

**292987 v2.0**

Aptean Ltd  
Copyright © 2011-2025.

# Contents

<b>1 292987.....</b>	<b>1</b>
1.1 Client Requirement.....	2
1.2 Solution.....	2
1.3 Scope.....	4
<b>2 Set-up.....</b>	<b>5</b>
2.1 Pre-requisites.....	5
2.2 Menu Structure.....	5
2.3 Data.....	5
2.4 Implementation Advice.....	5
<b>3 Functional Description.....</b>	<b>6</b>
3.1 C-TMS Activity Extract (Trip Level).....	6
3.2 Delivery Overview (Order Level).....	7
3.3 Provider Revenue (Carrier Level).....	8
<b>4 AUTHORISED BY.....</b>	<b>11</b>

1 292987

**A**aptean

DHL C-TMS

## C-TMS reports to be produced

### FUNCTIONAL SPECIFICATION - 10.7

01/11/11 - 2.0

Reference: 292987-TH-8MLJTQ



## Contents

### 1.1 Client Requirement

Development of 3 new extracts.

- C-TMS Activity Extract
- C-TMS Delivery Overview
- C-TMS Provider Revenue

### 1.2 Solution

Three new CSV extract reports will be developed to be available to be run from the Exports screen the details of each extract are described below.

#### C-TMS Activity Extract(Trip Level)

The following fields are required in the extract

Name	C-TMS field	C-TMS Table
TO_PLANNING_REG	PLANNING_REGION	GEO_LOCATION record for the to_loc of the order
SCHEDULE DATE	SCHED_NAME	SCH_TRIP
TRIP ID	TRIP_ID	SCH_TRIP
CARRIER	CARRIER_ID	SCH_TRIP
CARRIER_TYPE	CARRIER_TYPE_ID	RES_CARRIER
DRIVER ID	FORNAME SURNAME	SCH_TRIP.driver_id join to RES_PERSON.id
VEHICLE ID	TRACTOR_ID	SCH_TRIP
TRAILER ID	TRAILER_ID	SCH_TRIP
TRAILER TYPE	DESCRIPTION	RES_TRAILER_TYPE
FROM LOC	LOCATION_ID	SCH_TRIP_STOP (the start location of the trip)
FROM TOWN	TOWN	GEO_LOCATION (town for the above location id)
FROM POSTCODE	POSTCODE	GEO_LOCATION(postcode for the above location id)
TO LOC	LOCATION_ID	SCH_TRIP_STOP(location id of the last delivery on the trip)
TO LOC TOWN	TOWN	GEO_LOCATION(town for the above location id)
TO LOC POSTCODE	POSTCODE	GEO_LOCATION(postcode for the above location id)
FIRST DEL TIME	EARLY_DEL	SCH_ORD(indicates the early del Date/Time of the first order on the trip)
SHIPPMENT	ROUTE_CODE	SCH_TRIP
TOTAL DEL NOTES	CALCULATED	Total count of orders on the trip
COMMENTS	ORDER_COMMENTS	SCH_ORD(taken from the first order on the trip)
RPE	CALCULATED	Sum of the SCH_ORD.total_rpe for all orders on the trip
REVENUE	CALCULATED	Sum of the SCH_ORD.ord_revenue for all orders on the trip
COST	CALCULATED	Sum of the SCH_ORD.ord_cost for all orders on the trip
POD RECEIVED	DERIVED	Will only be set if all orders on the trip have the SCH_ORD.POD set

There will be three parameters available to control the records selected,

- From Schedule
- To Schedule (the schedules will be used to select trips with a schedule name between these 2 values)
- Depot will restrict orders selected by Owning Depot. If the user only has one owning depot the list will default to this else the owning depot can be selected from the list. If no depot is selected all relevant depots will be used



## Delivery Overview (Order Level)

The Following fields are required in the extract

Name	C-TMS field	C-TMS Table
TO_PLANNING_REG	PLANNING_REGION	GEO_LOCATION planning region of the to_loc of the order
SCHEDULE DATE	SCHED_NAME	SCH_TRIP
TRIP ID	TRIP_ID	SCH_TRIP
DROP NUMBER	CALCULATED	Taken from trip details so delivery 1 is drop one, delivery 2 is drop 2 etc
CARRIER	CARRIER_ID	SCH_TRIP
CARRIER_TYPE_ID	CARRIER_TYPE_ID	RES_CARRIER
DRIVER ID	FORNAME SURNAME	SCH_TRIP.driver_id join to RES_PERSON.id
VEHICLE ID	TRACTOR_ID	SCH_TRIP
TRAILER ID	TRAILER_ID	SCH_TRIP
TRAILER TYPE	TRAILER_TYPE	RES_TRAILER_TYPE
SHIPPMENT	ROUTE_CODE	SCH_TRIP
DEL NOTE	EXTERNAL_REF	SCH_ORD
DEL TYPE	DELIVERY_TYPE_ID	SCH_ORD
FROM LOC	FROM_LOC	SCH_ORD
FROM TOWN	TOWN	GEO_LOCATION(town of above location id)
FROM POSTCODE	POSTCODE	GEO_LOCATION(postcode of above location id)
TO LOC	TO_LOC	SCH_ORD
TO LOC TOWN	TOWN	GEO_LOCATION(town of above location id)
TO POSTCODE	POSTCODE	GEO_LOCATION(postcode of above location id)
FIRST DEL TIME	EARLY_DEL	SCH_ORD
MILES	DISTANCE	SCH_ORD(this value may be in km?s and will need to be calculated correctly)
WEIGHT	TOTAL_WEIGHT	SCH_ORD
RPE	TOTAL_RPE	SCH_ORD
ORDER REVENUE	ORD_REVENUE	SCH_ORD
TRIP COST	TRIP_COST	SCH_TRIP(for multiple order trips this will only be displayed on the first order)
POD CONFIRMED	POD	SCH_ORD
MANUAL FINANCE	DERIVED	This field will be set to ?Y ?if any manual finance has been associated with this order/trip

There will be four parameters available to control the records selected,

- From Schedule
- To Schedule (the schedules will be used to select trips with a schedule name between these 2 values)
- Depot will restrict the trips selected by Owning Depot . If the user only has one owning depot the list will default to this else the owning depot can be selected from the list. If no depot is selected all relevant depots will be used
- POD confirmed will restrict orders selected by proof of delivery the value can be Yes/No/Both

## Provider Revenue (Carrier Level)

The Following fields are required in the extract

Name	C-TMS field	C-TMS Table
DEPOT	OWING_DEPOT	SCH_TRIP
CARRIER ID	CARRIER_ID	SCH_TRIP
CARRIER_TYPE_ID	CARRIER_TYPE_ID	RES_CARRIER
TOTAL TRIPS	CALCULATED	Total trips for the carrier within schedule range
TOTAL ORDERS	CALCULATED	Total of orders on the trips
TOTAL REVENUE	CALCULATED	Total SCH_ORD.ord_revenue of orders



TOTAL COST      CALCULATED      Total SCH\_ORD.ord\_cost of orders

MARGIN      CALCULATED      TOTAL REVENUE - TOTAL COST

% MARGIN      CALCULATED      TOTAL REVENUE - TOTAL COST as a % value

There will be Three parameters available to control the records selected all totals will be produced based on the trip carrier,

- From Schedule
- To Schedule (the schedules will be used to select trips with a schedule name between these 2 values)
- Depot will restrict the trips selected by Owning Depot .If the user only has one owning depot the list will default to this else the owning depot can be selected from the list. If no depot is selected all relevant depots will be used

## 1.3 Scope

This change will be applied to system version 10.7



## 2 Set-up

### 2.1 Pre-requisites

None

### 2.2 Menu Structure

Unchanged

### 2.3 Data

New entries will be inserted into the REP\_REPORT and REP\_REPORT\_PARAMS tables to control the reports.

### 2.4 Implementation Advice

A system super user will be required to grant access to the required extracts

Group Name	Description	Enabled
ADMIN	Administrator of MTS	<input checked="" type="checkbox"/>
CUSTOMER_CHARGES	Administrators of Customer Fuel and Premium Charges	<input checked="" type="checkbox"/>
EDI IMPLEMENTOR	EDI Administration - EDI_OWNER ONLY	<input checked="" type="checkbox"/>
EXPORTS	Exports	<input checked="" type="checkbox"/>
IMPLEMENTORS	MTS Implementation Team	<input checked="" type="checkbox"/>
MTS PROJECT IMPLEMENTORS	Non Competency Centre Implementors	<input checked="" type="checkbox"/>
MTS USER ADMIN	MTS User Administrator	<input checked="" type="checkbox"/>
OBS IMPLEMENTORS	OBS Implementation Team	<input checked="" type="checkbox"/>

Extracts authorised for use by this Group

- BGW\_DEBRIEF
- Dunelm Manual Loaded Items Extract

Extracts Available

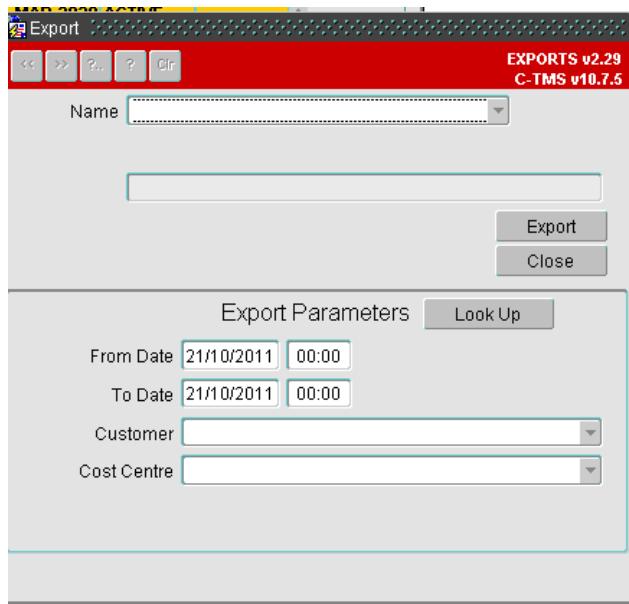
- 10.6 Customer By Product Export
- 10.6 Customer Location Export
- 10.6 Customer Product By Arrival
- 10.6 Data Extract
- 10.6 Load Status Export
- 10.6 Matthew Clark Data Extract



## 3 Functional Description

### 3.1 C-TMS Activity Extract (Trip Level)

An extract will be developed and will be available to be run from the Exports Screen and example of which is shown below



The user will be prompted to enter From Schedule, To Schedule and the Owning Depot.

The following information will be required in the extract.

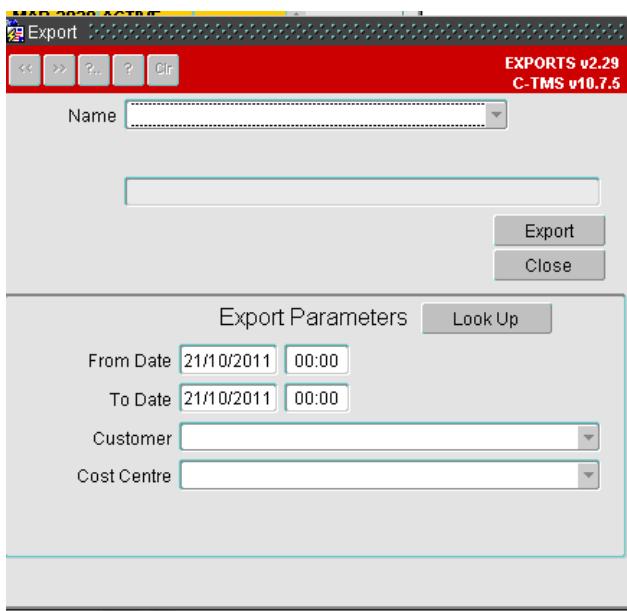
Name	C-TMS field	C-TMS Table
TO_PLANNING_REG	PLANNING_REGION	GEO_LOCATION for to_loc of the order
SCHEDULE DATE	SCHED_NAME	SCH_TRIP
TRIP ID	TRIP_ID	SCH_TRIP
CARRIER	CARRIER_ID	SCH_TRIP
CARRIER_TYPE_ID	CARRIER_TYPE_ID	RES_CARRIER
DRIVER ID	FORNAME SURNAME	SCH_TRIP.driver_id join to RES_PERSON.id
VEHICLE ID	TRACTOR_ID	SCH_TRIP
TRAILER ID	TRAILER_ID	SCH_TRIP
TRAILER TYPE	DESCRIPTION	RES_TRAILER_TYPE
FROM LOC	LOCATION_ID	SCH_TRIP_STOP (the start location of the trip)
FROM TOWN	TOWN	GEO_LOCATION (town for the above location id)
FROM POSTCODE	POSTCODE	GEO_LOCATION(postcode for the above location id)
TO LOC	LOCATION_ID	SCH_TRIP_STOP(location id of the last delivery on the trip)
TO LOC TOWN	TOWN	GEO_LOCATION(town for the above location id)
TO LOC POSTCODE	POSTCODE	GEO_LOCATION(postcode for the above location id)
FIRST DEL TIME	EARLY_DEL	SCH_ORD(indicates the early del Date/Time of the first order on the trip)
SHIPPMENT	ROUTE_CODE	SCH_TRIP
TOTAL DEL NOTES	CALCULATED	Total count of orders on the trip
COMMENTS	ORDER_COMMENTS	SCH_ORD(taken from the first order on the trip)
RPE	CALCULATED	Sum of the SCH_ORD.total_rpe for all orders on the trip
REVENUE	CALCULATED	Sum of the SCH_ORD.ord_revenue for all orders on the trip
COST	CALCULATED	Sum of the SCH_ORD.ord_cost for all orders on the trip



POD RECEIVED      DERIVED      Will only be set if all orders on the trip have the SCH\_ORD.POD set  
 The extract will be developed into the standard CSV processing package. A new procedure to select the data and a new procedure to write the data to the extract file will be added to the existing program. The schedules and the owning depot parameters will be used to control the information selected. If the user only has one owning depot the list will default to this value. If no owning depot is selected all depots relevant to the user will be used.

## 3.2 Delivery Overview (Order Level)

An extract will be developed and will be available to be run from the Exports Screen and example of which is shown below:



The user will be required to enter the From and To Schedule, the Owning Depot and optionally a Proof of Delivery Y/N/ALL option.

The following information is required in the extract

Name	C-TMS field	C-TMS Table
TO_PLANNING_REG	PLANNING_REGION	GEO_LOCATION planning region of to_loc
SCHEDULE DATE	SCHED_NAME	SCH_TRIP
TRIP ID	TRIP_ID	SCH_TRIP
DROP NUMBER	CALCULATED	Taken from trip details so delivery 1 is drop one, delivery 2 is drop 2 etc
CARRIER	CARRIER_ID	SCH_TRIP
CARRIER_TYPE_ID	CARRIER_TYPE_ID	RES_CARRIER
DRIVER ID	FORENAME SURNAME	SCH_TRIP.driver_id join to RES_PERSON.id
VEHICLE ID	TRACTOR_ID	SCH_TRIP
TRAILER ID	TRAILER_ID	SCH_TRIP
TRAILER TYPE	TRAILER_TYPE	RES_TRAILER_TYPE
SHIPPMENT	ROUTE_CODE	SCH_TRIP
DEL NOTE	EXTERNAL_REF	SCH_ORD
DEL TYPE	DELIVERY_TYPE_ID	SCH_ORD
FROM LOC	FROM_LOC	SCH_ORD
FROM TOWN	TOWN	GEO_LOCATION(town of above location id)
FROM POSTCODE	POSTCODE	GEO_LOCATION(postcode of above location id)

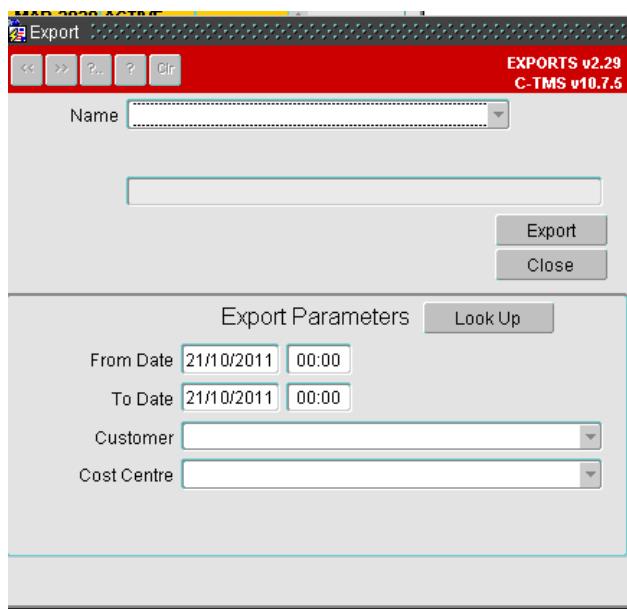


TO LOC	TO_LOC	SCH_ORD
TO LOC TOWN	TOWN	GEO_LOCATION(town of above location id)
TO POSTCODE	POSTCODE	GEO_LOCATION(postcode of above location id)
FIRST DEL TIME	EARLY_DEL	SCH_ORD
MILES	DISTANCE	SCH_ORD(this value may be in km?s and will need to be calculated correctly)
WEIGHT	TOTAL_WEIGHT	SCH_ORD
RPE	TOTAL_RPE	SCH_ORD
ORDER REVENUE	ORD_REVENUE	SCH_ORD
TRIP COST	TRIP_COST	SCH_TRIP(for multiple order trips this will only be displayed on the first order)
POD CONFIRMED	POD	SCH_ORD
MANUAL FINANCE	DERIVED	This field will be set to ?Y ?if any manual finance has been associated with this order/trip

The extract will be developed into the standard CSV processing package. A new procedure to select the data and a new procedure to write the data to the extract file will be added to the existing program. The schedules and the owning depot parameters will be used to control the information selected. If the user only has one owning depot the list will default to this value. If no owning depot is selected all depots relevant to the user will be used. The Proof of delivery parameter will be set to Yes/No or Both this will control records extracted by proof of delivery.

### 3.3 Provider Revenue (Carrier Level)

An extract will be developed and will be available to be run from the Exports Screen and example of which is shown below:



The user will be required to enter the From and To Schedules and the Owning Depot.

The extract will contain the following information:

Name	C-TMS field	C-TMS Table
DEPOT	OWNING_DEPOT	SCH_TRIP
CARRIER ID	CARRIER_ID	SCH_TRIP
CARRIER_TYPE_ID	CARRIER_TYPE_ID	RES_CARRIER
TOTAL TRIPS	CALCULATED	Total trips for the carrier within schedule range
TOTAL ORDERS	CALCULATED	Total of orders on the trips



TOTAL REVENUE	CALCULATED	Total SCH_ORD.ord_revenue of orders
TOTAL COST	CALCULATED	Total SCH_ORD.ord_cost of orders
MARGIN	CALCULATED	TOTAL REVENUE - TOTAL COST
% MARGIN	CALCULATED	TOTAL REVENUE - TOTAL COST as a % value

The extract will be developed into the standard CSV processing package. A new procedure to select the data and a new procedure to write the data to the extract file will be added to the existing program. The schedules and the owning depot parameters will be used to control the information selected. If the user only has one owning depot the list will default to this value. If no owning depot is selected all depots relevant to the user will be used. All totals produced will be created at carrier level within the specified parameters

## Table Updates Required

DATA\_REP\_REPORT\_292987.SQL

```
insert into rep_report (name, report_type, filename, printer_type, show_pform, orientation, proc_name) values
('ACTIVITY_EXTRACT','CSV','ACTIVITY_EXTRACT','Laser','F','LANDSCAPE', 'dp_csv5.activity_extract');
```

```
insert into rep_report (name, report_type, filename, printer_type, show_pform, orientation, proc_name) values
('DELIVERY_OVERVIEW','CSV','DELIVERY_OVERVIEW','Laser','F','LANDSCAPE', 'dp_csv5.delivery_overview');
```

```
insert into rep_report (name, report_type, filename, printer_type, show_pform, orientation, proc_name) values
('PROVIDER_REVENUE','CSV','PROVIDER_REVENUE','Laser','F','LANDSCAPE', 'dp_csv5.provider_revenue');
```

DATA\_REP\_REPORT\_PARAM\_292987.SQL

```
INSERT INTO rep_report_param(report_name, report_type, param_type, param_name, conditional, param_title,
report_list) values ('ACTIVITY_EXTRACT','CSV','P_START_SCHED','P_START_SCHED','M','From Schedule','N');
```

```
INSERT INTO rep_report_param(report_name, report_type, param_type, param_name, conditional, param_title,
report_list) values ('ACTIVITY_EXTRACT','CSV','P_END_SCHED','P_END_SCHED','M','To Schedule','N');
```

```
INSERT INTO rep_report_param(report_name, report_type, param_type, param_name, conditional, param_title,
sql_string, report_list) values ('ACTIVITY_EXTRACT','CSV','PG_SELECT_LIST1','P_OWNING_DEPOT','M','Owning
Depot', 'SELECT distinct(gl.location_id) COL1, gl.location_name COL2 FROM geo_location gl, adm_user_param aup
WHERE gl.location_id = aup.value AND aup.username = (SELECT user FROM dual) AND aup.param_type =
BASED_AT AND gl.depot = RDC and NVL(gl.inactive,N) = 'N' UNION SELECT distinct (gl.location_id) COL1,
gl.location_name COL2 FROM geo_location gl, adm_user_param aup WHERE gl.location_name like
DECODE(aup.value,Y,%%) AND aup.username = (SELECT user FROM dual) and aup.param_type = ALL_DEPOTS and
gl.depot = RDC and NVL(gl.inactive,N) = N,'N');
```

```
INSERT INTO rep_report_param(report_name, report_type, param_type, param_name, conditional, param_title,
report_list) values ('DELIVERY_OVERVIEW','CSV','P_START_SCHED','P_START_SCHED','M','From Schedule','N');
```

```
INSERT INTO rep_report_param(report_name, report_type, param_type, param_name, conditional, param_title,
report_list) values ('DELIVERY_OVERVIEW','CSV','P_END_SCHED','P_END_SCHED','M','To Schedule','N');
```

```
INSERT INTO rep_report_param(report_name, report_type, param_type, param_name, conditional, param_title,
sql_string, report_list) values ('DELIVERY_OVERVIEW','CSV','PG_SELECT_LIST1','P_OWNING_DEPOT','M','Owning
Depot', 'SELECT distinct(gl.location_id) COL1, gl.location_name COL2 FROM geo_location gl, adm_user_param aup
WHERE gl.location_id = aup.value AND aup.username = (SELECT user FROM dual) AND aup.param_type =
BASED_AT AND gl.depot = RDC and NVL(gl.inactive,N) = 'N' UNION SELECT distinct (gl.location_id) COL1,
gl.location_name COL2 FROM geo_location gl, adm_user_param aup WHERE gl.location_name like
DECODE(aup.value,Y,%%) AND aup.username = (SELECT user FROM dual) and aup.param_type = ALL_DEPOTS and
gl.depot = RDC and NVL(gl.inactive,N) = N,'N');
```

```
INSERT INTO rep_report_param(report_name, report_type, param_type, param_name, conditional, param_title,
default_value_type,default_value report_list) values ('DELIVERY_OVERVIEW','CSV','PG_FFCHAR1','PFPOD','M','POD
Confirmed','STRING','Y','N');
```

```
INSERT INTO rep_report_param(report_name, report_type, param_type, param_name, conditional, param_title,
report_list) values ('PROVIDER_REVENUE','CSV','P_START_SCHED','P_START_SCHED','M','From Schedule','N');
```



INSERT INTO rep\_report\_param(report\_name, report\_type, param\_type, param\_name, conditional, param\_title, report\_list) values ('PROVIDER\_REVENUE','CSV','P\_END\_SCHED','P\_END\_SCHED','M','To Schedule','N');

INSERT INTO rep\_report\_param(report\_name, report\_type, param\_type, param\_name, conditional, param\_title, sql\_string, report\_list) values ('PROVIDER\_REVENUE','CSV','PG\_SELECT\_LIST1','P\_OWNING\_DEPOT','M','Owning Depot', 'SELECT distinct(gl.location\_id) COL1, gl.location\_name COL2 FROM geo\_location gl, adm\_user\_param aup WHERE gl.location\_id = aup.value AND aup.username = (SELECT user FROM dual) AND aup.param\_type = BASED\_AT AND gl.depot = RDC and NVL(gl.inactive,N) = 'N' UNION SELECT distinct (gl.location\_id) COL1, gl.location\_name COL2 FROM geo\_location gl, adm\_user\_param aup WHERE gl.location\_name like DECODE(aup.value,Y,%) AND aup.username = (SELECT user FROM dual) and aup.param\_type = ALL\_DEPOTS and gl.depot = RDC and NVL(gl.inactive,N) = 'N','N');

## Modules to be changed

Module Name	Module Type	Notes
DP_CSV5.sql	Package	Add new extracts

## References

Ref No	Document Title & ID	Version	Date
1	EST-292987 TH-8MLJTQ	0.1	18/10/11

## Glossary

Term or Acronym	Meaning
C-TMS	Calidus TMS

## Document History

Version	Date	Status	Reason	Initials
0.1	21/10/11	Draft	Initial version	CAK
0.2	24/10/11	Draft	Reviewed	MJC
1.0	24/10/11	Issue	Issued	MJC
1.1	27/10/11	Draft	Revised	CAK
1.2	31/10/11	Draft	Revised	CAK
2.0	01/11/11	Issue	Reviewed and Issued	MJC



## 4 AUTHORISED BY

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

