291360 v2.1

Aptean Ltd

Contents

1 291360

aptean

DHL C-TMS

Services

FUNCTIONAL SPECIFICATION - 10.7

20/10/11 - 2.1
Reference: FS 291360 - MS-8KNGFM

1.1 Client Requirement

Change Request Summary:

4 - Services. Paul McGoran/Manchester/UK/NFC

Change Request Details:

Maintain master service and surcharge types and associated charging attributes. Capture revenue surcharges from Order interface and manual additions, inserts, amends and deletes. Apply cost surcharges to trip - stored at carrier level.

Benefits identified as a result of the change:

Required for implementation

1.2 Solution

Purpose

Beyond costing for transportation services provided by carriers and charging transportation services to NR, DHL will arrange and manage other related services and associated cost and charges. There are many examples including put-away services, HIAB (unloading by carrier), Banksman, PTS and vehicle escorts. Note that some of these additional services relate to carrier only (and so are not recharged to NR) and some relate to customer account only (so are not received as additional cost from carrier service provider). Although unusual, it is possible that a service might be captured in C-TMS with no financial implications. Also, the system functionality to support service and related surcharge will also be used to manage penalty charges / costs (demurrage).

Services and the related surcharges are used to define additional revenue charges to customers (accounts) and define additional cost of service provision from carriers (additional to transport)

Service Master Maintenance

A new table and maintenance screen will be created to add, edit and remove service records in master data. The maintenance screen will be a new tab in the existing Accounts Maintenance screen and will be based on a new table called ACC_SERVICES; the columns of the new table are defined below:

SERVICE_ID	Unique Code
SERVICE_NAME	Unique Description
SERVICE_EVENT	?BOTH?,?TRIP?,?ORDER?
INITIATED_FROM_TASK	?Y? or ?N?

Services Capture (Orders & Trips)

Services will be allocated to orders as part of order creation from EDI and CSV upload and manual entry and will be saved on a new order tab called Services. Services, at order level, will generate revenue surcharges to the customer (account) of the order. Once an order is scheduled onto a trip, the trip will inherit any services from the order. Services, at trip level, will generate estimated cost surcharge from carrier. Trip services will be stored in a new tab called Services in Trip Manipulation and in Trip Debrief.

The service codes held at order level will be generated onto the trip depending on the service event of the master record (if BOTH). The respective revenue surcharge and cost surcharge can be different depending on the calculation rules described below. The revenue and cost might also vary by customer (account) and carrier and the respective rates set into the charge rules.

Maintaining Charge and Cost Rates & Rating Engine

To accommodate this charge and cost functionality, a new table ACC_SERVICE_RATES will be created.

This table will allow revenue surcharge rates by customer (account) to be defined and will allow cost surcharge rates by carrier to be defined.

The fields in the new table are defined below:

DEBIT_ACCOUNT
CREDIT_ACCOUNT
SERVICE_ID
EFFECTIVE_DATE
CHARGE_TYPE
AMOUNT
A new tab will be added to Accounts Maintenance. The tab screen will be used to maintain revenue and cost surcharges by service. Like other tabs in Accounts Maintenance the parties involved will be maintained using a Credit Account and a Debit Account. For revenue surcharges, the debit account will be customer (account) and credit account the NR cost centre. For cost surcharges, the debit account will be the NR cost centre and credit account the carrier. The tab screen will allow a ?wildcard? value for either customer or carrier for a standing surcharge record covering either all customers (accounts) or all carriers. When calculating surcharges, the rating engine will select a specific service record for carrier or customer respectively for cost or for revenue; if no record exists a wildcard record for the service will be selected. If no rating records for the service is found the service will be rated as zero for manual update.

The charge type for revenue and cost surcharges will be one of FIXED, QTY or HOURS.

If the charge type is set to FIXED, the surcharge (the payment record) generated for the service will be a fixed value defined in amount field. The surcharge value will be displayed in the Order Services and / or Trip Services tab screens.

If the charge type is set to QTY or HOURS, the surcharge (the payment record) generated for the service will be based on the units captured on the services record in the Orders and / or Trip tab screens. The surcharge value will be displayed in the Order Services and / or Trip Services tab screens.

If the charge type is QTY, the service payment will be regenerated each time the order or trip are validated to ensure any change in quantity is captured and reflected in the payment amount.

When an order is created with services attached, the customer (account) on the order will identify which record in the ACC_SERVICE_RATES table is used to generate the payment. The payment will be created between the customer and the cost centre and a payment _type set to the SERVICE_ID.

When a trip is set to ACCEPTED status, all services inherited from orders will be used to generate costs. The carrier id assigned to the trip will identify which record in ACC_SERVICE_RATES is used to generate the payment. The payment will be generated between the carrier and the cost centre with a payment type set to the SERVICE_ID.

The finance tabs on the Order and Trip screens will be amended to include any service payments where applicable. The user will be able to use the existing command button to drill down to the payment record.

A command button will be added to the order and trip services tab to allow manual services records to be created and rated.

Services will be defined with a service event, which will allow users to control when a payment should be generated. The values allowed will be TRIP, ORDER or BOTH. Multiple orders delivered at the same stop, will generate multiple service charges at order level but one service charge at trip level.

Microlise Tasks

Service charges may also be assigned to Trips and Orders through trip stop tasks, received from MICROLISE in PocPod messages. This information will be stored in the task tab of the Trip debrief screen and the relevant service records and payments will be generated against the trip and orders. If a service is assigned from a task and the service event is set to ?BOTH? or ?ORDER?, the relevant service record will be added to the associated orders.

If required a de-code record will be created to translate Microlise task names into C-TMS service types. This de-code translation will be processed during the interface upload of PodPoc (in OBS XML format) messages into C-TMS

(Note that this functionality will support Put-Away service surcharge which is done by the DHL (own fleet) driver, hence the surcharge will be configured as ?ORDER? only)

Audit

An audit trail for creation, amend and delete of service records and the related surcharge values will be kept against orders and against trips showing the date and time of the audit record and the user responsible.

CSV Import

Two new imports will be created, a service master and a service rates. The imports will populate the two new service tables ACC_SERVICES and ACC_SERVICE_RATES.

Reporting & Export

Any service and surcharge reporting and export requirements will be covered separately in RIO MS-8KNHMH (OBSL ref 291373)

1.3 Scope

This change will be applied to system version 10.7 on INDTST and once approved INDPRD.

2 Set-up

2.1 Pre-requisites

None

2.2 Menu Structure

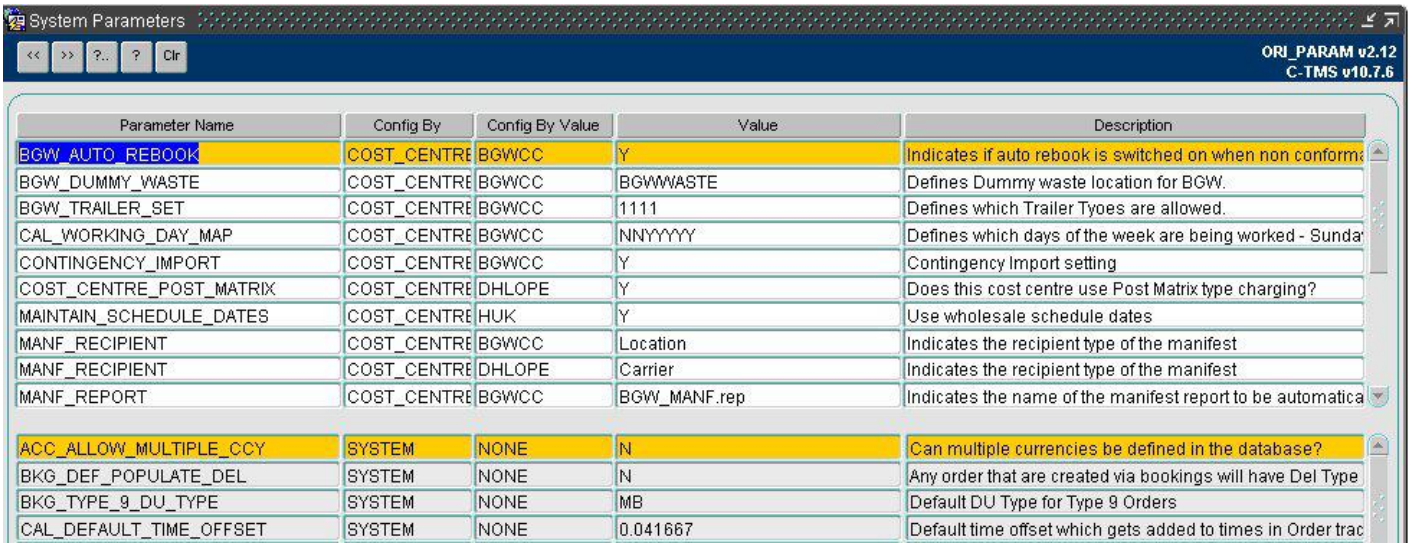
Unchanged

2.3 Data

The following new database tables will be created: ACC_SERVICE, ACC_SERVICE_RATE, ACC_SERVICE_AUDIT, SCH_ORD_SERVICES

2.4 Implementation Advice

All functionality as described in the functional spec will be controlled by a new COST_CENTRE parameter called ?SERVICES?. A super user will be required to set the value of this parameter to ?Y? for the NR cost centre, using the system parameter maintenance screen:



3 Functional Description

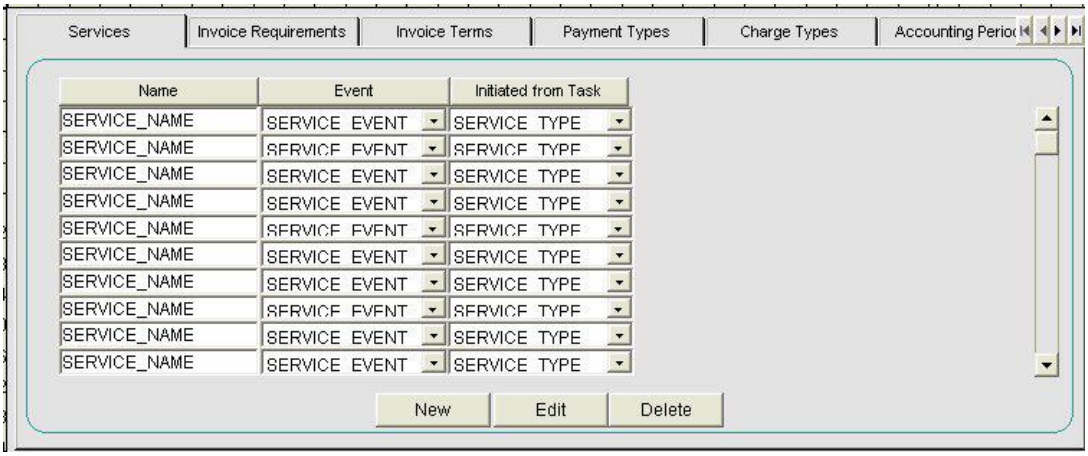
3.1 Accounts Maintenance

The Accounts Maintenance screen will be altered to include a new ?Services? tab. This will allow the user to create and maintain master details related to how additional service charges are linked to orders and trips via customers and carriers.

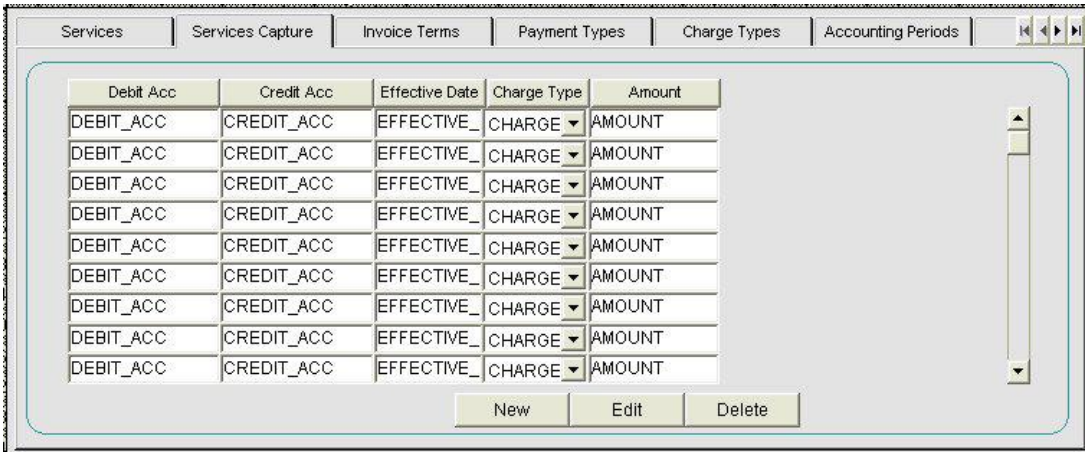
A new table will be created called ACC_SERVICES

ACC_SERVICES	
SERVICE_ID	VARCHAR2(12)
SERVICE_NAME	VARCHAR2(100)
SERVICE_EVENT	VARCHAR2(12)
INITIATED_FROM_TASK	VARCHAR2(1)

The ?Services? tab will contain a simple cluster that allows the user to enter a ?Service Name?, ?Service Event? (either ?ORDER?, ?TRIP? or ?BOTH?). The ?Service Name? will be expected to be a unique description of the service charge to be applied. The ?Initiated From Task? field will be a display item to indicate if the services is allocated from a Microlise task.



A second tab will be added to the screen labelled as ?Services Capture?:



Service charges can be further defined, to indicate how payment records should be derived. The debit account and credit account allow users to define when the payment created is a revenue charge to a customer or a transport cost from a carrier.

ACC_SERVICE_RATES	
DEBIT_ACC	VARCHAR2(12)
CREDIT_ACC	VARCHAR2(12)
SERVICE_ID	VARCHAR2(12)
EFFECTIVE_DATE	DATE
CHARGE_TYPE	VARCHAR2(12)
AMOUNT	NUMBER

The information recorded is:

DEBIT_ACC	For revenue service charges, this will be set to the Customer For cost service charges, this will be set to the DHL Cost centre For revenue service charges where the same charge is applied to all customers, this will be set to ?ALL?
CREDIT_ACC	For revenue service charges, this will be set to the DHL Cost Centre For cost service charges, this will be set to the Carrier For cost service charges where the same cost is applied by all carriers, this will be set to ?ALL?
SERVICE_ID	Identifies the service type of the record.
EFFECTIVE_DATE	Service charges may change over time, this date will allow the system to determine which ?current? service charge should be applied.
CHARGE_TYPE	Charge type can be set to FIXED or HOURS. The setting of this field will determine how the payment record is calculated.
AMOUNT	A numeric value, combined with the charge type to generate a payment record.

There will be occasion when a revenue service charge differs between customers and occasion when a transport cost service charge differs between carriers. The table structure allows users to define each service type specifically for a carrier or customer.

Where a service charge is a standard charge for all customers, the user will be able to enter a single record, using a debit account ?ALL? rather than generate the same record for every customer. The same functionality exists for Carrier costs, where the user sets the Credit account to ?ALL?.

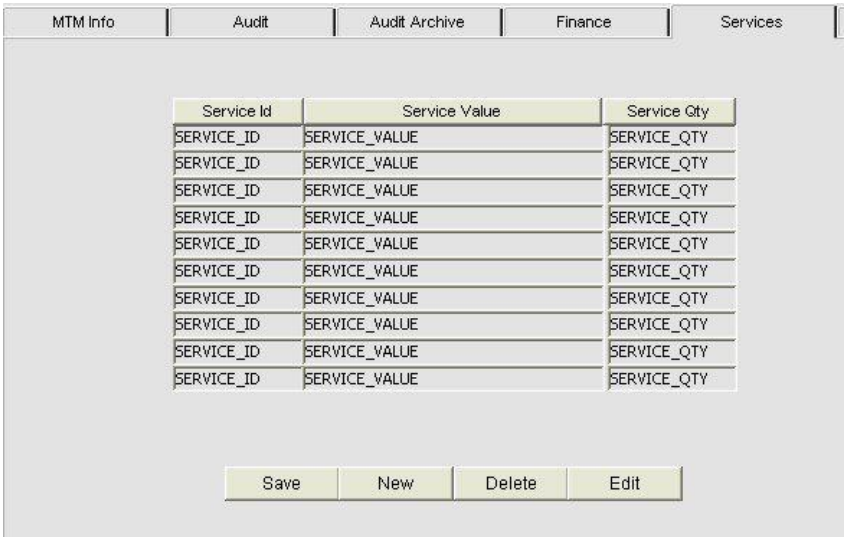
3.2 Service Records

In addition to revenue and cost generated from Customer and Carrier contracts, NR will also capture service charges which may result in additional revenue from the Customer or additional cost from the Carrier.

Services are attached to Orders and can be received as part of the EDI and CSV order creation or manually added in the Order screen. A new table will be created to store service information called SCH_ORD_SERVICES.

SCH_ORD_SERVICES	
OMS_REF	VARCHAR2(12)
SERVICE_ID	VARCHAR2(12)
SERVICE_QTY	VARCHAR2(12)
SERVICE_VALUE	VARCHAR2(12)
INHERITED	VARCHAR2(1)

A new tab will be added to the orders screen which will allow users to view, add and edit the service records.



Services at order level will generate additional revenue payments for the Customer (Customers will be defined as account types (HEAVY, NON-HEAVY etc). If the service has been defined as being assigned to ?BOTH? it may also generate additional cost from the Carrier once the order has been scheduled onto a trip.

In addition to applying order services to a trip, services can also be assigned directly to trips as Microlise Trip stop tasks. The service records will be received from Microlise in POC/POD messages. If a service is assigned to a trip from Microlise and the event of the service is set to Order or Both, the service will be applied to all the orders at the stop.

The services applied to a trip will be displayed in a new tab screen in the Trip Manipulation and trip planning screens. A new table will be created to store the service records at trip level.

SCH_TRIP_SERVICES	
SCHED_NAME	VARCHAR2(6)
TRIP_ID	VARCHAR2(12)
SERVICE_ID	VARCHAR2(12)
SERVICE_QTY	VARCHAR2(12)
SERVICE_VALUE	VARCHAR2(12)
INHERITED	VARCHAR2(1)

In the execution screen, users are able to select a trip and right click and select ?Show Trip Details?, to display the trip details tab as displayed in the Trip Manipulation screen. The new tab will also be added into this screen when accessed from Execution.

In Trip Debrief, there is currently a tab for displaying Trip stop tasks. Trip Service records will be sent in the XML format using the trip stop tasks fields but will be differentiated as services using the task name. Services will be displayed in two new tabs in the debrief screen called Trip Services and Order Services.

To accommodate this, as part of the Microlise EDI flow, the system must be able to distinguish between service tasks and general tasks, this will be handled as part of the Microlise development (219372 MS-8KNHJA Microlise).

The Trip Services tab will be based on the SCH_TRIP_SERVICES table, where the trip_id and schedule match the trip id and schedule being debriefed. The Order Services tab will be based on the SCH_ORD_SERVICES table and will display all service records for each order loaded on the trip. The OMS reference will be displayed to allow users to identify which services link to each order. The records will be displayed in OMS_REF order.

3.3 Payment Creation

3.3.1 Order Service Payments

Records are added into SCH_ORD_SERVICES via the EDIT order flow, the CSV import, manual entry or inherited from Trips. Validate_Order will be amended to call a new procedure in the Rating engine called Validate_Order_Services.

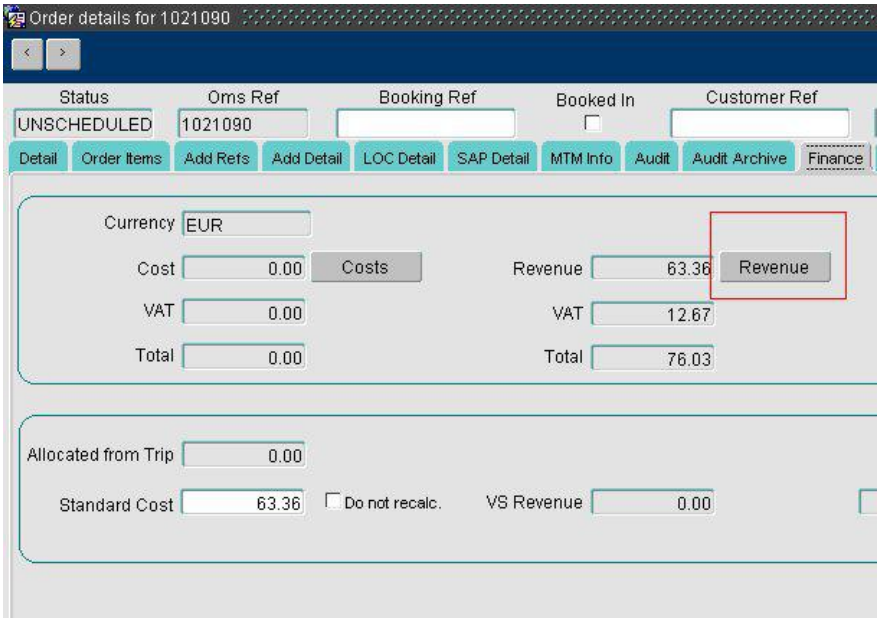
Validate Order services will identify any records in SCH_ORD_SERVICES table where the processed field is set to N. Using information from SCH_ORD_SERVICES and the order details, the new procedure will generate payment records.

The payment records will be generated based on the following information:

Payment Field	Data Source
DEBIT_ACC	SCH_ORD.CUSTOMER
CREDIT_ACC	SCH_ORD.COST_CENTRE
PAYMENT_TYPE	SCH_ORD_SERVICES.SERVICE_ID
EVENT_REF	SCH_ORD.OMS_REF
ENTERED_DATE	SYSTEM DATE
REVENUE_DATE	SYSTEM DATE
AMOUNT	Calculated from acc_services and sch_ord_services
STATUS	?F?
VAT	
VAT COUNTRY	?GB?
NARRATIVE_1	?SERVICE?

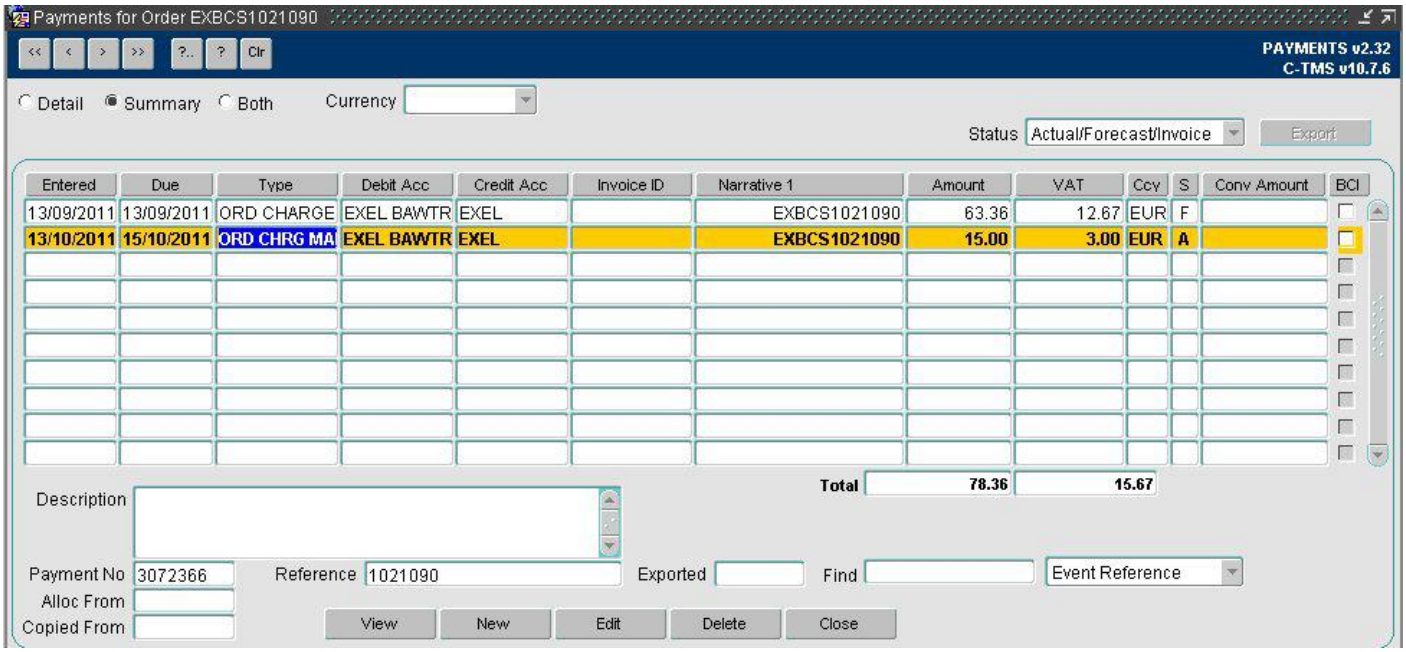
If an order service is no longer required for an order, it will be a manual exercise to remove the service record from SCH_ORD_SERVICES and delete the related payment.

Payments will be displayed by selecting the Revenue button on the finance tab of the orders screen:



Selecting revenue will display all the revenue payments held against the order. From this screen, users are able to delete

and edit payment records.



If a service is removed from a scheduled order, the user may be required to manually remove the service from the trip and delete the payment record, if it is not longer valid.

3.3.2 Trip Service Payments

There are three ways a trip service can be created; via a task from microlise, inherited from a scheduled order and manual entry.

In all three methods, a record will be added into SCH_TRIP_SERVICES and the processed flag will be set to ?N?. The following events will look for unprocessed service records and generate payments:

- Trip Status change (Planned to Accepted and upwards)
- Receiving a service task from Microlise
- When an order is scheduled onto a trip
- A button will be added to the services tab of the Trip screens to allow users to force the Services process to run.

When a trip is set to Accepted, a new procedure called RATE.CREATE_INHERITED will identify all distinct services assigned to orders scheduled on the trip were the service event is set to ?TRIP? or ?BOTH?. For each service identified, a single record will be added to the SCH_TRIP_SERVICES table. (See Inherited Services)

When all of the service records have been added, the system will call the RATE.VALIDATE_SERVICES procedure to generate the payment records, using the data from the SCH_TRIP_SERVICES table and the trip details.

Payment Field	Data Source
DEBIT_ACC	SCH_TRIP.COST_CENTRE
CREDIT_ACC	SCH_TRIP.CARRIER_ID
PAYMENT_TYPE	SCH_TRIP_SERVICES.SERVICE_ID
EVENT_REF	SCH_TRIP.TRIP_ID
ENTERED_DATE	SYSTEM DATE
REVENUE_DATE	SYSTEM DATE
AMOUNT	Calculated from acc_services and sch_ord_services
STATUS	?F?
VAT	
VAT COUNTRY	?GB?
NARRATIVE_1	?SERVICE?

If a service record is no longer required against a trip, the user will be required to manually remove the service record and delete the associated payment. The service record can be removed using the service tab and the payment can be deleted using the cost button on the finance tab.

3.4 Payment Details

3.4.1 Payment Type

Currently when creating a payment record, the payment type assigned must exist in the ACC_PAYMENT_TYPE reference table.

To prevent users from having to set up service types in ACC_SERVICES and also set up records in ACC_PAYMENT_TYPE, when a new record is added to ACC_SERVICES a trigger will automatically add a record for the new Service to ACC_PAYMENT_TYPE.

This will allow payment to be created with a payment type based on the service ids.

3.4.2 Payment Amount

Payment amount will be calculated based on the values of the following fields:

ACC_SERVICE_RATES.CHARGE_TYPE

ACC_SERVICE_RATES.AMOUNT

SCH_ORD_SERVICES.QTY or SCH_TRIP_SERVICES.QTY

If the service record to be processed into a payment is in the SCH_ORD_SERVICES table, C-TMS will look for a record in ACC_SERVICE_RATES based on the service id where the DEBIT_ACC is the customer on the order. If the system is unable to find a record for the service id and customer combination, the system will look for a record in ACC_SERVICE_RATES where the debit_acc =?ALL? based on service id.

Where the service record to be processed into a payment is in the SCH_TRIP_SERVICES table, the system will look for a record in ACC_SERVICE_RATES based on the service id, where the CREDIT_ACC is the carrier on the trip. Again if the system is unable to find a suitable record for the carrier service_id combination, the system will look for a record where the CREDIT_ACC is equal to ?ALL? based on the service_id.

Once an ACC_SERVICE_RATES record has been selected, the charge type and amount will be used with the quantity from the service record to calculate the payment amount.

Some examples of the calculation are shown below;

ACC_SERVICE_RATES.	ACC_SERVICE_	SCH_{*trip/ord*}_	ACC_PAYMENT.
CHARGE_TYPE	RATES.AMOUNT	SERVICES.QTY	AMOUNT
FIXED	100	1	100
HOURS	15	3	45

If no qty is sent in the service record for an ?HOURS? service, the qty will be set to 0, creating a payment for 0. Users will be able to edit the payment and the amount when the number of hours required is known. The hours input field will be decimal and allow part hours to be entered i.e. 1.75 hours.

If the quantity in the service record for a FIXED service is null, the system will set the quantity to 1 and create a payment for the fixed amount.

3.5 Inherited Services

This is a two way process, where orders can inherit services from Trips and Trips can inherit services from orders. Service types are inherited based on the value of ACC_SERVICES.SERVICE_EVENT, which can be set to TRIP, ORDER or BOTH.

When an order is scheduled onto a trip, any services which have been assigned to the order with a service event of TRIP or BOTH will be inherited by the trip. A new service record will be added to the SCH_TRIP_SERVICES table if the service is not already on the trip and the INHERITED flag will be set to Y.

Trips can be assigned services through the Microlise update. When the service record is for service event ORDER or BOTH, the service will be inherited by all orders scheduled on the trip; a record will be added to SCH_ORD_SERVICES for each order and the INHERITED flag will be set to ?Y?.

Once inherited services are created in the service tables, payments will be generated based on stated trigger points. The system must be able to track inherited payments and establish when inherited services are no longer valid.

EXAMPLE : An order is unscheduled from a trip

Any trip services which the order inherited from being scheduled on the trip should be removed and any order services that the trip inherited from scheduling the order should be removed.

For services inherited by a trip, further checks are required as un-scheduling an order from the trip may not necessarily result in the service no longer being valid if other orders with the same service remain scheduled on the trip:

TRIP : MAN-00001234 4 SCHEDULED ORDERS : 000123,000234,000345,000456

The four orders on the trip have been assigned the following order services:

OMS_REF	SERVICE_ID	SERVICE_EVENT
000123	BANKSMAN	BOTH
000123	POLICE ESCORT	BOTH
000345	BANKSMAN	BOTH
000456	PTS	ORDER

Based on the Order services, the CREATE_INHERITED procedure creates two new service records in SCH_TRIP_SERVICES:

TRIP_ID	SERVICE_ID	INHERITED
MAN-00001234	BANKSMAN	Y
MAN-00001234	POLICE ESCORT	Y

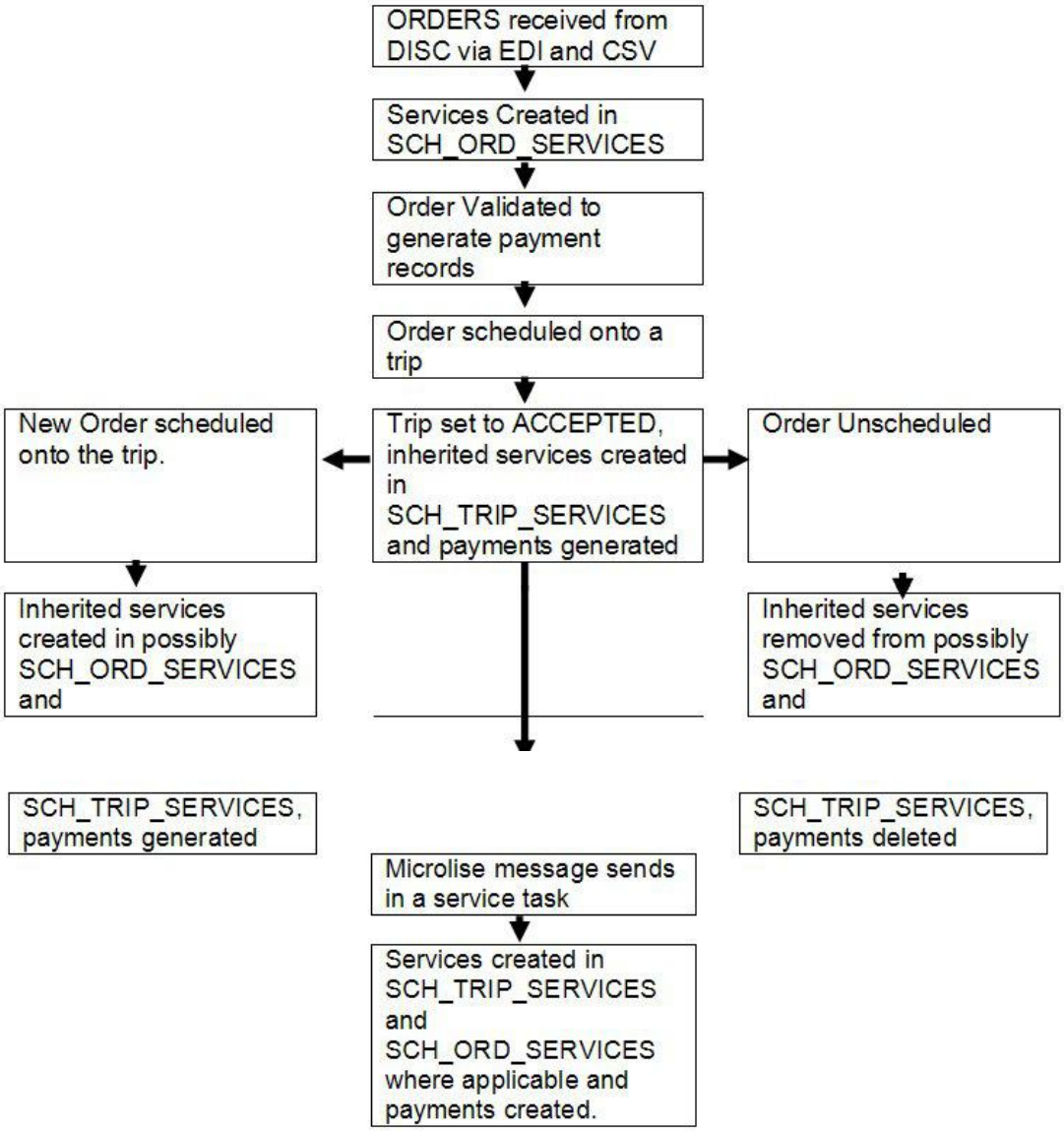
If order 000123 is unscheduled form the trip, only the POLIC ESCORT service should be removed from the trip. If order 000345 is unscheduled, no services should be removed from the trip. The BANKSMAN service should remain on the trip until all orders scheduled on the trip with a BANKSMAN service assigned are removed.

Every time an order is scheduled or unscheduled onto an ACCEPTED trip, the system will call RATE.CREATE_INHERITED. The procedure will also be called when a trip is set from PLANNED to ACCEPTED. In addition to creating inherited services, this procedure will also remove invalid redundant services.

After the Create_Inherited procedure is run, Validate_Services will be called to generate payments from any unprocessed services.

Command buttons will be added to the Order Services tab and the Trip services tab called Create Inherited Services. This will allow users to force the procedure to run.

3.6 Services: Data Flow Diagram



3.7 Payment Events

Currently in C-TMS, when a trip is changed (status change, new orders, orders removed, stop details entered) all payments generated by the system for the trip and any orders on the trip are deleted and recreated.

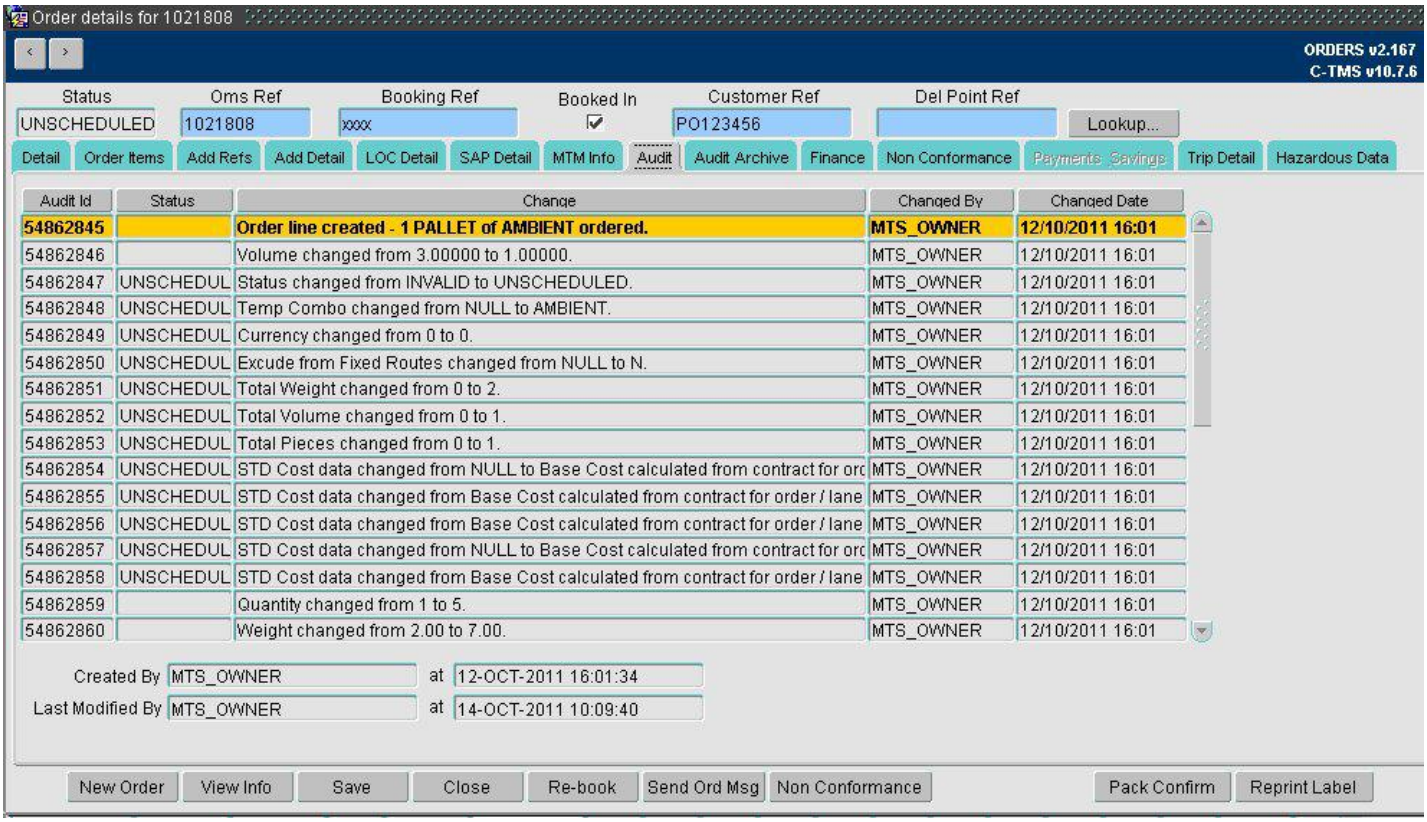
This process is run based on the event reference of the payment which is set to the trip id or the oms_ref. As service payments are generated by the system, they would currently be removed and regenerated. Service payment amounts may be entered manually or overwritten by the user and regenerating the payments would loose this information.

To ensure the system does not overwrite the service payments, the remove payments by event code in the accounts package will be amended to exclude Service payments.

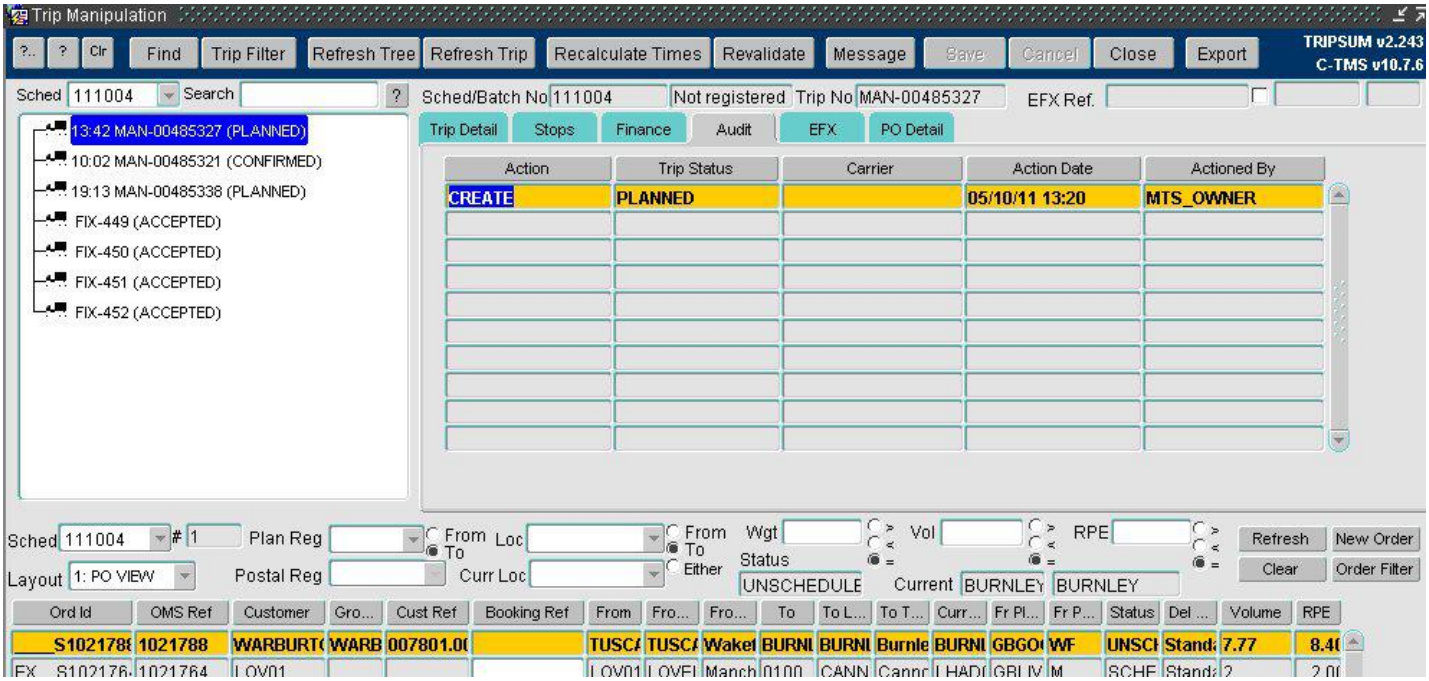
Developer Notes: Service payment records will be identified by the NARRATIVE_1 field which will be set to the value ?SERVICE?.

3.8 Service Audit

Every time a record is added , edited or deleted a record will be added to the relevant audit table. Record events in SCH_ORD_SERVICES will be recorded in SCH_ORD_AUDIT.



The status will retain the status of the order and the Change field will describe if a service has been added, removed or the service quantity has been changed. Record events in SCH_TRIP_SERVICES will be recorded in SCH_TRIP_AUDIT.



The action will be set to NEW-SERVICE, DEL_SERVICE or QTY_SERVICE (to indicate the quantity in the service record was changed). The Trip status field will display the SERVICE which has been added, removed or edited.

3.9 CSV Import

Two new CSV imports will be created to populate the ACC_SERVICES table and the ACC_SERVICE_RATES table. The imports will be created using the existing system structure for defining imports.

Any import to ACC_SERVICES will check if the service id and effective date already exist. Where a record is found, the details of the record will be amended. If the service id exists, but the effective date is different, a new record will be created.

ACC_SERVICE_RATES import will append all records from the import, setting the effective date to the date provided in the CSV. If no effective date is provided, the system will set the effective date to today?s date.

Two new procedures will be added to the IMP_2 package and the imports will be available to run from the standard import screen.

Table updates Required

New Tables:

SCH_TRIP_SERVICES	
SCHED_NAME	VARCHAR2(6)
TRIP_ID	VARCHAR2(12)
SERVICE_ID	VARCHAR2(12)
SERVICE_QTY	VARCHAR2(12)
SERVICE_VALUE	VARCHAR2(12)
INHERITED	VARCHAR2(1)
SCH_ORD_SERVICES	
OMS_REF	VARCHAR2(12)
SERVICE_ID	VARCHAR2(12)
SERVICE_QTY	VARCHAR2(12)
SERVICE_VALUE	VARCHAR2(12)
INHERITED	VARCHAR2(1)
ACC_SERVICES	
SERVICE_ID	VARCHAR2(12)
SERVICE_NAME	VARCHAR2(100)
SERVICE_EVENT	VARCHAR2(12)
INITIATED_FROM_TASK	VARCHAR2(1)
ACC_SERVICE_RATES	
DEBIT_ACC	VARCHAR2(12)
CREDIT_ACC	VARCHAR2(12)
SERVICE_ID	VARCHAR2(12)
EFFECTIVE_DATE	DATE
CHARGE_TYPE	VARCHAR2(12)
AMOUNT	NUMBER

Modules to be changed

Module Name	Module Type	Notes
ORDER.fmb	Orders Screen	New SERVICES tab
TRIP_PLAN.fmb	Trip Planning Screen	New SERVICES tab
TRIPSUM.fmb	Trip Manipulation Screen	New SERVICES tab
EXECUTION.fmb	Trip Execution Screen	New SERVICES tab
TRIP_DTL.fmb	Trip Debrief screen	New SERVICES tab
IMP_2.sql	New Import package	2 New import process
ACC.sql	Accounts package	Amend remove payments code
RATE.sql	Rating engine package	New procedures for services
ACC.fmb	Accounts screen	2 new tabs for ACC_SERVICES and ACC_SERVICE_RATES.
TUI_SCH_ORD_SERVICES	New trigger	
TUI_SCH_TRIP_SERVICES	New trigger	

TRM.sql	Trip manipulation package	Add in service procedures to set_status procedure
OMS.sql	Order Management package	Add in service procedures to Validate Order.

References

Ref No	Document Title & ID	Version	Date
1	EST 291360 MS-8KNGFM Services v1.0	1.0	21/09/11

Glossary

Term or Acronym	Meaning
C-TMS	Calidus TMS

Document History

Version	Date	Status	Reason	Initials
0.1	14/10/11	Draft	Initial version	SEW
1.0	17/10/11	Issue	Reviewed and Issued	DJM
2.0	18/10/11	Issue	Re-issued	MJC
2.1	20/10/11	Amend	Amended following client comments	SEW

4 AUTHORISED BY

Matt Crisford	Development Manager
Peter Greer	TMSCC MTS Product Manager