285989 v1.0

Aptean Ltd

Contents

1 285989

aptean

DHL MTS

Develop Asset Tracking Functionality

FUNCTIONAL SPECIFICATION - 10.6

- 1.0

Reference: FS 285989-AD-8DQSF3

1.1 Client Requirement

1.2 Solution

OBS will develop the following to provide a fully integrated Asset Tracking solution for the Auto Alliance operation.

C-TMS Database Changes−

It is expected the following Tables will need to be added to the C-TMS Database:

ASSET_STATUSES - To hold a list of Asset Statuses i.e. FIT, REPAIR, DAMAGED

ASSET_ORIGINS - To Hold where each Customer?s Assets should return to i.e. Mercedes = DHLMILT

ASSET_DETAILS - To hold all information regarding an Asset i.e. type, current location, status etc

ASSET_HISTORY - To hold historic data, what happened and when and by who.

Asset Management Screen

A new screen will be developed and added to the Administration Menu -> Asset Mgmt

The new screen will allow the creation / amendment (i.e. current location) and deletion of assets to be tracked through the Auto Alliance Network.

The screen will be able to filter data on a number of options such as Customer (Merc/Volvo/Colt), Originating Site (DHLMILT/DHLHATF/DHLCIRN), Current Location.

See attached for proposed look and feel and detailed proposed functionality.

Import Maintenance / Process

A new import process will be needed to upload large quantities of assets quickly from a spreadsheet. Data will be added to the Import Maintenance tables, new code will need to be added to support this within the Import Process (IMP package). The import should be an ADD/UPDATE process.

Microlise Interface Changes

On receipt of POCPOD messages from Microlise the following processes will need to be added to existing logic:

1. If known order and known asset is received, (i.e. known asset is being delivered or planned return collection):

    1. Update Asset Table and Asset History accordingly, i.e. current location, arrival date/time.
2. If unknown order and known asset is received (i.e. known asset is being collected (ad-hoc)) :
    1. Create New OMS reference on C-TMS
    2. Update Asset Table and Asset History accordingly, i.e. In Transit, departure date/time.
    3. Automatically schedule new order onto trip id within Microlise Message to the last DHL Depot on Trip. - This will leave order in a status of ?SCHED_COLL?
    4. Send Message to WCS to trigger unload at last DHL Depot on Trip
3. If unknown order and unknown asset is received (i.e. unknown asset is being collected (ad-hoc)) :
    1. Create New OMS reference on C-TMS
    2. Create New Asset on Asset Table and Asset History accordingly, i.e. In Transit, departure date/time.(assumption is Location Id of Collection will drive Origin and therefore where the ?To loc?. DU Type will be assumed to be ?LRC?, and Asset Status will be set to status NEW.)
    3. Automatically schedule new order onto trip id within Microlise Message to the last DHL Depot on Trip. - This will leave order in a status of ?SCHED_COLL?
    4. Send Message to WCS to trigger unload at last DHL Depot on Trip
4. If known order and unknown asset is received (i.e. unknown asset is being collected on planned order (i.e. Return Collection) :
    1. Create New Asset on Asset Table and Asset History accordingly, i.e. In Transit, departure date/time.(assumption is Location Id of Collection will drive Origin and therefore where the ?To loc?. DU Type will be assumed to be ?LRC?, and Asset Status will be set to status NEW.)
    2. Automatically schedule new order onto trip id within Microlise Message to the last DHL Depot on Trip. - This will leave order in a status of ?SCHED_COLL?
    3. Send Message to WCS to trigger unload at last DHL Depot on Trip

The following assumptions have been made to collate all information required for creating new orders in C-TMS:

1. No Customer references will be captured. A new unique OMS reference will be generated by C-TMS.
2. Customer will be derived using the Customer Group set up for the <STOP_LOCATION_ID> sent in the Microlise Message.
3. Cost Centre will default to a new system param: MIC_DEF_COST (?DHLAA?)
4. Collect From will be set to <STOP_LOCATION_ID> sent in the Microlise Message.
5. IF UNKNOWN ITEM:
    1. Deliver To will be set to a derived value based on the Collect From?s Customer Group, Envisaged this will be one of three DHL depots: DHLMILT, DHLHATF, DHLCIRN
6. IF KNOWN ITEM
    1. Deliver To will be derived from the Asset Details table.
7. Delivery Times will be set to the above + a parameterised number of Days i.e. 5 days. (MIC_DEL_DAY_OFFSET)
8. Product Type will be defaulted to a system parameter: MIC_DEF_PRODUCT (?AMBIENT?)
9. IF UNKNOWN ITEM:
    1. DU Type will be set to parameter MIC_DEF_DU_TYPE (?LRC?)
10. IF KNOWN ITEM
    1. DU Type will be derived from the Asset Details table.
11. DU Qty will be set to total number of assets received on message assumption is 1
12. Schedule and RPE will be derived as per system standards
13. Order Items will be populated with <ITEM_IDENTIFIER> and <DELIVERED> into Identifier and ?To Deliver ? qty.
14. Reason Code information will populate the relevant Non conformance records in C-TMS.

WCS Changes

Both Load and Unload Processes will need to be amended to update the Asset Details and Asset History tables when Assets are positively loaded and unloaded from vehicles.

The Order Creation process will also have to be modified so that when new assets are encountered they are generated automatically based on scan location = origin, selected type = Du Type, created by etc should be WCS user etc.

1.3 Scope

This change will be applied to system version 10.6.0 on AAMTST and once approved AAMPRD.

2 Set-up

2.1 Pre-requisites

None

2.2 Menu Structure

?Unchanged?

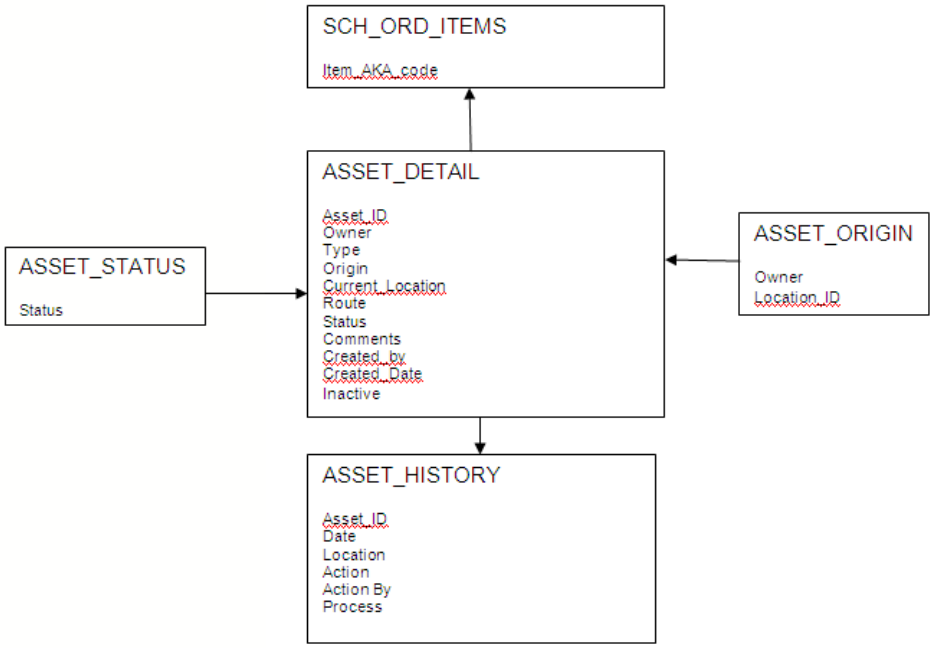
2.3 Data

3 Functional Description

Asset ids will be held against the order items in the ITEM_AKA_CODE. Two new tables will be created to store properties of the ASSET and the history of the ASSET. The new tables will allow users to track assets and understand when they are used.

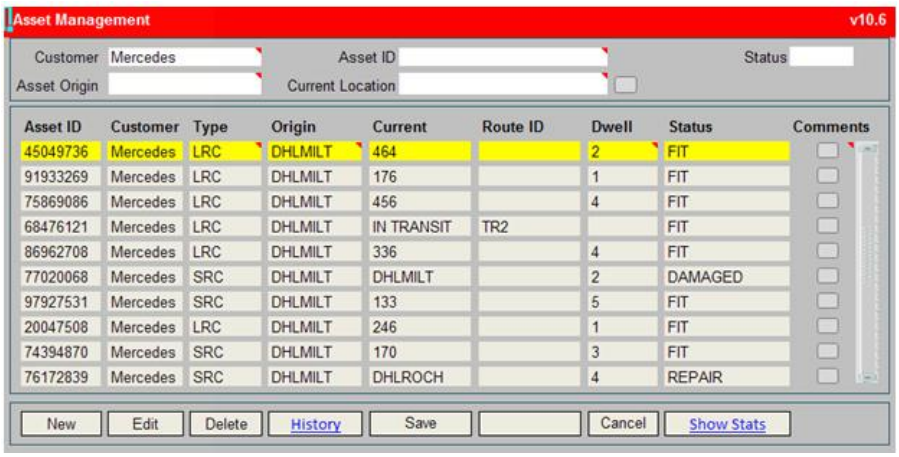
The detail table , ASSET_DETAIL, will accept manual entry via a new Asset Management screen. The history table, ASSET_HISTORY, will be automatically populated at different trigger points and specific fields in the Detail table will be updated at the same time.

Two new reference tables are required to store the ASSET STATUS values and the ASSET ORIGIN values.



3.1 Asset Management

A new screen will be developed to allow users to add, edit and delete ASSET Detail records. From within this screen, users will also be able to view the history of each asset. The history information will be view only and will not be updateable.



The main asset management screen will be based on the ASSET_DETAIL table. The created date and created by fields will not be visible on the screen.

The header of the Asset Management screen will allow users to filter the ASSET data based on the following fields:

- Customer
- Origin
- Asset_id
- Current Location
- Status

All of the search fields will be drop down lists but will also allow the user to enter text. The search fields can be used in combination. A new query button will be available in the header, once the selection criteria has been populated, selecting the command button will dynamically write the where clause based on the values selected.

The New, Edit and Delete buttons will be enabled based on the user. If the user is a SUPERUSER, the buttons will be enabled, if the user is not a SUPERUSER, the buttons will be disabled. Supe users will also have access to a right click option allowing them to maintain the records in ASSET_STATUS.

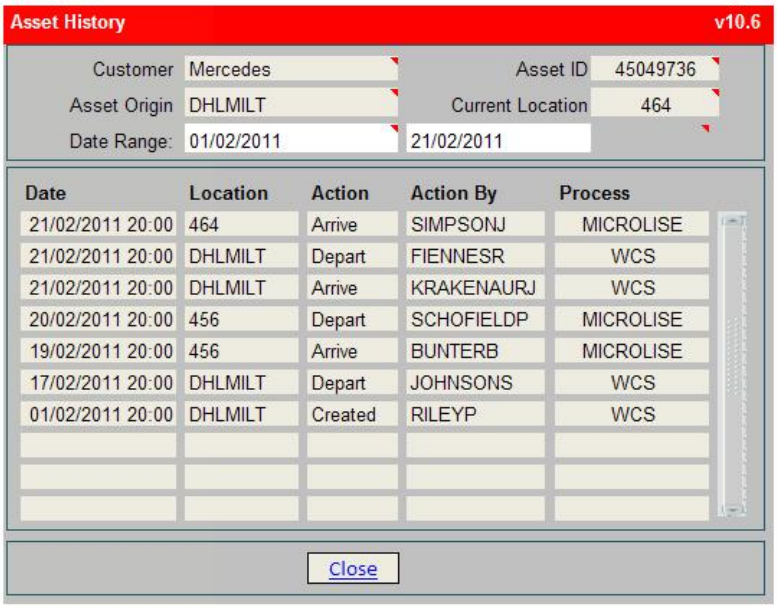
The comments field will be displayed in the block as a command button. Pressing the command button will display a pop up screen where the user can view, edit, add or remove comments.

The DWELL field will be a calculated field based on the DATE field from the ASSET_HISTORY table. DWELL not be calculated for ASSETS which are at a status of IN TRANSIT. If the current location of an asset is not IN TRANSIT, the latest ASSET_HISTORY record will be selected. The DATE value will be compared to SYSDATE and rounded to fully days. If SYSDATE - DATE = 1.3, this will be rounded to 2 days.

The block headings in the data block will be command buttons which when selected will sort the data in ascending and descending order

Asset History

Selecting the History button will open the ASSET HISTORY canvas for the selected ASSET in the ASSER_DETAIL screen.



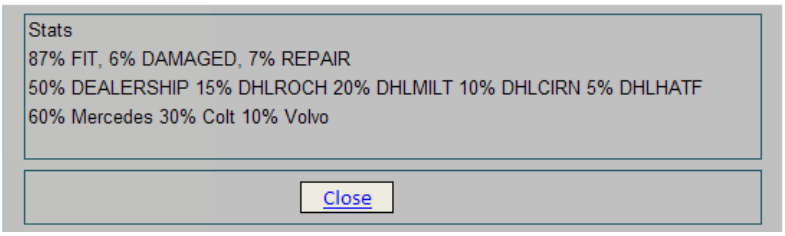
The header of the asset history canvas will allow users to limit the selection of data displayed in the block. Like the ASSET DETAIL screen, users will be able to populate more than one selection screen and select a ?refresh? command button to see the results.

The selection fields will be drop down lists which will also allow text entry.

The records in the data block will not be updatable.

Selecting Show Stats will display a new pop up canvas Asset Stats. The new canvas will display real time statistics, the proportion of assets at each status, the proportion of assets at each location and the proportion of assets belonging to each owner. The stats will be collated from the ASSET DETAIL table , using the STATUS, CURRENT LOCATION and OWNER fields.

The proportion of assets at each location will differentiate between RDCs but will group all dealerships together.



3.2 Import Maintenance Process

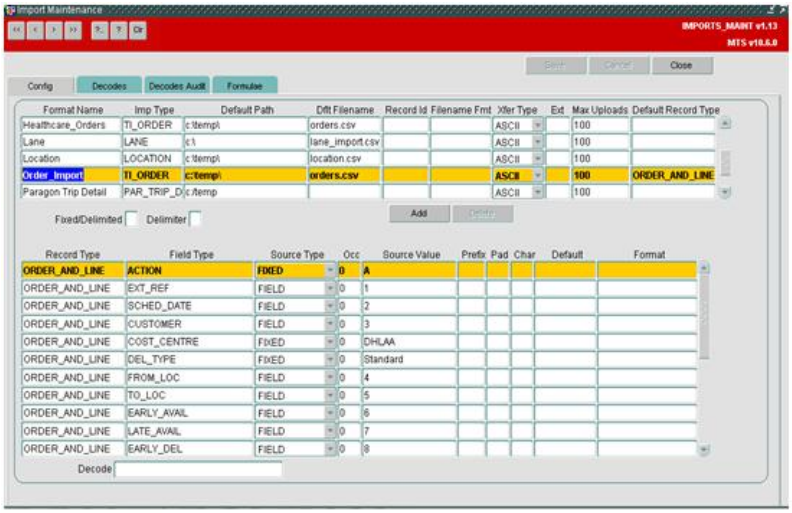
A new procedure called PROCESS_ASSET will be added to the IMP package. This will be based on a new IMPORT configuration added to the IMPORTS MAINTENANCE screen.

The new import will allow records to be added and modified in the ASSET_DETAILS table, so the following fields must be specified as part of the import configuration:

ASSET_ID, CUSTOMER, ASSET_TYPE, ORIGIN, CURRENT_LOCATION, ROUTE, STATUS, COMMENTS, CREATED_BY, CREATED_DATE, INACTIVE.

An action field will be added to allow users to specifiy Add or Modify, this field will be set to A, for each record, a cursor will check if the asset already exists. If a record is found, the action field will be set to M, otherwise it will remain as A.

If the action is set to A, the record will be inserted, if the action is set to M, all fields excluding the ASSET_ID, CREATED_BY and CREATED_DATE fields will be updated.



3.3 Microlise Interface Changes

There are 4 different scenarios we can encounter when receiving a POC/POD message from Microlise:

- Known Order and Known Asset
- Known Order and Unknown Asset
- Unknown Order and Known Asset
- Unknown Order and Unknown Asset.

A POC/POD message will be the only file received with order item details.

The asset code will be sent in the AKA_CODE which is specified in the latest XSD but is not currently populated as part of the POC/POD MICROLISE Inbound message. Work will be required with ESI to include this data in the message received from MICROLISE.

Unknown orders will only be encountered for empty assets being transported through the network. If the asset is unknown, a record will be added to the Asset Detail table and a record will be added to the Asset History table. If the asset is known, the record on the Asset Detail table will be amended , updating the current location and the a new record will be added to the Asset History table.

If the order is unknown, a new record will be added to the SCH_ORD and SCH_ORD_ITEMS tables.

Scenario	ASSET_DETAIL	ASSET_HISTORY	ORDER TABLES
Known Asset and Order	Update	Add	No action
Known Asset and unknown order	Update	Add	Add
Unknown Asset and Order	Add	Add	Add
Unknown Asset and known order	Add	Add	No action

Code will be added to the existing procedure INT_XML_OUT2.Process_MIC_Trip_XML_IN. Currently is a null oms_ref is

received an error is processed . To allow ad hoc collections this will be changed so that if the oms_ref is null and the AKA_CODE is not null a new order will be created on TMS.

The order will be derived using the following information

OMS_REF	SEQ.nextval
CUSTOMER	If the asset is known, the CUSTOMER will be set to the owner from the ASSET DETAILS table. If the ASSET is unknown, the CUSTOMER will be set based on the load location(dealership) and the data stored in the GEO_LOCATION_USAGE table.
COST CENTRE	New system parameter MIC_DEF_COST
COLLECT_FROM	Stop location id
DELIVER_TO	For known assets, this will be the ORIGIN value from the ASSET_DETAIL table. For unknown assets, this will be derived using the CUSTOMER and the new table ASSET_ORIGINS table to link the customer to the ORIGIN.
EARLY COLL DATE	STOP ACTUAL ARRIVE
LATE COLL DATE	STOP ACTUAL DEPART DATE
EARLY DEL DATE	EARLY_COLL_DATE + new system parameter MIC_DEL_DAY_OFFSET
LATE DEL DATE	LATE_COLL_DATE + new system parameter MIC_DEL_DAY_OFFSET
PRODUCT_TYPE	New system param MIC_DEF_PRODUCT
DU_TYPE	If unknown the value will be determined from the new system param MIC_DEF_DU_TYPE, otherwise the value will be set to the type field on the ASSET DETAIL table
SCHED_NAME	Derived using MTS CODE
RPE	Derived using MTS CODE
ORDER ITEMS	ITEM_IDENTIFIER, DELIVERED, AKA_CODE
REASON CODE	

Once the order has been created, OMS.VALIDATE_ORDER will be called to set the sched_name, status and RPE .

The new Asset Detail record will be derived as follows

ASSET_ID	AKA_CODE
OWNER	Customer from the order record
TYPE	DU Type from the order record
ORIGIN	DELIVER_TO from the order record
CURRENT−_LOCATION	?IN TRANSIT?
ROUTE	Route code of the trip
STATUS	NEW
COMMENTS	
CREATED_BY	MTS_OWNER
CREATED_DATE	SYSDATE
INACTIVE	

In all instances, a new record will be added to the ASSET_HISTORY table as follows

ASSET_ID	AKA_CODE
LOCATION_DATE	SYSDATE
LOCATION	STOP_LOCATION_ID
ACTION	DEPART
ACTION_BY	DRIVER NAME if available otherwise MICROLISE
PROCESS	MICROLISE

3.4 Load and Unload Scanning

If an order is successfully loaded onto a trip, the item_identifier we will used to identify the asset using item_aka_code on the sch_ord_items table.

A successful load will be processed in the procedure DP_RDT_GOODS_OUT..RECVD_WCS__LOAD_CONF. In addition to the processing currently done, asset information will also be amended and added.

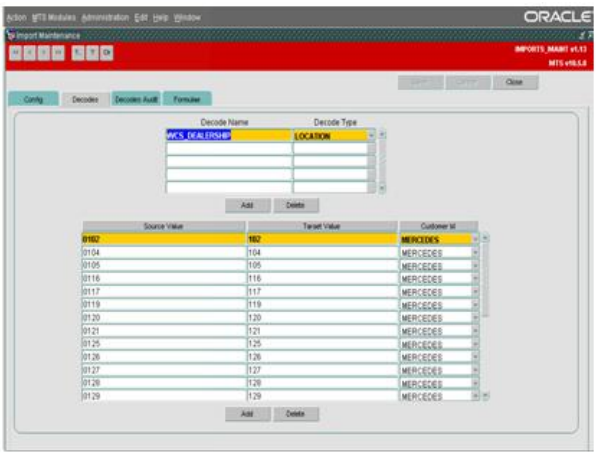
The ASSET_DETAIL record will be updated to reflect the CURRENT_LOCATION. The current_location will be set to IN_TRANSIT and the route code will be updated based on the trip the item has been loaded onto.

If the ASSET does not exist on the system, a new record will be added to the ASSET_DETAIL table

ASSET_ID	ITEM_AKA_CODE
CUSTOMER	Customer from the order record
TYPE	DU Type from the order record
ORIGIN	From location of the order record
CURRENT−_LOCATION	?IN TRANSIT?
ROUTE	Route code of the trip
STATUS	NEW
COMMENTS	
CREATED_BY	WCS user*
CREATED_DATE	SYSDATE
INACTIVE	

In the ASSET_HISTORY table a new record will be inserted

ASSET_ID	AKA_CODE
LOCATION_DATE	SYSDATE
LOCATION	FROM_LOC on the order, identified from the item_identifier
ACTION	DEPART
ACTION_BY	WCS user*
PROCESS	WCS



If an item is successfully unloaded from a trip, the ASSET_DEAILS will be updated and a new record will be added to ASSET_HISTORY. The current location will be updated to the unload location and the route code will be set to null as the asset is no longer in transit.

A new record will be added to the ASSET_HISTORY table

ASSET_ID	AKA_CODE
LOCATION_DATE	SYSDATE
LOCATION	Unload location
ACTION	ARRIVE
ACTION_BY	WCS user* (as discussed above)
PROCESS	WCS

Table Updates Required

Create new tables

ASSET_DETAIL

ASSET_ID	VARCHAR2(30)
OWNER	VARCHAR2(12)
ASSET_TYPE	VARCHAR2(10)
ORIGIN	VARCHAR2(12)
CURRENT_LOCATION	VARCHAR2(12)
ROUTE	VARCHAR2(12)
STATUS	VARCHAR2(12)
COMMENTS	VARCHAR2(150)
CREATED_BY	VARCHAR2(50)
CREATED_DATE	DATE
INACTIVE	VARCHAR2(1)

ASSET_HISTORY

ASSET_ID	VARCHAR2(30)
DATE	DATE
LOCATION	VARCHAR2(12)
ACTION	VARCHAR2(12)
ACTION_BY	VARCHAR2(50)
PROCESS	VARCHAR2(30)

ASSET_STATUS

STATUS VARCHAR2(12)

ASSET_ORIGINS

LOCATION VARCHAR2(12)

OWNER VARCHAR2(12)

References

Ref No	Document Title & ID	Version	Date
	AD-8DQSF3 285989 Develop Asset Tracking Functionality	1.0	28/02/11

Document History

Version	Date	Status	Reason	Initials
0.1	07/03/11	Draft	Initial version	SW
1.0	16/03/11	Issue	Reviewed and Issued	MJC

4 AUTHORISED BY

Matt Crisford	Development Manager
Peter Greer	TMSCC MTS Product Manager