

WEBFLEET Interface

Aptean Ltd
Copyright © 2011-2026.

Contents

1 WEBFLEET Interface.....	1
1.1 Solution.....	1
1.2 Functional Description.....	1

1 WEBFLEET Interface

This page describes the interfaces between C-TMS and TomTom to handle sending out orders and receiving back debrief information at stop location level.

1.1 Solution

WEBFLEET will be configured to notify C-TMS of specific events that have occurred by the driver, through a queue that will be configured at implementation. The events that will be captured are:

- Acceptance
- Completion
- Automatic Arrival/Depart through GeoFence breaks at location (if these have been set up in WEBFLEET).

Each event will capture:

- Date and Time
- Vehicle
- Driver
- ODO reading
- Fuel Drawn
- ETA (for acceptance)

This information is periodically uploaded into C-TMS and will be stored against the trip and trip stops. Modifications will be required to this interface.

Fuel Drawn will be captured through this interface. This must automatically detect and calculate the fuel drawn for the trip. To confirm, this term means fuel used by the engine, rather than drawn into the vehicle fuel tank from a bunker or pump.

The date and time information for arrival and departures against the stops will always over-write any information that may already be in the database for this - this is to recognise that TomTom updates are the prime data for these times, as opposed to C-EPOD.

If all arrive/depart information is entered automatically by this process, C-TMS will automatically change the status of the trip to "COMPLETED". Care must be taken with the interaction between the C-EPOD and TomTom automated update processes, that the successful completion of one does not prevent the other from updating data.

Note: where non-conformance reason codes are updated into C-TMS from TomTom or C-EPOD or manually entered, C-TMS will be developed so if the reason code is configured to require an email alert to be generated, this output will be generated. This relies on a modification to the reason codes maintenance. The recipient email addresses will be setup against locations corresponding to the owning depot of the trip.

1.2 Functional Description

The TomTom web service offers some possible actions for sending and retrieving data to interact with Webfleet.

C-TMS will be developed to use the following available actions to pass trip stop information to TomTom and to retrieve the debrief information from TomTom for updating and storing in C-TMS.

?sendDestinationOrderExtern? will be used to send the orders (stops) to TomTom and will be used in principle to initiate navigation instructions to the driver.



?createQueueExtern?

will be used to create a queue initially (or via web page)

?popQueueMessagesExtern?

will be used to extract debrief messages from the TomTom queue.

?ackQueueMessagesExtern?

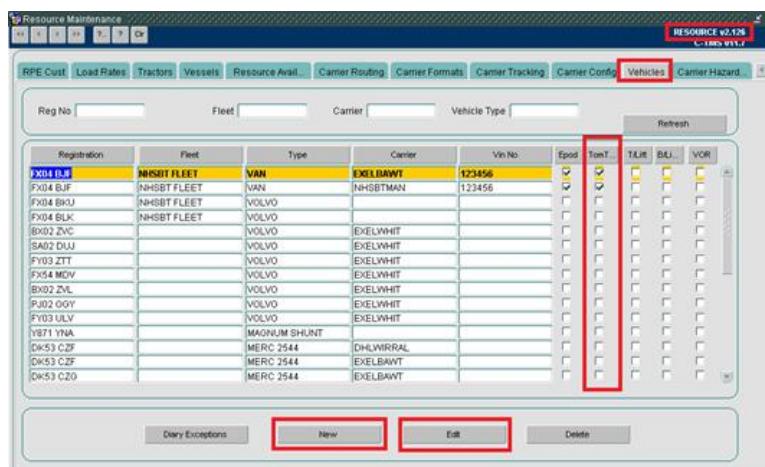
will be used to delete the messages from the TomTom queue.

?deleteOrderExtern?

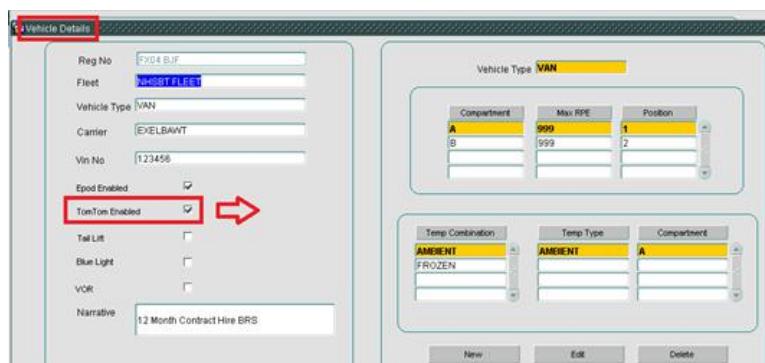
will be used for the deletion of orders (stops) from TomTom when trips are changed after they have already been sent

1.2.1 TomTom Control

Controlling whether a trip requires a TomTom message to be sent will be controlled at vehicle level so the vehicles tab on the Resource Maintenance form (RESOURCE.fmb) will be modified to add TOMTOM as a Tracking System option (as a flag) when adding or editing vehicles..



In addition the system will hold the external system reference for each vehicle. This will be added alongside the TomTom Enabled flag (new field TRACK_REF). This is the unique reference for the vehicle as configured in Webfleet.



1.2.2 Send Orders to TomTom

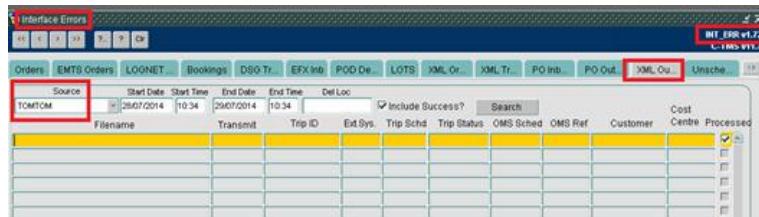
Webfleet and TomTom use the concept of Orders to provide driver instructions to navigate and complete an activity. C-TMS will generate an order into Webfleet, which updates the TomTom device for the vehicle, for each trip stop of each trip planned in C-TMS for the vehicle assigned to the trip.

The C-EPOD solution will be used to confirm order fulfilment, proof of collection loading and proof of delivery.



The initial trigger point for sending the orders from C-TMS to Webfleet will be at trip status change when the trip status is set to ?ACCEPTED?.

The generated control record can be seen on the Interface Errors forms on the XML Outbound tab in the screen shown below:-



A new package called DP_TOMTOM will be developed that will contain a procedure (SEND_ORDERS). This will be setup as a database job during the implementation to run at regular intervals (say every 1 minute).

This procedure will retrieve all of the unprocessed control records for the external system TomTom, build an appropriate html request per trip stop (for those with no actual times already set - see later section of this document covering updates), send the request to the TomTom web service, wait for a response and then process the response by updating the control record accordingly once all of the stops have been sent.

NB) System parameters will be utilized to control the web service url commands, the account, the username and the password. Also any other parameters required to generate the http calls.

New System Parameters

Name	Description	Value
GEO_TOMTOM_PROXY	Proxy server setting	
GEO_TOMTOM_TIMEOUT	HTTP wait for response before timeout	10
GEO_TOMTOM_ACCOUNT	Account name for the webservice	logistics-obs
GEO_TOMTOM_USERNAME	User name for TomTom webservice	gritter
GEO_TOMTOM_PASSWORD	Password for TomTom webservice	Password123
GEO_TOMTOM_SENDORD_URL	URL for sending orders to TomTom	https://connect.webfleet.tomtomwork.com/extern?
GEO_TOMTOM_GETMSGS_URL	URL for getting the messages from the TomTom queue	https://connect.webfleet.tomtomwork.com/extern?

The format for the http request will be as follows:-

```
https://connect.webfleet.tomtomwork.com/extern?account=logistics-obs&username=gritter&password=Password123
&lang=en&action=sendDestinationOrderExtern&objectno=002&orderid=LM0003&ordertext=NewObject
&latitude=53349616&longitude=-02856274&country=GB&zip=L24 9HZ&city=Liverpool&street=Speke Boulevard
```

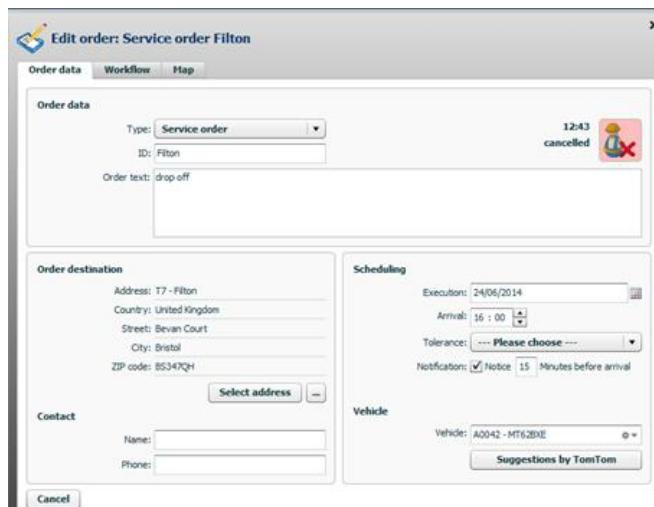
Send Orders (sendDestinationOrderExtern) Parameters

Tag	Description	C-TMS Table	Field
objectno	Vehicle registration of the trip (e.g. TomTom Device)	SCH_TRIP_STOP to RES_TRACTOR	TRAILER_ID to TRACK_REF
orderid	Unique Reference	SCH_TRIP_STOP	STOP_ID
ordertext	Free Text	See below	See below
latitude	Latitude for stop location	GEO_LOCATION	LATITUDE



longitude	Longitude for stop location	GEO_LOCATION	LONGITUDE
country	Country for stop location	GEO_LOCATION	
zip	Post Code	GEO_LOCATION	POSTCODE
city	Town	GEO_LOCATION	TOWN
street	Address Line 1	GEO_LOCATION	ADDRESS_LINE1
ordertype	2=collect (load), 3=delivery (unload)	Calculated from SCH_TRIP_STOP	STOP_TYPE
orderdate	Date (YYYY-MM-DD)	SCH_TRIP_STOP	ARRIVE
ordertime	Time(HH:MI:SS)	SCH_TRIP_STOP	ARRIVE
arrivaltolerance	Late arrival (0,15,30,60,120)	Override default ETA message sending at trip stop level.	
notify_enabled	0 (No) or 1 (Yes)	Override default notification flag of an arrival at trip stop level.	
notify_leadtime	Minutes	Override default notification minutes of an arrival at trip stop level.	
contact	Contact name	GEO_CONTACT	FORENAME / SURNAME
contacttel	Contact phone number	GEO_CONTACT	PHONE

The orders (stops) data sent over the API will be visible from within Webfleet on the order screen which will look like:-



The Order Text will contain several pieces of information useful for the driver;

- The name for the location (GEO_LOCATION.NAME) of the stop.
- The route code for the trip (SCH_TRIP.ROUTE_CODE).
- A list of distinct order types at the stop (SCH_ORD.DELIVERY_TYPE_ID).

When a collection has also been planned at the same location as a delivery then there is no requirement to send any details about the collection. This integration function is used to instruct the driver on the TomTom device (via the Webfleet back office system) to navigate to the next location, so not required for the subsequent collection event at the same location.

Note that the vehicle included in the interface message will be derived from the trip stop so that vehicle swap scenarios will be supported. In other words, stop 1 and 2 of a trip can be for vehicle A and stop 3 and 4 of a trip can be for vehicle B.

1.2.3 Send Trip Updates

When trips are modified after having already been previously sent to the Webfleet system (trip must be at either status ?ACCEPTED? or ?EN-ROUTE?) then it will be necessary to mark the existing data within Webfleet as deleted before sending the new revised information.

This process will utilise the ?deleteOrderExtern? web service call.

Send Delete Order (deleteOrderExtern)

Parameters

Tag	Description	C-TMS Table	Field
orderid	Unique Reference	SCH_TRIP_STOP	STOP_ID
mark_deleted	?1? to delete		

Currently, C-TMS is coded to manage retransmission of replacement update messages to execution systems where trips, stops of trips and orders have changed through the use of triggers at various data levels (trip, stop, haulage activity, order line, items and reason codes).

For the TomTom interface, changes to order data or adding new orders to existing stops doesn't affect the contents of the message previously sent (except possibly changing what was sent in the order text) so the only existing triggers that need to be active are Trip (status change - TRG_SCH_TRIP_XML_INT) and Stop (Insert, delete, amending planned arrival time -TRG_STS_XML).

The interface solution includes populating the stop_id on the control records for the stop trigger only.

The trip trigger will create a record at trip level so the processing of the message will split this trip control record into control records for each stop.

This means that only the stops that are changed (detected by the trigger code) are required to be resent rather than the entire trip.

When the interface logic determines that a stop has already previously been sent to TomTom then the process will be to send a delete message (?deleteOrderExtern?) for the stop (with an appropriate suffix) before sending the new create message (?sendDestinationOrderExtern?) for the stop (with an appropriate suffix if required) with the new and revised details.

The interface logic for amendments will be written to consider changes to the vehicle (trailer_id on stop). The stop trigger (TRG_STS_XML_) will be modified to recognise a stop level vehicle change and initiate the stop delete and stop recreate messaging described above.

1.2.4 Message Queues - Retrieve Debrief Data

Webfleet will be configured to make available execution updates on a message queue. This configuration is a one-off setup that ODSL will action. C-TMS will be developed to interrogate the Webfleet message queue every few minutes, extract and upload the order actual activity events and update the C-TMS trip debrief information accordingly.

The message queue will be created manually via the Webfleet web page (or using the createQueueExtern webservice call) with the appropriate level of message class (to be confirmed but assumed to be level 4 Order Related) as a one-off implementation task.

The possible message class values supported by Webfleet are:-

0 - All messages (includes tracking-only messages and 4,5,7,8)

2 - All except position messages (includes 4,5,7,8, does not include tracking-only)

4 - Order related messages



5 - Driver related messages

7 - Status messages

8 - Text messages

15 - LINK.connect messages

Webfleet data captured from the TomTom devices in the vehicles will then be exposed and made available for extract on the message queue ready to be extracted by C-TMS initiated web service calls.

The messages will be retrieved by a ?popQueueMessagesExtern? call to the Webfleet API webservice passing the appropriate message class.

A new procedure (GET_MESSAGES) will be developed and added to the new TomTom package in C-TMS.

This procedure will be setup as a database job and configured to run at regular intervals, probably every few minutes.

The procedure will retrieve all outstanding messages from the Webfleet message queue (up to 500 messages per request) and for each (relevant) message it will process the results and create an appropriate audit record (INT_XML_TRIP_STOP) for each suitable message and update the appropriate actual_arrive, actual_depart or planned ETA time for the appropriate trip stop in C-TMS debrief.

These updates extracted and received into C-TMS can be viewed via the existing interface errors form on the XML Trip Stops tab using the source system of ?TOMTOM?.

Within each message the following data will be extracted:-

popQueueMessagesExtent Possible Useable Response fields

Tag	Description	C-TMS Processing
msg_type	Last 3 digits show type	To be confirmed (see below)
objectno	Vehicle (Tractor, Device)	Not required
odometer	also start_odometer and end_odometer	Trip Start and end odometer
pos_time	Date/Time	Actual arrive or depart on stop



orderno	Unique reference	Trip stop id
order_state	See below	Arrive or depart
order_type	2=Pickup,3=Delivery	Used with order_state
eta	Expected time of arrival	Stop_ETA_Time
msg_time	Date/Time	Instead of pos_time
surplus_data	Additional Data	See later.

The following message types can be used to determine times:-

Possible Message Types to be used for NHSBT

msg type Description

739 Arrival at site (Unload and Load)
 742 Departure from site (Unload and Load)
 440,450 Arrival and departing area
 714,715 Start and end breaks
 747 Arrival at Depot

or the following order state values can be used.

Possible order state values to be used instead of msg_type and order_type

order_state Description

- 221 Pick up started
- 222 Arrived at pick up (load)
- 225 Departed from pick up (load)
- 241 Delivery started
- 242 Arrived at delivery (unload)
- 245 Departed from delivery (unload)
- 401 Finished

Currently C-TMS uses the `order_state ?401?` message to identify the arrival time at the stop and the `?221? / ?241?` messages to identify the depart from the previous stop.

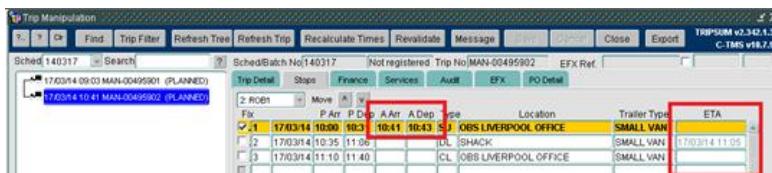
1.2.4.1 ETA Date

As part of some of the messages available from the Webfleet message queue, calculated by TomTom navigation, the current ETA date for the next stop location will be provided.

The latest ETA derived from the message queue for the stop provided will be used to populate the corresponding trip stops ETA date within C-TMS.

The outbound order message from C-TMS will set the arrivaltolerance /notify_enabled and /notify_leadtime parameters for each order if no default can be set on the Webfleet system to control ETA.

The ETA date and time, actual arrive date and time, and actual depart date and time once extracted from the message queue and updated in C-TMS can be viewed within the configurable section of the trip stops:-



And will be used to automatically update the trip debrief in C-TMS at trip stop level.



1.2.4.2 ODO Readings

As part of some of the messages available from the Webfleet message queue, the current ODOMeter reading of the vehicle is provided.

This will be used to populate the ODO start and end values for the trip which can be seen on the trip debrief form:-

The screenshot shows the 'Trip Debrief' screen with various buttons and fields. The 'ODO Start' and 'End' fields are highlighted with red boxes. Other fields like 'Fuel Drawn', 'Emissions', and 'MPG' are also highlighted. The screen displays trip details such as 'Trip Start' (10-04-2014 00:00) and 'Trip End' (11-04-2014 08:13), and trip times.

1.2.4.3 Messages Cleardown

Once C-TMS retrieves data from the Webfleet message queue the messages can then be deleted from the queue by issuing an ?ackQueueMessageExtern? call to the web service.

NB) The only parameter in addition to the authentication information for these calls is ?msgclass? (message class) as set when the queue was first created manually.

The response also includes the number of outstanding message which should be zero.

1.2.5 TomTom Non-Conformance

Where non-conformance reason codes are updated into C-TMS from TomTom or C-EPOD or manually entered, C-TMS will be developed so if the reason code is configured to require an email alert to be generated, this output will be generated. This relies on a modification to the reason codes maintenance. The recipient email addresses will be setup against locations corresponding to the owning depot of the trip.

Two new system parameters will be created and used to control under what circumstances a late reason will be generated from the TomTom updates.

The first parameter ?TOMTOM_LATE_MINS? will define the number of minutes that once exceeded will then generate a late reason against the orders on a stop and subsequently send out an e-mail event. This will be user configurable for the system.

The second parameter ?TOMTOM_LATE_REASON? and will define the reason code value to be used for the TomTom late reason.

When the actual arrival time is written to a stop in C-TMS from the TomTom interface, if it is after the planned time by more than the value of minutes in the ?TOMTOM_LATE_MINS? parameter then a late reason code will be generated (determined from the ?TOMTOM_LATE_REASON? parameter) for all of the orders at the stop and create a new event to send an e-mail.



An e-mail alert has been specified within the C-TMS - C-EPOD interface functional specification (see ?317907 NHSBT EPOD Interface? for more details) for when orders are not fully delivered and will be used as a basis for the new TomTom non-conformance e-mail.

