

269496

Aptean Ltd
Copyright © 2011-2026.

Contents

1 269496.....	1
2 269496 - AT-7W5J4V/ New MTS-Paragon Interface.....	2
3 FUNCTIONAL OVERVIEW.....	3
3.1 Client Requirement.....	3
3.2 Solution.....	3
3.3 Scope.....	4
4 FUNCTIONAL DESCRIPTION.....	5
4.1 MTS to Paragon Order Flow.....	5
4.2 Paragon to MTS Trip Flow.....	6
5 REFERENCES.....	10
6 DOCUMENT HISTORY.....	11
7 AUTHORISED BY.....	12

1 269496

1 Ready for **What's Next, Now™**



2 269496 - AT-7W5J4V/ New MTS-Paragon Interface

Copyright OBS Logistics © 2010

The information contained herein is the property of OBS Logistics and is supplied without liability for errors or omissions. No part may be reproduced or used except as authorised by contract or other written permission. The copyright and foregoing restriction on reproduction and use extend to all media in which the information may be embodied



3 FUNCTIONAL OVERVIEW

3.1 Client Requirement

Use the new OBS XML v2.5 Trip Order XML message format via ESI.

New Paragon-MTS interfaces for Baxter client.

3.2 Solution

The new flows between MTS and Paragon will be based upon the existing Dixon's code.

We already have a system registry called ?PARAGON_INSTALLED? which is set to ?YES? for Dixon's (manual csv flows) and ?XML? for M&S (manual XML orders out and automatic XML trips in).

This will be extended to include ?HCR? for the new Baxter's csv flows.

The standard PAR_INT form will be used (will need minor changes for new system registry value).

The order outbound procedure to paragon (CSV.Write_Orders_to_Paragon) will be amended to check the system registry :-

If ?YES? then the layout will be as Dixon's existing requirements

but if ?HCR? then the new layout will be provided.

Other than the layout the rest of the order outbound code will remain unchanged (i.e. Export History, Orders to ?NEW? etc.) and will use the existing directories and naming conventions.

The trip inbound process will also be very similar to Dixon's (IMP.process_paragon_trip_data followed by PAR.process_paragon_data).

The additional fields (to be confirmed by Paragon) will be added to the dropdown list for paragon trip import type (PAR_TRIP_DTL) and added to required imports.

The IMP.process_paragon_trip_data will be changed to extract this new data and populate the equivalent new fields on the intermediate table (PAR_TRIP_DTL).

The PAR.process_paragon_data will be changed to take these additional fields into account and use them as required.

This development will also include an enhancement to how cross-dock paths will be processed and connecting trunking trips created in MTS to support the radial last leg schedule provided by Paragon.

Currently there are 3 key fields used to define the cross-dock paths :-

From Loc (of order), To Loc (off order) and Paragon Loc (provided by the paragon file as the delivery / outbase depot)

The enhancement will allow the cross dock path entries to be set up with the To Loc as the delivering depot / outbase. This can be considered as a wildcard input to avoid having to create a specific cross dock path for every delivery location.

This means the current functionality can be utilised with cross dock paths created to specific To Locs (like hospitals) or a default cross dock path created (for any final location not otherwise specified) via the specific delivery / outbase depot that Paragon schedules the order to be delivered from.

As the Paragon file is processed, MTS will first reference the cross dock paths data looking for a specific To Loc. If no entry exists, the cross dock paths will be referenced again with the delivery depot code as the To Loc. The objective here is to define the connecting trunk trip or trips and associated trunking carriers for each delivery (or group of deliveries).



3.3 Scope

This change will be applied to system version 10.5



4 FUNCTIONAL DESCRIPTION

The following changes are required to the existing Dixon?mts-Paragon csv interfaces in order to satisfy the new requirements for the Baxter client.

4.1 MTS to Paragon Order Flow

4.1.1 PAR_INT - Order Export

The current outbound csv flow is manually initiated and displayed from the PAR_INT form (Order Export tab) see below:-

No changes will be made to the appearance of this form.

The code behind the export button will be changed so that if the system registry ?PARAGON_INSTALLED? is set to either ?YES? or ?HCR? then the CSV.Orders_to_Paragon package will get invoked.

Another change is that the Healthcare system will initially be setup for 2 independent cost centres (HUK and BAX). Access to these cost centres will be controlled via the standard user maintenance and the drop down list is already populated accordingly. To ensure that a user does not run the export for both cost centres then if the ?PARAGON_INSTALLED? is set to ?HCR? then it will ensure that a cost centre is always populated before allowing the export to be run.

The control of access to this form and the control about which mandatory cost centre is selectable should ensure that the wrong cost centre's orders are not exported to Paragon.

The rest of the code will remain unchanged and will behave the same for Baxter?s as it does for Dixon?s.

NB) Error trapping for allowing the form to run also needs amending to allow value ?HCR? in system registry.

4.1.2 CSV Package

The procedure `Write_Orders_to_Paragon` within the package `CSV` will need changing to take into account the system registry `?PARAGON_INSTALLED?` to determine which format the csv output file is to be produced in.

If ?YES? then it will use the code that is already in place and produce a file in the current Dixon? format.

If ?HCR? then the file produced will be in the new Baxter format (See appendix A).

All of the additional data required can be added and then retrieved from the existing cursors (on SCH_ORD and SCH_ORD_LINE) apart from :-

The location information can be retrieved from **GEO_LOCATION** for the delivery location (**SCH_ORD.TO_LOC**).

The unloading rate code can also be obtained from the above location (GEO_LOCATION.UNLOADING_RATE) which is then linked to RES_LOAD_RATE to obtain the fixed unloading time (FIXED_MINS) and the default DU type variable unloading time (DEFAULT_MINS_PER_DU).



The existing cursor on SOL (first line only) will be used to obtain the DU type for the export (can an order have multiple lines with different DU types?).

In order to calculate the variable unloading time it will be necessary to create a new cursor to loop through all of the SOL and optionally link using the DU type to RES_LOAD_RATE_DTL. If a record is found then it will use MINS_PER_DU as the rate otherwise it will use the DEFAULT_MINS_PER_DU. The unload time will be calculated as the above rate * SOL.QUANTITY. These will be summed to provide the total variable unloading minutes.

Another new cursor is required to obtain the contact information (phone number and e-mail). It will again use the delivery location to get the newest contact (first record from GEO_CONTACT when order by ID desc).

The holiday flag will be set to ?Y? if SCH_ORD.DELIVERY_TYPE_ID contains the string ?TRAVEL? once upshifted otherwise it will be ?N?.

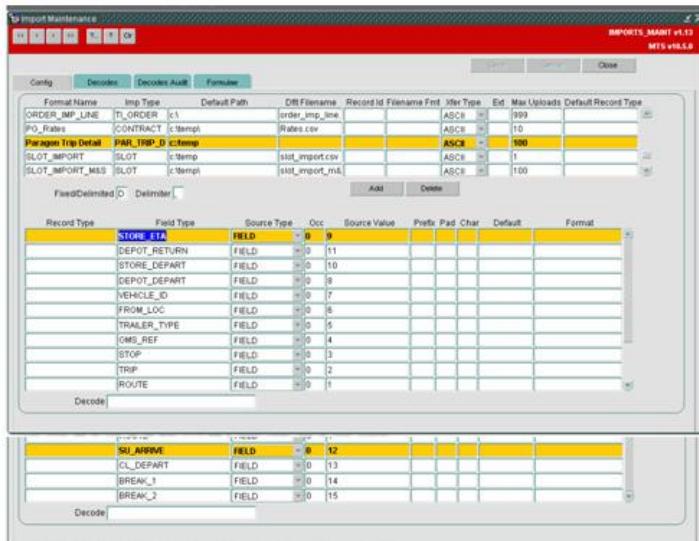
Apart from the changes to the format (Headings and Data) then the rest of the code will remain unchanged.

The file to be produced will contain a single header record to specify the field names for paragon then a detail line per order

4.2 Paragon to MTS Trip Flow

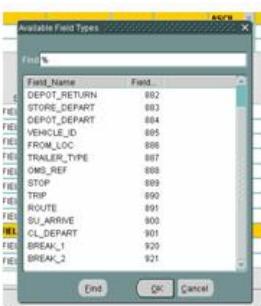
4.2.1 Import Format

The import of trips from paragon for the client Dixon?s currently looks like:-



The additional fields required for Baxter?s are:-

These will be added to the list available for adding to detail.



These can then be added in the required positions (source value) on the export.



4.2.2 Cross Dock Maintenance

The current paragon cross-dock maintenance looks like :-

From Loc	To Loc	Paragon From Loc	Delivery Leg Carrier
EXELSAVTR	A0101OUT	EXEL1NOW	EXEL1NOWCBF
EXELBAVTR	BLUEACCR	EXEL1NOW	EXEL1NOWCBF

Seq	X-dock Loc	Trunk Carrier Id
1	EXEL1NOW	EXELBAVTR

For every delivery location (final delivery location from the order - SCH_ORD.TO_LOC) then a corresponding cross-dock path has to be setup.

For the Baxter client then lots of the orders will be for home deliveries from a standard warehouse meaning that there will be a lot of ?To Loc? (final delivery locations) requiring cross-dock paths to be setup for the same ?From Loc? (warehouse) which are being delivered by the same ?Paragon From Loc? (depot responsible for the final delivery leg).

Also the new CIM order interface to MTS will be able to create new locations if they do not already exist in MTS so if the current method was used then someone would have to manually add the cross-dock path for the new locations before the order could be loaded onto a trip and sent back from Paragon.

The proposed solution is to firstly check, as now, to see if a specific cross-dock path has been setup for the final delivery location.

If not then a secondary check will be done (only if the system registry indicates that it is the Baxter flow so as not to interfere with the current code for Dixon?s but this check may be removed later as Dixon?s are also unhappy with the amount of data setup that is required for cross-docks) to see if a generic cross-dock has been setup for the delivery depot (Paragon From Loc).

So using the data above then a new cross-dock will be added to replace the above records as :-

From Loc	To Loc	Paragon From Loc	Delivery Leg Carrier
EXELSAVTR	EXEL1NOW	EXEL1NOW	EXEL1NOWCBF

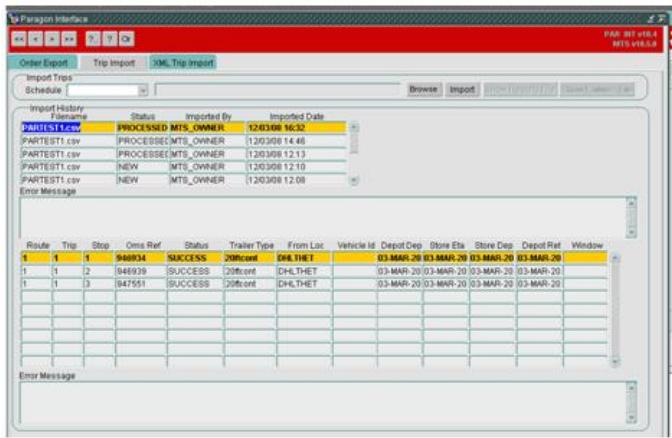
Seq	X-dock Loc	Trunk Carrier Id
1	EXEL1NOW	EXELBAVTR

This new cross-dock path can be used for all deliveries from the Warehouse being delivered by the same depot regardless of the final delivery address.

4.2.3 PAR_INT - Trip Import

The current paragon trip import is manually initiated and displayed from the PAR_INT form (Trip Import tab) :-





The only changes to the form are to ensure that only orders for the cost centre's provided in the file from Paragon are reset back to status ?UNSCHEDULED? from ?NEW? rather than all orders as now.

NB) This is just in case in future multiple cost centre's on the same database all use there own Paragon interfaces.

A new cursor will extract the cost centre's for the orders that have been added to trips (PAR_TRIP_DTL) by this file. It will then only reset these orders back to status ?NEW?.

The only other changes required will be internally to the packages IMP and PAR that are called by this form.

4.2.4 IMP Package

A few extra fields have been added to the Baxter version of the csv file to be provided by Paragon to MTS for generating trips.

These new fields are :-

(See appendix B for full layout).

These extra fields will need adding to the intermediate paragon table (PAR_TRIP_DTL).

ALTER TABLE PAR_TRIP_DTL ???

The procedure ?process_paragon_trip_data? within the IMP package will need changing to extract these new fields from the provided csv file (as configured in the import maintenance screen), do any necessary validation (size, type etc.) and then store them in the appropriate columns on the intermediate table.

NB) Dixon's will not have added these fields to there import layout so the code can be done regardless of which flow is being run.

4.2.5 PAR Package

None of the additional fields on the intermediate table (PAR_TRIP_DTL) will have been populated by the Dixon's data flow so the new code required for Baxter can simply be added for when the fields are set to a value on the intermediate table.

The additional fields will be used by the procedure ?process_paragon_data? within the PAR package to update the following fields in MTS :-

As well as the above changes to the procedure it will also need to check for cross-dock paths differently but only for the client Baxter.

When the system registry ?PARAGON_INSTALLED? is set to ?HCR? then the code which checks for paragon cross-dock paths using the collection location (SCH_ORD.FROM_LOC), the final delivery location (SCH_ORD.TO_LOC)



and the extracted field ?FROM_LOC? (delivery depot) from the provided csv file will still be run and used to check against the fields FROM_LOC, TO_LOC and PARAGON_FROM_LOC on the paragon cross-dock paths table (PAR_XDOCK_PATH).

If this returns no data then an additional check will be made against the paragon cross-dock paths table using the delivery depot provided in the csv file as both the delivery depot (PARAGON_FROM_LOC) and the final delivery location (TO_LOC).

If this new additional check finds any cross-dock paths then the code will continue the processing as now using the values found to generate the trunk leg trips.



5 REFERENCES

Not Available



6 DOCUMENT HISTORY

Version	Date	Status	Reason	Initials
0.1	28/08/09	Draft	Initial version	DRM



7 AUTHORISED BY

Dave Meir	Development Manager
Peter Greer	TMSCC MTS Product Manager

