

286485 v1.0

Aptean Ltd
Copyright © 2011-2025.

Contents

- 1 286485.....1**
 - 1.1 Client Requirement.....2
 - 1.2 Solution.....2
 - 1.3 Scope.....2
- 2 Set-up.....3**
 - 2.1 Pre-requisites.....3
 - 2.2 Menu Structure.....3
 - 2.3 Data.....3
- 3 Functional Description.....4**
 - 3.1 ?Import? Screen (IMPORTS_EXEC).....4
 - 3.2 Triggers.....5
 - 3.3 ?INT_XML_IN? Package.....5
- 4 AUTHORISED BY.....8**

1 286485



DHL MTS

Ethos - Unique Location ID

FUNCTIONAL SPECIFICATION - 10.6

18/03/2011 - 1.0
Reference: 286485 FS-8EEKUY



1.1 Client Requirement

Change Request Summary:

Unique Location ID - Currently there is no unique location ID which comes in the file from EcoTherm. Request to build something into the system to validate information and create the location ID.

Change Request Details:

Unique Location ID - Currently there is no unique location ID which comes in the file from EcoTherm. Request to build something into the system to validate information and create the location ID.

Benefits identified as a result of the change:

Locations imported into the system with Unique Location ID.

1.2 Solution

Currently any locations that are sent into C-TMS without an Address ID are rejected by the interface. If a location is sent into C-TMS with a location ID that doesn't already exist then a location is created (either using the data from the file or as a temporary New Location to be updated manually).

As the standard XML flow is being used a new EDI import process parameter will be created that will control how the locations are handled by the interface. If this registry is set then the interface will allow addresses on an order that contain a location ID of 'Unknown'. The location name and postcode will then be used as to identify this location, as these two items should make the location unique. The EDI import process parameters may be entered for 'Inbound' directional flows via the 'Params' button in the 'Import' screen.

When the order is created in C-TMS the locations are currently checked and if they are not present then they are added to the standing data tables. A new check will be created that will use the location name and postcode to identify whether this location already exists. If it does not then a new location ID will be generated using a sequence. Using the location name with the sequence to easily identify these locations. The first address line of the location also sent in the file and will be a hard coded value assigned before the file is sent to C-TMS

This location id will then be stored against the order and will be used by the other functionality within C-TMS (trip creation, etc). If the location exists in C-TMS (based on the location name and postcode) then the existing location id will be stored against the Order.

N.B. Any output from the system will contain this location ID and address information as this is all we have stored in C-TMS.

1.3 Scope

This change will be applied to system version 10.6..



2 Set-up

2.1 Pre-requisites

A new EDI import process parameter called ?CREATE_UNKNOWN_LOCS? exists.

Two new columns exist on table ?EDI_PROCESS_TRIGGERS?.

2.2 Menu Structure

Unchanged

2.3 Data

The new EDI import process parameter called ?CREATE_UNKNOWN_LOCS? should be set to ?Y?.

Table ?EDI_PROCESS_TRIGGERS? will need two new columns and have them added to its primary key and thus index (see Appendix A).

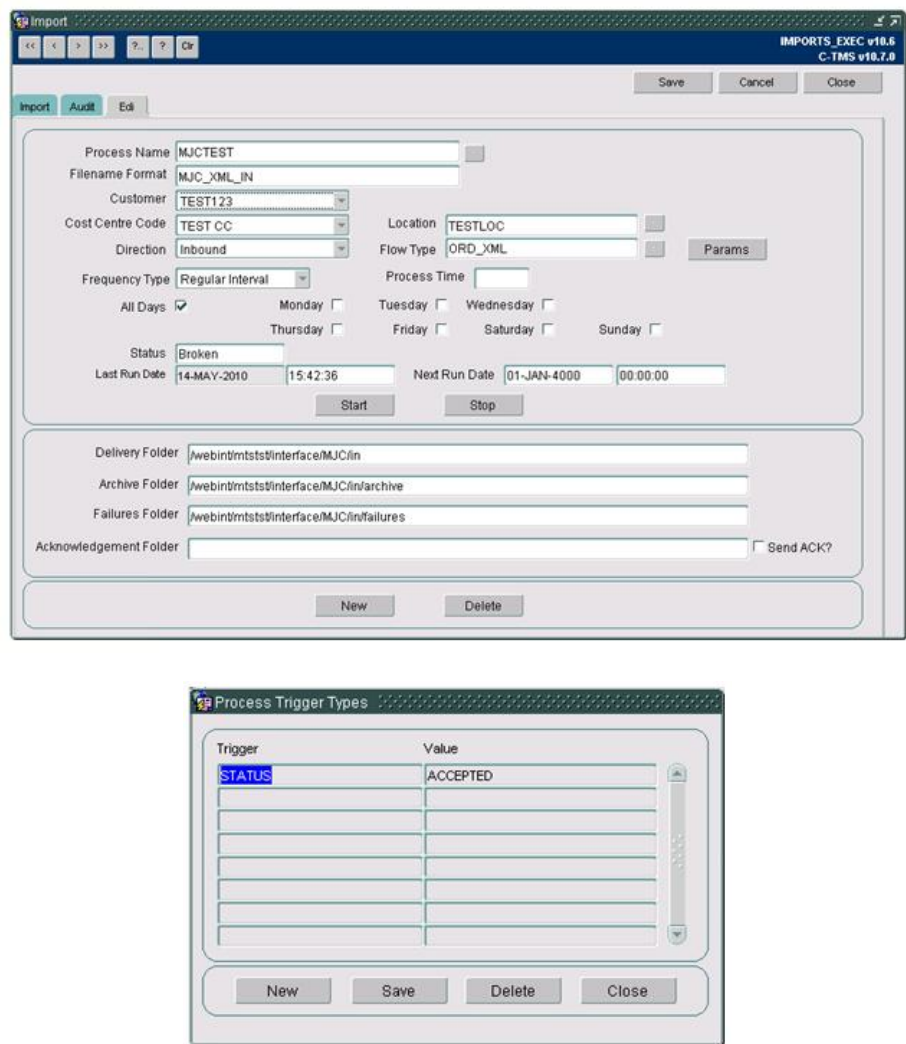


3 Functional Description

3.1 ?Import? Screen (IMPORTS_EXEC)

The ?Edi? tab page of the ?Import? screen will be changed to enable parameters to be entered for the inbound EDI flows.

The ?Params? button will be modified to allow parameters to be entered where the direction of the flow is ?Inbound? (if the direction of the flow is ?Outbound? then the existing canvas will be displayed with the title ?Process Trigger Types?):



If the direction of the flow is ?Inbound? then the name of the canvas displayed will be ?Process Parameters? and the columns will have the headings ?Parameter? and ?Value?.

The trigger types and the process parameters will be stored on the same table ?EDI_PROCESS_TRIGGERS? as follows:

PROCESS_NAME	TRIGGER_TYPE	TRIGGER_VALUE	DIRECTION	TYPE
?Process Name?	?Trigger? or ?Parameter?	?Value?	?O? or ?I?	?T? or ?P?

The ?DIRECTION? will be ?O? for ?outbound? or ?I? for ?Inbound? and the ?TYPE? will be ?T? for ?Trigger? or ?P? for ?Parameter?. These columns may account for future functionality.

In this canvas a process parameter called ?CREATE_UNKNOWN_LOCS? may be entered to determine if ?Unknown? locations may create new source and destination locations via the XML order entry process: if it is set to ?Y? then the new locations may be created otherwise the existing process to create temporary locations will be followed.

The description (not stored) of the process parameter will be:

?May the XML orders create a new source or destination location if it is unknown (Y/N)??

3.2 Triggers

The existing data on the table ?EDI_PROCESS_TRIGGERS? need not be changed at present, however, it will be necessary for the directional flow to be checked to ensure that it is an outbound process.

Developer Notes: Triggers ?TRG_SCH_TRIP_XML_GEN?, ?TRG_SHA_XML_GEN? and ?TRG_STS_XML_GEN? will be changed to check that the direction is ?O? and the type is ?T? of the record on table ?EDI_PROCESS_TRIGGERS? in cursor ?c_check_status?.

3.3 ?INT_XML_IN? Package

3.3.1 ?PROCESS_ORD_XML_IN? Procedure

The procedure ?PROCESS_ORD_XML_IN? will be changed to enable the unknown locations to be processed differently.

If the process parameter ?CREATE_UNKNOWN_LOCS? does not exist or is not set to ?Y?, or the location ID is not ?Unknown?, then the existing functionality will be retained: the existing functionality will enable temporary locations to be added to the database to be updated at a later time once the actual location details are known.

If the location ID provided does not exist in C-TMS then the location will be created for that location ID and with a location name of ?NEW LOCATION {SEQUENCE_NUMBER}? and a depot type of ?BRANCH?; the sequence number will be obtained from database sequence ?SEQ_NEW_LOC?.

The location ID generated will then be used as the source (i.e. ?SCH_ORD.FROM_LOC?) or destination (i.e. ?SCH_ORD.TO_LOC?) location when the order is created.

If the process parameter ?CREATE_UNKNOWN_LOCS? is set to ?Y? and the location ID is ?Unknown? then the functionality that enables unknown locations to be added to the database will be changed so that the location created is related to the location?s name and postcode.

The location?s name and postcode will form a unique combination for the unknown location ID and the following cursor will be used to assess whether a location ID already exists for the combination:

Developers Notes:

CURSOR c_check_location_comb (ci_name geo_location.location_name%TYPE, ci_postcode geo_location.postcode%TYPE) IS

SELECT location_id

FROM geo_location

WHERE location_name = ci_name

AND postcode = ci_postcode;

The cursor ?c_check_location_comb? will be called after it has been checked that the location ID has been received (i.e. after error message ?To Location(DEL) cannot be NULL? may be generated) and the values obtained from XML tags <ADDRESS_NAME> and <ADDRESS_POSTCODE> will be passed to this cursor.

If the cursor obtains a location ID stored for the combination then it will be used as the location ID for the order, however, if a location ID is not obtained then a location ID will be generated based on the location name.

As the location ID is limited to 12 characters and the location name may be up to 50 characters in length then only the first eight characters of the location name will be used to generate the location ID with a three-digit sequence number appended to make it unique; the following cursor will be called to obtain the latest sequence number for the location name:

Developers Notes:

CURSOR c_check_location_seq (ci_name geo_location.location_name%TYPE) IS



```

SELECT max(location_id)

FROM geo_location

WHERE substr(location_name, 1, 8) || '?-' = substr(ci_name, 1, 8) || '?-';

```

The location ID will be updated from 'Unknown' using the statement below:

For the source location (i.e. address type 'DEP'):

```
t_address_id_dep := substr(location_name, 1, 8) || '?-' || to_char(to_number(substr(tv_max_loc, 10, 3)) + 1);
```

For the destination location (i.e. address type 'DEL'):

```
t_address_id_del := substr(location_name, 1, 8) || '?-' || to_char(to_number(substr(tv_max_loc, 10, 3)) + 1);
```

If the location received is 'Unknown' and the process parameter 'CREATE_UNKNOWN_LOCS' is 'Y' then the location mapping, to also assess the address account number received in XML tag <ADDRESS_AC>, will not be performed.

Once the temporary variables 't_address_id_dep' and 't_address_id_del' have been updated they may be stored on table 'INT_XML_ORD_HEADER' in columns 'LOCATION_ID_DEP' and 'LOCATION_ID_DEL' as at present.

If a new location ID has been generated then it will be stored on table 'GEO_LOCATION' with the address information provided in the XML tags:

Column	XML Tag / Value
LOCATION_ID	t_address_id_dep or t_address_id_del
DEPOT	'BRANCH'
LOCATION_NAME	ADDRESS_NAME
ADDRESS_LINE_1	ADDRESS_LINE1
ADDRESS_LINE_2	ADDRESS_LINE2
ADDRESS_LINE_3	ADDRESS_LINE3
TOWN	ADDRESS_TOWN
COUNTY	ADDRESS_COUNTY
COUNTRY_CODE	ADDRESS_COUNTRY_CODE
POSTCODE	ADDRESS_POSTCODE
INACTIVE	'N'

N.B. The new location will be created without a latitude, longitude, postal region, planning region, loading rate, unloading rate, etc and these values will need to be updated prior to trip planning of the order.

Procedure 'PROCESS_ORDER' will then be called and will obtain the updated location IDs from table 'INT_XML_ORD_HEADER', they will then be passed to procedures 'F_PROCESS_ORDER_RECORD' and 'POST_ORDER' and used for the order created.

The order and subsequent trips will then be able to reference the new location ID when necessary.

3.3.2 'POST_ORDER' Procedure

Procedure 'POST_ORDER' will not need to be changed because the new location IDs will have been stored on table 'GEO_LOCATION' in procedure 'PROCESS_XML_ORD_IN'.

This procedure will still assess the location ID received but will find the existing record.

N.B. The check for the active location will be retained (i.e. if the location is inactive then access to the function 'GEO_ACTIVATE_LOCATION' must be present to enable it to be validated).



N.B. The validation of the original source location and the final destination will not be changed because they do not create new locations as they must exist and be active.

Table Updates Required

The table ?EDI_PROCESS_TRIGGERS? will include two new columns and the primary key will be extended to include these columns:

Glossary

Term or Acronym	Meaning
C-TMS	Calidus TMS

References

Ref No	Document Title & ID	Version	Date
1	EST-286485 FS-8EEKUY Ethos - Unique Location ID v1.0.doc	1.0	08/03/11

Document History

Version	Date	Status	Reason	Initials
0.1	18/03/11	Draft	Initial version	PDR
0.2	18/03/11	Draft	Reviewed	MJC
1.0	18/03/11	Issued	Issued	MJC



4 AUTHORISED BY

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

