

Opturion Dynamic Transport Optimiser
Data Formats
9 September 2026



OPTURION
Intelligent Decision Support

© 2017–2026 Opturion Pty Ltd. All rights reserved.

Opturion is a registered trademark of Opturion Pty Ltd. All other trademarks mentioned in this document are the property of their respective owners.

Contents

1 Overview **1**

2 Data Model **2**

2.1 Scenarios . . . 2
2.2 Solutions . . . 2
2.3 Identifiers . . . 2
2.4 Attributes . . . 2
2.5 Weights and Volumes . . . 2
2.6 Time . . . 2
2.7 Costs and Hidden Costs . . . 2
2.8 Global Parameters . . . 4
2.8.1 Iterations . . . 4
2.8.2 Batched Loads . . . 4
2.8.3 Colocated Pickups . . . 4
2.8.4 Sticky Deliveries . . . 4
2.8.5 Scaling Factors . . . 4
2.8.6 Geofencing . . . 4
2.8.7 Maximum Neighbourhood Size . . . 4
2.8.8 Preferred-Vehicle Rewards . . . 5
2.8.9 Unassigned Penalty Scheme . . . 6
2.8.10 Order Splitting Strategy . . . 7
2.8.11 Heavy Order Threshold . . . 8
2.9 Replanning . . . 9
2.10 Locations . . . 10
2.11 Vehicles . . . 12
2.11.1 Vehicle Costs . . . 12
2.11.2 Vehicle Locations . . . 13
2.11.3 Vehicle Times . . . 13
2.11.4 Vehicle Capacities . . . 13
2.11.5 Vehicle Compartments . . . 13
2.11.6 Maximum Loads . . . 13
2.11.7 Maximum Drops per Load . . . 14
2.11.8 Fatigue Management Scheme . . . 14
2.11.9 Loaded Breaks . . . 14
2.11.10 Vehicle Speed Scale . . . 14
2.11.11 Vehicle Types . . . 14
2.11.12 Load Ordering . . . 15
2.12 Orders . . . 16
2.13 Fatigue Management . . . 18
2.13.1 Standard Fatigue Management . . . 18
2.13.2 Simple Fatigue Management . . . 18
2.14 Iteration Schemes . . . 19
2.15 Time Windows . . . 20
2.15.1 Arrival Only in Time Window . . . 20
2.16 Soft Time Windows . . . 20
2.17 Compatibility . . . 21
2.18 Locked Orders . . . 22
2.19 Exclusive Orders . . . 23
2.20 Preloading . . . 24
2.21 Stops . . . 25
2.22 Loads . . . 25
2.23 Solution Times . . . 25
2.24 Unassigned Orders . . . 26
2.25 Grouped Pickup . . . 27

2.26 Pickup Segments . . . 28
2.27 Delivery Segments . . . 28
2.28 Finish Segments . . . 28
2.29 Loaded Distance Penalties . . . 29
2.30 Split Solving . . . 30
2.30.1 Example . . . 30

3 Microsoft Excel Input **31**
3.1 Identifiers in Excel . . . 31
3.2 Times in Excel . . . 31
3.3 Sheet: General . . . 32
3.3.1 Column: Parameter . . . 32
3.3.2 Column: Value . . . 32
3.3.3 Parameter: Scenario Name . . . 32
3.3.4 Parameter: Iterations . . . 32
3.3.5 Parameter: Unassigned Penalty Scheme . . . 32
3.3.6 Parameter: Fixed Unassigned Penalty . . . 32
3.3.7 Parameter: Rush Hour . . . 32
3.3.8 Parameter: Use Miles . . . 32
3.3.9 Parameter: Iteration Scheme . . . 33
3.3.10 Parameter: Colocated Pickups . . . 33
3.3.11 Parameter: Geofence North . . . 33
3.3.12 Parameter: Geofence East . . . 33
3.3.13 Parameter: Geofence South . . . 33
3.3.14 Parameter: Geofence West . . . 33
3.3.15 Parameter: Hard Geofence Errors . . . 33
3.3.16 Parameter: Batched Loads . . . 33
3.3.17 Parameter: Sticky Deliveries . . . 33
3.3.18 Parameter: Cost Scale . . . 34
3.3.19 Parameter: Weight Scale . . . 34
3.3.20 Parameter: Volume Scale . . . 34
3.3.21 Parameter: Maximum Neighbourhood Size . . . 34
3.3.22 Parameter: Disable Replanning . . . 34
3.3.23 Parameter: Previous-Vehicle Fixed Reward . . . 34
3.3.24 Parameter: Previous-Vehicle Weight Reward . . . 34
3.3.25 Parameter: Previous-Vehicle Volume Reward . . . 34
3.3.26 Parameter: Forbid Unrecognised Columns . . . 35
3.3.27 Parameter: Arrival Only in Time Window . . . 35
3.3.28 Parameter: Order Splitting Strategy . . . 35
3.3.29 Parameter: Split Solve . . . 35
3.3.30 Parameter: Split Solve Jobs . . . 35
3.4 Sheet: Excel Report Configuration . . . 36
3.4.1 Column: Parameter . . . 36
3.4.2 Column: Value . . . 36
3.4.3 Parameter: Unassigned Order Attributes . . . 36
3.4.4 Parameter: Order Vehicle Compatibility . . . 36
3.5 Sheet: Locations . . . 37
3.5.1 Column: Id . . . 37
3.5.2 Column: Name . . . 37
3.5.3 Column: Attributes . . . 37
3.5.4 Column: Address . . . 37
3.5.5 Column: Longitude . . . 37
3.5.6 Column: Latitude . . . 37
3.5.7 Column: Site Time . . . 37
3.5.8 Column: Load Time . . . 37
3.5.9 Column: Unload Time . . . 37
3.5.10 Column: Opening Time . . . 38
3.5.11 Column: Closing Time . . . 38

3.5.12 Column: Operating Time Windows . . . 38
3.5.13 Column: Cost per Visit . . . 38
3.5.14 Column: Hidden Cost per Visit . . . 38
3.5.15 Column: Loading Bays . . . 38
3.5.16 Column: Loading Bay Gap . . . 38
3.5.17 Column: Allow Preloading . . . 38
3.5.18 Column: Allow Splitting . . . 39
3.6 Sheet: Orders . . . 40
3.6.1 Column: Id . . . 40
3.6.2 Column: Name . . . 40
3.6.3 Column: Customer Name . . . 40
3.6.4 Column: Pickup Location . . . 40
3.6.5 Column: Pickup Name . . . 40
3.6.6 Column: Pickup Service Time . . . 40
3.6.7 Column: Earliest Pickup Time . . . 40
3.6.8 Column: Latest Pickup Time . . . 40
3.6.9 Column: Pickup Time Windows . . . 40
3.6.10 Column: Pickup Soft Time Windows . . . 41
3.6.11 Column: Soft Pickup Constant . . . 41
3.6.12 Column: Soft Pickup Variable . . . 41
3.6.13 Column: Delivery Location . . . 41
3.6.14 Column: Delivery Name . . . 41
3.6.15 Column: Delivery Service Time . . . 41
3.6.16 Column: Earliest Delivery Time . . . 41
3.6.17 Column: Latest Delivery Time . . . 41
3.6.18 Column: Delivery Time Windows . . . 41
3.6.19 Column: Delivery Soft Time Windows . . . 42
3.6.20 Column: Soft Delivery Constant . . . 42
3.6.21 Column: Soft Delivery Variable . . . 42
3.6.22 Column: Maximum Duration . . . 42
3.6.23 Column: Weight . . . 42
3.6.24 Column: Volume . . . 42
3.6.25 Column: Assign Cost . . . 42
3.6.26 Column: Hidden Assign Cost . . . 42
3.6.27 Column: Revenue . . . 43
3.6.28 Column: Attributes . . . 43
3.6.29 Column: Allow Splitting . . . 43
3.7 Sheet: Vehicle Type . . . 44
3.7.1 Column: Id . . . 44
3.7.2 Column: Start Location . . . 44
3.7.3 Column: Finish Location . . . 44
3.7.4 Column: Maximum Weight . . . 44
3.7.5 Column: Maximum Volume . . . 44
3.7.6 Column: Earliest Start Time . . . 44
3.7.7 Column: Latest Start Time . . . 44
3.7.8 Column: Latest Finish Time . . . 44
3.7.9 Column: Minimum Paid Time . . . 45
3.7.10 Column: Maximum Drive Time . . . 45
3.7.11 Column: Maximum Work Time . . . 45
3.7.12 Column: Load Time . . . 45
3.7.13 Column: Unload Time . . . 45
3.7.14 Column: Speed Scale . . . 45
3.7.15 Column: Cost per Use . . . 45
3.7.16 Column: Cost per Hour . . . 45
3.7.17 Column: Cost per Kilometre . . . 46
3.7.18 Column: Cost per Mile . . . 46
3.7.19 Column: Cost per Load . . . 46
3.7.20 Column: Hidden Cost per Use . . . 46

3.7.21 Column: Hidden Cost per Hour . . . 46
3.7.22 Column: Hidden Cost per Kilometre . . . 46
3.7.23 Column: Hidden Cost per Mile . . . 46
3.7.24 Column: Hidden Cost per Load . . . 46
3.7.25 Column: Finish Segment Distance Penalty . . . 47
3.7.26 Column: Finish Segment Per Hour Penalty . . . 47
3.7.27 Column: Pickup Segment Fixed Penalty . . . 47
3.7.28 Column: Pickup Segment Distance Penalty . . . 47
3.7.29 Column: Delivery Segment Fixed Penalty . . . 47
3.7.30 Column: Delivery Segment Distance Penalty . . . 47
3.7.31 Column: Loaded Weight Distance Penalty . . . 47
3.7.32 Column: Loaded Volume Distance Penalty . . . 47
3.7.33 Column: Attributes . . . 48
3.7.34 Column: Break Scheme . . . 48
3.7.35 Column: Forbid Loaded Breaks . . . 48
3.7.36 Column: Maximum Loads . . . 48
3.7.37 Column: Compartments . . . 48
3.7.38 Column: Allow Preloading . . . 48
3.7.39 Column: Load Order . . . 48
3.7.40 Column: Heavy Order Threshold . . . 48
3.8 Sheet: Fleet . . . 49
3.8.1 Column: Id . . . 49
3.8.2 Column: Ignored . . . 49
3.8.3 Column: Vehicle Type . . . 49
3.8.4 Column: Name . . . 49
3.8.5 Column: Driver Name . . . 49
3.8.6 Column: Start Location . . . 49
3.8.7 Column: Finish Location . . . 49
3.8.8 Column: Maximum Weight . . . 49
3.8.9 Column: Maximum Volume . . . 50
3.8.10 Column: Earliest Start Time . . . 50
3.8.11 Column: Latest Start Time . . . 50
3.8.12 Column: Latest Finish Time . . . 50
3.8.13 Column: Minimum Paid Time . . . 50
3.8.14 Column: Maximum Drive Time . . . 50
3.8.15 Column: Maximum Work Time . . . 50
3.8.16 Column: Load Time . . . 50
3.8.17 Column: Unload Time . . . 51
3.8.18 Column: Speed Scale . . . 51
3.8.19 Column: Cost per Use . . . 51
3.8.20 Column: Cost per Hour . . . 51
3.8.21 Column: Cost per Kilometre . . . 51
3.8.22 Column: Cost per Mile . . . 51
3.8.23 Column: Cost per Load . . . 51
3.8.24 Column: Hidden Cost per Use . . . 51
3.8.25 Column: Hidden Cost per Hour . . . 52
3.8.26 Column: Hidden Cost per Kilometre . . . 52
3.8.27 Column: Hidden Cost per Mile . . . 52
3.8.28 Column: Hidden Cost per Load . . . 52
3.8.29 Column: Finish Segment Distance Penalty . . . 52
3.8.30 Column: Finish Segment Per Hour Penalty . . . 52
3.8.31 Column: Pickup Segment Fixed Penalty . . . 52
3.8.32 Column: Pickup Segment Distance Penalty . . . 52
3.8.33 Column: Delivery Segment Fixed Penalty . . . 53
3.8.34 Column: Delivery Segment Distance Penalty . . . 53
3.8.35 Column: Loaded Weight Distance Penalty . . . 53
3.8.36 Column: Loaded Volume Distance Penalty . . . 53
3.8.37 Column: Attributes . . . 53

3.8.38 Column: Break Scheme . . . 53
3.8.39 Column: Forbid Loaded Breaks . . . 53
3.8.40 Column: Maximum Loads . . . 53
3.8.41 Column: Maximum Drops per Load . . . 53
3.8.42 Column: Compartments . . . 54
3.8.43 Column: Allow Preloading . . . 54
3.8.44 Column: Load Order . . . 54
3.8.45 Column: Heavy Order Threshold . . . 54
3.9 Sheet: Standard Fatigue Management . . . 55
3.9.1 Column: Id . . . 55
3.9.2 Column: Number . . . 55
3.9.3 Column: Duration . . . 55
3.9.4 Column: Interval . . . 55
3.10 Sheet: Simple Fatigue Management . . . 56
3.10.1 Column: Id . . . 56
3.10.2 Column: Earliest . . . 56
3.10.3 Column: Preferred . . . 56
3.10.4 Column: Latest . . . 56
3.10.5 Column: Duration . . . 56
3.11 Sheet: Time Windows . . . 57
3.11.1 Column: Id . . . 57
3.11.2 Column: Start . . . 57
3.11.3 Column: End . . . 57
3.12 Sheet: Compartments . . . 58
3.12.1 Column: Scheme Id . . . 58
3.12.2 Column: Id . . . 58
3.12.3 Column: Name . . . 58
3.12.4 Column: Maximum Weight . . . 58
3.12.5 Column: Maximum Volume . . . 58
3.12.6 Column: Attributes . . . 58
3.13 Sheet: Rush Hour . . . 59
3.13.1 Column: Start . . . 59
3.13.2 Column: End . . . 59
3.13.3 Column: Speed Scale . . . 59
3.14 Sheet: Iteration Schemes . . . 60
3.14.1 Column: Id . . . 60
3.14.2 Column: Iterations . . . 60
3.14.3 Column: Maximum Neighbourhood Size . . . 60
3.15 Sheet: *Order-Vehicle* Compatibility . . . 61
3.15.1 Column `A`: Order Attribute . . . 61
3.15.2 Row `1`: Vehicle Attribute . . . 61
3.15.3 All other cells (except `A1`) . . . 61
3.16 Sheet: *Order-Compartment* Compatibility . . . 62
3.16.1 Column `A`: Order Attribute . . . 62
3.16.2 Row `1`: Compartment Attribute . . . 62
3.16.3 All other cells (except `A1`) . . . 62
3.17 Sheet: *Order-Order* Compatibility . . . 63
3.17.1 Column `A`: Order Attribute . . . 63
3.17.2 Row `1`: Order Attribute . . . 63
3.17.3 All other cells (except `A1`) . . . 63
3.18 Sheet: *Location-Vehicle* Compatibility . . . 64
3.18.1 Cell A1 . . . 64
3.18.2 Column A: Location Attribute . . . 64
3.18.3 Row 1: Vehicle Attribute . . . 64
3.18.4 All other cells . . . 64
3.19 Sheet: *Order-Vehicle* Costs . . . 65
3.19.1 Column: Order . . . 65
3.19.2 Column: Vehicle . . . 65

3.19.3 Column: Cost . . . 65
3.19.4 Column: Hidden Cost . . . 65
3.20 Sheet: *Order-Vehicle* Preferences . . . 66
3.20.1 Column: Order . . . 66
3.20.2 Column: Vehicle . . . 66
3.20.3 Column: Fixed Reward . . . 66
3.20.4 Column: Weight Reward . . . 66
3.20.5 Column: Volume Reward . . . 66
3.21 Sheet: Order Precedences . . . 67
3.21.1 Column: Before . . . 67
3.21.2 Column: After . . . 67
3.21.3 Column: Gap . . . 67
3.22 Sheet: Grouped Pickups . . . 68
3.22.1 All Rows . . . 68
3.23 Sheet: Locked Orders . . . 69
3.23.1 Column: Order . . . 69
3.23.2 Column: Vehicle . . . 69
3.24 Sheet: Exclusive Orders . . . 70
3.25 Sheet: Previous Solution . . . 71
3.25.1 Column: Order . . . 71
3.25.2 Column: Vehicle . . . 71
3.25.3 Column: Stop . . . 71
3.26 Sheet: Time Matrix . . . 72
3.26.1 The Matrix . . . 72
3.27 Sheet: Distance Matrix . . . 72
3.27.1 The Matrix . . . 72
3.28 Note on Matrices . . . 72
3.29 Sheet: Matrix Override . . . 73
3.29.1 Column: Location 1 . . . 73
3.29.2 Column: Location 2 . . . 73
3.29.3 Column: Time Forwards . . . 73
3.29.4 Column: Time Backwards . . . 73
3.29.5 Column: Distance Forwards . . . 73
3.29.6 Column: Distance Backwards . . . 73

4 Microsoft Excel Output **74**
4.1 Sheet: Runsheet . . . 75
4.1.1 Column: Vehicle . . . 75
4.1.2 Column: Load . . . 75
4.1.3 Column: Stop Number . . . 75
4.1.4 Column: ETA . . . 75
4.1.5 Column: ETD . . . 75
4.1.6 Column: Stop . . . 75
4.1.7 Column: Order . . . 75
4.1.8 Column: Customer Name . . . 75
4.1.9 Column: Delivery Name . . . 75
4.1.10 Column: Compartment . . . 75
4.1.11 Column: Address . . . 75
4.1.12 Column: Volume . . . 76
4.1.13 Column: Delta Volume . . . 76
4.1.14 Column: Weight . . . 76
4.1.15 Column: Delta Weight . . . 76
4.1.16 Column: Site Time . . . 76
4.1.17 Column: Load Time . . . 76
4.1.18 Column: Unload Time . . . 76
4.1.19 Column: Service Time . . . 76
4.1.20 Column: Transit Time . . . 76
4.1.21 Column: Transit Distance . . . 76

4.1.22 Column: Soft Time Window Start . . . 76
4.1.23 Column: Soft Time Window End . . . 76
4.1.24 Column: Soft Time Window Delta . . . 77
4.2 Sheet: Vehicle Summary . . . 78
4.2.1 Column: Vehicle . . . 78
4.2.2 Column: Used . . . 78
4.2.3 Column: Cost . . . 78
4.2.4 Column: Revenue . . . 78
4.2.5 Column: Profit . . . 78
4.2.6 Column: Peak Volume . . . 78
4.2.7 Column: Total Volume . . . 78
4.2.8 Column: Volume Capacity . . . 78
4.2.9 Column: Peak Weight . . . 78
4.2.10 Column: Total Weight . . . 78
4.2.11 Column: Weight Capacity . . . 78
4.2.12 Column: Number Of Loads . . . 78
4.2.13 Column: Transit Distance . . . 79
4.2.14 Column: Loaded Distance . . . 79
4.2.15 Column: Empty Distance . . . 79
4.2.16 Column: Actual Start Time . . . 79
4.2.17 Column: Actual Finish Time . . . 79
4.2.18 Column: Work Time . . . 79
4.2.19 Column: Allowed Work Time . . . 79
4.2.20 Column: Transit Time . . . 79
4.2.21 Column: Allowed Transit Time . . . 79
4.2.22 Column: Site Time . . . 79
4.2.23 Column: Load Time . . . 79
4.2.24 Column: Unload Time . . . 79
4.2.25 Column: Service Time . . . 80
4.2.26 Column: Idle Time . . . 80
4.2.27 Column: Break Time . . . 80
4.2.28 Column: Number Of Drops . . . 80
4.2.29 Column: Number Of Orders . . . 80
4.2.30 Column: Average Transit Speed . . . 80
4.2.31 Column: Pickup Soft Time Window Early . . . 80
4.2.32 Column: Pickup Soft Time Window Late . . . 80
4.2.33 Column: Delivery Soft Time Window Early . . . 80
4.2.34 Column: Delivery Soft Time Window Late . . . 80
4.3 Sheet: Load Summary . . . 81
4.3.1 Column: Vehicle . . . 81
4.3.2 Column: Load Number . . . 81
4.3.3 Column: Peak Volume . . . 81
4.3.4 Column: Volume Capacity . . . 81
4.3.5 Column: Peak Weight . . . 81
4.3.6 Column: Weight Capacity . . . 81
4.3.7 Column: Loaded Distance . . . 81
4.3.8 Column: Empty Distance Before . . . 81
4.3.9 Column: Empty Distance After . . . 81
4.3.10 Column: Work Time . . . 81
4.3.11 Column: Transit Time . . . 81
4.3.12 Column: Site Time . . . 82
4.3.13 Column: Load Time . . . 82
4.3.14 Column: Unload Time . . . 82
4.3.15 Column: Service Time . . . 82
4.3.16 Column: Idle Time . . . 82
4.3.17 Column: Break Time . . . 82
4.3.18 Column: Number Of Drops . . . 82
4.3.19 Column: Average Transit Speed . . . 82

4.3.20 Column: Pickup Soft Time Window Early . . . 82
4.3.21 Column: Pickup Soft Time Window Late . . . 82
4.3.22 Column: Delivery Soft Time Window Early . . . 82
4.3.23 Column: Delivery Soft Time Window Late . . . 83
4.4 Sheet: Solution Summary . . . 84
4.4.1 Scenario Name . . . 84
4.4.2 Total Iterations . . . 84
4.4.3 Total Cost . . . 84
4.4.4 Total Revenue . . . 84
4.4.5 Total Profit . . . 84
4.4.6 Total Distance . . . 84
4.4.7 Loaded Distance . . . 84
4.4.8 Empty Distance . . . 84
4.4.9 Average Transit Speed (km/h) . . . 84
4.4.10 Average Transit Speed (mph) . . . 84
4.4.11 Loads . . . 84
4.4.12 Km Per Load . . . 85
4.4.13 Miles Per Load . . . 85
4.4.14 Hours Per Load . . . 85
4.4.15 Drops Per Load . . . 85
4.4.16 Work Time . . . 85
4.4.17 Transit Time . . . 85
4.4.18 Break Time . . . 85
4.4.19 Idle Time . . . 85
4.4.20 Total Weight . . . 85
4.4.21 Total Volume . . . 85
4.4.22 Assigned Orders . . . 85
4.4.23 Fleet Usage . . . 85
4.4.24 Column: Pickup Soft Time Window Early . . . 86
4.4.25 Column: Pickup Soft Time Window Late . . . 86
4.4.26 Column: Delivery Soft Time Window Early . . . 86
4.4.27 Column: Delivery Soft Time Window Late . . . 86
4.5 Sheet: Unassigned Orders . . . 87
4.5.1 Column: Order . . . 87
4.5.2 Column: Customer Name . . . 87
4.5.3 Column: Weight . . . 87
4.5.4 Column: Volume . . . 87
4.5.5 Column: Pickup Location . . . 87
4.5.6 Column: Delivery Location . . . 87
4.5.7 Column: Reason . . . 87
4.6 Sheet: Unassigned Compatibility . . . 88
4.6.1 Column `A`: Order . . . 88
4.6.2 Row `1`: Vehicle . . . 88
4.6.3 All other cells (except `A1`) . . . 88
4.7 Sheet: Replan Changes . . . 89
4.7.1 Column: Order . . . 89
4.7.2 Column: Weight . . . 89
4.7.3 Column: Volume . . . 89
4.7.4 Column: Previous Vehicle . . . 89
4.7.5 Column: New Vehicle . . . 89
4.7.6 Column: Previous Pickup Time . . . 89
4.7.7 Column: New Pickup Time . . . 89
4.7.8 Column: Previous Delivery Time . . . 89
4.7.9 Column: New Delivery Time . . . 89
4.8 Sheet: Exclusive Order Summary . . . 90
4.8.1 Column: Assigned . . . 90
4.8.2 Column: Unassigned . . . 90

5 JSON Input **91**
5.1 Scenario Object . . . 91
5.2 General Object . . . 94
5.3 Unassigned Penalty Scheme Codes . . . 97
5.4 Order Splitting Strategy Codes . . . 98
5.5 Load Order Codes . . . 99
5.6 Excel Report Configuration . . . 100
5.7 Location Object . . . 101
5.8 Order Object . . . 103
5.9 Vehicle Object . . . 106
5.10 Standard Break Object . . . 111
5.11 Simple Break Object . . . 112
5.12 Compartment Object . . . 113
5.13 Time Window Object . . . 114
5.14 Soft Time Window Object . . . 115
5.15 Attribute Pair Object . . . 116
5.16 Rush Hour Object . . . 117
5.17 Iteration Batch Object . . . 118
5.18 Vehicle Cost Object . . . 119
5.19 Vehicle Preference Object . . . 120
5.20 Order Precedence Object . . . 121
5.21 Grouped Pickup Object . . . 122
5.22 Locked Order Object . . . 123
5.23 Exclusive Order Object . . . 124
5.24 Matrix Override Object . . . 125

6 JSON Output **126**
6.1 Solution Object . . . 126
6.2 Route Object . . . 129
6.3 Stop Object . . . 131
6.4 Stop Type Codes . . . 133
6.5 Route KPI Object . . . 134
6.6 Load KPI Object . . . 136
6.7 Unassigned Reason Object . . . 138
6.8 Exclusive Order Summary Object . . . 139

Appendices **140**

A JSON Input Example **140**

B JSON Output Example **143**

C Unsupported Excel Functions **147**

1 Overview

This document specifies the input and output formats of the Opturion Dynamic Transport Optimiser (DTO). The input and output to the DTO can be given either as a Microsoft Excel workbook (DTO-Excel, see Sections 3 and 4) or as a JSON document (DTO-JSON, see Sections 5 and 6).

The DTO-Excel format is intended for use as a standalone modelling interface.

The DTO-JSON interface is intended for use when interacting with the DTO via a web service.

Examples of data input and output in JSON format are provided in Appendices A and B.

2 Data Model

This section provides an overview of the major concepts in the DTO data model.

2.1 Scenarios

An input to the DTO takes the form of a set of parameters, locations, vehicles and orders, called a *scenario*. A scenario is a representation of the transport optimisation problem you wish to solve. It includes a description of all the orders that need servicing, all available vehicles and their properties, location data and any order-vehicle restrictions as well as other optional data which is described in the following sections.

2.2 Solutions

A *solution* is a plan created by the optimiser. It consists of a schedule that assigns orders to vehicles.

2.3 Identifiers

Entities such as orders, locations and vehicles within a scenario must have a unique *identifier* (Id) that is used to refer to them.

Identifiers must contain at least one non-whitespace character and must not contain the semicolon (`;`) character. They are not case or whitespace sensitive.

For example, "`DEPOT2`", "`depot2`" and "`Depot 2 `" will all be treated as identifying the same entity by the optimiser because they only differ by case or the amount of whitespace they contain.

Even though they may be composed of digits, identifiers are not numeric. For example, the optimiser will treat "`1`" and "`001`" as being distinct identifiers.

(Users of the DTO-Excel format should be aware that there are Excel-specific restrictions on the specification of identifiers; see Section 3.1 for details.)

All references to an entity are made using its identifier.

2.4 Attributes

Attributes are labels that are attached to entities like orders, locations and vehicles to describe their properties. Attribute names, like identifiers, must contain at least one non-whitespace character and must not contain the semicolon (`;`) character. They are not case or whitespace sensitive.

Attributes may be used to define relationships between the entities in a scenario. (See Section 2.17 for further details).

Note that an entity's identifier is automatically considered to be one of its attributes.

2.5 Weights and Volumes

Weights and volumes in a scenario can be specified in arbitrary units, however, the selected units must be used consistently throughout the scenario. Mixing units, giving order weights in kilograms and vehicle capacities in tons for example, will lead to errors.

2.6 Time

Time is specified in *HH:MM* format.

The hour component of a time may extend beyond `23`, allowing times beyond a single day to be specified. For example, `36:00` would be midday on the second day.

2.7 Costs and Hidden Costs

The optimiser deals with two kinds of cost: *actual* (i.e. monetary) costs and *hidden* costs.

Vehicles, orders and locations can have both actual and hidden costs associated with them.

Hidden costs are considered by the optimiser when making decisions, but they do *not* contribute to the actual costs reported by the optimiser. Hidden costs are nominally measured in the same units as actual costs.

Hidden costs have many uses, for example they can be used to cause the optimiser to prefer the use of a particular subset of the fleet or to affect the "shape" of routes.

2.8 Global Parameters

2.8.1 Iterations

The number of iterations the optimiser will run for. Each iteration may result in an improved solution and hence a higher number of iterations increases the probability of finding a better solution. This comes at the cost of increasing the duration the optimiser will run for.

2.8.2 Batched Loads

Batched loads is a scheduling constraint that enforces a strict "Pickup - Delivery" cycle where once a vehicle begins making deliveries, it cannot pick up new orders until it has delivered all its current orders. (See Section 2.22 for further information.)

2.8.3 Colocated Pickups

Colocated pickups require all orders in a load to originate from the same location.

2.8.4 Sticky Deliveries

When enabled, *sticky deliveries* encourages the optimiser to assign all orders for a specific location to the same vehicle. The intention of this is to reduce the number of different vehicles that make deliveries at a location.

2.8.5 Scaling Factors

A scaling factor can be applied to speeds, costs, weights and volumes in the scenario. This allows the user to increase the effect that a particular variable has on the optimiser. A value of `0` means that no scaling is applied. The scaling factors include speed scale, cost scale, weight scale and volume scale.

2.8.6 Geofencing

The optimiser can be restricted to only consider orders and vehicles that are located within a specific geographic area. This area is called a *geofence* and it takes the form of a bounding box that is defined by the latitudes of its southern and northern boundaries and by the longitudes of its western and eastern boundaries.

The geofence can operate in one of two modes: *hard* and *soft*.

If the geofence is in *hard* mode, then the optimiser will report an error for any order that has pickup or delivery locations outside the geofence. It will also report an error for any vehicle that has start or finish locations outside of the geofence. In the event of such errors being detected, optimisation will *not* proceed.

If the geofence is in *soft* mode, then the optimiser will not assign any order whose pickup or delivery location is outside the geofence. It will not use any vehicle whose start or finish location is outside the geofence. In this mode optimisation will proceed using orders and vehicles that are within the geofence.

2.8.7 Maximum Neighbourhood Size

The maximum size of the neighbourhood for the optimiser. At each iteration of the optimiser, the solution is unpacked and changed to find a new solution with the neighbourhood being the amount to unpack. Smaller neighbourhoods are quicker to change but may result in worse quality solutions, while larger neighbourhoods need longer running time but may lead to better quality solutions. Note that the maximum neighbourhood size must be at least ten.

2.8.8 Preferred-Vehicle Rewards

If preferred vehicles are specified for an order, then there are three types of rewards that can be specified.

1. **Preferred-Vehicle Fixed Reward**

   This is a fixed reward for assigning the order to one of its preferred vehicles.

2. **Preferred-Vehicle Weight Reward**

   This reward is multiplied by the weight of the order if it is assigned to one of its preferred vehicles. It can be used to give a higher priority to assigning heavier orders to their preferred vehicles.

3. **Preferred-Vehicle Volume Reward**

   This reward is multiplied by the volume of the order if it is assigned to one of its preferred vehicles. It can be used to give a higher priority to assigning larger orders to their preferred vehicles.

2.8.9 Unassigned Penalty Scheme

Each order that is *not* assigned by the optimiser incurs a hidden cost, called the *unassigned penalty*. Typically, you want the optimiser to assign all the orders in the input. Consequently, the unassigned penalty is usually very large relative to any other costs (monetary or hidden) in the scenario. There are multiple different methods for setting the value of the unassigned penalty for the orders in a scenario. Each method is called an *unassigned penalty scheme*.

The following unassigned penalty schemes are supported:

1. **Fixed**

   There is a fixed unassigned penalty for each order. You can set the size of this penalty in the input. This scheme is the default.

2. **Legacy**

   The unassigned penalty for each order is `4,500,000` divided by the number of orders in the scenario. This scheme is deprecated and will be deleted in a future release.

2.8.10 Order Splitting Strategy

Some orders may be too large or heavy to fit on a *compatible* vehicle in the fleet. *Order splitting* lets the optimiser automatically split up such orders into smaller orders that will fit on a vehicle. How this occurs depends upon the composition of the orders and is controlled by the optimiser's *order splitting strategy*.

Currently, the following order splitting strategies are supported:

1. **None**

   No order splitting is allowed.

2. **Split By Weight**

   Orders will be split into multiple orders, each of which will have weight equal to the maximum weight capacity of the largest compatible vehicle. Any remaining weight will be allocated to an additional order.

   Note: All orders *must* have zero volume to use this scheme.

3. **Split By Volume**

   Orders will be split into multiple orders, each of which will have volume equal to the maximum volume capacity of the largest compatible vehicle. Any remaining volume will be allocated to an additional order.

   Note: All orders *must* have zero weight to use this scheme.

Example:

Consider the case where we have an order with identifier `Order_1` and volume `24`. The largest vehicle in the fleet that is compatible with `Order_1` has a maximum volume capacity of `10`. If splitting by volume is enabled, then this order will be split into the three separate orders shown in Table 1. Each of the split orders is assigned a new identifier that consists of the identifier of the original order, followed by an underscore ("_") and then an integer "split index".

Id	Volume
Order_1_0	10
Order_1_1	10
Order_1_2	4

Table 1: Order Splitting Example

2.8.11 Heavy Order Threshold

If a vehicle is using the weight category *load order*, then any order with weight greater than or equal to this value will be considered a *heavy* order.

2.9 Replanning

Sometimes it is necessary to update or extend a solution generated by the optimiser. Reasons for this include:

- Orders being added.
- Orders being cancelled.
- Order details (e.g. weights, volumes, time windows) changing.
- Vehicles being added to the fleet.
- Vehicles being removed from the fleet.
- Vehicle / driver availability changing.

When this happens, it is often desirable to maintain as much of the existing solution as practicable. For example, when loads are being assembled at the loading docks of a warehouse, altering the original plan too much may unduly affect warehouse operations.

This usage of the optimiser is called *replanning*. When replanning, an updated scenario is extended with information about the assignments and sequencing in a previous solution. The optimiser can be configured to minimise the differences between the solution it generates and the previous solution while incorporating any changes.

The following global parameters control the degree and manner to which deviations from the existing solution can occur.

1. **Previous-Vehicle Fixed Reward**

   This parameter gives the strength of the preference for orders to remain on their previously assigned vehicles.

2. **Previous-Vehicle Weight Reward**

   This parameter gives the strength of the preference for heavier orders to remain on their previously assigned vehicles.

3. **Previous-Vehicle Volume Reward**

   This parameter gives the strength of the preference for larger orders to remain on their previously assigned vehicles.

Each of the above rewards is a form of hidden cost and their effect is cumulative.

2.10 Locations

1. **Identifier**

   Each location must have a unique identifier. This identifier acts as a reference to the location when referred to from other data (e.g. orders).

2. **Name**

   Each location may have a name associated with it, for example a store or warehouse name.

3. **Address**

   Each location may have a street address associated with it.

4. **Coordinates**

   Each location *must* be provided with its geographic coordinates specified as latitude and longitude in decimal degrees.

5. **Site Time**

   Site time is the time incurred by the vehicle when it arrives at the location. It is *only* incurred when the vehicle arrives at a location. Multiple consecutive pickups or deliveries at the *same* location without any intervening travel do not individually incur site time.

6. **Load Time**

   Load time is the time incurred by the vehicle when it visits the location for a pickup or pickups. It is incurred only once per sequence of consecutive pickups at the *same* location. If the vehicle also has a positive load time, then the vehicle's load time will override this value.

7. **Unload Time**

   Unload time is the time incurred by the vehicle when it visits the location for a delivery or deliveries. It is incurred only once per sequence of consecutive deliveries at the *same* location. If the vehicle also has a positive unload time, then the vehicle's unload time will override this value.

8. **Opening Time**

   The earliest time a vehicle can pickup or deliver at the location.

9. **Closing Time**

   The latest time a vehicle can pickup or deliver at the location.

10. **Time Windows**

    Time windows define spans of time when a vehicle can pickup or deliver at the location. Time windows may be a single span of time, such as 9:00-17:00 or multiple spans, such as 9:00-12:00 and 14:00-17:00.

11. **Operating Times**

    The *operating times* are times at which a vehicle can pickup or deliver at the location. Operating times are defined by the opening time, closing time and any time windows.

12. **Cost per Visit**

    This is a cost that is incurred whenever a vehicle visits the location.

13. **Hidden Cost per Visit**

    This is a hidden cost that is incurred whenever a vehicle visits the location. This is used in the computation of a solution but will not be seen in the final output.

14. **Loading Bays**

    The number of loading bays at a location limits how many vehicles can load or unload at the location at the same time. If an arriving vehicle exceeds this limit, then it will be required to wait until a loading bay becomes unoccupied. At locations with no loading bays, any number of vehicles can load or unload at the same time.

15. **Loading Bay Gap**

    This specifies the duration that must elapse between a vehicle leaving a loading bay and the arrival of the next vehicle at the same bay. This is used to model the time required to manoeuvre vehicles in and out of position.

2.11 Vehicles

2.11.1 Vehicle Costs

Vehicle costs are specified in arbitrary units. The units used must be consistent across the scenario.

1. **Cost per Use**

   This is a cost incurred whenever the vehicle is used (i.e. has orders assigned to it) in a solution.

2. **Hidden Cost per Use**

   The hidden version of the cost per use. This is used in the computation of a solution, but it will not be seen in the final output.

3. **Cost per Hour**

   This is the cost-per-hour for the vehicle.

4. **Hidden Cost per Hour**

   The hidden version of the cost per hour. This is used in the computation of a solution, but it will not be seen in the final output.

5. **Cost per Kilometre**

   This is the cost-per-kilometre for the vehicle.

6. **Hidden Cost per Kilometre**

   The hidden version of the cost per km.

7. **Cost per Mile**

   This is the cost-per-mile for the vehicle.

8. **Hidden Cost per Mile**

   The hidden version of the cost per mile.

9. **Cost per Load**

   This is a cost that is incurred for each load a vehicle has in the solution.

10. **Hidden Cost per Load**

    The hidden version of the cost per load. This is used in the computation of a solution, but it will not be seen in the final output.

Costs and hidden costs per distance may be specified using either kilometres or miles, but *not* both.

The following additional hidden costs (penalties) for vehicles are useful for controlling the "shape" of a vehicle's route.

1. Pickup segment penalties.
2. Delivery segment penalties.
3. Finish segment penalties.
4. Loaded volume distance penalties.
5. Loaded weight distance penalties.

2.11.2 Vehicle Locations

1. **Starting Location**

   This is the location of the vehicle at the start of its route.

   If this location differs from the first pickup location, then the vehicle is required to travel from the start location to the first pickup location. In such a case, the travel time and distance to the first pickup location are included in the solution.

   If no start location is specified, then the vehicle begins its route at the location of its first pickup.

2. **Finishing Location**

   This is a location to which the vehicle must return at the end of its route. The travel time and distance to return to this location are considered in the solution.

   If no finish location is specified, then the vehicle ends its route at the location of its last delivery.

2.11.3 Vehicle Times

1. **Earliest Start**

   This is the earliest time that a vehicle can start its route.

2. **Latest Start**

   This is the latest time that a vehicle can start its route.

3. **Latest Finish**

   This is the latest time that the vehicle can finish its route. It includes the time taken for the vehicle to return to its finish location.

4. **Maximum Drive Time**

   This is the maximum time that a vehicle can spend in transit during a route.

5. **Maximum Work Time**

   This is the maximum time allowed between the start of the route and its completion. This includes the time needed for a vehicle to return to its finish location at the end of its route.

6. **Load Time**

   Load time is the time incurred by the vehicle when it begins a sequence of pickups at a location. A vehicle's load time will override a location's load time if *both* the vehicle's and location's load time are greater than zero.

7. **Unload Time**

   Unload time is the time incurred by the vehicle when it begins a sequence of deliveries at a location. A vehicle's unload time will override a location's unload time if *both* the vehicle's and location's unload times are greater than zero.

2.11.4 Vehicle Capacities

Vehicles may have both a maximum weight capacity and maximum volume capacity. These are upper bounds on how much the vehicle can carry.

2.11.5 Vehicle Compartments

Vehicles may be divided into multiple *compartments*. Each compartment has its own capacities (e.g. maximum weight, maximum volume) and attributes. A *compartment scheme* is a list of compartments identified using a unique identifier. An individual compartment scheme can be applied to multiple vehicles.

2.11.6 Maximum Loads

If batched loads is enabled, then this is an upper bound on the number of loads the vehicle is allowed to do.

2.11.7 Maximum Drops per Load

If batched loads is enabled, then this is an upper bound on the number of locations a vehicle can deliver to per load.

2.11.8 Fatigue Management Scheme

This is an identifier that specifies the fatigue management rules that apply to the vehicle. Driver breaks inserted by the optimiser will satisfy all rules in the fatigue management scheme.

2.11.9 Loaded Breaks

A *loaded break* is a driver break that is scheduled while one or more orders are loaded on the vehicle. The optimiser can optionally prevent loaded breaks from being scheduled (i.e. driver breaks will only occur while the vehicle is not loaded).

2.11.10 Vehicle Speed Scale

As all vehicles use the same time and distance matrices in computation of transit times and distance, applying a scaling factor allows the user to account for vehicles with varying speed. For example, if a vehicle was known to be 30% slower than the average vehicle in the fleet, a scaling factor of `0.7` (70%) could be applied. Similarly, if a vehicle was known to be 10% faster, a scaling factor of `1.10` (110%) could be applied.

2.11.11 Vehicle Types

To avoid unnecessary repetition when defining vehicles, DTO-Excel (only) allows for the definition of vehicle types. All vehicles of the same type share the properties defined for that type, such as maximum weight, speed scale etc. Note that defining both a vehicle type and vehicle properties in the Fleet sheet is not allowed.

2.11.12 Load Ordering

A vehicle's *load ordering* is an additional constraint on the pattern of pickups and deliveries made by the vehicle.

The following *load orders* are supported:

- **None:** There is no load order constraint.
- **LIFO:** A last-in-first-out ordering. This can be used to model vehicles which are loaded from the back.

  Example:

Stop Number	Stop Type	Order
1	PICKUP	Order1
2	PICKUP	Order2
3	PICKUP	Order3
4	DELIVERY	Order3
5	DELIVERY	Order2
6	DELIVERY	Order1

- **Weight Category:** The load ordering is weight sensitive where heavier items are picked up first but delivered last. The definition of a *heavy* item is specified using the heavy order threshold in the scenario.

  Example:

Stop Number	Stop Type	Order	Weight Category
1	PICKUP	Order1	Heavy
2	PICKUP	Order2	Heavy
3	PICKUP	Order3	Light
4	PICKUP	Order4	Light
5	PICKUP	Order5	Light
6	DELIVERY	Order4	Light
7	DELIVERY	Order3	Light
8	DELIVERY	Order5	Light
9	DELIVERY	Order1	Heavy
10	DELIVERY	Order2	Heavy

2.12 Orders

1. **Locations**

   The pickup and delivery locations must be given for each order and must match a location identifier defined in the set of locations. (See Section 2.10.)

2. **Times**

   (a) **Earliest and Latest Times**

   The earliest pickup time is the earliest time that the order will be available at its pickup location. The earliest delivery time is the earliest time the same order is able to be delivered. This is analogous for latest times.

   (b) **Time Windows**

   The times when an order can be picked up and delivered can also be specified by time windows. If both earliest and latest times and time windows are specified, then *both* must be satisfied for the order to be picked up or delivered.

   (c) **Service Time**

   Each pickup and delivery may have an amount of service time associated with it. This is the amount of time incurred by the vehicle when the order is picked up or delivered. This time is *in addition* to any site, load or unload time incurred because of the location or vehicle.

3. **Soft Time Window Penalties**

   If an order has soft time windows, then both fixed and variable penalties are incurred if these time windows are missed.

   (a) **Fixed Penalty**

   A constant penalty that is incurred if the soft time windows are missed. Note that if multiple soft time windows are defined, this penalty is still only incurred once.

   (b) **Variable Penalty**

   A penalty incurred per minute the time window is missed. If multiple time windows are present, then the penalty is calculated based on the nearest soft time window boundary.

4. **Maximum Duration**

   Orders can optionally have a *maximum duration* specified. The optimiser will not generate solutions that assign such orders in a way that the time needed to service them exceeds their maximum duration. (The *duration* of an order is defined to be the time that elapses from when its pickup commences up until the time when its delivery is complete.)

   If the minimum service time needed by an order exceeds its maximum duration, then the order cannot be assigned by the optimiser.

5. **Weight and Volume**

   The weight and volume of the order are given in arbitrary units which must be consistent with the respective vehicle capacities previously defined. For example, the volume may be measured in cubic metres, number of pallets or roll cages.

6. **Assign Cost**

   This is a monetary cost associated with servicing the order. Negative costs can also be used; these cause the optimiser to prefer certain orders. (A negative assign cost leads the optimiser to the same solutions as an equivalent revenue; the difference is that using revenue keeps the cost and revenue figures separate in the output.)

7. **Hidden Assign Cost**

   The hidden version of the assign cost. While this is used in the computation of a solution, it will not be seen in the final output.

8. **Revenue**

   The revenue earned for servicing this order. The optimiser will prefer orders with higher revenue. (See also Assign Cost.)

9. **Vehicle-Specific Assign Cost**

   An order can have costs associated with it that are only incurred when the order is assigned to a specific vehicle or set of vehicles. These costs are incurred in addition to any assign cost for the order.

10. **Vehicle-Specific Hidden Assign Cost**

    An order can have hidden costs associated with it that are only incurred when the order is assigned to a specific vehicle or set of vehicles. These hidden costs are incurred in addition to any hidden assign cost for the order.

11. **Preferred Vehicles**

    An order can have a list of preferred vehicles. A reward is obtained on assigning the order to one of its preferred vehicles. This reward is calculated using a fixed reward, a volume dependent reward and a weight dependent reward. See Global Parameters for an explanation of these rewards.

12. **Order Precedences**

    Two orders can be associated with a precedence. This requires that the *delivery* of the first order must be completed before the *pickup* of the second. An optional duration that must elapse between the delivery and the pickup can also be specified. This duration is called the *gap*.

    Precedences are often used to model cross-docking, where the first leg of the order must be delivered before the second leg can commence.

2.13 Fatigue Management

A fatigue management scheme is a set of rules that specify the number of breaks of a certain duration that must occur within a given time interval (period).

There are two fatigue management formats.

2.13.1 Standard Fatigue Management

Each *standard* rule consists of the following:

1. **Number of Breaks**

   The number of breaks required by the rule.

2. **Duration of Break**

   The duration of the break.

3. **Interval of Break**

   The time interval (period) in which the break must occur.

Example A fatigue management scheme may have the rule: A single `15` minute break every `2` hours. Here, the number of breaks is `1`, the duration of the break is `15` minutes and the interval in which the break must occur is `2` hours.

2.13.2 Simple Fatigue Management

Each *simple* rule consists of the following:

1. **Duration of Break**

   The duration of the break.

2. **Earliest Start**

   The earliest time after the vehicle's start that the break can commence.

3. **Preferred Start**

   The preferred time after the vehicle's start that the break can commence.

4. **Latest Start**

   The latest time after the vehicle's start that the break can commence.

2.14 Iteration Schemes

Iteration schemes provide finer-grained control over how the optimiser explores its solution space. An iteration scheme allows a single invocation of the optimiser to be split into a sequence of "batches", where each batch runs for a specified number of iterations with a specified maximum neighbourhood size. In the absence of an iteration scheme the optimiser will run with a single fixed maximum neighbourhood size for all the iterations.

An example iteration scheme is:

- `1000` iterations with a maximum neighbourhood size of `200`.
- `2000` iterations with a maximum neighbourhood size of `150`.
- `4000` iterations with a maximum neighbourhood size of `100`.

In this scheme, the optimiser will run for a total of `7000` iterations. For the first `1000` iterations the optimiser will use a maximum neighbourhood size of `200`; for the next `2000` it will use a maximum neighbourhood size of `150`; for the last `4000` it will use a maximum neighbourhood size of `100`.

Using an iteration scheme *may* improve performance as the optimiser is allowed a greater degree of freedom in the choices it makes at the beginning of a batch. Having multiple batches effectively increases the amount of time that the optimiser is afforded this greater degree of freedom.

Reducing the maximum neighbourhood size in later batches (as in the above example) reduces the likelihood of the optimiser straying too far from a good solution and failing to find any further improvements.

2.15 Time Windows

Time windows represent a period of time within which either an order can be serviced or a location is open depending on the context.

Example: a location is open between 6:00-12:00, closed for lunch and open again between 13:00-17:00. Figure 1 shows how to represent this using time windows.

```
[
  {
    "start": "06:00",
    "end": "12:00"
  },
  {
    "start": "13:00",
    "end": "17:00"
  }
]
```

	A	B	C
1	**ID**	**Start**	**End**
2	DAY	06:00	12:00
3	DAY	13:00	17:00

Figure 1: Example Time Window in JSON format (top) and Excel format (bottom).

2.15.1 Arrival Only in Time Window

Time windows have two modes of operation. The first ensures that every pickup or delivery occurs within the specified time window. This is the default behaviour. The second is called "Arrival Only in Time Window". This mode of operation makes it so that the time window is satisfied for each pickup (or delivery) if the time window was satisfied upon the arrival of the vehicle at the location. This second mode of operation is only enabled when Batched Loads is also enabled.

	A	B	C
1	Stop	Time	Time Window
2	START	9:00	
3	PICKUP	9:20	9:00 - 10:00
4	PICKUP	9:25	9:00 - 9:20
5	PICKUP	9:30	8:30 - 9:15
6	DELIVERY	11:00	9:00 - 12:00
7	BREAK	11:05	
8	DELIVERY	11:35	9:00 - 11:30
9	DELIVERY	11:40	9:00 - 11:30
10	FINISH	12:00	

Figure 2: Example route with time windows to illustrate "Arrival Only in Time Window" behaviour. All pickups and all deliveries are at the same location. Time windows shaded green are satisfied normally. If "Arrival Only in Time Window" is enabled, then time windows shaded yellow are also satisfied, since the arrival stop is within the time window. The time window shaded red is violated with both schemes, since neither the stop nor the arrival stop are within the time window.

2.16 Soft Time Windows

Soft time windows apply only to orders. They represent a preferred time of service (i.e. pickup or delivery). The optimiser may schedule pickups or deliveries outside of any soft time windows but will incur a penalty for doing so.

A soft time window penalty has two components: a *fixed* component that is incurred if the soft time window is missed and a *variable* component that is a per-minute rate based on the number of minutes from the nearest time window boundary. The soft time window penalty acts as an additional form of hidden cost.

Example Assume the time windows in Figure 1 are delivery soft time windows and we have `200` and `5` as the fixed and variable penalty components respectively. In this case the hard time window exists from 6:00-17:00. If the order was delivered at 12:20, then we would incur a fixed penalty of `200` and a variable penalty of $5 \times 20 = 100$ for a total penalty of `300`.

2.17 Compatibility

Compatibility relationships are used to restrict what decisions the optimiser may make when generating a solution. The following kinds of compatibility relationships are supported:

- **Order-vehicle compatibility**: an order may only be assigned to a vehicle with which it is compatible.
- **Order-compartment compatibility**: an order may only be loaded in a compartment with which it is compatible.
- **Order-order compatibility**: orders may only be simultaneously loaded into a compartment if they are compatible.
- **Load Order-order compatibility**: orders may only be simultaneously loaded on to a vehicle if they are compatible.
- **Location-vehicle compatibility**: a vehicle must only pickup or deliver orders to locations with which it is compatible.

Compatibility relationships are specified using attributes. As an example, consider a fleet that consists of both refrigerated and non-refrigerated vehicles. To restrict an order that requires refrigeration to be assigned to only a refrigerated vehicle, we can label the order with the attribute REFRIGERATED and also label all the refrigerated vehicles with the attribute REFRIGERATED. We can then specify an *order-vehicle* compatibility relationship such that orders with the attribute REFRIGERATED must only be assigned to vehicles with the attribute REFRIGERATED.

Order\Vehicle	REFRIGERATED	TALL	LONG	VEHICLE1
REFRIGERATED	true			
TALL		true	true	
ORDER7				true

Table 2: Order-Vehicle Compatibility.

Table 2 is an example of an *order-vehicle* compatibility relationship, where `true` means that the attribute pair is compatible and an empty cell means that the attribute pair is *not* compatible.

Table 2 is interpreted as follows: for orders with the attribute REFRIGERATED, the compatible vehicles must have the attribute REFRIGERATED; for orders with the attribute TALL, the compatible vehicles must have **either** the attribute TALL or the attribute LONG; for the orders with the attribute ORDER7, the compatible vehicles must have the attribute VEHICLE1. For the last case, if ORDER7 is one of the order identifiers and VEHICLE1 is one of the vehicle identifiers, then this will restrict ORDER7 to only be compatible with VEHICLE1. If ORDER7 has other attributes that VEHICLE1 is not compatible with, then this means that the order cannot be assigned to any vehicle. (See **Conjunction Rule** below).

Three rules are applied when interpreting attributes and their compatibility. Here we use *order-vehicle* compatibility to illustrate them.

1. **Null Rule**: An order attribute that is not part of a *order-vehicle* compatibility relationship imposes no restriction on which vehicle the order can be assigned to. We can ignore this attribute or, equivalently, consider it compatible with all vehicle attributes.
2. **Disjunction Rule**: If an order attribute has multiple compatible vehicle attributes, then any vehicle having *one* of the attributes is a candidate compatible vehicle.
3. **Conjunction Rule**: If an order has multiple attributes, then a compatible vehicle for the order must have a compatible attribute for each of the order's attributes.

These rules are applied in only one direction: attributes are interpreted as requirements for orders and features for vehicles. An order does not need to have a compatible attribute for each of a vehicle's attributes to be compatible with that vehicle. For example, in Table 2, an order that has only one attribute, REFRIGERATED, is compatible with a vehicle that has attributes REFRIGERATED and LONG. These rules apply similarly to *order-compartment* and *location-vehicle* compatibility.

For *order-order* and *load order-order* compatibility, these rules must be applied in *both* directions, that is: for two compatible orders, each of them must have a compatible attribute for each attribute of the other order.

2.18 Locked Orders

An order may be optionally *locked* to a vehicle. This has two effects:

1. If the order is assigned, then it can only be assigned to the vehicle that it is locked to.
2. The order will be prioritised for assignment by the optimiser over non-locked orders.

Locked orders must satisfy *all* other constraints they are subject to. As such, locking an order does *not* guarantee that an order will be assigned by the optimiser.

2.19 Exclusive Orders

Exclusive orders are sets of orders from which at most one order can be assigned. They can be used to model situations where order have multiple possible pickup or delivery locations.

Exclusive orders must satisfy *all* other constraints they are subject to. As such, there is *no* guarantee that one of the orders will be assigned by the optimiser.

2.20 Preloading

Preloading is when a vehicle starts its plan already loaded and can depart from its start location immediately.

The following conditions are required to allow a vehicle to preload at a location:

- The location allows preloading.
- The vehicle allows preloading.
- The vehicle's *start location* is at the location.

When preloading occurs, no pickup service times and load time are incurred for the initial sequence of pickups. Site time *is* incurred as normal. This is because site time is intended for modelling activities that the driver needs to undertake regardless of whether the vehicle is preloaded or not (e.g. paperwork, briefings or safety inspections).

The number of vehicles that can be preloaded at a location is not restricted by the number of loading bays at that location.

2.21 Stops

In a solution, a vehicle's route consists of a sequence of stops, each with a time-of-arrival and time-of-departure. The different stop types are:

1. `START`: Denotes the start of the route.
2. `FINISH`: Denotes the end of the route.
3. `PICKUP`: Denotes an order pickup.
4. `DELIVERY`: Denotes an order delivery.
5. `BREAK`: Denotes a break for the vehicle.

A route must begin with a `START` stop and end with a `FINISH` stop. These are the only positions in the route where `START` and `FINISH` stops can appear. Every `PICKUP` stop in the route must have a matching `DELIVERY` stop.

2.22 Loads

A *load* is defined to be a sequence of `PICKUP` stops followed by a sequence of `DELIVERY` stops such that once the first `DELIVERY` stop occurs, no further `PICKUP` stops can occur until the vehicle is empty.

Example:

Stop Number	Load	Not a Load
1	`PICKUP`	`PICKUP`
2	`PICKUP`	`PICKUP`
3	`PICKUP`	`DELIVERY`
4	`DELIVERY`	`PICKUP`
5	`DELIVERY`	`DELIVERY`
6	`DELIVERY`	`DELIVERY`

2.23 Solution Times

In the solution generated by the optimiser, we have the following notions of times:

1. **Work Time**

   In the vehicle summary, this is the time between leaving the `START` stop and arriving at the `FINISH` stop in a vehicle's route. A vehicle's work time is the total time taken to complete the route, while a load's work time is the total time taken to complete the load.

2. **Site Time**

   A vehicle's site time is the sum of all site time incurred over the entire route, while a load's site time is the sum of all site time incurred during the load.

3. **Load Time**

   A vehicle's load time is the sum of all load times incurred over the entire route. If batched loads are enabled, then this measure is analogous for each of the vehicle's loads.

4. **Unload Time**

   A vehicle's unload time is the sum of all unload time incurred over the entire route. If batched loads are enabled, then this measure is analogous for each of the vehicle's loads.

5. **Service Time**

   A vehicle's service time is the sum of all service time incurred over the entire route. If batched loads are enabled, then this measure is analogous for each of the vehicle's loads.

6. **Idle Time**

   Idle time (wait time) is incurred by a vehicle if it arrives at a location outside of the location's time windows or if it arrives outside of the order's time windows. A vehicle's idle time is the sum of all idle time incurred over the entire route. If batched loads are enabled, then this measure is analogous for each of the vehicle's loads.

7. **Break Time**

   Break time is incurred if the vehicle has a fatigue management scheme specified (See Section 2.13). A vehicle's break time is the sum of all break time incurred over the entire route. If batched loads are enabled, then this measure is analogous for each of the vehicle's loads.

2.24 Unassigned Orders

If the solution generated by the optimiser contains orders that are not assigned to a vehicle, then the optimiser will do an analysis to try to determine the reasons for those orders not being assigned.

Examples of such reasons include:

- Orders not being compatible with any of the vehicles in the fleet.
- Orders have a weight or volume demand that exceeds the maximum vehicle weight or volume capacity of any vehicle in the fleet.
- Orders that need to be picked up or delivered *before* the start time of any of the vehicles in the fleet.
- Orders that require more time to complete than the maximum working time of any vehicle in the fleet.

Note that the focus of the analysis is on finding reasons that are likely to indicate errors or omissions in the input data. It is possible that the analysis will not find any definite reason for an order being unassigned.

2.25 Grouped Pickup

A *grouped pickup* is an optional constraint which can be imposed to enforce orders that must be picked up together. No order can be part of multiple sets of grouped orders and every order within a group *must* share a pickup location.

2.26 Pickup Segments

A *pickup segment* is defined to be any two consecutive `PICKUP` stops in a route. (A route or load with only a single `PICKUP` stop has no pickup segments.)

The optimiser can optionally apply a penalty based on the pickup segments in a route. This penalty has two components: a *fixed* component based on the number of pickup segments in the route and a *variable* component proportional to the transit distance along the pickup segments. These penalties take the form of an additional hidden cost.

The pickup segment penalty is useful for "shaping" the route. The fixed component of the penalty may be used to reduce the number of pickup stops per load, while the variable component may be used to encourage the optimiser to cluster pickups more closely together.

The pickup segment penalty is only applied if the Batched Loads parameter is enabled.

2.27 Delivery Segments

A *delivery segment* is defined to be any two consecutive `DELIVERY` stops in a route. (A route or load with only a single `DELIVERY` stop has no delivery segments.)

The optimiser can optionally apply a penalty based on the delivery segments in a route. This penalty has two components: a *fixed* component based on the number of delivery segments in the route and a *variable* component proportional to the transit distance along the delivery segments. These penalties take the form of an additional hidden cost.

The delivery segment penalty is useful for "shaping" the route. The fixed component of the penalty may be used to reduce the number of delivery stops per load, while the variable component may be used to encourage the optimiser to cluster deliveries more closely together.

The delivery segment penalty is only applied if the Batched Loads parameter is enabled.

2.28 Finish Segments

The *finish segment* of a route is defined as the part of the route from the last `DELIVERY` stop to the `FINISH` stop. There may be intervening `BREAK` stops between these two stops.

The optimiser can optionally apply a penalty based on the transit time or distance of the finish segment of a route. These penalties take the form of an additional hidden cost.

Penalties applied to the finish segment are used to tell the optimiser to prefer solutions in which vehicles end their routes "close" to their finish locations.

2.29 Loaded Distance Penalties

A *loaded distance penalty* is a hidden cost that is scaled by both the distance travelled by a vehicle and the amount loaded on the vehicle while travelling.

More precisely, the loaded distance penalty for a vehicle over a single travel segment is defined as:

$$TransitDistance \times LoadedQuantity \times PenaltyFactor$$

where $TransitDistance$ is the distance travelled in that segment (in either kilometres or miles), $LoadedQuantity$ is the amount loaded on the vehicle over that segment and $PenaltyFactor$ is a user-supplied value. The loaded distance penalty of a vehicle's route is the sum of the loaded distance penalties over all its individual travel segments.

Loaded distance penalties can be applied to either the weight loaded on the vehicle, the volume loaded on the vehicle or both.

2.30 Split Solving

Split solve is a divide-and-conquer strategy for the solver. Instead of solving one large scenario as a single optimisation problem, it partitions the scenario into several independent sub-scenarios, solves each one independently, and then combines the sub-solutions into one final solution.

Partitioning is done via time-window overlap. Orders and vehicles are grouped into categories whose maximal time windows overlap; groups that don't overlap in time end up in separate sub-scenarios, because they can never share a vehicle's route.

Split solve may be beneficial primarily due to the reduction in combinatorial complexity. A scenario which is twice as large may take more than twice the time to solve. Partitioning a large scenario into independent sub-scenarios means each solver searches a much smaller space, and the sum of these small searches finishes far faster than one search over the whole. Additionally, the sub-scenarios may be solved concurrently, so there may be additional wall-clock gain in addition to the search-space gain.

2.30.1 Example

A regional carrier runs a fleet doing two disjoint shifts. Consider a day-plus-night operation: a morning depot has 40 delivery orders all with time windows between 06:00 and 12:00, served by a set of vehicles whose shifts also sit in the morning. A separate batch of 40 orders has windows between 18:00 and 23:00, served by an evening fleet whose shifts sit in the evening. There's a midday gap where nothing overlaps.

Here the time windows partition cleanly into a morning sub-scenario and an evening sub-scenario with no intersection.

3 Microsoft Excel Input

This section describes the Microsoft Excel input format for the optimiser.

Excel input must be either an Office Open XML workbook (`.xlsx`) or macro-enabled workbook (`.xlsm`). Office Open XML binary workbooks (`.xlsb`) are *not* currently supported. Workbooks in the old Excel binary file format (`.xls`) are *not* supported. Workbooks in the Strict OOXML file format are *not* supported.

The workbook should contain the sheets described in the following sections. Those sheets that are specified as "required" must be present in the workbook. The workbook can contain sheets other than those recognised by the DTO-Excel format. Such extra sheets are ignored by the optimiser.

Unless otherwise noted, sheets recognised by the optimiser may contain columns other than those required by the DTO-Excel format. Such extra columns are ignored by the optimiser.

Columns recognised by the optimiser may appear in any order.

All sheets must have a non-empty first row.

The optimiser will evaluate any formulas used in the workbook when it is loaded. Formulas must *not* refer to sheets in other workbooks.

Not all Excel functions are supported by the optimiser; appendix C lists the unsupported Excel functions.

3.1 Identifiers in Excel

Identifiers and attributes in DTO-Excel may be provided as either text or numeric data. Identifiers given in text cells are taken *as-is* subject only to the rules concerning case and whitespace sensitivity. Identifiers given in numeric cells are *only* accepted by the optimiser if they are integer (i.e. their fractional part is zero). Formulas that evaluate to either text or (integer) numeric data may also be used to specify identifiers in Excel.

3.2 Times in Excel

In Excel, times are *stored* as decimal numbers, where `0` is `00:00:00` and `1` is `24:00:00`.

The integer component of the decimal represents a 24-hour period and is used to represent times greater than `23:59:59`. When using this representation, it is useful to think of the integer component as the day on which the time occurs, with times less than `24:00:00` occurring on *day zero*. For example, `1.50` would be `36:00`, or `12:00` on the second day.

In Excel, cells containing times may be formatted in various ways, for example, `hh:mm`, `d hh:mm` or `h:mm AM/PM`. Times may be given to the optimiser in any of the Excel time formats.

	hh:mm	**[hh];mm**	**dd/mm/yyyy hh:mm**	**d hh:mm**
0.5	12:00	12:00	00/01/1900 12:00	0 12:00
1.5	12:00	36:00	01/01/1900 12:00	1 12:00

Table 3: Excel time formatting examples.

Note here that if using the `dd/mm/yyyy hh:mm` format, *day zero* occurs on `00/01/1900`, which is Excel's earliest recognised date.

3.3 Sheet: General

This sheet is required.

3.3.1 Column: Parameter

Required: Yes.
Type: Text.
Description: Parameters that control the settings for the scenario.

3.3.2 Column: Value

Required: Yes.
Type: Text, Boolean or Number.
Description: Values of the parameters. Can be a number or text depending on the parameter.

3.3.3 Parameter: Scenario Name

Required: No.
Value Type: Text.
Default: No name.
Description: A name for the scenario.

3.3.4 Parameter: Iterations

Required: No.
Value Type: Number (positive integer).
Default: `3000`.
Description: The number of iterations the optimiser will run for. (This parameter is ignored if the Iteration Scheme parameter is also specified.)

3.3.5 Parameter: Unassigned Penalty Scheme

Required: No.
Value Type: String (one of `FIXED` or `LEGACY`).
Default: `FIXED`.
Description: The unassigned penalty scheme to use.

3.3.6 Parameter: Fixed Unassigned Penalty

Required: No.
Value Type: Number (positive integer).
Default: `30000`.
Description: The value of the penalty for use by the `FIXED` unassigned penalty scheme.

3.3.7 Parameter: Rush Hour

Required: No.
Value Type: Text (Boolean).
Default: `FALSE`.
Description: If `TRUE`, then use the rush hour periods defined in the Rush Hour sheet.

3.3.8 Parameter: Use Miles

Required: No.
Value Type: Text (Boolean).
Default: `FALSE`.
Description: If `TRUE`, then distance values in the solution will be reported in miles instead of kilometres.

3.3.9 Parameter: Iteration Scheme

Required: No.
Value Type: Text.
Default: No iterations scheme used.
Description: The identifier of an iteration scheme defined in Iteration Schemes sheet for the optimiser to use.

3.3.10 Parameter: Colocated Pickups

Required: No.
Value Type: Text (Boolean).
Default: `FALSE`.
Description: If `TRUE`, then enforce the Colocated Pickups constraint (Section 2.8).

3.3.11 Parameter: Geofence North

Required: No.
Value Type: Number in [`-90`, `90`].
Default: none.
Description: Latitude of the northern boundary for locations in the scenario.

3.3.12 Parameter: Geofence East

Required: No.
Value Type: Number in [`-180`, `180`].
Default: none.
Description: Longitude of the eastern boundary for locations in the scenario.

3.3.13 Parameter: Geofence South

Required: No.
Value Type: Number in [`-90`, `90`].
Default: none.
Description: Latitude of the southern boundary for locations in the scenario.

3.3.14 Parameter: Geofence West

Required: No.
Value Type: Number in [`-180`, `180`].
Default: none.
Description: Longitude of the western boundary for locations in the scenario.

3.3.15 Parameter: Hard Geofence Errors

Required: No.
Value Type: Boolean.
Default: `FALSE`.
Description: If `TRUE`, then treat any errors produced by geofence violations as being hard.

3.3.16 Parameter: Batched Loads

Required: No.
Value Type: Text (Boolean).
Default: `TRUE`.
Description: If `TRUE`, then enforce the Batched Loads constraint.

3.3.17 Parameter: Sticky Deliveries

Required: No.
Value Type: Text (Boolean).
Default: `FALSE`.
Description: If `TRUE`, then enable the Sticky Deliveries constraint.

3.3.18 Parameter: Cost Scale

Required: No.
Value Type: Number (positive).
Default: `0` (no scaling).
Description: Multiply all cost values by this number before solving. (See Scaling Factors in Section 2.8.)

3.3.19 Parameter: Weight Scale

Required: No.
Value Type: Number (positive).
Default: `0` (no scaling).
Description: Multiply all weight values by this number before solving. (See Scaling Factors in Section 2.8.)

3.3.20 Parameter: Volume Scale

Required: No.
Value Type: Number (positive).
Default: `0` (no scaling).
Description: Multiply all volume values by this number before solving. (See Scaling Factors in Section 2.8.)

3.3.21 Parameter: Maximum Neighbourhood Size

Required: No.
Value Type: Number (positive integer).
Default: `200`.
Description: The value of the maximum neighbourhood size for the optimiser. (See Maximum Neighbourhood Size in Section 2.8.)

3.3.22 Parameter: Disable Replanning

Required: No.
Value Type: Text (Boolean).
Default: `FALSE`.
Description: If `TRUE`, then ignore the Previous Solution sheet if it is present.

3.3.23 Parameter: Previous-Vehicle Fixed Reward

Required: No.
Value Type: Number (nonnegative).
Default: `0`.
Description: The strength of the preference for orders to remain on their previously assigned vehicles when replanning.

3.3.24 Parameter: Previous-Vehicle Weight Reward

Required: No.
Value Type: Number (nonnegative).
Default: `0`.
Description: The strength of preference for heavier orders to remain on their previously assigned vehicles when replanning.

3.3.25 Parameter: Previous-Vehicle Volume Reward

Required: No.
Value Type: Number (nonnegative).
Default: `0`.
Description: The strength of preference for larger orders to remain on their previously assigned vehicles when replanning.

3.3.26 Parameter: Forbid Unrecognised Columns

Required: No.
Value Type: Boolean.
Default: `FALSE`.
Description: If `TRUE`, then treat any unrecognised column as an error.

3.3.27 Parameter: Arrival Only in Time Window

Required: No.
Value Type: Boolean.
Default: `FALSE`.
Description: If `TRUE`, then the optimiser will enable the Arrival Only in Time Window behaviour.

3.3.28 Parameter: Order Splitting Strategy

Required: No.
Value Type: Text (one of the Order Splitting Strategy codes).
Default: `None`.
Description: The order splitting strategy to use.

3.3.29 Parameter: Split Solve

Required: No.
Value Type: Boolean.
Default: `FALSE`.
Description: If `TRUE`, then the optimiser will attempt to split solve the scenario.

3.3.30 Parameter: Split Solve Jobs

Required: No.
Value Type: Number (positive integer).
Default: 2.
Description: If split solve is `TRUE`, then the number of sub-scenarios to solve concurrently.
Note: This value is limited according to the installation and any value above this will be reduced to this limit.

3.4 Sheet: Excel Report Configuration

This sheet is not required.

3.4.1 Column: Parameter

Required: Yes.
Type: Text.
Description: Parameters that control the Excel report configuration.

3.4.2 Column: Value

Required: Yes.
Type: Text, Boolean or Number.
Description: Values of the parameters. Can be a number or text depending on the parameter.

3.4.3 Parameter: Unassigned Order Attributes

Required: No.
Value Type: Boolean.
Default: `FALSE`.
Description: If `TRUE`, then the Unassigned Order sheet will contain the `Attributes` column. This column contains order attributes for the unassigned orders.

3.4.4 Parameter: Order Vehicle Compatibility

Required: No.
Value Type: Boolean.
Default: `FALSE`.
Description: If `TRUE`, then include the Unassigned Compatibility sheet in the output.

3.5 Sheet: Locations

This sheet must be defined and contain at least one location.

3.5.1 Column: Id

Required: Yes.
Type: Text or (integer) number.
Description: A unique identifier for the location.

3.5.2 Column: Name

Required: No.
Type: Text or (integer) number.
Description: A name for the location (e.g. a store or facility name).

3.5.3 Column: Attributes

Required: No.
Type: Text or (integer) number.
Description: Multiple attributes are separated by semicolons. (Multiple attributes can only be specified in a text cell.)

3.5.4 Column: Address

Required: No.
Type: Any.
Description: The full street address of the location.

3.5.5 Column: Longitude

Required: Yes.
Type: Number.
Description: Longitude of the location in decimal degrees between `-180` and `180`.

3.5.6 Column: Latitude

Required: Yes.
Type: Number.
Description: Latitude of the location in decimal degrees between `-90` and `90`.

3.5.7 Column: Site Time

Required: No.
Type: Text (*HH:MM*).
Default: `00:00`.
Description: Minimum time spent at the location. (See Site Time in Section 2.10.)

3.5.8 Column: Load Time

Required: No.
Type: Text (*HH:MM*).
Default: `00:00`.
Description: Time spent for loading at the location. (See Load Time in Section 2.10.)

3.5.9 Column: Unload Time

Required: No.
Type: Text (*HH:MM*).
Default: `00:00`.
Description: Time spent for unloading at the location. (See Unload Time in Section 2.10.)

3.5.10 Column: Opening Time

Required: No.
Type: Excel time.
Default: `00:00`.
Description: Earliest opening time for the location.

3.5.11 Column: Closing Time

Required: No.
Type: Excel time.
Default: No closing time.
Description: Latest closing time for the location.

3.5.12 Column: Operating Time Windows

Required: No.
Type: Text or (integer) number.
Default: No time windows.
Description: The identifier of the time window as defined in the Time Windows sheet. (See Location Time Windows in Section 2.10).

3.5.13 Column: Cost per Visit

Required: No.
Type: Number.
Default: `0`.
Description: The cost incurred when a vehicle visits this location. This may be negative to represent gains associated with visiting the location. (See Cost per Visit in Section 2.10.)

3.5.14 Column: Hidden Cost per Visit

Required: No.
Type: Number.
Default: `0`.
Description: An additional cost incurred when a vehicle visits this location. This may be negative and is not included in the cost calculation for the solution. (See Hidden Cost per Visit in Section 2.10.)

3.5.15 Column: Loading Bays

Required: No.
Type: Number (nonnegative integer).
Default: `0`.
Description: The number of loading bays at this location.

3.5.16 Column: Loading Bay Gap

Required: No.
Type: Excel time.
Default: `00:00`.
Description: The loading bay gap at this location.

3.5.17 Column: Allow Preloading

Required: No.
Type: Text (Boolean).
Description: If `TRUE`, then vehicles can preload at this location.
Note: See Preloading for more information.

3.5.18 Column: Allow Splitting

Required: No.
Type: Text (Boolean).
Description: If `TRUE`, then orders that are picked up or delivered to this location can be split according to the order splitting strategy.
Note: The order must also allow splitting for this to take effect.

3.6 Sheet: Orders

This sheet must be defined and contain at least one order.

3.6.1 Column: Id

Required: Yes.
Type: Text or (integer) number.
Description: A unique identifier for the order. An order's identifier also acts as an attribute name.

3.6.2 Column: Name

Required: No.
Type: Text or (integer) number.
Description: A descriptive name for the order (e.g. a purchase order).

3.6.3 Column: Customer Name

Required: No.
Type: Text or (integer) number.
Description: The name of the order's customer.

3.6.4 Column: Pickup Location

Required: Yes.
Type: Text or (integer) number.
Description: An identifier from the Locations sheet.

3.6.5 Column: Pickup Name

Required: No.
Type: Text or (integer) number.
Description: A descriptive name for the pickup.

3.6.6 Column: Pickup Service Time

Required: No.
Type: Excel time.
Default: `00:00`.
Description: A nonnegative time. (See Service Time in Section 2.12.)

3.6.7 Column: Earliest Pickup Time

Required: No.
Type: Excel time.
Default: `00:00`.
Description: A nonnegative time.

3.6.8 Column: Latest Pickup Time

Required: No.
Type: Excel time.
Default: Inferred from data.
Description: A nonnegative time.

3.6.9 Column: Pickup Time Windows

Required: No.
Type: Text.
Default: Can occur at any time subject to the pickup location's and vehicles' time windows.
Description: The identifier of the time window as defined in the Time Windows sheet. (See Order Time Windows in section 2.12.)

3.6.10 Column: Pickup Soft Time Windows

Required: No.
Type: Text.
Default: Can occur at any time subject to the pickup location's, order's and vehicles' time windows.
Description: The identifier of the time window as defined in the Time Windows sheet.

3.6.11 Column: Soft Pickup Constant

Required: No.
Type: Number.
Default: `0`.
Description: The fixed penalty for the order's pickup soft time windows.

3.6.12 Column: Soft Pickup Variable

Required: No.
Type: Number.
Default: `0`.
Description: The variable penalty for the order's pickup soft time windows.

3.6.13 Column: Delivery Location

Required: Yes.
Type: Text or (integer) number.
Description: A location identifier from the Locations sheet.

3.6.14 Column: Delivery Name

Required: No.
Type: Text or (integer) number.
Description: A descriptive name for the delivery.

3.6.15 Column: Delivery Service Time

Required: No.
Type: Excel time.
Default: `00:00`.
Description: A nonnegative time. (See Service Time in Section 2.12.)

3.6.16 Column: Earliest Delivery Time

Required: No.
Type: Excel time.
Default: `00:00`.
Description: A nonnegative time.

3.6.17 Column: Latest Delivery Time

Required: No.
Type: Excel time.
Default: No latest delivery time.
Description: A nonnegative time.

3.6.18 Column: Delivery Time Windows

Required: No.
Type: Text.
Default: Can occur at any time subject to locations' and vehicles' time windows.
Description: The identifier of the time window as defined in the Time Windows sheet. (See Order Time Windows in section 2.12.)

3.6.19 Column: Delivery Soft Time Windows

Required: No.
Type: Text.
Default: Can occur at any time subject to the delivery location's, order's and vehicles' time windows.
Description: The identifier of the time window as defined in the Time Windows sheet.

3.6.20 Column: Soft Delivery Constant

Required: No.
Type: Number.
Default: `0`.
Description: The fixed penalty for any delivery soft time windows associated with the order.

3.6.21 Column: Soft Delivery Variable

Required: No.
Type: Number.
Default: `0`.
Description: The variable penalty for any delivery soft time windows associated with the order.

3.6.22 Column: Maximum Duration

Required: No.
Type: Excel time.
Default: No maximum duration.
Description: The maximum duration for the order.

3.6.23 Column: Weight

Required: No, but if omitted then the Volume column *must* be present.
Type: Number.
Default: `0`.
Description: Weight of the order with units consistent with vehicles' weight capacity.

3.6.24 Column: Volume

Required: No, but if omitted then the Weight column *must* be present.
Type: Number.
Default: `0`.
Description: Volume of the order with units consistent with vehicles' volume capacity.

3.6.25 Column: Assign Cost

Required: No.
Type: Number.
Default: `0`.
Description: The cost incurred when servicing the order. This can be negative to represent gains associated with servicing the order. (See Assign Cost in Section 2.12.)

3.6.26 Column: Hidden Assign Cost

Required: No.
Type: Number.
Default: `0`.
Description: Additional fixed cost to service the order. This can be negative and is not included in the cost calculation for the solution and may be used to prioritise certain orders. (See Hidden Assign Cost in Section 2.12.)

3.6.27 Column: Revenue

Required: No.
Type: Number (nonnegative).
Default: `0`.
Description: The revenue earned for servicing the order.

3.6.28 Column: Attributes

Required: No.
Type: Text or (integer) number.
Description: Multiple attributes are separated by semicolons. (Multiple attributes can only be specified in a text cell.)

3.6.29 Column: Allow Splitting

Required: No.
Type: Text (Boolean).
Description: If `TRUE`, then this order can be split according to the order splitting strategy.
Note: The pickup and delivery locations must *both* allow splitting for this to take effect.

3.7 Sheet: Vehicle Type

The **Vehicle Type** sheet may be used to define a type of vehicle. It may be omitted if vehicle types are not used.

3.7.1 Column: Id

Required: Yes.
Type: Text or (integer) number.
Description: Each vehicle type must have a unique identifier.

3.7.2 Column: Start Location

Required: No.
Type: Text or (integer) number.
Default: The vehicle begins its route at its first pickup location.
Description: The identifier of the location in the Locations sheet where the vehicle starts its route.

3.7.3 Column: Finish Location

Required: No.
Type: Text or (integer) number.
Default: The vehicle finishes its route at its last delivery location.
Description: The identifier of the location in the Locations sheet where the vehicle *must* end its route. The transit time and distance associated with returning to this location after the last delivery are considered by the solution.

3.7.4 Column: Maximum Weight

Required: Yes.
Type: Number.
Default: `0`.
Description: The maximum weight the vehicle can carry. Units are unspecified but must be consistent with orders' weights. Must be nonnegative.

3.7.5 Column: Maximum Volume

Required: Yes.
Type: Number.
Default: `0`.
Description: The maximum volume the vehicle can carry. Units are unspecified but must be consistent with orders' volumes. Must be nonnegative.

3.7.6 Column: Earliest Start Time

Required: No.
Type: Text (*HH:MM*).
Default: `00:00`.
Description: Earliest time a vehicle can start at the start location.

3.7.7 Column: Latest Start Time

Required: No.
Type: Excel time.
Default: No latest start time.
Description: Latest time a vehicle can start at the start location.

3.7.8 Column: Latest Finish Time

Required: No.
Type: Excel time.
Default: No latest finish time.
Description: Latest time a vehicle must arrive at the finish location.

3.7.9 Column: Minimum Paid Time

Required: No.
Type: Excel time.
Default: `00:00`.
Description: A minimum amount of time that this vehicle's cost-per-hour will be incurred. If the vehicle's actual working time is less than this, then the hourly cost will be this value multiplied by the cost-per-hour instead.

3.7.10 Column: Maximum Drive Time

Required: No.
Type: Excel time.
Default: Inferred from data.
Description: Maximum time a vehicle can drive for. Must be nonnegative.

3.7.11 Column: Maximum Work Time

Required: No.
Type: Excel time.
Default: Inferred from data.
Description: Maximum time a vehicle can work for, calculated from the start time to finish time. Must be nonnegative.

3.7.12 Column: Load Time

Required: No.
Type: Excel time.
Default: `00:00`.
Description: Time needed to load vehicle. (See Load Time in Section 2.11.)
Must be nonnegative.

3.7.13 Column: Unload Time

Required: No.
Type: Excel time.
Default: `00:00`.
Description: Time needed to unload the vehicle. (See Unload Time in Section 2.11.)
Must be nonnegative.

3.7.14 Column: Speed Scale

Required: No.
Type: Number.
Default: `0` (means no speed scale).
Description: Nonnegative speed scale multiplier for the vehicle. (See Speed Scale in Section 2.11.)

3.7.15 Column: Cost per Use

Required: No.
Type: Number.
Default: `0`.
Description: A fixed cost for using the vehicle.

3.7.16 Column: Cost per Hour

Required: No.
Type: Number.
Default: `0`.
Description: Cost incurred per hour of vehicle's work time.

3.7.17 Column: Cost per Kilometre

Required: No.
Type: Number.
Default: `0`.
Description: Cost incurred per kilometre of transit distance.
Note: If the parameter Use Miles is `TRUE` in the General sheet, then this column must *not* be present.

3.7.18 Column: Cost per Mile

Required: No.
Type: Number.
Default: `0`.
Description: Cost incurred per mile of transit distance.
Note: If the parameter Use Miles is `FALSE` in the General sheet, then this column must *not* be present.

3.7.19 Column: Cost per Load

Required: No.
Type: Number.
Default: `0`.
Description: Cost incurred for each load. Only applied if batched loads is enabled.

3.7.20 Column: Hidden Cost per Use

Required: No.
Type: Number.
Default: `0`.
Description: Hidden "cost" for using the vehicle. This is not included in the cost calculation for the solution and could be used to prioritise certain vehicles.

3.7.21 Column: Hidden Cost per Hour

Required: No.
Type: Number.
Default: `0`.
Description: Additional cost incurred per hour of vehicle use. This is not included in the cost calculation for the solution and could be used to prioritise vehicles.

3.7.22 Column: Hidden Cost per Kilometre

Required: No.
Type: Number.
Default: `0`.
Description: Additional cost incurred per km of vehicle use. This is not included in the cost calculation for the solution and could be used to prioritise certain vehicles.
Note: If the parameter Use Miles is `TRUE` in the General sheet, then this column must *not* be present.

3.7.23 Column: Hidden Cost per Mile

Required: No.
Type: Number.
Default: `0`.
Description: Additional cost incurred per mile of vehicle use. This is not included in the cost calculation for the solution and could be used to prioritise certain vehicles.
Note: If the parameter Use Miles is `FALSE` in the General sheet, then this column **must** be present.

3.7.24 Column: Hidden Cost per Load

Required: No.
Type: Number.
Default: `0`.
Description: Additional cost incurred for each load. This is not included in the cost calculation for the solution. Only applied if batched loads is enabled.

3.7.25 Column: Finish Segment Distance Penalty

Required: No.
Type: Number.
Default: 0.
Description: The per-unit distance component of the finish segment penalty for this vehicle. Must be nonnegative.

3.7.26 Column: Finish Segment Per Hour Penalty

Required: No.
Type: Number.
Default: 0.
Description: The per hour component of the finish segment penalty for this vehicle. Must be nonnegative.

3.7.27 Column: Pickup Segment Fixed Penalty

Required: No.
Type: Number.
Default: 0.
Description: The fixed component of the pickup segment penalty for this vehicle type. Must be nonnegative.

3.7.28 Column: Pickup Segment Distance Penalty

Required: No.
Type: Number.
Default: 0.
Description: The per-unit distance component of the pickup segment penalty for this vehicle type. Must be nonnegative.

3.7.29 Column: Delivery Segment Fixed Penalty

Required: No.
Type: Number.
Default: 0.
Description: The fixed component of the delivery segment penalty for this vehicle type. Must be nonnegative.

3.7.30 Column: Delivery Segment Distance Penalty

Required: No.
Type: Number.
Default: 0.
Description: The per-unit distance component of the delivery segment penalty for this vehicle type. Must be nonnegative.

3.7.31 Column: Loaded Weight Distance Penalty

Required: No.
Type: Number.
Default: 0.
Description: The weight scaled loaded distance penalty factor for this vehicle type. Must be nonnegative.

3.7.32 Column: Loaded Volume Distance Penalty

Required: No.
Type: Number.
Default: 0.
Description: The volume scaled loaded distance penalty factor for this vehicle type. Must be nonnegative.

3.7.33 Column: Attributes

Required: No.
Type: Text or (integer) number.
Description: Multiple attributes are separated by semicolons. (Multiple attributes can only be specified in a text cell.)

3.7.34 Column: Break Scheme

Required: No.
Type: Text or (integer) number.
Description: The break scheme requirements for the vehicle. Must match a break scheme identifier defined in either the Standard Fatigue Management or Simple Fatigue Management sheets.

3.7.35 Column: Forbid Loaded Breaks

Required: No.
Type: Boolean.
Description: If set to `TRUE`, then loaded breaks will not be scheduled for this vehicle type.

3.7.36 Column: Maximum Loads

Required: No.
Type: Number (positive integer).
Default: `0` (means no constraint on the maximum number of loads).
Description: The maximum number of loads this vehicle type can do. Must be nonnegative. This field is only applicable if Batched Loads is enabled.

3.7.37 Column: Compartments

Required: No.
Type: Text or (integer) number.
Description: The compartment scheme for the vehicle type. If given, then this must match the identifier of a compartment scheme defined in the Compartments sheet.

3.7.38 Column: Allow Preloading

Required: No.
Type: Boolean.
Description: If set to `TRUE`, then this vehicle can preload orders at its start location.
Note: A non-empty Start Location must be defined.
Note 2: See Preloading for more information.

3.7.39 Column: Load Order

Required: No.
Value Type: Text (one of the Load Order codes).
Default: `None`.
Description: The load order strategy to use.

3.7.40 Column: Heavy Order Threshold

Required: No.
Value Type: Number (nonnegative integer).
Description: Any order with weight greater than or equal to this value is considered a *heavy* order for this vehicle type if it is using a weight category *load order*. (See Heavy Order Threshold in Section 2.8.)

3.8 Sheet: Fleet

This sheet must be defined and contain at least one vehicle.

3.8.1 Column: Id

Required: Yes.
Type: Text or (integer) number.
Description: Each vehicle must have a unique identifier. Note that a vehicle's identifier can also function as an attribute name.

3.8.2 Column: Ignored

Required: No.
Type: Boolean.
Description: If set to `TRUE`, then this vehicle will be ignored. Errors will still be reported.

3.8.3 Column: Vehicle Type

Required: No.
Type: Text or (integer) number.
Description: The identifier of a vehicle type defined in the Vehicle Type sheet.

3.8.4 Column: Name

Required: No.
Type: Text or (integer) number.
Default: No name.
Description: A descriptive name for the vehicle.

3.8.5 Column: Driver Name

Required: No.
Type: Text.
Default: No driver name.
Description: The name, or other identifying information, of the driver of the vehicle.

3.8.6 Column: Start Location

Required: No.
Type: Text or (integer) number.
Default: The vehicle begins its route at its first pickup location.
Description: The identifier of the location in the Locations sheet where the vehicle starts its route.

3.8.7 Column: Finish Location

Required: No.
Type: Text or (integer) number.
Default: The vehicle finishes its route at its last delivery location.
Description: The identifier of the location in the Locations sheet where the vehicle *must* end its route. The transit time and distance associated with returning to this location after the last delivery are considered by the solution.

3.8.8 Column: Maximum Weight

Required: Yes.
Type: Number.
Default: `0`.
Description: Maximum weight the vehicle can carry. Units are unspecified but must be consistent with orders' weights. Must be nonnegative.

3.8.9 Column: Maximum Volume

Required: Yes.
Type: Number.
Default: `0`.
Description: Maximum volume the vehicle can carry. Units are unspecified but must be consistent with orders' volumes. Must be nonnegative.

3.8.10 Column: Earliest Start Time

Required: No.
Type: Text (*HH:MM*).
Default: `00:00`.
Description: Earliest time a vehicle can start at the start location.

3.8.11 Column: Latest Start Time

Required: No.
Type: Excel time.
Default: No latest start time.
Description: Latest time a vehicle can start at the start location.

3.8.12 Column: Latest Finish Time

Required: No.
Type: Excel time.
Default: No latest finish time.
Description: Latest time a vehicle must arrive at the finish location.

3.8.13 Column: Minimum Paid Time

Required: No.
Type: Excel time.
Default: `00:00`.
Description: A minimum amount of time that this vehicle's cost-per-hour will be incurred. If the vehicle's actual working time is less than this, then the hourly cost will be this value multiplied by the cost-per-hour instead.

3.8.14 Column: Maximum Drive Time

Required: No.
Type: Excel time.
Default: Inferred from data.
Description: Maximum time a vehicle can drive for. Must be nonnegative.

3.8.15 Column: Maximum Work Time

Required: No.
Type: Excel time.
Default: Inferred from data.
Description: Maximum time a vehicle can work for, calculated from the start time to finish time. Must be nonnegative.

3.8.16 Column: Load Time

Required: No.
Type: Excel time.
Default: `00:00`.
Description: Time needed to load vehicle. (See Load Time in Section 2.11.)
Must be nonnegative.

3.8.17 Column: Unload Time

Required: No.
Type: Excel time.
Default: `00:00`.
Description: Time needed to unload the vehicle. (See Unload Time in Section 2.11.)
Must be nonnegative.

3.8.18 Column: Speed Scale

Required: No.
Type: Number.
Default: `0` (means no speed scale).
Description: Nonnegative speed scale multiplier for the vehicle. (See Speed Scale in Section 2.11.)

3.8.19 Column: Cost per Use

Required: No.
Type: Number.
Default: `0`.
Description: A fixed cost for using the vehicle.

3.8.20 Column: Cost per Hour

Required: No.
Type: Number.
Default: `0`.
Description: Cost incurred per hour of vehicle's work time.

3.8.21 Column: Cost per Kilometre

Required: No.
Type: Number.
Default: `0`.
Description: Cost incurred per kilometre of transit distance.
Note: If the parameter Use Miles is `TRUE` in the General sheet, then this column must *not* be present.

3.8.22 Column: Cost per Mile

Required: No.
Type: Number.
Default: `0`.
Description: Cost incurred per mile of transit distance.
Note: If the parameter Use Miles is `FALSE` in the General sheet, then this column must *not* be present.

3.8.23 Column: Cost per Load

Required: No.
Type: Number.
Default: `0`.
Description: Cost incurred for each load. Only applied if batched loads is enabled.

3.8.24 Column: Hidden Cost per Use

Required: No.
Type: Number.
Default: `0`.
Description: Hidden "cost" for using the vehicle. This is not included in the cost calculation for the solution and could be used to prioritise certain vehicles.

3.8.25 Column: Hidden Cost per Hour

Required: No.
Type: Number.
Default: `0`.
Description: Additional cost incurred per hour of vehicle use. This is not included in the cost calculation for the solution and could be used to prioritise vehicles.

3.8.26 Column: Hidden Cost per Kilometre

Required: No.
Type: Number.
Default: `0`.
Description: Additional cost incurred per km of vehicle use. This is not included in the cost calculation for the solution and could be used to prioritise certain vehicles.
Note: If the parameter Use Miles is `TRUE` in the General sheet, then this column must *not* be present.

3.8.27 Column: Hidden Cost per Mile

Required: No.
Type: Number.
Default: `0`.
Description: Additional cost incurred per mile of vehicle use. This is not included in the cost calculation for the solution and could be used to prioritise certain vehicles.
Note: If the parameter Use Miles is `FALSE` in the General sheet, then this column **cannot** be present.

3.8.28 Column: Hidden Cost per Load

Required: No.
Type: Number.
Default: `0`.
Description: Additional cost incurred for each load. This is not included in the cost calculation for the solution. Only applied if batched loads is enabled.

3.8.29 Column: Finish Segment Distance Penalty

Required: No.
Type: Number.
Default: `0`.
Description: The per-unit distance component of the finish segment penalty for this vehicle. Must be nonnegative.

3.8.30 Column: Finish Segment Per Hour Penalty

Required: No.
Type: Number.
Default: `0`.
Description: The per hour component of the finish segment penalty for this vehicle. Must be nonnegative.

3.8.31 Column: Pickup Segment Fixed Penalty

Required: No.
Type: Number.
Default: `0`.
Description: The fixed component of the pickup segment penalty for this vehicle type. Must be nonnegative.

3.8.32 Column: Pickup Segment Distance Penalty

Required: No.
Type: Number.
Default: `0`.
Description: The per-unit distance component of the pickup segment penalty for this vehicle type. Must be nonnegative.

3.8.33 Column: Delivery Segment Fixed Penalty

Required: No.
Type: Number.
Default: `0`.
Description: The fixed component of the delivery segment penalty for this vehicle. Must be nonnegative.

3.8.34 Column: Delivery Segment Distance Penalty

Required: No.
Type: Number.
Default: `0`.
Description: The per-unit distance component of the delivery segment penalty for this vehicle. Must be nonnegative.

3.8.35 Column: Loaded Weight Distance Penalty

Required: No.
Type: Number.
Default: `0`.
Description: The weight scaled loaded distance penalty factor for this vehicle. Must be nonnegative.

3.8.36 Column: Loaded Volume Distance Penalty

Required: No.
Type: Number.
Default: `0`.
Description: The volume scaled loaded distance penalty factor for this vehicle. Must be nonnegative.

3.8.37 Column: Attributes

Required: No.
Type: Text or (integer) number.
Description: Multiple attributes are separated by semicolons. (Multiple attributes can only be specified in a text cell.)

3.8.38 Column: Break Scheme

Required: No.
Type: Text or (integer) number.
Description: The break scheme requirements for the vehicle. Must match a break scheme identifier defined in either the Standard Fatigue Management or Simple Fatigue Management sheets.

3.8.39 Column: Forbid Loaded Breaks

Required: No.
Type: Boolean.
Description: If set to `TRUE`, then loaded breaks will not be scheduled for the vehicle.

3.8.40 Column: Maximum Loads

Required: No.
Type: Number (positive integer).
Default: `0` (means no constraint on the maximum number of loads).
Description: The maximum number of loads the vehicle can do. Must be nonnegative. This field is only applicable if batched loads is enabled.

3.8.41 Column: Maximum Drops per Load

Required: No.
Type: Number (positive integer).

Default: `0` (means no constraint on the maximum number of drops per load).
Description: The maximum number of delivery locations the vehicle can visit on a given load. Must be nonnegative. This field is only applicable if batched loads is enabled.

3.8.42 Column: Compartments

Required: No.
Type: Text or (integer) number.
Description: The compartment scheme for the vehicle. If given, must match the identifier of a compartment scheme defined in the Compartments sheet.

3.8.43 Column: Allow Preloading

Required: No.
Type: Boolean.
Description: If set to `TRUE`, then this vehicle can preload orders at its start location.
Note: A non-empty Start Location must be defined.
Note 2: See Preloading for more information.

3.8.44 Column: Load Order

Required: No.
Value Type: Text (one of the Load Order codes).
Default: `None`.
Description: The load order strategy to use.

3.8.45 Column: Heavy Order Threshold

Required: No.
Value Type: Number (nonnegative integer).
Description: Any order with weight greater than or equal to this value is considered a *heavy* order for this vehicle if it is using a weight category *load order*. (See Heavy Order Threshold in Section 2.8.)

3.9 Sheet: Standard Fatigue Management

This sheet may be omitted if no standard fatigue management schemes are required. (See Section 2.13 for further information.)

3.9.1 Column: Id

Required: Yes.
Type: Text or (integer) number.
Description: Each fatigue management scheme must have a unique identifier.
Note: Ids must be unique across all *simple* **and** *standard* schemes.

3.9.2 Column: Number

Required: Yes.
Type: Number (integer).
Description: The number of breaks required by the scheme in the given interval in the Interval column. Must be nonnegative.

3.9.3 Column: Duration

Required: Yes.
Type: Excel time.
Description: The duration of the break. Must be nonnegative.

3.9.4 Column: Interval

Required: Yes.
Type: Excel time.
Description: The interval in which the number of blocks specified in Duration must occur as breaks. Must be nonnegative.

3.10 Sheet: Simple Fatigue Management

This sheet may be omitted if no simple fatigue management schemes are required. (See Section 2.13 for further information.)

3.10.1 Column: Id

Required: Yes.
Type: Text or (integer) number.
Description: Each fatigue management scheme must have a unique identifier.
Note: Ids must be unique across all *simple* **and** *standard* schemes.

3.10.2 Column: Earliest

Required: Yes.
Type: Excel time.
Description: The earliest time after the vehicle's start that the break can commence. Must be nonnegative.

3.10.3 Column: Preferred

Required: Yes.
Type: Excel time.
Description: The preferred time after the vehicle's start that the break can commence. Must be nonnegative.

3.10.4 Column: Latest

Required: Yes.
Type: Excel time.
Description: The latest time after the vehicle's start that the break can commence. Must be nonnegative.

3.10.5 Column: Duration

Required: Yes.
Type: Excel time.
Description: The duration of the break. Must be nonnegative.

3.11 Sheet: Time Windows

The **Time Windows** sheet may be used to define groups of time windows that can be associated with pickups, deliveries and locations.

This sheet can be omitted if no named groups of time windows are required.

3.11.1 Column: Id

Required: Yes.
Type: Text or (integer) number.
Description: An identifier giving the named group of time windows that this window is part of.

3.11.2 Column: Start

Required: Yes.
Type: Excel time.
Description: Starting time of the window. Must occur before the End time and be nonnegative.

3.11.3 Column: End

Required: Yes.
Type: Excel time.
Description: Finishing time of the window. Must occur after the Start time and be nonnegative.

3.12 Sheet: Compartments

This sheet may be omitted if no vehicle has compartments.

3.12.1 Column: Scheme Id

Required: Yes.
Type: Text or (integer) number.
Description: A unique identifier for the compartment scheme.

3.12.2 Column: Id

Required: Yes.
Type: Text or (integer) number.
Description: A unique identifier for the compartment. Must be unique within the compartment scheme.

3.12.3 Column: Name

Required: No.
Type: Text or (integer) number.
Description: A descriptive name for the compartment.

3.12.4 Column: Maximum Weight

Required: Yes.
Type: Number.
Description: The maximum weight that the compartment can carry. Units are unspecified but must be consistent with the orders' weights. Must be nonnegative.

3.12.5 Column: Maximum Volume

Required: Yes.
Type: Number.
Description: The maximum volume that the compartment can carry. Units are unspecified but must be consistent with the orders' volumes. Must be nonnegative.

3.12.6 Column: Attributes

Required: No.
Type: Text or (integer) number.
Description: Multiple attributes are separated by semicolons. (Multiple attributes can only be specified in a text cell.)

3.13 Sheet: Rush Hour

The **Rush Hour** sheet is used to define time periods where speed scaling takes effect. Note that this feature must be enabled by setting the value of the Rush Hour parameter to `TRUE`.

This sheet can be omitted if rush hour periods are not required.

3.13.1 Column: Start

Required: Yes.
Type: Excel time.
Description: Starting time of the rush hour period.

3.13.2 Column: End

Required: Yes.
Type: Excel time.
Description: Finishing time of the rush hour period. Must occur after Start Time and be nonnegative.

3.13.3 Column: Speed Scale

Required: Yes.
Type: Number.
Description: The speed scaling factor to apply to transit times within the rush hour period.

3.14 Sheet: Iteration Schemes

The **Iteration Schemes** sheet is used to define iteration schemes.

Each row in this sheet defines an iteration scheme batch. The order of the batches in each scheme is defined by the order of the rows. Note that this sheet can define multiple iteration schemes, although the optimiser can only use one at a time.

This sheet can be omitted.

3.14.1 Column: Id

Required: Yes.
Type: Text or (integer) number.
Description: The identifier of the iteration scheme to which this batch belongs.

3.14.2 Column: Iterations

Required: Yes.
Type: Number (integer).
Description: The nonnegative number of iterations in the batch.

3.14.3 Column: Maximum Neighbourhood Size

Required: Yes.
Type: Number (integer).
Description: The maximum size of the neighbourhood of the batch. Must be at least 10.

3.15 Sheet: *Order-Vehicle* Compatibility

This sheet may be omitted if *order-vehicle* compatibility is not needed. It takes the form of a matrix where the first column specifies the order attributes and the first row specifies the vehicle attributes. The other cells define the compatibility: they must be `TRUE` or empty (i.e. `FALSE`). The upper-left corner of the matrix (i.e. cell `A1`) must be empty.

Note that this sheet may not contain extra columns that are *not* recognised by the optimiser.

3.15.1 Column `A`: Order Attribute

Required: Yes.
Type: Text or (integer) number.
Description: Order attributes that are needed in the *order-vehicle* compatibility are listed in column `A`, starting from the row `2`.

3.15.2 Row `1`: Vehicle Attribute

Required: Yes.
Type: Text or (integer) number.
Description: Vehicle attributes that are needed in the *order-vehicle* compatibility are listed in row `1`, starting from the column `B`.

3.15.3 All other cells (except `A1`)

Required: Yes (only for `TRUE`).
Type: Boolean.
Description: If the corresponding order attribute (on column `A`) and vehicle attribute (on row `1`) are compatible, then the cell value is `TRUE`; otherwise, empty.

3.16 Sheet: *Order-Compartment* Compatibility

This sheet may be omitted if *order-compartment* compatibility is not needed. It is in matrix format with the first column for order attributes, the first row for compartment attributes, the upper-left corner being empty and all other cells being `TRUE` or empty.

Note that this sheet may not contain extra columns that are *not* recognised by the optimiser.

3.16.1 Column `A`: Order Attribute

Required: Yes.
Type: Text or (integer) Number.
Description: Order attributes that are needed in the *order-compartment* compatibility are listed on the column `A`, starting from row `2`.

3.16.2 Row `1`: Compartment Attribute

Required: Yes.
Type: Text or (integer) Number.
Description: Compartment attributes that are needed in the *order-compartment* compatibility are listed on the row `1`, starting from column `B`.

3.16.3 All other cells (except `A1`)

Required: Yes (only for `TRUE`).
Type: Boolean.
Description: If the corresponding order attribute (on column `A`) and compartment attribute (on row `1`) are compatible, then the cell value is `TRUE`; otherwise, empty.

3.17 Sheet: *Order-Order* Compatibility

This sheet may be omitted if *order-order* compatibility is not needed. It is in matrix format with both the first column and the first row for order attributes, the upper-left corner being empty and all other cells being `TRUE` or empty.

Note that this sheet may not contain extra columns that are *not* recognised by the optimiser.

3.17.1 Column `A`: Order Attribute

Required: Yes.
Type: Text or (integer) Number.
Description: Order attributes that are needed in the *order-order* compatibility are listed on the column `A`, starting from the row `2`.

3.17.2 Row `1`: Order Attribute

Required: Yes.
Type: Text or (integer) Number.
Description: Starting from the column `B`, the row `1` has the same sequence of order attributes as the one on the column `A`.

3.17.3 All other cells (except `A1`)

Required: Yes (only for `TRUE` and the upper triangular matrix).
Type: Boolean.
Description: If the corresponding order attribute (on column `A`) and order attribute (on row `1`) are compatible, then the cell value is `TRUE`; otherwise, empty. Since the matrix is symmetrical, only the values of the diagonal and above the diagonal are read by the optimiser.

3.18 Sheet: *Location-Vehicle* Compatibility

This sheet may be omitted if location-vehicle compatibility is not needed.

3.18.1 Cell A1

Required: No.
Type: any type.
Description: The content is not read by the DTO, so any content is acceptable, for example "Vehicle\Location".

3.18.2 Column A: Location Attribute

Required: Yes.
Type: Text.
Description: Location attributes that are needed in the location-vehicle compatibility are listed on the column `A`, starting from the row `2`.

3.18.3 Row 1: Vehicle Attribute

Required: Yes.
Type: Text.
Description: Vehicle attributes that are needed in the location-vehicle compatibility are listed on the row `1`, starting from the column `B`.

3.18.4 All other cells

Required: Yes (only for `true`).
Type: Boolean.
Description: If the corresponding location attribute (on column `A`) and vehicle attribute (on row `1`) are compatible, then the cell value is `true`; otherwise, empty.

3.19 Sheet: *Order-Vehicle* Costs

This sheet is used to specify vehicle-specific assign costs and hidden assign costs for orders. This sheet may be omitted if *order-vehicle* costs are not needed.

3.19.1 Column: Order

Required: Yes.
Type: Text or (integer) number.
Description: The identifier of an order from the Orders sheet.

3.19.2 Column: Vehicle

Required: Yes.
Type: Text or (integer) number.
Description: The identifier of a vehicle from the Fleet sheet.

3.19.3 Column: Cost

Required: No.
Type: Number.
Default: `0`.
Description: The cost incurred if the order is assigned to the vehicle.

3.19.4 Column: Hidden Cost

Required: No.
Type: Number.
Default: `0`.
Description: The hidden cost incurred if the order is assigned to the vehicle.

3.20 Sheet: *Order-Vehicle* Preferences

This sheet is used to specify preferred vehicles for orders. This sheet may be omitted if *order-vehicle* preferences are not needed.

3.20.1 Column: Order

Required: Yes.
Type: Text or (integer) number.
Description: The identifier of an order from the Orders sheet.

3.20.2 Column: Vehicle

Required: Yes.
Type: Text or (integer) number.
Description: The identifier of a vehicle from the Fleet sheet.

3.20.3 Column: Fixed Reward

Required: No.
Value Type: Number (positive integer).
Description: The fixed reward for assigning the order to this vehicle.

3.20.4 Column: Weight Reward

Required: No.
Value Type: Number (positive integer).
Description: The weight reward for assigning the order to this vehicle.

3.20.5 Column: Volume Reward

Required: No.
Value Type: Number (positive integer).
Description: The volume reward for assigning the order to this vehicle.

3.21 Sheet: Order Precedences

This sheet is used to specify precedences between orders. This sheet may be omitted if order precedences are not needed.

3.21.1 Column: Before

Required: Yes.
Type: Text or (integer) number.
Description: The identifier of an order from the Orders sheet.

3.21.2 Column: After

Required: Yes.
Type: Text or (integer) number.
Description: The identifier of an order from the Orders sheet.

3.21.3 Column: Gap

Required: No.
Type: Excel time.
Description: The duration between the departure from the delivery stop of the *before* order and the arrival at the pickup stop of the *after* order.
Default: `00:00`, meaning the pickup of the *after* order may occur immediately after the delivery of the *before* order.

3.22 Sheet: Grouped Pickups

This sheet is used to specify any grouped pickups. This sheet may be omitted if grouped pickups are not needed.

3.22.1 All Rows

Required: Yes.
Type: Text or (integer) number.
Description: A group of orders that must be picked up together. Each cell is an order from the Orders sheet.

3.23 Sheet: Locked Orders

This sheet is used to specify any locked orders. This sheet may be omitted if locked orders are not needed.

3.23.1 Column: Order

Required: Yes.
Type: Text or (integer) number.
Description: The identifier of an order.

3.23.2 Column: Vehicle

Required: Yes.
Type: Text or (integer) number.
Description: The identifier of vehicle.

3.24 Sheet: Exclusive Orders

This sheet is used to specify any exclusive orders. This sheet may be omitted if exclusive orders are not needed.

Each row in the sheet defines a set of exclusive orders. Each cell in the row *must* contain the identifier of an order.

Example:

Table 4 defines three sets of exclusive orders:

1. Order_1A, Order_1B and Order_1C
2. Order_2A, Order_2B and Order_2C
3. Order_3A, Order_3B and Order_3C

Order_1A	Order_1B	Order_1C
Order_2A	Order_2B	Order_2C
Order_3A	Order_3B	Order_3C

Table 4: Exclusive Order example.

3.25 Sheet: Previous Solution

This sheet is used to specify assignments and sequencing from a previous solution for replanning. This sheet can be omitted if replanning is not being used. It is ignored if the global parameter disable replanning is set to `TRUE` in the General sheet.

The columns in this sheet are a subset of those in the Runsheet sheet of DTO-Excel output workbooks. It is possible to copy that sheet directly from an output workbook. To facilitate this, the sheet name "Runsheet" is recognised as a synonym for "Previous Solution".

3.25.1 Column: Order

Required: Yes.
Type: Text or (integer) number.
Description: An identifier for an order. A row containing an identifier that is *not* in the Orders sheet will be ignored.

3.25.2 Column: Vehicle

Required: Yes.
Type: Text or (integer) number.
Description: An identifier for a vehicle. A row containing an identifier that is *not* in the Fleet sheet will be ignored.

3.25.3 Column: Stop

Required: Yes.
Value Type: Text (stop type).
Description: The type of stop. See Section 2.21 for further information.

3.26 Sheet: Time Matrix

The **Time Matrix** sheet may be used to define the transit times, in minutes, between each pair of locations in the Locations sheet.

This sheet is optional. If omitted, the optimiser will compute transit times.

3.26.1 The Matrix

Description: The matrix must be square and contain all locations provided in the Locations sheet. The location identifiers are used to reference the locations and they must be included. The row locations are listed in column `A` beginning in the second row, while the column locations are in row `1` beginning in the second column. Note that negative values are not allowed and that blank cells will mean that it is impossible to transit between locations.

3.27 Sheet: Distance Matrix

The **Distance Matrix** sheet may be used to define the transit distances in **kilometres** between the locations given in the Locations sheet. Note that kilometres are used even if Use Miles is `true` in the General sheet.

This sheet is optional. If omitted, the optimiser will compute the transit distances.

3.27.1 The Matrix

Description: The matrix must be square and contain all locations provided in the Locations sheet. The location identifiers are used to refer to locations and they must be included. The row locations are listed in column `A`, beginning in the second row, while the column locations in row `1`, beginning in the second column. Note that negative values are not allowed and that blank cells will mean that it is impossible to transit between locations.

3.28 Note on Matrices

Providing matrices is not necessary to use the DTO. They are automatically calculated if not provided. However, if you want to use your own then they can be defined here.

	A	B	C	D	E	F	G
1		DEPOT	LOC1	LOC2	LOC3	LOC4	LOC5
2	DEPOT	0.00	15.27	12.90	4.00	4.68	4.68
3	LOC1	16.17	0.00	3.60	13.05	12.72	13.23
4	LOC2	12.57	2.97	0.00	9.47	9.12	9.63
5	LOC3	3.80	12.02	9.67	0.00	1.82	3.48
6	LOC4	4.78	11.77	9.40	1.97	0.00	1.72
7	LOC5	4.57	12.38	10.02	3.38	1.75	0.00

Table 5: Matrix Example

3.29 Sheet: Matrix Override

The **Matrix Override** sheet may be used to override the routing times and distances between pairs of locations in the Locations sheet.

Times in route matrix overrides are affected by speed scaling and rush hour constraints as normal.

This sheet is optional. If omitted, then the optimiser will either use the provided matrices or compute them.

3.29.1 Column: Location 1

Required: Yes.
Type: Text or (integer) number.
Description: The identifier of location from the Location sheet.

3.29.2 Column: Location 2

Required: Yes.
Type: Text or (integer) number.
Description: The identifier of location from the Location sheet.

3.29.3 Column: Time Forwards

Required: No.
Type: Number (positive).
Description: The transit time, in minutes, when travelling from "Location 1" to "Location 2".

3.29.4 Column: Time Backwards

Required: No.
Type: Number (positive).
Description: The transit time, in minutes, when travelling from "Location 2" to "Location 1".

3.29.5 Column: Distance Forwards

Required: No.
Type: Number (positive).
Description: The transit distance, in kilometres, when travelling from "Location 1" to "Location 2".

3.29.6 Column: Distance Backwards

Required: No.
Type: Number (positive).
Description: The transit distance, in kilometres, when travelling from "Location 2" to "Location 1".

Location 1	Location 2	Time Forwards	Time Backwards	Distance Forwards	Distance Backwards
DEPOT	Hallam	0	0	0	0
DEPOT	Skye	0	0	0	0
DEPOT	Rowville	0	0	0	0

Table 6: Matrix Override Example

4 Microsoft Excel Output

This section describes the Microsoft Excel output format for the optimiser.

The Excel output will contain references (e.g. location identifiers, order identifiers) to the scenario from which it was generated.

The optimiser generates output workbooks in Office OpenXML format (`.xlsx`).

4.1 Sheet: Runsheet

4.1.1 Column: Vehicle

Type: Text.
Description: The identifier of the vehicle as given in `Id` column of the Fleet sheet.

4.1.2 Column: Load

Type: Number.
Description: If batched loads is enabled, then this is the index of the current load.

4.1.3 Column: Stop Number

Type: Number.
Description: Numerical identification of a stop generated by the optimiser. Stops are numbered from `0` in increasing order with each stop number being unique.

4.1.4 Column: ETA

Type: Excel time.
Description: The "Estimated Time of Arrival" of the vehicle at the location.

4.1.5 Column: ETD

Type: Excel time.
Description: The "Estimated Time of Departure" of the vehicle from the location.

4.1.6 Column: Stop

Type: Text.
Description: The type of stop. See Section 2.21 for further information.

4.1.7 Column: Order

Type: Text.
Description: If the stop is a `PICKUP` or `DELIVERY` stop, then this contains the identifier of the order in the Orders sheet. For `START`, `FINISH` and `BREAK` stops this will be blank.

4.1.8 Column: Customer Name

Type: Text.
Description: If the stop is a `PICKUP` or `DELIVERY` stop, then this contains the name of the customer for the order from the Orders sheet. This column will be omitted if all of the orders have blank customer names.

4.1.9 Column: Delivery Name

Type: Text.
Description: If the stop is a `PICKUP` or `DELIVERY` stop, then this contains the name of the delivery for the order from the Orders sheet. This column will be omitted if all of the orders have blank delivery names.

4.1.10 Column: Compartment

Type: Text.
Description: If the vehicle has compartments and the stop is a `PICKUP` or `DELIVERY` stop, then this contains the identifier of the compartment that the order is assigned to. For `START`, `FINISH` and `BREAK` stops this will be blank.

4.1.11 Column: Address

Type: Text.
Description: The identifier of the stop location as defined in the Locations sheet.

4.1.12 Column: Volume

Type: Number.
Description: The total volume on the vehicle at the stop.

4.1.13 Column: Delta Volume

Type: Number.
Description: The change in volume at the stop. Positive values represent an increase in volume, while negative values represent a decrease.

4.1.14 Column: Weight

Type: Number.
Description: The total weight on the vehicle at the stop.

4.1.15 Column: Delta Weight

Type: Number.
Description: The change in weight at the stop. Positive values represent an increase in weight, while negative values represent a decrease.

4.1.16 Column: Site Time

Type: Excel time.
Description: The site time incurred at the stop. (See Site Time in Section 2.10.)

4.1.17 Column: Load Time

Type: Excel time.
Description: The load time incurred at the stop. (See Load Time in Section 2.10.)

4.1.18 Column: Unload Time

Type: Excel time.
Description: The unload time incurred at the stop. (See Unload Time in Section 2.10.)

4.1.19 Column: Service Time

Type: Excel time.
Description: The service time incurred at the stop. (See Service Time in Section 2.12.)

4.1.20 Column: Transit Time

Type: Excel time.
Description: The transit time between the previous stop and this one.

4.1.21 Column: Transit Distance

Type: Number.
Description: The transit distance between the previous stop and this one.

4.1.22 Column: Soft Time Window Start

Type: Excel time.
Description: If the order has a soft time window for the current stop type, then the start time of the window closest to the stop arrival time.

4.1.23 Column: Soft Time Window End

Type: Excel time.
Description: If the order has a soft time window for the current stop type, then the end time of the window closest to the stop arrival time.

4.1.24 Column: Soft Time Window Delta

Type: Excel time.
Description: If the order has a soft time window for the current stop type, then the time difference between the stop arrival time and the closest boundary of the time window defined by the `Soft Time Window Start` and `Soft Time Window End` columns.

4.2 Sheet: Vehicle Summary

This sheet is a summary of the Runsheet data for each vehicle.

4.2.1 Column: Vehicle

Type: Text.
Description: The identifier of the vehicle as given by the `Id` column of the Fleet sheet.

4.2.2 Column: Used

Type: Text (Boolean).
Description: `TRUE` if the vehicle has at least one order assigned to it.

4.2.3 Column: Cost

Type: Number.
Description: The cost incurred by the vehicle.

4.2.4 Column: Revenue

Type: Number.
Description: The revenue earned by the vehicle.

4.2.5 Column: Profit

Type: Number.
Description: The profit or loss for the vehicle. Profit is the difference between revenue and cost.

4.2.6 Column: Peak Volume

Type: Number.
Description: The highest volume achieved during the vehicle's route.

4.2.7 Column: Total Volume

Type: Number.
Description: The total volume carried during the vehicle's route.

4.2.8 Column: Volume Capacity

Type: Number.
Description: The maximum volume capacity of the vehicle as defined in the Fleet sheet.

4.2.9 Column: Peak Weight

Type: Number.
Description: The highest weight achieved during the vehicle's route.

4.2.10 Column: Total Weight

Type: Number.
Description: The total weight carried during the vehicle's route.

4.2.11 Column: Weight Capacity

Type: Number.
Description: The maximum weight capacity of the vehicle as defined in the Fleet sheet.

4.2.12 Column: Number Of Loads

Type: Number.
Description: If batched loads is enabled, this is the number of loads in the vehicle's route. (See Section 2.22.)

4.2.13 Column: Transit Distance

Type: Number.
Description: The total transit distance of the vehicle in the route.

4.2.14 Column: Loaded Distance

Type: Number.
Description: The total transit distance of the vehicle while it is carrying at least one order.

4.2.15 Column: Empty Distance

Type: Number.
Description: The total transit distance of the vehicle while it is empty.

4.2.16 Column: Actual Start Time

Type: Excel time.
Description: The time of arrival at the `START` stop in the route for the vehicle.

4.2.17 Column: Actual Finish Time

Type: Excel time.
Description: The time of departure at the `FINISH` stop in the route for the vehicle.

4.2.18 Column: Work Time

Type: Excel time.
Description: The total time spent between the `START` and `FINISH` stops in the route for the vehicle.

4.2.19 Column: Allowed Work Time

Type: Excel time.
Description: The maximum work time for the vehicle as defined in the Fleet sheet. See Vehicle Times for further information.

4.2.20 Column: Transit Time

Type: Excel time.
Description: The total time spent in transit by the vehicle.

4.2.21 Column: Allowed Transit Time

Type: Excel time.
Description: The maximum time that a vehicle can spend in transit during a route as defined in the Fleet sheet. See Vehicle Times for further information.

4.2.22 Column: Site Time

Type: Excel time.
Description: The total site time incurred by the vehicle. (See Site Time in Section 2.10.)

4.2.23 Column: Load Time

Type: Excel time.
Description: The total load time incurred by the vehicle. (See Load Time in Section 2.10.)

4.2.24 Column: Unload Time

Type: Excel time.
Description: The total unload time incurred by the vehicle. (See Unload Time in Section 2.10.)

4.2.25 Column: Service Time

Type: Excel time.
Description: The total service time incurred by the vehicle. (See Service Time in Section 2.12.)

4.2.26 Column: Idle Time

Type: Excel time.
Description: The total time spent idle by the vehicle.

4.2.27 Column: Break Time

Type: Excel time.
Description: The total time spent on breaks by the vehicle.

4.2.28 Column: Number Of Drops

Type: Number.
Description: The number of deliveries such that consecutive deliveries at the same location are only counted once.

4.2.29 Column: Number Of Orders

Type: Number.
Description: The number of orders assigned to the vehicle.

4.2.30 Column: Average Transit Speed

Type: Number.
Description: The average speed of the vehicle. This is derived by dividing the total transit distance by the total transit time. Units are in Km/h by default and Mph if Use Miles in the General sheet is `TRUE`.

4.2.31 Column: Pickup Soft Time Window Early

Type: Excel time.
Description: The total time the vehicle serviced an order's pickup before the start of its closest soft time window.

4.2.32 Column: Pickup Soft Time Window Late

Type: Excel time.
Description: The total time the vehicle serviced an order's pickup after the end of its closest soft time window.

4.2.33 Column: Delivery Soft Time Window Early

Type: Excel time.
Description: The total time the vehicle serviced an order's delivery before the start of its closest soft time window.

4.2.34 Column: Delivery Soft Time Window Late

Type: Excel time.
Description: The total time the vehicle serviced an order's delivery after the end of its closest soft time window.

4.3 Sheet: Load Summary

If batched loads is enabled, then this sheet is a summary of each load for each vehicle.

4.3.1 Column: Vehicle

Type: Text.
Description: The identifier of the vehicle as given by the `Id` column of the Fleet sheet.

4.3.2 Column: Load Number

Type: Number.
Description: The number of the load.

4.3.3 Column: Peak Volume

Type: Number.
Description: The highest volume achieved during the load.

4.3.4 Column: Volume Capacity

Type: Number.
Description: The maximum volume capacity of the vehicle as defined in the Fleet sheet.

4.3.5 Column: Peak Weight

Type: Number.
Description: The highest weight achieved during the load.

4.3.6 Column: Weight Capacity

Type: Number.
Description: The maximum weight capacity of the vehicle as defined in the Fleet sheet.

4.3.7 Column: Loaded Distance

Type: Number.
Description: The transit distance from the first pickup to the last delivery in the load.

4.3.8 Column: Empty Distance Before

Type: Number.
Description: For the first load, this is the distance from the `START` stop to the first pickup; otherwise it is the distance from the final delivery of the previous load to the first pickup of this load.

4.3.9 Column: Empty Distance After

Type: Number.
Description: For the final load, this is the distance from the final delivery to the `FINISH` stop; otherwise it is the distance from the final delivery of this load to the first pickup of the next load.

4.3.10 Column: Work Time

Type: Excel time.
Description: The total time spent between the first stop and last stop in the load by the vehicle.

4.3.11 Column: Transit Time

Type: Excel time.
Description: The time spent in transit by the vehicle for this load.

4.3.12 Column: Site Time

Type: Excel time.
Description: The site time incurred by the vehicle for this load. (See Site Time in Section 2.10.)

4.3.13 Column: Load Time

Type: Excel time.
Description: The total load time incurred by the vehicle for this load. (See Load Time in Section 2.10.)

4.3.14 Column: Unload Time

Type: Excel time.
Description: The total unload time incurred by the vehicle for this load. (See Unload Time in Section 2.10.)

4.3.15 Column: Service Time

Type: Excel time.
Description: The total service time incurred by the vehicle for this load. (See Service Time in Section 2.12.)

4.3.16 Column: Idle Time

Type: Excel time.
Description: The time spent idle by the vehicle for this load.

4.3.17 Column: Break Time

Type: Excel time.
Description: The time spent on breaks by the vehicle for this load.

4.3.18 Column: Number Of Drops

Type: Number.
Description: The number of deliveries in the load such that consecutive deliveries at the same location are only counted once.

4.3.19 Column: Average Transit Speed

Type: Number.
Description: The average speed of the vehicle during the load. This is derived by dividing the total transit distance by the total transit time. Units are in Km/h by default and Mph if Use Miles in the General sheet is `true`.

4.3.20 Column: Pickup Soft Time Window Early

Type: Excel time.
Description: The total time the vehicle serviced an order's pickup before the start of its closest soft time window in the load.

4.3.21 Column: Pickup Soft Time Window Late

Type: Excel time.
Description: The total time the vehicle serviced an order's pickup after the end of its closest soft time window in the load.

4.3.22 Column: Delivery Soft Time Window Early

Type: Excel time.
Description: The total time the vehicle serviced an order's delivery before the start of its closest soft time window in the load.

4.3.23 Column: Delivery Soft Time Window Late

Type: Excel time.
Description: The total time the vehicle serviced an order's delivery after the end of its closest soft time window in the load.

4.4 Sheet: Solution Summary

4.4.1 Scenario Name

Type: Text.
Description: The name of the scenario from which the solution was generated. Matches the `name` field of the General sheet.

4.4.2 Total Iterations

Type: Number.
Description: The actual number of iterations completed by the optimiser before termination.

4.4.3 Total Cost

Type: Number.
Description: The total cost for all routes in the solution.

4.4.4 Total Revenue

Type: Number.
Description: The total revenue earned for all routes in the solution.

4.4.5 Total Profit

Type: Number.
Description: The total profit or loss for all routes in the solution. Profit is the difference between revenue and cost.

4.4.6 Total Distance

Type: Number.
Description: The total transit distance of all routes in the solution.

4.4.7 Loaded Distance

Type: Number.
Description: The total transit distance travelled in all routes while the vehicles are carrying at least one order.

4.4.8 Empty Distance

Type: Number.
Description: The total transit distance travelled in all routes while the vehicles are empty.

4.4.9 Average Transit Speed (km/h)

Type: Number.
Description: The average speed of the vehicles in the solution. This is derived by dividing the total transit distance by the total transit time. Only present if Use Miles in the General sheet is false.

4.4.10 Average Transit Speed (mph)

Type: Number.
Description: The average speed of the vehicles in the solution. This is derived by dividing the total transit distance by the total transit time. Only present if Use Miles in the General sheet is true.

4.4.11 Loads

Type: Number.
Description: If Batched Loads is true, then the total number of loads in the solution.

4.4.12 Km Per Load

Type: Number.
Description: If Batched Loads is true and Use Miles is false, then this is the average kilometres per load in the solution.

4.4.13 Miles Per Load

Type: Number.
Description: If Batched Loads and Use Miles are both true, then this is the average miles per load in the solution.

4.4.14 Hours Per Load

Type: Excel time.
Description: If Batched Loads is true then the average time per load in the solution.

4.4.15 Drops Per Load

Type: Number.
Description: If Batched Loads is true then the total average number of drops per load in the solution.

4.4.16 Work Time

Type: Excel time.
Description: The total work time for all routes in the solution.

4.4.17 Transit Time

Type: Excel time.
Description: The total transit time for all routes in the solution.

4.4.18 Break Time

Type: Excel time.
Description: The total break time for all routes in the solution.

4.4.19 Idle Time

Type: Excel time.
Description: The total idle time for all routes in the solution.

4.4.20 Total Weight

Type: Number.
Description: The cumulative weight carried in all routes in the solution.

4.4.21 Total Volume

Type: Number.
Description: The cumulative volume carried in all routes in the solution.

4.4.22 Assigned Orders

Type: Number.
Description: The number of assigned orders represented as a fraction of the total number of orders.

4.4.23 Fleet Usage

Type: Number.
Description: The number of vehicles used in the solution represented as a fraction of the total number of vehicles.

4.4.24 Column: Pickup Soft Time Window Early

Type: Excel time.
Description: The total time an order's pickup occurred before the start of its closest soft time window.

4.4.25 Column: Pickup Soft Time Window Late

Type: Excel time.
Description: The total time an order's pickup occurred after the end of its closest soft time window.

4.4.26 Column: Delivery Soft Time Window Early

Type: Excel time.
Description: The total time an order's delivery occurred before the start of its closest soft time window.

4.4.27 Column: Delivery Soft Time Window Late

Type: Excel time.
Description: The total time an order's delivery occurred after the end of its closest soft time window.

4.5 Sheet: Unassigned Orders

4.5.1 Column: Order

Type: Text.
Description: The identifier of the unassigned order as given by the Id column of the Orders sheet.

4.5.2 Column: Customer Name

Type: Text.
Description: The name of the customer for the order as given in the Orders sheet. This column will be omitted if all of the orders have blank customer names.

4.5.3 Column: Weight

Type: Number.
Description: The weight of the unassigned order as given by the Weight column of the Orders sheet. This column will be omitted if there are no orders with non-zero weight.

4.5.4 Column: Volume

Type: Number.
Description: The volume of the unassigned order as given by the volume column of the Orders sheet. This column will be omitted if there are no orders with non-zero volume.

4.5.5 Column: Pickup Location

Type: Text.
Description: The identifier of the pickup location of the unassigned order as given by the Id column of the Locations sheet.

4.5.6 Column: Delivery Location

Type: Text.
Description: The identifier of the delivery location of the unassigned order as given by the Id column of the Locations sheet.

4.5.7 Column: Reason

Type: Text.
Description: A reason why the order was not assigned.
Note: There may be multiple reason columns. Reasons may not be provided for some unassigned orders; this usually indicates a lack of fleet capacity or insufficient available working hours.

4.6 Sheet: Unassigned Compatibility

This sheet describes what vehicles are compatible with each unassigned order.

This sheet is optional and will only be included in the output workbook if Order Vehicle Compatibility output is enabled. Even if that parameter is enabled, it will be omitted if no *order-vehicle* compatibility relationships are present or if there are no unassigned orders.

4.6.1 Column A: Order

Type: Text.
Description: The identifier of the unassigned order as given by the Id column of the Orders sheet.

4.6.2 Row 1: Vehicle

Required: Yes.
Type: Text.
Description: A merged header cell with the text `Vehicle`.

4.6.3 All other cells (except A1)

Required: Yes.
Type: Text.
Description: The identifier of a compatible vehicle as given by the Id column of the Fleet sheet.
Note: If the order is compatible with all of the vehicles in the fleet, then this row will be empty and the background of the *Order* cell will be *green*.
Note 2: If the order is compatible with none of the vehicles in the fleet, then this row will be empty and the background of the *Order* cell will be *red*.

4.7 Sheet: Replan Changes

This sheet is only generated when replanning. It contains a summary of differences between the previous solution and this one.

4.7.1 Column: Order

Type: Text.
Description: The identifier of an order whose vehicle assignment has changed between this solution and the previous one. Orders that are no longer present in the scenario are marked as `<CANCELLED>`.

4.7.2 Column: Weight

Type: Number or Text.
Description: The weight of the order in this solution. For orders that are no longer present, this is reported as `N/A`.

4.7.3 Column: Volume

Type: Number or Text.
Description: The volume of the order in this solution. For orders that are no longer present, this is reported as `N/A`.

4.7.4 Column: Previous Vehicle

Type: Text.
Description: The identifier of the vehicle the order was assigned to in the previous solution. Vehicles that are no longer present in the scenario are marked as `<UNAVAILABLE>`.

4.7.5 Column: New Vehicle

Type: Text.
Description: The identifier of the vehicle the order is assigned to in this solution.

4.7.6 Column: Previous Pickup Time

Type: Excel Time.
Description: The pickup time of the order in the previous solution.

4.7.7 Column: New Pickup Time

Type: Excel Time.
Description: The pickup time of the order in this solution. For orders that are no longer present, this is reported as `N/A`.

4.7.8 Column: Previous Delivery Time

Type: Excel Time.
Description: The delivery time of the order in the previous solution.

4.7.9 Column: New Delivery Time

Type: Excel Time.
Description: The delivery time of the order in this solution. For orders that are no longer present, this is reported as `N/A`.

4.8 Sheet: Exclusive Order Summary

Each row in this sheet corresponds to a set of exclusive orders as given by a row in the Exclusive Orders sheet.

4.8.1 Column: Assigned

Type: Text.
Description: The identifier of the assigned order as given by the Id column of the Orders sheet.
Note: This cell may be empty if none of the orders were assigned.

4.8.2 Column: Unassigned

Type: Text.
Description: A comma separated list of identifiers of the orders which were not assigned, as given by the Id column of the Orders sheet.

Table 7 contains an example of this sheet.

Assigned	Unassigned
Order_1A	Order_1B, Order_1C
	Order_2A, Order_2B, Order_2C
Order_3A	Order_3B, Order_3C

Table 7: Exclusive Order Summary Example.

5 JSON Input

This section describes the JSON input format to the optimiser.

Any fields that are omitted in the input take default values. Unless noted otherwise, these default values are `0` for numeric fields, `false` for Boolean fields and `null` for other types of field (including strings).

5.1 Scenario Object

The input to the optimiser is a **scenario** object. It has the following fields:

1. **Field:** `id`
   Required: No.
   Type: String.
   Description: An identifier for this scenario assigned by the client. The form of this identifier is client dependent, for example it may be a database id or a UUID.

2. **Field:** `tags`
   Required: No.
   Type: Array of strings.
   Description: Client-specific information associated with the scenario. This field is ignored by the optimiser.

3. **Field:** `general`
   Required: Yes.
   Type: A general object.
   Description: Contains global optimiser parameters and settings. Must not be `null`.

4. **Field:** `excel_report_configuration`
   Required: No.
   Type: An Excel Report Configuration object.
   Description: Contains configuration parameters for the Microsoft Excel Output.

5. **Field:** `iteration_schemes`
   Required: No.
   Type: Array of iteration batch objects.
   Description: Defines iteration schemes. The order of the batches in each scheme is defined by the order of this array.

6. **Field:** `rush_hour`
   Required: No.
   Type: Array of rush hour objects.
   Description: Contains the rush hour time scaling settings.

7. **Field:** `locations`
   Required: Yes.
   Type: Array of location objects.
   Description: The set of order pickup and delivery, and vehicle start and finish locations. Must not be `null` or empty.

8. **Field:** `orders`
   Required: Yes.
   Type: Array of order objects.
   Description: The set of orders that are to be assigned by the optimiser. Must not be `null` or empty.

9. **Field:** `fleet`
   Required: Yes.
   Type: Array of vehicle objects.
   Description: The set of vehicles making up the fleet. Must not be `null` or empty.

10. **Field:** `compartments`
    Required: No.
    Type: Array of compartment objects.
    Description: The set of compartments associated with the fleet.

11. **Field:** `fatigue_management`
    Required: No.
    Type: Array of standard break objects.
    Description: The set of standard break objects that describe how the drivers must take breaks.

12. **Field:** `simple_fatigue_management`
    Required: No.
    Type: Array of simple break objects.
    Description: The set of simple break objects that describe how the drivers must take breaks.

13. **Field:** `order_vehicle_compatibility`
    Required: No.
    Type: Array of attribute pair objects.
    Description: The rules describing the compatibility between orders and vehicles.

14. **Field:** `order_compartment_compatibility`
    Required: No.
    Type: Array of attribute pair objects.
    Description: The rules describing the compatibility between orders and compartments.

15. **Field:** `order_precedences`
    Required: No.
    Type: Array of order precedence objects.
    Description: The order precedences for the scenario.

16. **Field:** `order_order_compatibility`
    Required: No.
    Type: Array of attribute pair objects.
    Description: The rules describing the compatibility between different orders.

17. **Field:** `load_order_order_compatibility`
    Required: No.
    Type: Array of attribute pair objects.
    Description: The rules describing the compatibility between different orders within a load.

18. **Field:** `grouped_pickups`
    Required: No.
    Type: Array of grouped pickup objects.
    Description: The groups of orders that must be picked up together.

19. **Field:** `exclusive_orders`
    Required: No.
    Type: Array of exclusive order objects.
    Description: The exclusive orders in the scenario.

20. **Field:** `locked_orders`
    Required: No.
    Type: Array of locked order objects.
    Description: The locked orders for this scenario.

21. **Field:** `current_routes`
    Required: No (unless replanning).
    Type: Array of route objects.
    Description: If replanning, then these are the routes from the previous solution. These are ignored if not replanning.

22. **Field:** `time_matrix`
    Required: No.
    Type: Array of arrays of numbers which is a (square) matrix giving the transit times between all pairs of locations in the scenario measured in minutes. The rows and columns are ordered as per the order of the `locations` field. May be `null`, in which case the optimiser will compute the transit times itself.

23. **Field:** `distance_matrix`
Required: No.
Type: Array of arrays of numbers which is a (square) matrix giving the transit distances between all pairs of locations in the scenario measured in kilometres. The rows and columns are ordered as per the order of the `locations` field. May be `null`, in which case the optimiser will compute the transit distances itself.
24. **Field:** `matrix_overrides`
Required: No.
Type: Array of array of Matrix Override objects.
Description: The pairs of locations where the route time and distance matrix entries are to be overridden.

5.2 General Object

The **general** object contains settings for optimiser Global Parameters (see Section 2.8). It has the following fields:

1. **Field:** `name`
   Required: No.
   Type: String.
   Description: A name for the scenario.

2. **Field:** `iterations`
   Required: No.
   Type: Number.
   Default: `3000`.
   Description: The number of iterations that the optimiser runs for.

3. **Field:** `iteration_scheme`
   Required: No.
   Type: String.
   Default: `null`.
   Description: The identifier of the iteration scheme to be used by the optimiser.
   Note: If an iteration scheme is specified, then it overrides the values of the `iterations` and `maximum_neighbourhood_size` fields if they are also present.

4. **Field:** `maximum_neighbourhood_size`
   Required: No.
   Type: Number.
   Default: `200`.
   Description: The maximum size of a neighbourhood for the optimiser. (See Maximum Neighbourhood Size in Section 2.8.)

5. **Field:** `unassigned_penalty_scheme`
   Required: No.
   Type: String (one of the unassigned penalty scheme codes).
   Default: `FIXED`.
   Description: The unassigned penalty scheme to use.

6. **Field:** `fixed_unassigned_penalty`
   Required: No.
   Type: Number (positive integer).
   Default: `30000`.
   Description: The value of the penalty used with the `FIXED` unassigned penalty scheme.

7. **Field:** `geofence_north`
   Required: No.
   Type: Number in [`-90`, `90`].
   Description: The latitude of the northern boundary for locations in the scenario. Acts to enforce accuracy of geocodes.

8. **Field:** `geofence_east`
   Required: No.
   Type: Number in [`-180`, `180`].
   Description: The longitude of the eastern boundary for locations in the scenario. Acts to enforce accuracy of geocodes.

9. **Field:** `geofence_south`
   Required: No.
   Type: Number in [`-90`, `90`].
   Description: The latitude of the southern boundary for locations in the scenario. Acts to enforce accuracy of geocodes.

10. **Field:** `geofence_west`
    Required: No.
    Type: Number in [`-180`, `180`].
    Description: The longitude of the western boundary for locations in the scenario. Acts to enforce accuracy of geocodes.
11. **Field:** `hard_geofence_errors`
    Required: No.
    Type: Boolean.
    Default: `false`.
    Description: If `true`, then treat any errors produced by geofence violations as being hard.
12. **Field:** `colocated_pickups`
    Required: No.
    Type: Boolean.
    Default: `false`.
    Description: If `true`, then enforce the colocated pickups constraint.
13. **Field:** `batched_loads`
    Required: No.
    Type: Boolean.
    Default: `true`.
    Description: If `true`, then enforce the batched loads constraint.
14. **Field:** `use_miles`
    Required: No.
    Type: Boolean.
    Default: `false`.
    Description: If `true`, then distance values in the solution will be in miles rather than kilometres.
15. **Field:** `preferred_vehicle_fixed_reward`
    Required: No.
    Type: Number (positive).
    Default: `0`.
    Description: If replanning, a fixed reward for assigning an order to its originally planned vehicle.
16. **Field:** `preferred_vehicle_weight_reward`
    Required: No.
    Type: Number (positive).
    Default: `0`.
    Description: If replanning, a weight scaled reward for assigning an order to its originally planned vehicle.
17. **Field:** `preferred_vehicle_volume_reward`
    Required: No.
    Type: Number (positive).
    Default: `0`.
    Description: If replanning, a volume scaled reward for assigning an order to its originally planned vehicle.
18. **Field:** `sticky_deliveries`
    Required: No.
    Type: Boolean.
    Default: `false`.
    Description: If `true`, then enforce the Sticky Deliveries constraint.
19. **Field:** `cost_scale`
    Required: No.
    Type: Number.
    Default: `0` (Which means to not apply cost scaling.)
    Description: Scale costs by this amount before solving. (See Scaling Factors in Section 2.8).
20. **Field:** `weight_scale`
    Required: No.
    Type: Number.
    Default: `0` (Which means to not apply weight scaling.)
    Description: Scale weights by this amount before solving. (See Scaling Factors in Section 2.8).

21. **Field:** `volume_scale`
    Required: No.
    Type: Number.
    Default: `0` (Which means to not apply volume scaling.)
    Description: Scale volumes by this amount before solving. (See Scaling Factors in Section 2.8).

22. **Field:** `arrival_only_in_tw`
    Required: No.
    Type: Boolean.
    Default: `false`.
    Description: Enables the Arrival Only in Time Window behaviour.

23. **Field:** `order_splitting_strategy`
    Required: No.
    Type: String (one of the order splitting strategy codes).
    Default: `NONE`.
    Description: The order splitting strategy to use.

24. **Field:** `split_solve`
    Required: No.
    Value Type: Boolean.
    Default: `FALSE`.
    Description: If `TRUE`, then the optimiser will attempt to split solve the scenario.

25. **Field:** `split_solve_jobs`
    Required: No.
    Value Type: Number (positive integer).
    Default: `2`.
    Description: If split solve is `TRUE`, then the number of sub-scenarios to solve concurrently.
    Note: This value is limited according to the installation and any value above this will be reduced to this limit.

5.3 Unassigned Penalty Scheme Codes

Unassigned penalty scheme codes describe the different unassigned penalty schemes that are supported. The possible values are:

1. `FIXED`
2. `LEGACY`

5.4 Order Splitting Strategy Codes

Order splitting strategy codes describe the different order splitting strategies that are supported. The possible values are:

1. `NONE`
2. `SPLIT_BY_WEIGHT`
3. `SPLIT_BY_VOLUME`

5.5 Load Order Codes

Load order codes describe the different load order constraints that are supported. The possible values are:

1. `NONE`
2. `LIFO`
3. `WEIGHT_CATEGORY`

5.6 Excel Report Configuration

The **Excel report configuration** object contains configuration parameters for the Microsoft Excel Output. It has the following fields:

1. **Field:** `unassigned_order_attributes`
   Required: No.
   Type: Boolean.
   Default: `false`.
   Description: If `TRUE`, then the Unassigned Order sheet will contain the `Attributes` column.

2. **Field:** `unassigned_order_vehicle_compatibility`
   Required: No.
   Type: Boolean.
   Default: `false`.
   Description: If `TRUE`, then include the Unassigned Compatibility sheet in the output.

5.7 Location Object

Each location in the scenario is represented by a **location** object. It has the following fields:

1. **Field:** `id`
   Required: Yes.
   Type: String.
   Description: a unique identifier for the location.

2. **Field:** `name`
   Required: No.
   Type: String.
   Description: The name (e.g. a store or site name) for the location.

3. **Field:** `tags`
   Required: No.
   Type: Array of strings.
   Description: Client-specific information associated with the location. This field is ignored by the optimiser.

4. **Field:** `attributes`
   Required: No.
   Type: Array of strings.
   Description: An array containing the names of any attributes the location has.

5. **Field:** `address`
   Required: No.
   Type: String.
   Description: The street address of the location.

6. **Field:** `longitude`
   Required: Yes.
   Type: Number.
   Description: The longitude of the location in decimal degrees. Must be in [`-180`, `180`].

7. **Field:** `latitude`
   Required: Yes.
   Type: Number.
   Description: The latitude of the location in decimal degrees. Must be in [`-90`, `90`].

8. **Field:** `site_time`
   Required: No.
   Type: String (*HH:MM*).
   Default: `00:00`.
   Description: Site time for the location. (See Site Time in Section 2.10.)

9. **Field:** `opening_time`
   Required: No.
   Type: String (*HH:MM*).
   Default: `00:00`.
   Description: The opening time of a location. (See Opening Time in Section 2.10.)

10. **Field:** `closing_time`
    Required: No.
    Type: String (*HH:MM*).
    Default: `null` (meaning no closing time).
    Description: The closing time of a location. (See Closing Time in Section 2.10.)

11. **Field:** `time_windows`
    Required: No.
    Type: Array of time window objects.
    Default: Empty array (i.e. the location is *always* open).
    Description: An array containing the location's opening hours as time window objects.

12. **Field:** `load_time`
    Required: No.
    Type: String (*HH:MM*).
    Default: `00:00`.
    Description: Load time for a location. (See Load Time in Section 2.10.)
13. **Field:** `unload_time`
    Required: No.
    Type: String (*HH:MM*).
    Default: `00:00`.
    Description: Unload time for a location. (See Unload Time in Section 2.10.)
14. **Field:** `cost_per_visit`
    Required: No.
    Type: Number.
    Default: `0`.
    Description: The cost incurred when a vehicle visits this location. This may be negative to represent gains associated with visiting the location. (See Cost per Visit in Section 2.10.)
15. **Field:** `hidden_cost_per_visit`
    Required: No.
    Type: Number.
    Default: `0`.
    Description: An additional cost incurred when a vehicle visits this location. This may be negative and is not included in the cost calculation for the solution. (See Hidden Cost per Visit in Section 2.10.)
16. **Field:** `loading_bays`
    Required: No.
    Type: Number (nonnegative integer).
    Description: The number of loading bays at this location.
17. **Field:** `loading_bay_gap`
    Required: No.
    Type: String (*HH:MM*).
    Default: `00:00`.
    Description: The loading bay gap at this location.
18. **Field:** `allow_preloading`
    Required: No.
    Type: Boolean.
    Description: If set to `TRUE`, then vehicles can preload at this location.
    Note: See Preloading for more information.
19. **Field:** `allow_order_splitting`
    Required: No.
    Type: Boolean.
    Description: If set to `TRUE`, then orders that are picked up or delivered to this location can be split according to the order splitting strategy.
    Note: The order must also allow splitting for this to take effect.

5.8 Order Object

Each order in the scenario is represented by an **order** object, which has the following fields:

1. **Field:** `id`
   Required: Yes.
   Type: String.
   Description: the unique identifier for the order. An order's identifier also acts as an attribute name.
2. **Field:** `name`
   Required: No.
   Type: String.
   Description: A descriptive name for the order.
3. **Field:** `customerName`
   Required: No.
   Type: String.
   Description: The name of the order's customer.
4. **Field:** `tags`
   Required: No.
   Type: Array of strings.
   Description: Client-specific information associated with the order. This field is ignored by the optimiser.
5. **Field:** `attributes`
   Required: No.
   Type: Array of strings.
   Description: An array containing the names of any attributes the order has.
6. **Field:** `pickup_location`
   Required: Yes.
   Type: String.
   Description: The pickup location for the order. Must match the `id` of a location specified in a location object.
7. **Field:** `pickup_name`
   Required: No.
   Type: String.
   Description: A descriptive name for the pickup.
8. **Field:** `earliest_pickup_time`
   Required: No.
   Type: String (*HH:MM*).
   Default: `null` (meaning no earliest pickup time).
   Description: Earliest time the order can be picked up.
9. **Field:** `latest_pickup_time`
   Required: No.
   Type: String (*HH:MM*).
   Default: `null` (meaning no latest pickup time).
   Description: Latest time the order can be picked up.
10. **Field:** `pickup_time_windows`
    Required: No.
    Type: Array of time window objects.
    Default: `null` (meaning the pickup can occur at any time).
    Description: Pickup time windows for the order. (See Order Time Windows in section 2.12.) An empty array is treated as if the value were `null`.
11. **Field:** `pickup_soft_time_windows`
    Required: No.
    Type: A soft time window object.
    Default: `null` (meaning no soft pickup time windows).
    Description: An object that contains an array of soft time windows for the order's pickup as well as associated penalties.

12. **Field:** `pickup_service_time`
    Required: No.
    Type: String (*HH:MM*).
    Default: `00:00`.
    Description: Service time for the pickup. (See Service Time in Section 2.12.)
13. **Field:** `delivery_location`
    Required: Yes.
    Type: String.
    Description: The delivery location for the order. Must match the `id` of a location specified in a location object.
14. **Field:** `delivery_name`
    Required: No.
    Type: String.
    Description: A descriptive name for the delivery.
15. **Field:** `earliest_delivery_time`
    Required: No.
    Type: String (*HH:MM*).
    Default: `null` (meaning no earliest delivery time).
    Description: Earliest time the delivery can be made.
16. **Field:** `latest_delivery_time`
    Required: No.
    Type: String (*HH:MM*).
    Default: `null` (meaning no latest delivery time).
    Description: Latest time a delivery can be made.
17. **Field:** `delivery_time_windows`
    Required: No.
    Type: Array of time window objects.
    Default: `null` (meaning the delivery can occur at any time).
    Description: Delivery time windows for the order. An empty array is treated as if the value were `null`. (See Order Time Windows in section 2.12.)
18. **Field:** `delivery_soft_time_windows`
    Required: No.
    Type: A soft time window object.
    Default: `null` (meaning no soft delivery time windows).
    Description: An object that contains an array of soft time windows for the order's delivery as well as associated penalties.
19. **Field:** `delivery_service_time`
    Required: No.
    Type: String (*HH:MM*).
    Default: `00:00`.
    Description: Service time for the delivery. (See Service Time in Section 2.12).
20. **Field:** `maximum_duration`
    Required: No.
    Type: String (*HH:MM*).
    Default: `null` (This means there is no maximum duration).
    Description: The maximum duration of the order.
21. **Field:** `weight`
    Required: No.
    Type: Number.
    Default: `0`.
    Description: The weight of the order consistent with vehicle maximum weights. Must be nonnegative.

22. **Field:** `volume`
    Required: No.
    Type: Number.
    Default: `0`.
    Description: The volume of the order consistent with vehicle maximum volumes. Must be nonnegative.
23. **Field:** `assign_cost`
    Required: No.
    Type: Number.
    Default: `0`.
    Description: The assign cost for the order.
24. **Field:** `hidden_assign_cost`
    Required: No.
    Type: Number.
    Default: `0`.
    Description: The hidden assign cost for the order.
25. **Field:** `revenue`
    Required: No.
    Type: Number (nonnegative).
    Default: `0`.
    Description: The revenue earned for servicing this order.
26. **Field:** `vehicle_assign_cost`
    Required: No.
    Type: Array of vehicle cost objects.
    Default: No vehicle-specific assign costs.
    Description: The vehicle-specific assign costs for the order.
27. **Field:** `vehicle_hidden_assign_cost`
    Required: No.
    Type: Array of vehicle cost objects.
    Default: No vehicle-specific hidden assign costs.
    Description: The vehicle-specific hidden assign costs for the order.
28. **Field:** `vehicle_preference`
    Required: No.
    Type: Array of vehicle preference objects.
    Default: No vehicle-specific preferences.
    Description: The vehicle-specific rewards for the order.
29. **Field:** `allow_splitting`
    Required: No.
    Type: Boolean.
    Description: If set to `TRUE`, then this order can be split according to the order splitting strategy.
    Note: The pickup and delivery locations must also allow splitting for this to take effect.

5.9 Vehicle Object

Each vehicle in the fleet is represented by a **vehicle** object which has the following fields:

1. **Field:** `id`
   Required: Yes.
   Type: String.
   Description: A unique identifier for the vehicle. A vehicle's identifier also acts as an attribute name.
2. **Field:** `name`
   Required: No.
   Type: String.
   Description: A descriptive name for the vehicle.
3. **Field:** `driver_name`
   Required: Yes.
   Type: String.
   Description: The name, or other identifying information, of the driver of the vehicle.
4. **Field:** `tags`
   Required: No.
   Type: Array of strings.
   Description: Client-specific information associated with the vehicle. This field is ignored by the optimiser.
5. **Field:** `attributes`
   Required: No.
   Type: Array of strings.
   Description: An array containing the names of any attributes the vehicle has.
6. **Field:** `compartments`
   Required: No.
   Type: String.
   Default: `null` (No compartments).
   Description: The identifier of the compartment scheme used by this vehicle.
7. **Field:** `start_location`
   Required: Yes.
   Type: String.
   Default: `null` (This means the vehicle can start anywhere).
   Description: The start location of the vehicle. This must match the identifier of a location specified in a location object.
8. **Field:** `finish_location`
   Required: No.
   Type: String.
   Default: `null` (This means the vehicle can finish anywhere).
   Description: The finish location of the vehicle. This must match the identifier of a location specified in the location object.
9. **Field:** `maximum_weight`
   Required: No.
   Type: Number.
   Default: `0`.
   Description: The maximum weight capacity of the vehicle.
10. **Field:** `maximum_volume`
    Required: No.
    Type: Number.
    Default: `0`.
    Description: The maximum volume capacity of the vehicle.
11. **Field:** `earliest_start_time`
    Required: Yes.
    Type: String (*HH:MM*).
    Description: Earliest time a vehicle can start a route.

12. **Field:** `latest_start_time`
    Required: Yes.
    Type: String (*HH:MM*).
    Description: Latest time a vehicle can start a route.

13. **Field:** `latest_finish_time`
    Required: Yes.
    Type: String (*HH:MM*).
    Description: Latest time a vehicle can finish a route.

14. **Field:** `minimum_paid_time`
    Required: No.
    Type: String (*HH:MM*).
    Default: `null`.
    Description: A minimum amount of time that this vehicle's cost-per-hour will be incurred. If the vehicle's actual working time is less than this, then the hourly cost will be this value multiplied by the cost-per-hour instead. If this value is `null`, then the minimum paid time is `00:00`.

15. **Field:** `maximum_drive_time`
    Required: No.
    Type: String (*HH:MM*).
    Default: `null`.
    Description: Maximum time a vehicle can spend in transit. If this is `null`, then the value of `maximum_work_time` is used.

16. **Field:** `maximum_work_time`
    Required: No.
    Type: String (*HH:MM*).
    Default: `null`.
    Description: The maximum work time for the vehicle; if the value is `null`, then there is no limit on the work time.

17. **Field:** `speed_scale`
    Required: No.
    Type: Number.
    Default: `0` (No speed scale).
    Description: Nonnegative speed scale multiplier for the vehicle.

18. **Field:** `load_time`
    Required: No.
    Type: String (*HH:MM*).
    Default: `00:00`.
    Description: A vehicle-specific load time will override a location load time if applicable. (See Load Time in Section 2.11.)

19. **Field:** `unload_time`
    Required: No.
    Type: String (*HH:MM*).
    Default: `00:00`.
    Description: Unload time for a vehicle. (See Unload Time in Section 2.11.)

20. **Field:** `cost_per_use`
    Required: No.
    Type: Number.
    Default: `0`.
    Description: Flat cost to use the vehicle. Must be nonnegative. (See Cost Per Use in Section 2.11.)

21. **Field:** `cost_per_hour`
    Required: No.
    Type: Number.
    Default: `0`.
    Description: Cost incurred per hour of vehicle use. Must be nonnegative. (See Cost Per Hour in Section 2.11).

22. **Field:** `cost_per_km`
Required: No.
Type: Number.
Default: `0`.
Description: Cost incurred per kilometre of transit distance for the vehicle. Must be nonnegative. (See Cost Per KM in Section 2.11).
Note: If the Use Miles field is `true`, then this must be `0`.

23. **Field:** `cost_per_mile`
Required: No.
Type: Number.
Default: `0`.
Description: Cost incurred per mile of transit distance for the vehicle. Must be nonnegative.
Note: If the Use Miles field is `false`, then this must be `0`.

24. **Field:** `cost_per_load`
Required: No.
Type: Number.
Default: `0`.
Description: Cost incurred for each load. Must be nonnegative. Only applied if batched loads is enabled.

25. **Field:** `hidden_cost_per_use`
Required: No.
Type: Number.
Default: `0`.
Description: The hidden cost for using this vehicle.

26. **Field:** `hidden_cost_per_hour`
Required: No.
Type: Number.
Default: `0`.
Description: Additional cost incurred per hour of vehicle use. This can be negative and is not included in the cost calculation for the solution and could be used to prioritise certain vehicles. (See Hidden Cost Per Hour).

27. **Field:** `hidden_cost_per_km`
Required: No.
Type: Number.
Default: `0`.
Description: Additional cost incurred per kilometre of vehicle use. This can be negative and is not included in the cost calculation for the solution and could be used to prioritise certain vehicles. (See Hidden Cost Per Km).
Note: If the Use Miles field is `true`, then this must be `0`.

28. **Field:** `hidden_cost_per_mile`
Required: No.
Type: Number.
Default: `0`.
Description: Additional cost incurred per mile of vehicle use. This can be negative and is not included in the cost calculation for the solution and could be used to prioritise certain vehicles. (See Hidden Cost Per Mile).
Note: If the Use Miles field is `false`, then this must be `0`.

29. **Field:** `hidden_cost_per_load`
Required: No.
Type: Number.
Default: `0`.
Description: Additional cost incurred for each load. This can be negative and is not included in the cost calculation for the solution. Only applied if batched loads is enabled.

30. **Field:** `finish_segment_distance_penalty`
Required: No.
Type: Number.
Default: `0`.
Description: The per-unit distance component of the finish segment penalty for this vehicle. Must be nonnegative.

31. **Field:** `finish_segment_per_hour_penalty`
Required: No.
Type: Number.
Default: `0`.
Description: The per hour component of the finish segment penalty for this vehicle. Must be nonnegative.

32. **Field:** `pickup_segment_fixed_penalty`
Required: No.
Type: Number.
Default: `0`.
Description: The fixed component of the pickup segment penalty for this vehicle. Must be nonnegative.

33. **Field:** `pickup_segment_distance_penalty`
Required: No.
Type: Number.
Default: `0`.
Description: The per-unit distance component of the pickup segment penalty for this vehicle. Must be nonnegative.

34. **Field:** `delivery_segment_fixed_penalty`
Required: No.
Type: Number.
Default: `0`.
Description: The fixed component of the delivery segment penalty for this vehicle. Must be nonnegative.

35. **Field:** `delivery_segment_distance_penalty`
Required: No.
Type: Number.
Default: `0`.
Description: The per-unit distance component of the delivery segment penalty for this vehicle. Must be nonnegative.

36. **Field:** `loaded_weight_distance_penalty`
Required: No.
Type: Number.
Default: `0`.
Description: The weight scaled loaded distance penalty factor for this vehicle. Must be nonnegative.

37. **Field:** `loaded_volume_distance_penalty`
Required: No.
Type: Number.
Default: `0`.
Description: The volume scaled loaded distance penalty factor for this vehicle. Must be nonnegative.

38. **Field:** `break_scheme`
Required: No.
Type: String.
Description: An identifier that specifies the break rules that apply to this vehicle. (See Fatigue Management Scheme in Section 2.11.)

39. **Field:** `forbid_loaded_breaks`
Required: No.
Type: Boolean.
Default: `false`.
Description: If set to `true`, then loaded breaks will not be scheduled for the vehicle.

40. **Field:** `maximum_loads`
Required: No.
Type: Number (integer).
Default: `0`, which means no maximum number of loads.
Description: A nonnegative upper bound on the maximum number of loads for this vehicle which is only applied if batched loads is enabled. (See Maximum Loads in Section 2.11.)

41. **Field:** `max_drops_per_load`
Required: No.
Type: Number (integer).
Default: `0`, which means no maximum number of drops.
Description: A nonnegative upper bound on the maximum number of locations this vehicle can deliver to in a load. This is only applied if batched loads are enabled. (See Maximum Drops Per Load in Section 2.11.)

42. **Field:** `allow_preloading`
Required: No.
Type: Boolean.
Description: If set to `TRUE`, then this vehicle can preload orders at its start location.
Note: A non-empty `start_location` must be defined.
Note 2: See Preloading for more information.

43. **Field:** `load_order`
Required: No.
Value Type: Text (one of the Load Order codes).
Default: `None`.
Description: The load order strategy to use.

44. **Field:** `heavy_order_threshold`
Required: No.
Type: Number (nonnegative integer).
Default: 0.
Description: Any order with weight greater than or equal to this value is considered a *heavy* order for this vehicle if using a weight category *load order*. (See Heavy Order Threshold in Section 2.8.)

5.10 Standard Break Object

A **standard break** object describes a single rule governing how driver breaks. A scheme consists of at least one **standard break** object. (See Section 2.13.1 for further information). It has the following fields:

1. **Field:** `id`
   Required: Yes.
   Type: String.
   Description: An identifier giving the **break scheme** to which this rule belongs.

2. **Field:** `number`
   Required: Yes.
   Type: Number (integer).
   Description: The number of breaks required by the scheme.

3. **Field:** `duration`
   Required: Yes.
   Type: String (*HH:MM*).
   Description: The duration of the break.

4. **Field:** `interval`
   Required: Yes.
   Type: String (*HH:MM*).
   Description: The interval of time in which the specified `number` of breaks must occur.

5.11 Simple Break Object

A **simpmle break** object describes a single rule governing how driver breaks. A scheme consists of at least one **simple break** object. (See Section 2.13.2 for further information). It has the following fields:

1. **Field:** `id`
   Required: Yes.
   Type: String.
   Description: An identifier giving the **break scheme** to which this rule belongs.

2. **Field:** `duration`
   Required: Yes.
   Type: String (*HH:MM*).
   Description: The duration of the break.

3. **Field:** `earliest`
   Required: Yes.
   Type: String (*HH:MM*).
   Description: The earliest time after the vehicle's start that the break can commence.

4. **Field:** `preferred`
   Required: Yes.
   Type: String (*HH:MM*).
   Description: The preferred time after the vehicle's start that the break can commence.

5. **Field:** `latest`
   Required: Yes.
   Type: String (*HH:MM*).
   Description: The latest time after the vehicle's start that the break can commence.

5.12 Compartment Object

A **compartment** object has the following fields:

1. **Field:** `scheme_id`
   Required: Yes.
   Type: String.
   Description: The identifier of the compartment scheme to which this compartment belongs.

2. **Field:** `id`
   Required: Yes.
   Type: String.
   Description: The identifier for the compartment. This must be unique within the scheme to which this compartment belongs.

3. **Field:** `name`
   Required: No.
   Type: String.
   Description: A descriptive name for the compartment. This must be unique with the scheme to which this compartment belongs.

4. **Field:** `maximum_weight`
   Required: Yes.
   Type: Number.
   Description: The maximum weight capacity of the compartment.

5. **Field:** `maximum_volume`
   Required: Yes.
   Type: Number.
   Description: The maximum volume capacity of the compartment.

6. **Field:** `attributes`
   Required: No.
   Type: Array of strings.
   Description: Any attributes attached to this compartment.

5.13 Time Window Object

A **time window** object has the following fields:

1. **Field:** `start`
   Required: Yes.
   Type: String (*HH:MM*).
   Description: Starting time of the time window. Must not occur after the `end` time.

2. **Field:** `end`
   Required: Yes.
   Type: String (*HH:MM*).
   Description: Finishing time of the time window. Must be after `start` time.

5.14 Soft Time Window Object

A **soft time window** object has the following fields:

1. **Field:** `time_windows`
   Required: Yes.
   Type: Array of time window objects.
   Description: An array containing the order's soft delivery times as time window objects. An empty array means no soft time windows.

2. **Field:** `fixed_hidden_cost`
   Required: No.
   Type: Number.
   Default: 0.
   Description: The fixed cost for missing a soft time window. Must be nonnegative.

3. **Field:** `variable_hidden_cost`
   Required: No.
   Type: Number.
   Default: 0.
   Description: The variable cost for missing a soft time window. Must be nonnegative.

5.15 Attribute Pair Object

An **attribute pair** object is a pair of attributes and it is used for defining compatibility relationships between various entities. (See Section 2.17 for more information). It has the following fields:

1. **Field:** `attribute1`
   Required: Yes.
   Type: String.
   Description: It may be the order attribute in *order-vehicle* compatibility or order attribute in *order-order* compatibility.

2. **Field:** `attribute2`
   Required: Yes.
   Type: String.
   Description: It may be the vehicle attribute in *order-vehicle* compatibility or order attribute in *order-order* compatibility.

5.16 Rush Hour Object

This object holds the time dependent speed scaling information. It has the following fields:

1. **Field:** `start`
   Required: Yes.
   Type: String (*HH:MM*).
   Description: The start of the time period where the scaling factor is applied.
2. **Field:** `end`
   Required: Yes.
   Type: String (*HH:MM*).
   Description: The end of the time period where the scaling factor is applied.
3. **Field:** `speed_scale`
   Required: Yes.
   Type: Number.
   Description: The speed scaling factor to be applied.

5.17 Iteration Batch Object

An **iteration batch** object defines a single batch within an iteration scheme. It has the following fields:

1. **Field:** `id`
   Required: Yes.
   Type: String.
   Description: The identifier of the iteration scheme to which this batch belongs.

2. **Field:** `iterations`
   Required: Yes.
   Type: Number (positive integer).
   Description: The number of iterations that the optimiser runs for in this batch.

3. **Field:** `maximum_neighbourhood_size`
   Required: Yes.
   Type: Number (positive integer).
   Description: The maximum size of a neighbourhood of the optimiser for the duration of this batch.

5.18 Vehicle Cost Object

A **vehicle cost** associates a vehicle with a cost.

1. **Field:** `vehicle`
   Required: Yes.
   Type: String.
   Description: The `id` of a vehicle.

2. **Field:** `cost`
   Required: No.
   Type: Number.
   Default: `0`.
   Description: A cost to associate with the vehicle.

5.19 Vehicle Preference Object

A **vehicle preference** associates a vehicle with rewards. It has the following fields:

1. **Field:** `vehicle`
   Required: Yes.
   Type: String.
   Description: The `id` of a vehicle.

2. **Field:** `fixed_reward`
   Required: No.
   Type: Number.
   Default: `0`.
   Description: A fixed reward for assigning the order to the vehicle.

3. **Field:** `weight_reward`
   Required: No.
   Type: Number.
   Default: `0`.
   Description: A weight reward for assigning the order to the vehicle. This reward is multiplied by the weight of the order.

4. **Field:** `volume_reward`
   Required: No.
   Type: Number.
   Default: `0`.
   Description: A volume reward for assigning the order to the vehicle. This reward is multiplied by the volume of the order.

5.20 Order Precedence Object

An **order precedence** object defines a precedence relationship between two orders. It has the following fields:

1. **Field:** `before`
   Required: Yes.
   Type: String.
   Description: The `id` of the first order in the precedence relationship.

2. **Field:** `after`
   Required: Yes.
   Type: String.
   Description: The `id` of the second order in the precedence relationship.

3. **Field:** `gap`
   Required: No.
   Type: String (*HH:MM*).
   Description: The minimum duration that must elapse between the departure from the delivery stop of the *before* order and the arrival at the pickup stop of the *after* order.
   Default: `null`, the arrival at the pickup stop for the *after* order may occur immediately after the departure from the delivery stop for the *before* order.

5.21 Grouped Pickup Object

A **Grouped Pickup** object defines a group of orders that must be picked up together. It has the following fields:

1. **Field:** `orders`
   Required: Yes.
   Type: Array of strings.
   Description: A list of orders in the grouped pickup.

5.22 Locked Order Object

A **Locked Order** object locks an order to a vehicle. It has the following fields:

1. **Field:** `order`
   Required: Yes.
   Type: String.
   Description: The `id` of an order.

2. **Field:** `vehicle`
   Required: Yes.
   Type: String.
   Description: The `id` of a vehicle.

5.23 Exclusive Order Object

An **Exclusive Order** object defines a set of exclusive orders.

It has the following fields:

1. **Field:** `orders`
   Required: Yes.
   Type: Array of strings.
   Description: An array containing the order identifiers of orders in the group.

Here is an example of an exclusive order object.

```
{
  "orders": [
    "ORDER1",
    "ORDER2"
  ]
}
```

5.24 Matrix Override Object

A **Matrix Override** object is used to override the time and distances between two locations. It has the following fields:

1. **Field:** `location1`
   Required: Yes.
   Type: String.
   Description: The `id` of a location.
2. **Field:** `location2`
   Required: Yes.
   Type: String.
   Description: The `id` of a location.
3. **Field:** `time_forwards`
   Required: No.
   Type: Number (Positive).
   Description: The time from `location1` to `location2`.
4. **Field:** `time_backwards`
   Required: No.
   Type: Number (Positive).
   Description: The time from `location2` to `location1`.
5. **Field:** `distance_forwards`
   Required: No.
   Type: Number (Positive).
   Description: The distance from `location1` to `location2`.
6. **Field:** `distance_backwards`
   Required: No.
   Type: Number (Positive).
   Description: The distance from `location2` to `location1`.

6 JSON Output

This section describes the JSON output format for the optimiser.

6.1 Solution Object

Each solution object consists of the following fields:

1. **Field:** `routes`
   Type: Array of route objects.
   Description: An array of route objects, each of which describes the generated plan for a single vehicle.
2. **Field:** `route_kpis`
   Type: Array of route_kpi objects.
   Description: An array of route KPI objects, each of which describes the KPIs for a single vehicle.
3. **Field:** `cost`
   Type: Number.
   Description: The total cost for all routes in the solution.
4. **Field:** `revenue`
   Type: Number.
   Description: The total revenue for all routes in the solution.
5. **Field:** `profit`
   Type: Number.
   Description: The total profit or loss for all routes in the solution. Profit is the difference between revenue and cost.
6. **Field:** `distance`
   Type: Number.
   Description: The total transit distance of all routes in the solution.
7. **Field:** `delivery_distance`
   Type: Number.
   Description: The transit distance while making deliveries in all routes in the solution.
8. **Field:** `loaded_distance`
   Type: Number.
   Description: The total transit distance travelled in all routes while the vehicles are carrying at least one order.
9. **Field:** `empty_distance`
   Type: Number.
   Description: The total transit distance travelled in all routes while the vehicles are empty.
10. **Field:** `work_time`
    Type: String (*HH:MM*).
    Description: The total work time for all routes in the solution.
11. **Field:** `transit_time`
    Type: String (*HH:MM*).
    Description: The total transit time for all routes in the solution.
12. **Field:** `service_time`
    Type: String (*HH:MM*) .
    Description: The total service time for all routes in the solution. (See Service Time in Section 2.12.)
13. **Field:** `site_time`
    Type: String (*HH:MM*).
    Description: The site time incurred in all routes in the solution. (See Site Time in Section 2.10.)
14. **Field:** `idle_time`
    Type: String (*HH:MM*).
    Description: The total idle time for all routes in the solution.

15. **Field:** `break_time`
    Type: String (*HH:MM*).
    Description: The total break time for all routes in the solution.
16. **Field:** `weight`
    Type: Number.
    Description: The cumulative weight carried in all routes in the solution.
17. **Field:** `volume`
    Type: Number.
    Description: The cumulative volume carried in all routes in the solution.
18. **Field:** `assigned_orders`
    Type: Number (integer).
    Description: The number of orders that have been assigned to a vehicle by the optimiser.
19. **Field:** `unassigned_orders`
    Type: Array of strings.
    Description: An array containing the order identifiers of any unassigned orders. An empty array means that there are no unassigned orders.
20. **Field:** `unassigned_reasons`
    Type: Array of unassigned reason objects.
    Description: The results of the unassigned order analysis for any unassigned orders.
21. **Field:** `total_orders`
    Type: Number (integer).
    Description: The total number of orders in the scenario. This is also the length of the orders array in the scenario.
22. **Field:** `used_vehicles`
    Type: Number (integer).
    Description: The number of vehicles used in the solution.
23. **Field:** `total_vehicles`
    Type: Number (integer).
    Description: The total number of vehicles in the scenario.
24. **Field:** `loads`
    Type: Number (integer).
    Description: If batched loads is `true`, then the total number of loads in the solution.
25. **Field:** `km_per_load`
    Type: Number.
    Description: If batched loads is `true` and Use Miles is `false`, then this field gives the average number of kilometres per load in the solution.
26. **Field:** `miles_per_load`
    Type: Number.
    Description: If batched loads and Use Miles are both `true`, then this field gives the average number of miles per load in the solution.
27. **Field:** `drops_per_load`
    Type: Number.
    Description: If batched loads is `true`, then this field gives the average number of drops per load in the solution.
28. **Field:** `hours_per_load`
    Type: String (*HH:MM*).
    Description: If batched loads is `true`, then this field gives the average time per load.
29. **Field:** `pickup_soft_time_window_early`
    Type: String (*HH:MM*).
    Description: If there are orders with pickup soft time windows, then this field gives the total amount of time that orders were picked up early.

30. **Field:** `pickup_soft_time_window_late`
    Type: String (*HH:MM*).
    Description: If there are orders with pickup soft time windows, then this field gives the total amount of time that orders were picked up late.
31. **Field:** `delivery_soft_time_window_early`
    Type: String (*HH:MM*).
    Description: If there are orders with delivery soft time windows, then this field gives the total amount of time that orders were delivered early.
32. **Field:** `delivery_soft_time_window_late`
    Type: String (*HH:MM*).
    Description: If there are orders with delivery soft time windows, then this field gives the total amount of time that orders were delivered late.
33. **Field:** `iteration`
    Type: Number (integer).
    Description: The iteration that generated this solution.
34. **Field:** `total_iterations`
    Type: Number (integer).
    Description: The total number of iterations that the optimisation that generated this solution was set to run for.
35. **Field:** `errors`
    Type: Array of strings.
    Description: A list of errors in the solution; will be `null` or empty if there are no errors.
    Note: This field provides feedback when a solution is modified *after* optimisation.
36. **Field:** `warnings`
    Type: Array of strings.
    Description: A list of warnings about the solution; will be `null` or empty if there are no warnings.
37. **Field:** `date_generated`
    Type: String.
    Description: A string giving the day, date, time and year the solution was generated.
    Example: `Fri Jul 07 10:12:18 AEST 2017`
38. **Field:** `scenario`
    Type: String.
    Description: The name of the scenario from which the solution was generated. Matches the `name` field of the general object in the scenario object.
39. **Field:** `run_time`
    Type: String (*HH:MM:SS*).
    Description: The length of time the optimiser ran for.
40. **Field:** `exclusive_order_summary`
    Type: Array of exclusive order summary objects.
    Description: A summary of all exclusive orders in the scenario.

6.2 Route Object

A `route` object describes the planned route for a single vehicle. It has the following fields:

1. **Field:** `route_id`
   Type: Number (integer).
   Description: A number assigned to the route by the optimiser. It is the index of the route in the `routes` field of a solution object.

2. **Field:** `vehicle_id`
   Type: String.
   Description: The identifier of the vehicle assigned to this route.

3. **Field:** `stops`
   Type: Array of stop objects.
   Description: See Section 6.3 for further information.

   The following fields have been deprecated. See Section 6.5 for further information

4. **Field:** `cost`
   Type: Number.
   Description: The total cost of this route.

5. **Field:** `distance`
   Type: Number.
   Description: The total transit distance during the route.

6. **Field:** `loaded_distance`
   Type: Number.
   Description: The total transit distance travelled in the route while the vehicle is carrying at least one order.

7. **Field:** `empty_distance`
   Type: Number.
   Description: The total transit distance travelled in the route while the vehicle is empty.

8. **Field:** `delivery_distance`
   Type: Number.
   Description: The transit distance while making deliveries as part of the route.

9. **Field:** `work_time`
   Type: String (*HH:MM*).
   Description: The total time elapsed from the `START` to the `FINISH` stop in the route.

10. **Field:** `transit_time`
    Type: String (*HH:MM*).
    Description: The time spent in transit in the route.

11. **Field:** `service_time`
    Type: String (*HH:MM*).
    Description: The service time incurred in the route. (See Service Time in Section 2.12.)

12. **Field:** `site_time`
    Type: String (*HH:MM*).
    Description: The site time incurred in the route. (See Site Time in Section 2.10.)

13. **Field:** `idle_time`
    Type: String (*HH:MM*).
    Description: The time spent idle in the route.

14. `break_time`
    Type: String (*HH:MM*).
    Description: The time spent on breaks in the route.

15. **Field:** `actual_start_time`
    Type: String (*HH:MM*).
    Description: The time the vehicle started the route.

16. **Field:** `actual_finish_time`
    Type: String (*HH:MM*).
    Description: The time the vehicle finished the route.
17. **Field:** `weight`
    Type: Number.
    Description: The cumulative weight carried by the vehicle in the route.
18. **Field:** `volume`
    Type: Number.
    Description: The cumulative volume carried by the vehicle in the route.
19. **Field:** `assigned_orders`
    Type: Number.
    Description: The number of orders assigned to the vehicle in the route.
20. **Field:** `number_of_loads`
    Type: Number (integer).
    Description: If `batched_loads` is `true`, this is the number of loads in the route. (See Section 2.22).
21. **Field:** `loads`
    Type: If `batched_loads` is `true`, this is an array of load_kpi objects.
    Description: See Section 6.6 for further information.

6.3 Stop Object

Each `stop` object in the `stops` field of a route has the following fields:

1. **Field:** `stop_id`
   Type: Number (integer).
   Description: A number given to the stop by the optimiser. Matches the stop's location in the `stops` array.
2. **Field:** `stop_type`
   Type: String (a stop type code).
   Description: The type of stop.
3. **Field:** `order`
   Type: String.
   Description: For `stop_type` `PICKUP` or `DELIVERY` the identifier of the order being picked up or delivered. This matches the `id` of an order. For other types of stop this will be `null`.
4. **Field:** `location`
   Type: String.
   Description: The identifier of the location at which the `stop` occurs. This matches the `id` of a location.
5. **Field:** `arrival_time`
   Type: String (*HH:MM*).
   Description: The time at which the vehicle is planned to arrive at the stop location.
6. **Field:** `departure_time`
   Type: String (*HH:MM*).
   Description: The time at which the vehicle is planned to depart the stop location.
7. **Field:** `weight`
   Type: Number.
   Description: The total weight loaded on the vehicle at the departure from the stop.
8. **Field:** `volume`
   Type: Number.
   Description: The total volume loaded on the vehicle at the departure from the stop.
9. **Field:** `transit_time`
   Type: String (*HH:MM*).
   Description: The time spent in transit between the previous stop and this one.
   Note: If the `stop` has `stop_type` `START` or `BREAK`, then this will be `00:00`.
10. **Field:** `transit_distance`
    Type: Number.
    Description: The transit distance between the previous stop and this one.
    Note: If the `stop` has `stop_type` `START` or `BREAK`, then this will be `0`.
11. **Field:** `compartment`
    Type: String.
    Description: For `stop_type` `PICKUP` or `DELIVERY` where the compartment scheme (`compartments` field) for the vehicle is not `null`, this will be the identifier of the compartment that the order is assigned to. For other stop types, or where the vehicle lacks a compartment scheme, this will be `null`.
12. **Field:** `soft_time_window_start`
    Type: String (*HH:MM*).
    Description: For `PICKUP` stops where the order has pickup soft time windows, this field is the start time of the pickup soft time window closest to the stop's `arrival_time`. For `DELIVERY` stops where the order has delivery soft time windows, this field is the start time of the delivery soft time window closest to the stop's `arrival_time`. For other stop types, for `PICKUP` stops for orders with no pickup soft time windows or for `DELIVERY` stops for orders with no delivery soft time windows this field is `null`.

13. **Field:** `soft_time_window_end`
Type: String (*HH:MM*).
Description: For `PICKUP` stops where the order has pickup soft time windows, this field is the end time of the pickup soft time window closest to the stop's `arrival_time`. For `DELIVERY` stops where the order has delivery soft time windows, this field is the end time of the delivery soft time window closest to the stop's `arrival_time`. For other stop types, for `PICKUP` stops for orders with no pickup soft time windows or for `DELIVERY` stops for orders with no delivery soft time windows this field is `null`.

14. **Field:** `soft_time_window_delta`
Type: String (*HH:MM*).
Description: For `PICKUP` stops where `arrival_time` falls outside of the closest pickup soft time window, this field is either the amount of time from the arrival time to the start of the of the closest pickup soft time window (for early arrivals) or the amount of time from the end of the closest pickup soft time window (for late arrivals). This field is `00:00` if the arrival time is within a pickup soft time window. For `DELIVERY` stops, the value of this field is similarly defined except that it is with respect to delivery soft time windows. For other stop types, `PICKUP` stops with no pickup soft time windows, or `DELIVERY` stops with no delivery soft time windows this field is `null`.

6.4 Stop Type Codes

Stop type codes give the type of a stop. The possible values are:

1. `START`
2. `PICKUP`
3. `DELIVERY`
4. `BREAK`
5. `FINISH`

The names of the stop type codes and their meanings are the same as given in Section 2.21.

6.5 Route KPI Object

A `route KPI` object contains the route KPIs for a single vehicle. It has the following fields:

1. **Field:** `route_id`
   Type: Number (integer).
   Description: A number assigned to the route by the optimiser. It is the index of the route in the `routes` field of a solution object.
2. **Field:** `vehicle_id`
   Type: String.
   Description: The identifier of the vehicle assigned to this route.
3. **Field:** `cost`
   Type: Number.
   Description: The cost for this route.
4. **Field:** `revenue`
   Type: Number.
   Description: The revenue for this route.
5. **Field:** `profit`
   Type: Number.
   Description: The profit or loss for this route. Profit is the difference between revenue and cost.
6. **Field:** `distance`
   Type: Number.
   Description: The total transit distance during the route.
7. **Field:** `loaded_distance`
   Type: Number.
   Description: The total transit distance travelled in the route while the vehicle is carrying at least one order.
8. **Field:** `time_to_finish`
   Type: String (*HH:MM*).
   Description: The total time elapsed between the final `DELIVERY` to the `FINISH` stop in the route.
9. **Field:** `distance_to_finish`
   Type: Number.
   Description: The total transit distance travelled between the final `DELIVERY` to the `FINISH` stop in the route.
10. **Field:** `empty_distance`
    Type: Number.
    Description: The total transit distance travelled in the route while the vehicle is empty.
11. **Field:** `delivery_distance`
    Type: Number.
    Description: The transit distance while making deliveries as part of the route.
12. **Field:** `work_time`
    Type: String (*HH:MM*).
    Description: The total time elapsed from the `START` to the `FINISH` stop in the route.
13. **Field:** `transit_time`
    Type: String (*HH:MM*).
    Description: The time spent in transit in the route.
14. **Field:** `service_time`
    Type: String (*HH:MM*).
    Description: The service time incurred in the route. (See Service Time in Section 2.12.)
15. **Field:** `site_time`
    Type: String (*HH:MM*).
    Description: The site time incurred in the route. (See Site Time in Section 2.10.)

16. **Field:** `idle_time`
    Type: String (*HH:MM*).
    Description: The time spent idle in the route.
17. **Field:** `break_time`
    Type: String (*HH:MM*).
    Description: The time spent on breaks in the route.
18. **Field:** `actual_start_time`
    Type: String (*HH:MM*).
    Description: The time the vehicle started the route.
19. **Field:** `actual_finish_time`
    Type: String (*HH:MM*).
    Description: The time the vehicle finished the route.
20. **Field:** `weight`
    Type: Number.
    Description: The cumulative weight carried by the vehicle in the route.
21. **Field:** `volume`
    Type: Number.
    Description: The cumulative volume carried by the vehicle in the route.
22. **Field:** `assigned_orders`
    Type: Number.
    Description: The number of orders assigned to the vehicle in the route.
23. **Field:** `pickup_soft_time_window_early`
    Type: String (*HH:MM*).
    Description: If there are orders with pickup soft time windows, then this the total amount of time that orders were picked up early in the route.
24. **Field:** `pickup_soft_time_window_late`
    Type: String (*HH:MM*).
    Description: If there are orders with pickup soft time windows, then this is the total amount of time that orders were picked up late in the route.
25. **Field:** `delivery_soft_time_window_early`
    Type: String (*HH:MM*).
    Description: If there are orders with delivery soft time windows, then this is the total amount of time that orders were delivered early in the route.
26. **Field:** `delivery_soft_time_window_late`
    Type: String (*HH:MM*).
    Description: If there are orders with delivery soft time windows, then this is the total amount of time that orders were delivered late in the route.
27. **Field:** `number_of_loads`
    Type: Number (integer).
    Description: If `batched_loads` is `true`, this is the number of loads in the route. (See Section 2.22).
28. **Field:** `load_kpis`
    Type: If `batched_loads` is `true`, this is an array of load_kpi objects.
    Description: See Section 6.6 for further information.

6.6 Load KPI Object

A `load KPI` object contains the KPIs for an individual load within a single vehicle's route.

1. **Field:** `load_id`
   Type: Number (integer).
   Description: An integer assigned to each distinct load for a route by the optimiser. This is the index of the load in the `loads` field of a solution object.
2. **Field:** `start_stop_id`
   Type: Number (integer).
   Description: The `stop_id` of the stop which starts the load.
3. **Field:** `end_stop_id`
   Type: Number (integer).
   Description: The `stop_id` of the stop which ends the load.
4. **Field:** `cost`
   Type: Number.
   Description: The cost for the load.
5. **Field:** `revenue`
   Type: Number.
   Description: The revenue for the load.
6. **Field:** `profit`
   Type: Number.
   Description: The profit or loss for the load. Profit is the difference between revenue and cost.
7. **Field:** `loaded_distance`
   Type: Number.
   Description: The transit distance from the first pickup to the last delivery in the load.
8. **Field:** `empty_distance_before`
   Type: Number.
   Description: For the first load, this is the distance from the `START` stop to the first pickup; otherwise it is the distance from the final delivery of the previous load to the first pickup of this load.
9. **Field:** `empty_distance_after`
   Type: Number.
   Description: For the final load, this is the distance from the final delivery to the `FINISH` stop; otherwise it is the distance from the final delivery of this load to the first pickup of the next load.
10. **Field:** `delivery_distance`
    Type: Number.
    Description: The transit distance while making deliveries as part of the load.
11. **Field:** `work_time`
    Type: String (*HH:MM*).
    Description: The total time from the arrival at the first `PICKUP` stop to the departure of the last `DELIVERY` stop in the load.
12. **Field:** `transit_time`
    Type: String (*HH:MM*).
    Description: The time spent in transit during this load.
13. **Field:** `service_time`
    Type: String (*HH:MM*).
    Description: The service time during this load. (See Service Time in Section 2.12.)
14. **Field:** `site_time`
    Type: String (*HH:MM*).
    Description: The site time during this load. (See Site Time in Section 2.10.)
15. **Field:** `idle_time`
    Type: String (*HH:MM*).

Description: The time spent idle for this load.

16. **Field:** `break_time`
    Type: String (*HH:MM*).
    Description: The time spent on break during this load.
17. **Field:** `weight`
    Type: Number.
    Description: The cumulative weight carried by the vehicle in the load.
18. **Field:** `volume`
    Type: Number.
    Description: The cumulative volume carried by the vehicle in the load.
19. **Field:** `weight_utilization`
    Type: Number.
    Description: A number between [`0`, `1`] that represents the fraction of the vehicle's `maximum weight` used during the load.
20. **Field:** `volume_utilization`
    Type: Number.
    Description: A number between [`0`, `1`] that represents the fraction of the vehicle's `maximum volume` used during the load.
21. **Field:** `number_of_deliveries`
    Type: Number (integer).
    Description: The number of `DELIVERY` stops in the load.
22. **Field:** `assigned_orders`
    Type: Number (integer).
    Description: The number of orders assigned to this load.
23. **Field:** `pickup_soft_time_window_early`
    Type: String (*HH:MM*).
    Description: If there are orders with pickup soft time windows, then this is the total amount of time that orders were picked up early in the load.
24. **Field:** `pickup_soft_time_window_late`
    Type: String (*HH:MM*).
    Description: If there are orders with pickup soft time windows, then this is the total amount of time that orders were picked up late in the load.
25. **Field:** `delivery_soft_time_window_early`
    Type: String (*HH:MM*).
    Description: If there are orders with delivery soft time windows, then this is the total amount of time that orders were delivered early in the load.
26. **Field:** `delivery_soft_time_window_late`
    Type: String (*HH:MM*).
    Description: If there are orders with delivery soft time windows, then this is the total amount of time that orders were delivered late in the load.

6.7 Unassigned Reason Object

1. **Field:** `id`
   Type: String.
   Description: The identifier of the unassigned order.
2. **Field:** `reasons`
   Type: Array of strings.
   Description: Reasons why the order was not assigned.

6.8 Exclusive Order Summary Object

1. **Field:** `assigned`
   Type: String.
   Description: The identifier of the assigned order.

2. **Field:** `unassigned`
   Type: Array of strings.
   Description: The identifiers of the unassigned orders.

Here is an example of an exclusive order summary object.

```
{
  "assigned": "ORDER1",
  "unassigned": [
      "ORDER2"
  ]
}
```

Appendices

A JSON Input Example

This appendix contains an example of the JSON input required for creating a schedule for two vehicles and two orders to allocate. Site time is specified at each location. The vehicles operate using a cost-per-hour rate.

Time windows are specified at all locations (e.g. at the `Hallam` location, the time windows are 5AM to 10AM and 2PM to 6PM). There are also fatigue management rules in place (all drivers are required to take a single `35` minute break in every `4` hour period of work).


```
{
  "id": "8046e1d7-f713-447a-9fb0-3ce9115ade29",
  "general": {
    "name": "Example Scenario",
    "iterations": 3000,
    "iteration_scheme": "Batched",
    "colocated_pickups": true,
    "batched_loads": true,
    "rush_hour": true
  },
  "locations": [
    {
      "id": "DEPOT",
      "name": "Opturion Office",
      "address": "18 Kavanagh Street St, Southbank, VIC 3006",
      "longitude": 144.966167,
      "latitude": -37.82261,
      "site_time": "0:30",
      "attributes": [],
      "time_windows": [
        {
          "start": "5:00",
          "end": "10:00"
        },
        {
          "start": "11:00",
          "end": "18:00"
        }
      ]
    },
    {
      "id": "Hallam",
      "name": "Hallam ",
      "address": "11 Hallam Crescent",
      "site_time": "00:05",
      "longitude": 145.28085,
      "latitude": -38.018 ,
      "time_windows": [
        {
          "start": "5:00",
          "end": "10:00"
        },
        {
          "start": "11:00",
          "end": "18:00"
        }
      ]
```

```
    },
    {
      "id": "Skye",
      "name": "Skye",
      "address": "11 Skye Road",
      "site_time": "00:10",
      "longitude": 145.19373,
      "latitude": -38.12327,
      "time_windows": [
        {
          "start": "5:00",
          "end": "10:00"
        },
        {
          "start": "11:00",
          "end": "18:00"
        }
      ]
    }
  ],
  "orders": [
    {
      "id": "Order1",
      "pickup_location": "DEPOT",
      "delivery_location": "Hallam",
      "earliest_delivery_time": "10:30",
      "latest_delivery_time": "14:20",
      "delivery_service_time": "0:30",
      "pickup_time_windows":[
        {
          "start": "5:00",
          "end": "10:00"
        },
        {
          "start": "14:00",
          "end": "18:00"
        }
      ],
      "weight": 100,
      "volume": 3
    },
    {
      "id": "Order2",
      "pickup_location": "DEPOT",
      "delivery_location": "Skye",
      "earliest_delivery_time": "10:30",
      "latest_delivery_time": "14:20",
      "delivery_service_time": "0:30",
      "pickup_time_windows":[
        {
          "start": "05:00",
          "end": "10:00"
        },
        {
          "start": "11:00",
          "end": "18:00"
        }
      ],
```

```
      "weight": 300,
      "volume": 3
    }
  ],
  "fleet": [
    {
      "id": "Vehicle1",
      "registration": "Opturion-001",
      "driver": "D1",
      "start_location": "DEPOT",
      "finish_location": "DEPOT",
      "maximum_weight": 600,
      "maximum_volume": 10,
      "earliest_start_time": "6:00",
      "latest_start_time": "10:00",
      "latest_finish_time": "18:30",
      "maximum_drive_time": "8:00",
      "maximum_work_time": "9:00",
      "cost_per_use": 150,
      "cost_per_hour": 30.75,
      "cost_per_km": 0.2,
      "attributes": [],
      "break_scheme": "SH"
    },
    {
      "id": "Vehicle2",
      "registration": "Opturion-002",
      "driver": "D2",
      "start_location": "DEPOT",
      "finish_location": "DEPOT",
      "maximum_weight": 600,
      "maximum_volume": 10,
      "earliest_start_time": "6:00",
      "latest_start_time": "10:00",
      "latest_finish_time": "18:30",
      "maximum_drive_time": "8:00",
      "maximum_work_time": "9:00",
      "cost_per_use": 150,
      "cost_per_hour": 30.75,
      "cost_per_km": 0.2,
      "attributes": [],
      "break_scheme": "SH"
    }
  ],
  "fatigue_management": [
    {
      "id": "SH",
      "number": 1,
      "duration": "0:35",
      "interval": "4:00"
    }
  ]
}
```

B JSON Output Example

The appendix provides the JSON output example for the problem specified in Appendix A.


```
{
  "routes": [
    {
      "route_id": 0,
      "vehicle_id": "Vehicle1",
      "stops": [
        {
          "stop_id": 0,
          "stop_type": "START",
          "location": "DEPOT",
          "arrival_time": "09:55",
          "departure_time": "09:55",
          "transit_time": "00:00",
          "transit_distance": 0.0,
          "load_time": "00:00",
          "unload_time": "00:00",
          "site_time": "00:00",
          "service_time": "00:00"
        },
        {
          "stop_id": 1,
          "stop_type": "PICKUP",
          "order": "Order1",
          "location": "DEPOT",
          "arrival_time": "09:55",
          "departure_time": "10:25",
          "transit_time": "00:00",
          "transit_distance": 0.0,
          "load_time": "00:00",
          "unload_time": "00:00",
          "site_time": "00:30",
          "service_time": "00:00"
        },
        {
          "stop_id": 2,
          "stop_type": "PICKUP",
          "order": "Order2",
          "location": "DEPOT",
          "arrival_time": "10:25",
          "departure_time": "10:25",
          "transit_time": "00:00",
          "transit_distance": 0.0,
          "load_time": "00:00",
          "unload_time": "00:00",
          "site_time": "00:00",
          "service_time": "00:00"
        },
        {
          "stop_id": 3,
          "stop_type": "DELIVERY",
          "order": "Order1",
          "location": "Hallam",
          "arrival_time": "11:00",
          "departure_time": "11:35",
          "transit_time": "00:35",
          "transit_distance": 39.74,
          "load_time": "00:00",
          "unload_time": "00:00",
          "site_time": "00:05",
          "service_time": "00:30"
```

```
      },
      {
        "stop_id": 4,
        "stop_type": "DELIVERY",
        "order": "Order2",
        "location": "Skye",
        "arrival_time": "11:55",
        "departure_time": "12:35",
        "transit_time": "00:20",
        "transit_distance": 20.93,
        "load_time": "00:00",
        "unload_time": "00:00",
        "site_time": "00:10",
        "service_time": "00:30"
      },
      {
        "stop_id": 5,
        "stop_type": "BREAK",
        "arrival_time": "13:18",
        "departure_time": "13:53",
        "transit_time": "00:00",
        "transit_distance": 0.0,
        "load_time": "00:00",
        "unload_time": "00:00",
        "site_time": "00:00",
        "service_time": "00:00"
      },
      {
        "stop_id": 6,
        "stop_type": "FINISH",
        "location": "DEPOT",
        "arrival_time": "13:55",
        "departure_time": "13:55",
        "transit_time": "00:45",
        "transit_distance": 52.39,
        "load_time": "00:00",
        "unload_time": "00:00",
        "site_time": "00:00",
        "service_time": "00:00"
      }
    ]
  }
],
"route_kpis": [
  {
    "route_id": 0,
    "vehicle_id": "Vehicle1",
    "cost": 277.67,
    "revenue": 0,
    "profit": -277.67,
    "distance": 113.06,
    "delivery_distance": 60.67,
    "loaded_distance": 60.67,
    "empty_distance": 52.39,
    "work_time": "04:00",
    "transit_time": "01:40",
    "service_time": "01:00",
    "site_time": "00:45",
    "load_time": "00:00",
    "unload_time": "00:00",
    "idle_time": "00:00",
    "break_time": "00:35",
    "weight": 400.0,
    "volume": 6.0,
```

```
      "average_speed": 67.84,
      "peak_weight": 400.0,
      "peak_volume": 6.0,
      "assigned_orders": 2,
      "number_of_deliveries": 2,
      "number_of_loads": 1,
      "load_kpis": [
        {
          "load_id": 0,
          "start_stop_id": 1,
          "end_stop_id": 4,
          "cost": 244.13,
          "revenue": 0,
          "profit": -244.13,
          "distance": 60.67,
          "delivery_distance": 60.67,
          "maximum_delivery_distance": 39.74,
          "loaded_distance": 60.67,
          "empty_distance_before": 0.0,
          "empty_distance_after": 52.39,
          "work_time": "02:40",
          "transit_time": "00:55",
          "service_time": "01:00",
          "site_time": "00:35",
          "idle_time": "00:10",
          "break_time": "00:00",
          "load_time": "00:00",
          "unload_time": "00:00",
          "weight": 400.0,
          "volume": 6.0,
          "average_speed": 66.19,
          "weight_utilization": 0.67,
          "volume_utilization": 0.6,
          "number_of_deliveries": 2
        }
      ]
    }
  ],
  "cost": 277.67,
  "revenue": 0,
  "profit": -277.67,
  "distance": 113.06,
  "delivery_distance": 60.67,
  "maximum_delivery_distance": 60.67,
  "loaded_distance": 60.67,
  "empty_distance": 52.39,
  "work_time": "04:00",
  "transit_time": "01:40",
  "idle_time": "00:00",
  "break_time": "00:35",
  "weight": 400.0,
  "volume": 6.0,
  "average_speed": 67.84,
  "assigned_orders": 2,
  "unassigned_orders": [],
  "total_orders": 2,
  "used_vehicles": 1,
  "total_vehicles": 2,
  "loads": 1,
  "km_per_load": 113.06,
  "miles_per_load": 0.0,
  "hours_per_load": "04:00",
  "drops_per_load": 2.0,
  "scenario": "dtoExampleScenario",
```

```
  "objective": 295.0,
  "iteration": 3000,
  "total_iterations": 3000,
  "date_generated": "Thu Jul 19 15:29:34 AEST 2018",
  "run_time": "00:00"
}
```

C Unsupported Excel Functions

The following table lists all the Excel functions that are *not* supported by the optimiser.

ABSREF	DBCS	GET.CELL	LOGEST	RSQ
APP.TITLE	DDB	GET.DOCUMENT	LOGINV	SAVE.TOOLBAR
ARGUMENT	ENABLE.TOOL	GET.WINDOW	LOGNORMDIST	SEARCHB
ASC	END.IF	GET.WORKBOOK	MIDB	SKEW
BETADIST	ERROR	GET.WORKSPACE	N	SLN
BETAINV	EVALUATE	GETPIVOTDATA	NEGBINOMDIST	STEP
BINOMDIST	EXEC	GOTO	NUMBERSTRING	STEYX
CALL	EXPONDIST	GROWTH	PERMUT	SYD
CELL	FDIST	HARMEAN	PHONETIC	TINV
CHIDIST	FINDB	HYPGEOMDIST	PRESS.TOOL	TRIMMEAN
CHIINV	FINV	INFO	PROB	TTEST
CHITEST	FISHER	ISPMT	QUARTILE	TYPE
CONFIDENCE	FISHERINV	KURT	REGISTER.ID	USDOLLAR
CRITBINOM	FTEST	LAST.ERROR	RELREF	VDB
DATEDIF	GAMMADIST	LEFTB	REPLACEB	WEIBULL
DATESTRING	GAMMAINV	LENB	RETURN	WINDOW.TITLE
DB	GAMMALN	LINEST	RIGHTB	ZTEST