

271786 v1.0

Aptean Ltd
Copyright © 2011-2026.

Contents

1 271786.....	1
1.1 Client Requirement.....	2
1.2 Solution.....	2
1.3 Scope.....	3
2 Set-up.....	4
2.1 Pre-requisites.....	4
2.2 Database Changes.....	4
2.3 System Registries.....	4
2.4 Further Enhancements.....	4
3 Functional Description.....	6
3.1 Orders Export to Paragon.....	6
3.2 Trip Import from Paragon.....	6
4 AUTHORISED BY.....	9

1 271786

Aaptean

DHL MTS

Baxter Paragon Interface Improvements

FUNCTIONAL SPECIFICATION - 10.6

29/01/2010 - 1.0

Reference: FS 271786 NW-7Y2G6T



1.1 Client Requirement

Corrections to the Paragon Interface in the Healthcare environments.

The export from MTS needs to cater for multiple from locations as well as to locations, not just a single despatch location as is the current format.

Additionally the trunk legs created on the import from Paragon need to be consolidated by out-base rather than created as per radial delivery. Upon creation of these trunks they should be consolidated into single trips up to the capacity of a specified trailer.

1.2 Solution

Export from MTS :-

The export CSV from MTS has already been modified to cater for collection orders as well as delivery orders. For collections, the fields 1 through 10 in the attached specification are populated with the collection address data of the MTS order and field 17, the Order Type is set to value ?C?. For deliveries fields 1 through 10 are set as the delivery address and field 17 set to ?D?. This change has been delivered and tested successfully in HCRTST.

The MTS export CSV format design doesn't contain the original despatch location. It is understood that Paragon doesn't need this information while its focus is to plan the last leg delivery (or first leg collection). Paragon is aware of the network DC (or DCs) and out-bases and uses rules to define the DHL depot that will service the delivery or collection. MTS holds the order from and to locations so knows the despatch location when the Paragon response file is processed. While there is one central DC for the Baxter implementation, should multiple DCs be operated, this should cause issues with the integration solution in future.

Import into MTS :-

There are a number of issues arising from testing the CSV Trip import back into MTS which means the following solution enhancements are required;

MTS currently creates a trunk trip for each delivery route provided back from Paragon. MTS uses the Depot Code (field 9) to determine the delivering depot (this could be the NDC code DHLBAXT or out-base depot for example code DHLBELL) and the order From Loc to determine a trunk route. The Paragon X-Dock paths static data is then referenced to define the trunk routing so the number of legs and carrier for each.

Normally there will only be one trunk leg, however, the Paragon X-Dock paths does allow multiple trunk legs to be defined to support trailer swap at another depot or motorway service station.

Issues and enhancements;

Consolidated Trunk - Currently, MTS automatically creates a trunk trip for each radial delivery route. This logic will be modified to enable MTS to automatically create a consolidated trunk trip (or trips to the maximum capacity of the trailer utilised) from each origin location. The MTS Locations Slots static data (see point below regarding trunk SU times) will be utilised to define a trailer type for each trunk route to be defined; this enables MTS to understand the maximum capacity (weight, dims, RPE) of the consolidated trunk. The orders will be allocated to the trunk trip sorted by radial delivery route and drop within route. The consolidation will need to consider trailer swaps as defined in the Paragon X-Dock paths. If the consolidated trunk trip's capacity is exceeded, MTS will simply create another trip or trips until all the orders have been allocated.

Trunk Trip SU - Currently, the automatically generated trunk trips are scheduled to start based on the earliest collect date and time of the orders allocated. The Location Slot static data will be utilised to define a trunking schedule from the NDC to out-bases. This data structure allows a trunking schedule for each day of the week and slots will be entered with a minus day or days offset from the date of the Paragon radial delivery round. The slot data will allow the trunk start window at NDC and delivery window into out-base to be defined as well as the vehicle type (to control the capacity) and product type. The slot data will allow multiple trips to be defined for a particular out-base. The Paragon upload process will be modified to allocate orders to each trunk, earliest slot first and consolidate until the trailer capacity is reached before moving to the next slot.



Collections - MTS will be enhanced to allocate collections to trips from the information provided in the Paragon file. This means allocating an order onto a radial trip as a PK stop and setting the status to SCHED_COLL. MTS will not attempt to either create a trunk trip or allocate collections at out-bases to trunk trips back to the DC. The planners will allocate the SCHED_COLL collections to trunk vehicles (as the back-haul leg) manually.

Third Party Carriers - From testing, it has become apparent that Paragon uses field 9, described as the Depot Code, as either the out-base code when planning own fleet OR the name of the appointed third party if sub-contracted. MTS needs to know both these pieces of information to properly create trips so a change to the Paragon file is required. Field 9 should always define the DHL out-base and a new field should define the carrier code (not name). The Carrier code master data then needs to be synchronised in MTS and Paragon.

1.3 Scope

This change will be applied to system version 10.6.



2 Set-up

2.1 Pre-requisites

Latest version of Paragon Interface as changed under RIO AT-7W5J4V.

2.2 Database Changes

New Table PAR_TRIP_XDOCK to hold the staging post legs information for each order being passed from Paragon :-

Field	Type
ID	NUMBER
ROUTE	NUMBER
TRIP	NUMBER
OMS_REF	VARCHAR2(12)
XDOCK_SEQ	NUMBER
FROM_LOC	VARCHAR2(12)
TO_LOC	VARCHAR2(12)
RPE_QTY	NUMBER(8,2)
WEIGHT	NUMBER(20,2)

The table will have a unique primary key of ID, ROUTE, TRIP, OMS_REF, FROM_LOC, TO_LOC

The table PAR_TRIP_DTL also have the following column adding :-

Field	Type
CARRIER_ID	VARCHAR2(12)

Data also needs adding to the standing data table IMP_FIELD :-

Insert into IMP_FIELD (Field_Name, Import_Type) Values (?CARRIER?, ?PAR_TRIP_DTL?);

This new field can then be added manually in the appropriate position to the Paragon Trip Detail layout on the IMPORTS_MAINT form.

2.3 System Registries

No new system registries are required for this development.

2.4 Further Enhancements

The following are further enhancements being considered.

- Slot Updating outside of paragon interface (manual delete trip etc.)

Currently it is only the paragon interface that updates and uses the slot information for assigning and creating trunk trips. This means that if any manual changes are done to the PAR trips outside of the paragon interface (e.g. deleting trips, un-scheduling orders from trips etc.) then the slots are not maintained in line with these changes. This will result in slot space being retained in a state that is out of step with the trips and orders in MTS if these are manually changed post interface upload.

The original order slot table is then not compromised by manual changes to PAR trips and the bookings / Ti generation functionality needs no changes either]

- Slot Time using current time (now covered in the subsequent RIO NW-7ZELNS)



With the current solution, when assigning an order to a trunk trip, the upload process does not take into account the current time (i.e. the time that the interface file is being processed). The upload process is creating trunk trips according to a departure schedule. The upload process needs to understand some form of order cut-off for the trunk trips to allow for loading time and to ensure orders are not allocated to trunks that are, by their planned time, already en-route. In effect, each trunk needs to be closed to new orders at some agreed time ahead of loading and departure from the NDC.

- Schedule for radial trips not to be based on orders (also now covered in the subsequent RIO NW-7ZELNS)

During the Paragon file upload process, the schedule for each radial trip is determined using the schedule from the first order that is assigned to that radial trip (as per original MTS functionality carried forward). This causes failures when loading orders from different delivery days into a single paragon run and MTS trip schedule. The schedule for all trips generated from the paragon interface ought to be derived from the trip start date and time provided in the interface file from paragon (for each route which becomes a trip in MTS).

- Error trapping within generation radial and between trunk and radial

The paragon file is processed in two passes, one to create the trunks and one to create the radial delivery trips.

There is a possibility that if any errors occur in the trunk trip generation resulting in an order not being scheduled (SCCHED_COLL) onto the trunk, the order could still be considered for the radial delivery round. The resulting erroneous routing would show the out-base vehicle loading at the NDC in this event which could be undone manually.

Similarly, if all the available slots for the trunking schedule are exhausted, the same scenario will arise.



3 Functional Description

3.1 Orders Export to Paragon

The MTS export function was changed under the original Baxter RIO to match the extended output requirements for Paragon. One of the changes was to arrange quotes around each of the fields, this being a requirement of the version of Paragon in use.

Initially, MTS was developed to provide single quotes around each field which was later changed to double quotes. This method has proven to be a more robust solution and will deal with apostrophe in names, for example;

?Jack O?Neal? rather than ?Jack O?Neal?

The output CSV file to load paragon has already been changed to correctly interpret collections / returns. If the from location of an order in MTS is not a location type RDC, then the order is considered to be a collection / return. The record in the paragon file is created accordingly, setting the location codes and C/D flag for a collection.

3.2 Trip Import from Paragon

3.2.1 Import Package

The import package will be enhanced to process an additional column provided by Paragon. The additional column in position 10 will provide MTS with the carrier code of the nominated carrier for each radial trip whether own fleet or a third party.

The IMP package will be changed to extract the field from the provided Paragon CSV file.

The new field will then be validated via the RES_CARRIER table to ensure that it is a valid CARRIER_ID in MTS.

Once the entire line has been validated then the new field will be added to the creation of the row in the PAR_TRIP_DTL table populating the newly added column CARRIER_ID.

3.2.2 Paragon Package

The paragon package which represents the main upload functionality will be changed to use the newly added column CARRIER_ID from the PAR_TRIP_DTL table to populate the CARRIER_ID on the SCH_TRIP record for the radial trips. If a carrier code is not provided by paragon, then the process will continue to use the carrier from the corresponding Paragon X-Dock table as a default.

The existing trunk trip create function will continue to derive the CARRIER_ID from the corresponding Paragon X-Dock table.

Another much more significant change will be to the method and level of consolidation of orders to the trunk trips as they are generated.

Currently the trunk trips are only consolidated at radial trip level, meaning a trunk trip is created for each radial delivery trip. The requirement is to consolidate the trunk trips across the schedule allocating orders from multiple (complete) radial delivery trips to each trunk.

This new code will be activated using a configuration setting where the system registry value ?PARAGON_INSTALLED? is set to ?HCR?.

In order to group the radial trips together for trunking then the new table PAR_TRIP_XDOCK will be used. For every order on the paragon file the process will check the paragon cross-docks table for the ?orders from loc? and paragon ?delivery depot? and write a record to the new table for each trunk leg found along with the weight of the order. Also included on this table will be a sequence (trunk leg sequence) which will ensure that the trunk legs are processed in the correct order.

The existing slot master data associated with locations will be used to define the trunking schedule from the NDC to each out base. This data will be used by the paragon trunk leg planning process to determine which trunk trip will be generated and with what planned load and departure times (SU planned times).

Slots will be created at the unloading depot (meaning the out base) for each paragon cross-dock location with the secondary location being the loading depot (meaning NDC). The slot day will represent the day of the week that the radial



delivery is going to be made on as provided in the depot depart field by Paragon in the CSV upload file. The collect time and offset will be used to determine the time that the trunk leg departs after loading the orders. Against each slot, the trailer type which will be used to determine the maximum weight for each slot (each slot will represent a trunk trip on that day). It will also be used to populate the trailer type on the trunk trips. The Carrier will also be set on the slot and will be used to determine the Carrier for the trunk leg.



The PAR_TRIP_XDOCK table will then be summed at radial trip level for each trunk leg in trunk sequence. This will give a total weight for the radial trip for each trunk leg.

The delivery day for the trip along with the load and unload locations for the trunk leg will then be used to loop through all of the possible slots in collect window start time sequence. This will generate the first trunk trip for the schedule for the out base being processed.

Orders will be grouped by radial delivery round and the heaviest to lightest allocated to the trunk schedule in sequence. The process is designed to allocate groups of orders representing each radial delivery round to an individual trunk trip. This should minimise sorting at the out base.

The function that generates the radial trip will be enhanced to process collection orders as well as delivery orders.

The process will determine if the order is a collection or a delivery in the same way that the export package does based on from location not being an RDC (see above).

Only the first leg (loading at the orders ?From Loc? and unload at the delivery depot) will be planned on the radial trip for collections. This means only the collection into the out base will be automatically scheduled.

The trunks back from the delivery depot to the orders ?To Loc? will be planned manually in MTS on the next available return trunk leg to NDC - in essence the backhaul leg of the next trunk vehicle for the out base.

The enhancement for the radial trip and collections will be achieved by passing in the delivery depot as the ?To Loc? parameter when calling the standard TRM.Add_Ord_To_Trip procedure.

Also the times provided by Paragon in the CSV file for the arrival and departure from the location will need to update the load stop for collection orders and the unload stop for delivery orders.

Paragon Trip Detail (Import Maint)



Import Maintenance

IMPORTS_MAINT v1.13
MTS v10.5.0

Save Cancel Close

Config Decodes Decodes Audit Formulas

Format Name	Imp Type	Default Path	Diff Filename	Record Id	Filename Fmt	Xfer Type	Ext	Max Uploads	Default Record Type
Lane	LANE	c:\	lane_import.csv		ASCII	100			
Location	LOCATION	c:\temp\	location.csv		ASCII	100			
Order_Import	TL_ORDER	c:\temp\	orders.csv		ASCII	100			ORDER_AND_LIN
Paragon Trip Detail	PAR_TRIP_D	c:\temp\			ASCII	100			
TRAILER_TYPES	LOCATION	c:\temp\	trailer.csv		ASCII	100			

Fixed/Delimited Delimiter

Add Delete

Record Type	Field Type	Source Type	Occ	Source Value	Prefix	Pad	Char	Default	Format
STOP	FIELD	FIELD	0	4					
OMS_REF	FIELD	FIELD	0	5					
TRAILER_TYPE	FIELD	FIELD	0	7					
FROM_LOC	FIELD	FIELD	0	9					
DEPOT_DEPART	FIELD	FIELD	0	15					
STORE_DEPART	FIELD	FIELD	0	12					
DEPOT_RETURN	FIELD	FIELD	0	16					
STORE_ETA	FIELD	FIELD	0	11					
BREAK_2	FIELD	FIELD	0	20					
BREAK_1	FIELD	FIELD	0	19					
CARRIER	FIELD	FIELD	0	10					

Decodes

Document History

Version	Date	Status	Reason	Initials
0.1	13/01/10	Draft	Initial version	DRM
1.0	29/01/10	Issue	Reviewed and Issued	DJM



4 AUTHORISED BY

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

