_status: MANY_SEATS_AVAILABLE
    }
    [transit_realtime.carriage_seq_predictive_occupancy] {
      position_in_consist: 5
      departure_occupancy_status: MANY_SEATS_AVAILABLE
    }
    [transit_realtime.carriage_seq_predictive_occupancy] {
      position_in_consist: 6
      departure_occupancy_status: MANY_SEATS_AVAILABLE
    }
    [transit_realtime.carriage_seq_predictive_occupancy] {
      position_in_consist: 7
      departure_occupancy_status: MANY_SEATS_AVAILABLE
    }
    [transit_realtime.carriage_seq_predictive_occupancy] {
      position_in_consist: 8
      departure_occupancy_status: MANY_SEATS_AVAILABLE
    }
}
```

6 Appendix A – Duplicate Sydney Trains and NSW Trains services

The following list of trips represents services that appear in both the Sydney Trains realtime feed and NSW Trains intercity and regional realtime feed. For consumers using both feeds, TfNSW recommends filtering out these services from the Sydney Trains feed and preferentially using the NSW Trains feed.

Diesel NSW Trains service (NP, NT, V, SN, SP, ST, WN, WP, WT, KN, CN

Diesel NSW Trains service (NP, NT, V, SN, SP, ST, WN, WP, WT, KN, CN)

IC-Hunter Line - Up		
Scone / Dungog - Hamilton		
Monday to Friday		
Saturdays, Sundays and Public Holidays		
Ride Id:	Trip:	Consist:
10536	V618	2N
10537	V622	2N
10539	V638	2N
10543	V676	2N
10544	V682	2N
10545	V682	2N
10546	V700	2J
10548	V702	2J
10549	V702	2J
10551	V704	2J
10553	V706	2J
10554	V708	2J
10556	V710	2J
10557	V710	2J
10560	V712	2J
10562	V714	2J
10565	V716	2J
10569	V720	2J
10571	V722	2J
10572	V722	2J
10575	V724	2J
10578	V728	2J
10580	V730	2N
10581	V730	2J

IC-Hunter Line - Dn		
Hamilton - Scone / Dungog		
Monday to Friday		
Saturdays, Sundays and Public Holidays		
Ride Id:	Trip:	Consist:
10534	V603	2N
10535	V607	2N
10538	V625	2N
10540	V659	2N
10541	V669	2N
10542	V671	2N
11455	V701	2J
11456	V701	2J
11457	V703	2J
11458	V703	2J
10552	V705	2J
10555	V709	2J
10558	V711	2J
10559	V711	2J
10561	V713	2J
10563	V715	2J
10564	V715	2J
10566	V717	2J
10567	V719	2J
10568	V719	2J
10570	V721	2J
10573	V723	2J
10574	V723	2J
10576	V727	2J

10584	V732	2J
10585	V734	2J
10586	V734	2J
10589	V736	2J
10591	V738	2J
10594	V740	2J
10596	V742	2J
10597	V742	2J
10602	V744	2J
10603	V744	2J
10607	V748	2J
10608	V748	2J
10611	V750	2J
10614	V752	2J
10615	V754	2N
10616	V754	2J
10619	V756	2J
10621	V758	2J
10622	V758	2J
10623	V760	2J
10626	V762	2N
10627	V762	2J
10628	V764	2J
10630	V766	2J
10631	V766	2J
10632	V768	2J
10634	V770	2J
10635	V770	2J
10637	V772	2J
10639	V774	2J
10642	V776	2J
10644	V778	2J
10645	V778	2J
10648	V780	2J
10650	V782	2J
10653	V784	2J
10655	V786	2J
10657	V788	2J
10659	V790	2J
10660	V790	2J
10661	V792	2J
10667	V918	2N
10669	V926	2J
10671	V938	2J

10577	V727	2J
10579	V729	2J
10582	V731	2J
10583	V731	2J
10587	V735	2J
10588	V735	2J
10590	V737	2J
10592	V739	2J
10593	V739	2J
10595	V741	2N
10599	V743	2J
10600	V743	2J
10601	V743	2J
10604	V745	2J
10605	V747	2J
10606	V747	2J
10610	V749	2N
10612	V751	2J
10613	V751	2J
10617	V755	2J
10618	V755	2J
10620	V757	2N
10624	V761	2J
10625	V761	2J
10629	V765	2J
10633	V769	2J
10636	V771	2J
10638	V773	2J
10640	V775	2J
10641	V775	2J
10643	V777	2J
10646	V779	2J
10647	V779	2J
10649	V781	2J
10651	V783	2J
10652	V783	2J
10654	V785	2J
10656	V787	2J
10658	V789	2J
10662	V793	2J
10663	V793	2J
10664	V797	2J
10665	V907	2N
10666	V913	2J

10673	V946	2J
10675	V958	2J
10678	V966	2J
10680	V974	2J
10681	V974	2J

10668	V925	2J
10670	V933	2J
10672	V945	2J
10674	V953	2J
10676	V965	2J
10679	V967	2J

South Coast - diesels		
Monday to Friday		
Saturdays, Sundays and Public Holidays		
Ride Id:	Trip:	Consist:
10160	CN90	2N
10240	KN01	2N
10241	KN01	2N
10243	KN03	2N
10244	KN07	2N
10245	KN09	2N
10246	KN10	2N
10247	KN17	2N
10248	KN18	2N
10249	KN19	2N
10250	KN20	2N
10251	KN23	2N
10252	KN23	2N
10253	KN24	2N
10254	KN27	2N
10255	KN28	2N
10256	KN33	2N
10257	KN35	2N
10258	KN36	2N
10259	KN36	2N
10260	KN40	2N
10261	KN41	2N
10262	KN42	2N
10263	KN43	2N
10264	KN48	2N
10265	KN48	2N
10266	KN49	2N
10267	KN51	2N
10268	KN55	2N

Southern Highlands (diesels 2N or SN sets)		
Monday to Friday		
Saturdays, Sundays and Public Holidays		
Ride Id:	Trip:	Consist:
11386	SN20	2N
11387	SN20	2N
11364	SN21	2N
11366	SN23	2N
11388	SN24	2N
11389	SN24	2N
11390	SN25	2N
11391	SN25	2N
11367	SN26	2N
11392	SN27	2N
11393	SN28	2N
11394	SN28	2N
11395	SN29	2N
11396	SN30	2N
11397	SN31	2N
11398	SN31	2N
11369	SN32	2N
11399	SN32	2N
11400	SN33	2N
11401	SN34	2N
11402	SN34	2N
11403	SN35	2N
11404	SN35	2N
11405	SN36	2N
11406	SN38	2N
11407	SN39	2N
11408	SN39	2N
11409	SN40	2N
11370	SN40/2	2N/4N

10269	KN56	2N
10270	KN56	2N
10271	KN57	2N
10272	KN59	2N
10273	KN64	2N
10274	KN64	2N
10275	KN65	2N
10276	KN68	2N
10277	KN71	2N
10278	KN72	2N
10279	KN72	2N
10280	KN73	2N
10281	KN76	2N
10282	KN80	2N
10283	KN81	2N
10284	KN82	2N
10286	KN86	2N

11410	SN43	2N
11411	SN43	2N
11412	SN44	2N
11413	SN44	2N
11414	SN45	2N
11415	SN46	2N
11416	SN48	2N
11417	SN49	2N
11418	SN49	2N
11419	SN50	2N
11420	SN51	2N
11421	SN52	2N
11371	SN53	4N
11422	SN53	2N
11373	SN54	4N
11423	SN54	2N
11424	SN55	2N
11425	SN55	2N
11426	SN56	2N
11427	SN59	2N
11428	SN59	2N
11429	SN60	2N
11430	SN61	2N
11375	SN61/3	4N/2N
11431	SN64	2N
11432	SN65	2N
11433	SN67	2N
11434	SN68	2N
11376	SN69	2N
11435	SN70	2N
11436	SN71	2N
11437	SN72	2N
11438	SN73	2N
11439	SN74	2N
11440	SN75	2N
11441	SN76	2N
11442	SN77	2N
11443	SN78	2N
11444	SN79	2N
11377	SN80	2N
11378	SN82	2N
11445	SN85	2N
11446	SN85	2N

West Blue Mountains (diesels 2N or WN sets)		
Monday to Friday		
Saturdays, Sundays and Public Holidays		
Ride Id:	Trip:	Consist:
10007	WN11	2N
10003	WN12	2N
10008	WN15	2N
10004	WN16	2N
10005	WN17	2N
10006	WN17	2N
10009	WN18	2N
10010	WN18	2N
10011	WN18	2N

North Coast services	North West services	West and Broken Hill services	Canberra services	Griffith services	Melbourne services
Trip:	Trip:	Trip:	Trip:	Trip:	Trip:
NT33	NP23	WT27	SP31	SP41	ST23
NT35	NP43	WP45	SP33	SP41	ST21
NT31	NP24	WT28	SP35		ST22
NT34	NP44	WP46	SP32		ST24
NT36			SP34		
NT32			SP36		

7 Appendix B – Complete List of Vehicle Categories

vehicle_category_id	vehicle_category_name
A8	8 car Waratah
Acar	Individual Waratah car
B8	8 car Waratah Series 2
Bcar	Individual Waratah Series 2 car
C4	4 car C-set
C8	8 car C-set
Ccar	Individual C-set car
D10	10 car NIF
D4	4 car NIF
D6	6 car NIF
D8	8 car NIF
Dcar	Individual NIF car
H4	4 car Oscar
H8	8 car Oscar
Hcar	Individual Oscar car
J2	2 car Hunter
J4	4 car Hunter
Jcar	Individual Hunter car
K4	4 car K-set
K8	8 car K-set
Kcar	Individual K-set car
M4	4 car Millenium
M8	8 car Millenium
Mcar	Individual Millenium car
N2	2 car Endeavour
N4	4 car Endeavour
N6	6 car Endeavour
Ncar	Individual Endeavour car

P2	2 car Xplorer
P3	3 car Xplorer
P4	4 car Xplorer
P5	5 car Xplorer
P6	6 car Xplorer
P7	7 car Xplorer
Pcar	Individual Xplorer car
T4	4 car Tangara
T8	8 car Tangara
Tcar	Individual Tangara car
V4	4 car V-set
V8	8 car V-set
Vcar	Individual V-set car
X4	4 car XPT
X5	5 car XPT
X6	6 car XPT
X7	7 car XPT
Xcar	Individual XPT car