

**287419 v1.1**

Aptean Ltd  
Copyright © 2011-2025.

# Contents

<b>1 287419.....</b>	<b>1</b>
<b>2 Functional Overview.....</b>	<b>3</b>
2.1 Client Requirement.....	3
2.2 Solution.....	3
2.3 Scope.....	4
<b>3 Set-up.....</b>	<b>5</b>
3.1 Pre-requisites.....	5
3.2 System Registries.....	5
<b>4 Functional Description.....</b>	<b>6</b>
4.1 Outbound XML.....	6
4.2 Inbound XML.....	7
<b>5 AUTHORISED BY.....</b>	<b>8</b>

1 287419



DHL MTS

# Microlise Consolidate Stops

FUNCTIONAL SPECIFICATION - 10.6

- 1.1

Reference: FS 287419 AR-8FEFSS





## 2 Functional Overview

### 2.1 Client Requirement

On the flows into Microlise, consolidate CTMS delivery & collection stops into a single stop and consolidate stop orders into one consignment per stop.

Also need to deconsolidate the data on the return flows

In order to provide a usable view of data on the driver handhelds there is a requirement to consolidate delivery & collection stops in CTMS into a single stop.

In addition, in CTMS there is one cage per order. All the orders in a CTMS stop need to be consolidated into a single consignment in the Microlise interface.

This means that where CTMS has collection & delivery stops each with multiple orders, Microlise will have one stop with one collection consignment & one delivery consignment each containing the cages for collection or delivery at that stop.

These consolidated stops & orders will need to be deconsolidated on the return flows from Microlise.

As such adequate reference info needs to be used to support the interface processing

### 2.2 Solution

The solution below will make the interface consolidation an optional configuration using system parameters and so could be re-used going forward for other implementations.

#### Consolidation Stops

*Issue* - C-TMS trip building ensures that one activity, either loading or unloading is allocated to a trip stop. If the planner has both delivery orders and collection orders for the same location, C-TMS builds successive stops in the trip for the same location differentiated by the stop code DL delivery and PK collection respectively. Each trip stop in C-TMS has a unique stop sequence number that increments for each successive stop planned on the trip.

*Solution* - In C-TMS a new system parameter will be introduced called MICRO\_CONSOL\_STOPS. If set to Y, this will force consolidation of successive stops into the XML trip message created to send to Microlise via DHL Link. This in turn results in DHL Link mapping and transforming this into the Microlise journeyScheduleImport message as one site rather than two (with common site information).

During the message creation, C-TMS will read ahead to see if the subsequence stop of the trip is for the same location (and so adjacent in the stop schedule of the trip). If the configuration setting MICRO\_CONSOL\_STOPS is on, the XML will be generated with a single STOP segment comprising STOP\_HEADER, STOP\_DETAILS and ORDERS. The STOP\_SEQUENCE field definition will be extended and allow the two original 3 digit stop sequences (consolidated together on output) to appear concatenated as a six digit value. As an example, if stop 005 and 006, being DL and PK activity are for the same location, this will be interfaced in the STOP\_SEQUENCE as 005006. It is understood from reference information provided by Steve Bloomfield, that if the STOP\_SEQUENCE is mapped into the site attribute journeyDropReference, this value will be returned back to C-TMS via all the three Microlise flows podPoc, systemEvents and journeySummary.

When processing these message flows into C-TMS, the 6 digit STOP\_SEQUENCE will be detected and the Microlise updates (actual dates and times) made to both the two original C-TMS stops. In other words, using the example mentioned, both of the trip stops with sequence 005 and with sequence 006 will be updated with same site information from Microlise.

#### Consolidate Orders

*Issue* - In C-TMS, each BG transport order represents a single delivery unit (a cage). Ideally, the operation wants the driver to confirm all the cages delivered and then all the cages collected at a location under two over-arching consignments on the MDT for the site. It is assumed that this is regardless of the origin source location(s) of the cages



for deliveries and regardless of the final delivery location(s) of the cages for collections at the site.

*Solution* - In C-TMS a new system parameter will be introduced call MICRO\_CONSOL\_ORDERS. If set to Y will force consolidation of all orders on a trip stop in C-TMS into one order for the XML message created to send to Microlise via DHL Link. The ORDER\_HEADER segment will be generated with a new ORDER\_TYPE value ?S? meaning Shipment. The SO\_REF will be generated with a concatenation of D or C (either delivery or collection), TRIP REF and STOP SEQUENCE to give a unique value that can be mapped into orderRef on Microlise journeyScheduleImport. No other ORDER\_HEADER values will be sent (other than the mandatory ORDER\_TRANSACTION\_DATE) and none of the other ORDER\_HEADER.. segments will be populated.

Each order line (across all of the orders being consolidated in C-TMS) will be generated into the XML sub-ordinate to the ?S? shipment header described above. A new field will be added into the ORDER\_DETAIL segment of the OBSL TripOrder XSD called SHIPMENT\_DETAIL. This will be populated with a concatenation of STOP\_SEQUENCE+TRIP\_REF+ITEM\_IDENTIFER delimited with hypens. This value will be used to represent the method each order line was consolidated and will be mapped into consDetailInfo1 for the journeyScheduleImport. This means that all of the order details (cages) in C-TMS orders on stops will appear in Microlise as single delivery and single collection consignment per site and each cage will be represented as a consignment detail. The consDetailInfo1 field is returned from Microlise in the podPoc message and if mapped back into SHIPMENT\_DETAIL will allow C-TMS to update the correct original order and order line in C-TMS.

## 2.3 Scope

This change will be applied to system version 10.6



## 3 Set-up

### 3.1 Pre-requisites

None.

### 3.2 System Registries

?MIC\_CONSOL\_STOPS? - Whether Microlise should consolidate any collection and delivery stops onto a single stop for the outbound XML flow with the corresponding orders also consolidated but at the original stop level.

i.e. A single consolidated order for deliveries and another one for collections.



## 4 Functional Description

A new version 2.15 of the standard TripOrder.xsd will be used to accommodate the new tag.

### 4.1 Outbound XML

Rather than having 2 separate system registries as per the solution section we will just use the new ?MIC\_CONSOL\_STOPS? registry to control the new code.

Within the package INT\_XML\_OUT2 the procedure PROCESS\_XML\_OUTBOUND\_MIC which deals with the production of the outbound XML file for Microlise will be changed to check the new system registry.

All of the standard code remains as is until stop information is added to the message output.

If the new system registry is set then the first thing that the process now needs to do is to get the next planned trip stop record (add one to the STOP\_NO and retrieve the corresponding SCH\_TRIP\_STOP for the trip).

If this has the same location (LOCATION\_ID) as the current stop then it is assumed that both collections and deliveries are occurring at this same stop.

Once it has been identified that the stops need to be consolidated then the following logic will apply:-

*Developer Notes: A new flag will need to be set (T\_IGNORE\_NEXT\_STOP) so as when the next stop is processed on the next time through the loop then it knows that it has already been dealt with so the stop can be totally ignored.*

i.e Stops 5 and 6 would become ?005006?.

the appropriate values to represent a consolidated stop :-

The arrive and depart times, for both planned and actual, will be set from the arrival times at the first stop and the departure times from the second stop.

This section will only have 2 orders where pick ups and drop offs occur at the same location:

The first order will represent all of the orders being delivered at the first stop and the second order will represent all of the orders being collected at the second stop.

This applies to both the collections order and the deliveries order.

For collections this will be ?C-? plus the rest as above.

i.e. ?005-654321-EL8153?.

This completes the new process for consolidating stops and orders.

If the next stop was for a different location then the stops will not be consolidated but if the system parameter is set to consolidate stops then we still need to consolidate the orders to a single order on the stop.

The stop header and detail will be created as now.

It will then create a single order header in the same format as the consolidated order above with the order type being ?S? (shipment).

It will then loop through all of the order items at the stop and create a corresponding order details as per the consolidation above, including the new shipment detail tag.

This completes the new process for consolidating orders on a single stop.

Finally if we are not consolidating stops then the code will run as now.





## 4.2 Inbound XML

The existing procedure `PROCESS_MIC_TRIP_XML_IN` within the same package `INT_XML_OUT2` will need to be changed to bring it into line with the above changes for the consolidated stops and orders.

The standard code for extracting the data from the XML into the temporary fields does not need to change up to the point where it validates the stop.

Currently if the extracted stop sequence is 99 it retrieves the highest stop number on the trip in C-TMS and uses this number instead of the one provided in the XML. This code will be extended so that if the number is over 1000 then it will split the number into the 2 real stop numbers and then it will need to validate both of the stops.

It will also then need to update both of the stops with the appropriate information on both stops i.e. arrival date and time. But it will still only create a single stop interface record (`INT_XML_TRIP_STOP`).

A single order interface record (`INT_XML_ORD_HEADER`) will still be created but the `OMS_Ref` will be left blank.

NB) The new column `SHIPMENT_DETAIL` `varchar2(50)` will be added to the existing table `INT_XML_ORD_DETAILS` in order to store the new tag at order item level.

When checking that the order and its items are valid (using cursors `c_check_soi` and `c_ord`)

The rest of the code should be unaffected and will update the items and reason codes as current.

Some additional code is also required so that when the delivered quantity is updated on the items (`SCH_ORD_ITEMS.QTY_DELIVERED`) then this quantity is to be rolled up into the corresponding delivered quantity on the corresponding order line (`SCH_ORDER_LINE.ACTUAL_QUANTITY` for the same order and DU type).

This will only work if they have a single order line per DU type. If multiple lines exist for the same DU type then this updating will not know which order line to update.

### Document History

Version	Date	Status	Reason	Initials
0.1	06/04/11	Draft	Initial version	DRM
1.0	07/04/11	Issue	Reviewed and Issued	MJC
1.1	11/04/11	Draft	Changes following conference call	DRM



## 5 AUTHORISED BY

<b><i>Matt Crisford</i></b>	Development Manager
<b><i>Peter Greer</i></b>	TMSCC MTS Product Manager

