285532 v2.0

Contents

1 285532

aptean

DHL MTS

Stop Consolidation Options

FUNCTIONAL SPECIFICATION - 10.6

```
- 2.0
```

Reference: FS 285532 KM-8DEH2P

Ready for **What's Next, Now**™

Ready for **What's Next, Now**™

2 Functional Overview

2.1 Client Requirement

Change Request Summary:

Requirement to Consolidate orders into one stop where the consecutive orders are for the same destination. If however the orders have a return to base leg within the trip these must not be consolidated.Stuart Foster/Milton Keynes/UK/NFC

Change Request Details:

Requirement to Consolidate orders into one stop where the consecutive orders are for the same destination. Logic required to apply 30 mins plus 2 minutes per lift to calculate the window. This will then pass the detail to Microlise, and when the geofence is broken it will cover all the orders in one message. If however the orders have a return to base leg within the trip these must not be consolidated.

Benefits identified as a result of the change:

Will assist with LOTS / Microlise solution

2.2 Solution

Currently, when orders are consolidated at stops and there are no overlapping order windows, the stop times are set based on the latest order.

If the orders overlap the current functionality should be retained, however if the order windows do not overlap the stop times should be set based on the earliest order rather than the latest order.

Currently, the function STOP_WINDOW, determines the early arrive, late arrive, early depart and late depart for a stop based on the windows of the orders at the stop. To utilise the overlapping of windows, the greatest early arrive value is used. When orders do not overlap, this means that the latest order times are used,

Eg 1 |--------------x----------------------------|

|x--|

Eg 2|------------------------------------| x

|x---|

In example 1 where the order windows overlap, using the latest early collection time allows both orders to be collected within the order windows.

In example 2 where the orders do not overlap, using the latest early collection time means the first order is always late.

Within this functionality, we must determine if all the orders overlap, if so the existing functionality is applied. However if not all of the orders overlap, the earliest order windows should be applied. After the stop window has been determined, a cursor will be run to select any orders with windows outside of the stop window. If any orders are found, the stop window will be determined using the earliest window, otherwise the stop window calculated will be retained.

The additional checks will be controlled by a system parameter.

NB Stop consolidation only occurs when more than one order has the same activity at consecutive stops i.e. 2 loads or 2 unloads. If there is a load and an unload activity at the same location id the stops will NOT be consolidated

2.3 Scope

This change will be applied to system version 10.6.

Ready for **What's Next, Now**™

3 Set-up

3.1 Pre-requisites

None

3.2 Menu Structure

?Unchanged?

3.3 Data

Insert into ADM_SYSTEM_PARAM (param_name, value)

Values (?TRM_EARLIEST_STOP_WINDOW?,?N?)

Ready for **What's Next, Now**™

4 Functional Description

The STOP_WINDOW procedure was created to allow consolidation of orders at stops. This code was created to identify a stop window which allows all orders to be collected and delivered on time.

To achieve this, the stop window is always based on the latest early collection time and the earliest late collection time. The additional functionality allowing orders to be consolidated even when the windows do not overlap has resulted in the latest order window being applied for stops where the order window times do not overlap.

A change will be made to this procedure. The change will be controlled by a system parameter called TRM_EARLIEST_STOP_WINDOW. The existing parameter TRM_CONSOLIDATE_STOPS must also be set to Y.

If this functionality is customer specific, then the flag to indicate that the earliest window should be applied will be held at customer level rather than database level.

Added after development

When the original stop consolidation functionality is activated, for a consolidated trip stop to have the stop times based on the earliest time from the range of order collection/delivery windows as opposed to the latest time, the owning depot of the trip must be enabled to use early stop windows.

This setting will be held in the ?special? tab of the locations form. So, if we are creating a trip whose owning depot will be EXELBAWT, the location EXELBAWT will need this setting to be enabled in order for the trip to have early timings.

This setting is required along with the system parameters. If either the new TRM_EARLIEST_STOP_WINDOW parameter or the existing TRM_CONSOLIDATE_STOPS parameters are not activated, this location level setting will be ignored.
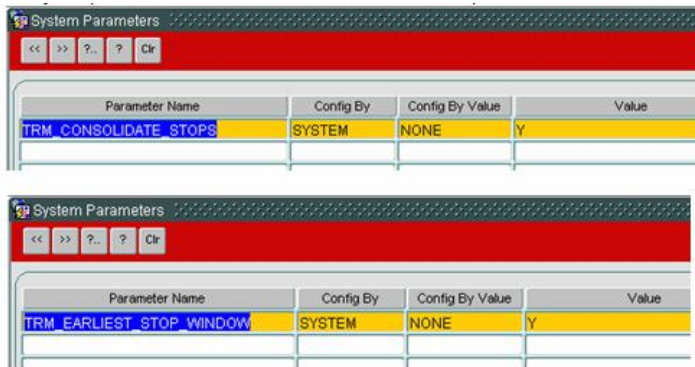
For example:

Trip with owning depot ?EXELBAWT?.

Ready for **What's Next, Now**™

Early Stop Window checkbox is checked for the trip:

The system wide parameters are also on; therefore the timings for the consolidated trip stop/s will be based on the earliest early.

The ?to? or ?from? locations for any orders on a trip do not need to be enabled for early stop windows. The setting must only be enabled for the owning depot and all stops on that trip will be consolidated where there are multiple orders being collected from / delivered to a particular location.

If all of the necessary activation parameters are set to Y, including the locations checkbox, and the trips can be successfully consolidated, a new cursor will run which will the collection or delivery windows for the orders at that particular stop with the stop window. The cursor will identify any orders outside of the stop window. If the orders is at the stop to be loaded, the stop window will be compared with the collection window for the order. If the stop is for unload activity, the stop window will be compared with the delivery window for each order.

If the stop is a combination of Load and Unload activity and there is more than one order for either or both activities, the load and unload cursor will be run to identify any orders which are outside of the stop window.

If there are any records returned from the cursor(s) , the stop_window will be recalculated based on the earliest order rather than the existing calculation.

Once the process establishes that order(s) fall outside of the calculated stop window, another cursor will run to identify the earliest collection or delivery order window to be applied.

Ready for **What's Next, Now**™

5 REFERENCES

Ref No	Document Title & ID	Version	Date
1	EST-285989 AD-8DQSF3 Develop Asset Tracking Functionality v1.0	1.0	N/A

Ready for **What's Next, Now**™

6 DOCUMENT HISTORY

Version	Date	Status	Reason	Initials
0.1	03/03/11	Draft	Initial version	SW
1.0	04/03/11	Issue	Reviewed and Issued	MJC
2.0	13/10/11	Issue	Reviewed After missing content	PC

Ready for **What's Next, Now**™

7 AUTHORISED BY

| Matt Crisford | Development Manager |
| Peter Greer | TMSCC MTS Product Manager |

Ready for **What's Next, Now**™

8 AUTHORISED BY

Matt Crisford	Development Manager
Peter Greer	TMSCC MTS Product Manager

Ready for **What's Next, Now**™