286642 v1.0

Contents

1 286642

DHL MTS

Integration Orders

FUNCTIONAL SPECIFICATION - 10.6

```
- 1.0
```

Reference: FS 286642 PM-8ELM6Z

Ready for **What's Next, Now**™

1.1 Client Requirement

Transport Orders GOLD to C-TMS. Transport order and address details for collect and deliver interfaced from GOLD to C-TMS via DHL Link covering order header and line details.

1.2 Solution

The data will be imported from ESI using the standard order flow

Reference Data

A new version of the flow will be created to allow the mapping of Reference data from OPENFIELD into C-TMS. Currently C-TMS does not store the reference data required so new fields will be created to store the information.

There are up to 19 different reference fields that can be populated for an order and the number of reference fields with data will vary from order to order.

The Standard flow has been amended to receive multiple sub reference records. A new item SUB_REF_IDENTIFIER will be populated with reference field name and a second item SUB_REFERENCE will be populated with the reference data, see TRIPORDER-2.13.xsd

A new table will be needed to store the passed reference data

Table SCH_ORD_REFERENCE

OMS_REF

SUB_REF_NAME

SUB_REF_VALUE

The records will be added to the SCH_ORD_REFERENCE table and displayed on screen as a grid of values. If the C-TMS code is required to retrieve a reference value, we look for the value of the SUB_REF_NAME being equal to a specific string.

Locations

The location id we receive in the flow will be a 19 digit value which is too big to store in the location id field, max (12). A decode record will be created to translate this code into a numeric location code for C-TMS.

When an address is received, the decode will be checked to see if the address id has already been created on the system. If a decode record exists the location id will be populated from C-TMS and the address details sent by OPENFIELD will be updated in the location table GEO_LOCATION.

The update to the location t will be carried out with a NOWAIT to ensure that users are not locked out of records due to the location being updated. A NOWAIT will only update the location record if it is not currently in use. If a user is accessing the location record via the maintenance screen or an order, the update will not be attempted. A exception will track if the NOWAIT caused the update to fail and a message will be recorded in INT_ERRORS to inform the user that the address was not updated.

Ready for **What's Next, Now**™

If a decode record is not found, a new geo_location record will be created. The new record will only be added after the address data has been through validation. The address will be required to include an address line, a town and a postcode. If any of the validation data is missing, the new record will not be created.

The location_id will be generated from a new sequence called SEQ_GEO_ID. The location details will be provided from the address details in the flow. When creating a new location on C-TMS, a record will also be added to the GEO_LOCATION_USAGE table which assigns locations to Customers.

LOCATION_ID (New location id generated form sequence)

USAGE CUSTOMER

USAGE_ID Customer Value

The customer value will be derived from an inbound parameter which will be populated in the GENERIC EDI screen and applied to the CUSTOMER field in the flow.

A new decode record will be added to convert the 19 digit OPENFIELD number to GEO_LOCATION.Location_id in C-TMS. To allow this a new decode type will be created called OPENFIELD_LOCS based on LOCATION. The source field will be populated with the VEL number and the target field will be populated with the newly generated LOCATION_ID. The decode type and decode field will be added to the generic EDI screen as parameters.

A new field called comments will be added to the GEO_LOCATION table which will store the VEL number. A new table will be created called GEO_LOCATION_REFERENCES which will store generic address references. The table will be 3 fields

Location_id

Reference_Name

Reference_Value

This table will be used to store the Assurance number, scheme and status . The standard flow will be amended to receive multiple address reference records per address.

The from and to addresses will be processed in the same way.

Generic EDI

The standard flow will be used to process the order data. The 2 new fields will be added to the table EDI_PROCESS_HEADER to allow the OPENFIELD data to be processed.

GROUP_NAME

UNKNOWN_LOCATIONS

DECODE_TYPE

DECODE_FIELD

The GROUP_NAME, DECODE_TYPE and DECODE_FIELD will be a VARCHAR2 which can be populated with any value and the UNKNOW_LOCATIONS will be a flag to indicate if new locations should be created.

The exisiting standard order xml process INT_XML_IN.PROCESS_ORD_XML_IN will be amended to process the OPENFIELD data

1.3 Scope

This change will be applied to system version 10.6.0 on INDTST and once approved INDPRD.

Ready for **What's Next, Now**™

2 Set-up

2.1 Pre-requisites

FS-286485 FS-8EEKUY Ethos - Unique Location ID

Part of this development will be an amendment to the EDI tab on the Import maintenance screen which will allow inbound parameters to be added to a process. These screen changes are a pre requisite to this development

2.2 Menu Structure

?Unchanged?

2.3 Data

CREATE TABLE SCH_ORD_REFERENCE

ADD COLUMN OMS_REF VARCHAR2(12) NOT NULL

ADD COLUMN SUB_REF_NAME VARCHAR2(200)

ADD COLUMN SUB_REF_VALUE VARCHAR2(200);

CREATE TABLE GEO_LOCATION_REFERENCES

ADD COLUMN LOCATION_ID VARCHAR2(12) NOT NULL

ADD COLUMN REFERENCE_NAME VARCHAR2(200)

ADD COLUMN REFERENCE_VALUE VARCHAR2(200);

CREATE TABLE INT_ORD_REFERENCES

INT_RECORD_ID VARCHAR2(12) NOT NULL

REFERENCE_NAME VARCHAR2(100)

REFERENCE_ID VARCHAR2(100)

Ready for **What's Next, Now**™

3 Functional Description

A new EDI process will be created for the import of GOLD orders into C-TMS. The process will be created using the EDI tab of the Import Maintenance screen as displayed below
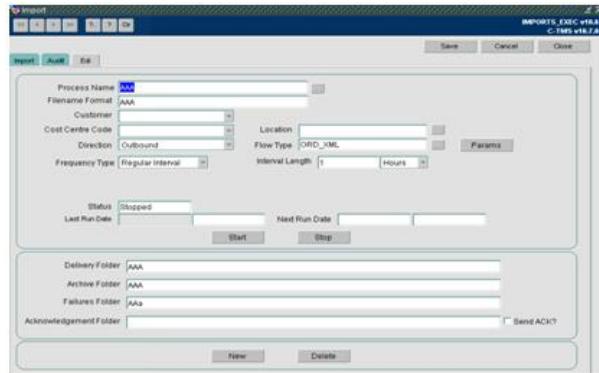
The screen will be used to state the delivery folders and the params command button will be used to define process flow parameters.

A new field will be added to the screen called group name, this will be stored in the EDI_PROCESS_HEADER table. Within the generic code, the GROUP_NAME field in C-TMS will be populated with the value from this field.

Four process flow records will be added into the EDI_PROCESS_TRIGGERS table for GOLD . These will be editable from the Params command button.

PROCESS_NAME	TRIGGER_TYPE	TRIGGER_VALUE	DIRECTION	TYPE
GOLD	GROUP_NAME	?GOLD?	I	P
GOLD	CREATE_UNKNOWN_LOCS	?Y?	I	P
GOLD	ADDRESS_ID	?GOLD_LOCS?	I	D
GOLD	CREATE_UNKNOWN_PRD	?Y?	I	P

3.1 Order References

Gold orders may have at least 19 order references supplied. C-TMS will be able to deal with 0 to an infinite number of reference records. Examples of references to be stored in this table are as follows:

1. ASSURANCE NO
2. HAULIER NO
3. FIXING NO
4. FIXING CUST REF
5. SALE GRADE
6. INTAKE REF
7. COMMODITY CODE
8. VARIETY CODE
9. CONTRACT COMMENTS
10. BIN BAY LINES
11. BIN BAY COMMENTS
12. PURCHASE NO
13. POOL
14. SALES NO
15. SALES COMMENT
16. PURCHASE GRADE
17. SALES COMMODITY
18. WEIGH BRIDGE TICKET
19. SPLIT ORDER

Ready for **What's Next, Now**™

The order reference data will be stored at the ORDER_SUB_REFS level of the XSD.

The INT_XML_IN.Process_Ord_Xml_In procedure will be amended to process the reference table. A new interface table will be created to store the reference data during processing, called INT_ORD_REFERENCES. Currently the reference values are concatenated into one field and stored against the order. This code will remain unchanged for non GOLD processes.

If the process Name is GOLD, the reference data will be processed differently. A cursor will run to select the xml tags and data within. A cursor loop will select the next reference record , storing the data in temporary variables before inserting into the new interface table INT_ORD_REFERENCE.

A new procedure within the INT_XML_IN package called POST_ORD_REFERENCES will be written to process the data from the interface table into the new C-TMS table SCH_ORD_REFERENCE. The new table can store from none to many reference records per order.

The reference data will be displayed in a new tab on the Order screen called REFERENCES. Displaying the tab will be controlled at user group level. See 286731 DK-8EME2Y Order Management for details of the screen change.

3.2 Locations

Within the PROCESS_ORD_XML_IN procedure, the address information is selected from the tags. Validation is carried out on the address id. If the id is null an error is returned, if the id is not null a decode is processed. Currently the decode is hard coded to a decode type ?CN MAPPING LOCATION?. If any value is returned from the decode , this is used as the address id.

For the GOLD process flow, a new decode called GOLD_LOCS will be added. As discussed earlier, the parameter information will be stored in the EDI_TRIGGER_PROCESSES table. The existing decode call will be amended to look for any type D records for address_id assigned to the process. An NVL to CN_MAPPING_LOCATION will maintain existing functionality.

If the parameter CREATE_UNKNOWN_LOCS has been assigned to the process and set to Y, additional checks will be carried out on the address name, town and postcode. If any of these three values are null, an error message will be written to the new variable V_ADDRESS_ERROR.

At the end of the address validation, if the decode process returns null and V_ADDRESS_ERROR is still null, a new decode record will be created for the address. This will create a location_id allowing a new location record and location usage record to be written. The location_id will be system generated based on a substring of the address name and a new location sequence. Including the address name will provide additional validation for the id based on manual inserts.

If the decode value already exists, the GEO_LOCATION record will be updated with the address information stored in the temporary variables. The update will be based on a cursor selecting the current location with a NO WAIT. This will ensure that the flow does not lock users in the system to perform an update.

Gold order addresses may have assurance records supplied. The assurance records will only be processed once the address had been validated and if necessary created in C-TMS. The assurance data will be stored at the OH_ADDRESS_REFERENCE level of the XSD.

The level is not mandatory and neither are the tags <add_ref_type> and <add_ref_value> The name of the reference data will be stored in the add_ref_type tag and the value will be stored in the add_ref_value tag.

The INT_XML_IN.Process_Ord_Xml_In procedure will be amended to process the reference table. As the location id will be available, the records can be inserted straight into C-TMS. A cursor will loop through the tag data and insert each record into the new C-TMS table GEO_LOCATION_REFERENCES.

3.3 Product Types

Product types are currently received as part of the Order Details in the XSD. For the new GOLD process, a new inbound parameter will be added called ?CREATE_UNKNOWN_PRD? which will be set to ?Y?.

Ready for **What's Next, Now**™

The parameter will allow the creation of unknown PRODUCT TYPES received from GOLD. Currently, if a product type is received which does not exist on C-TMS, an error is returned and the order is not processed.

Based on the new parameter being applied to the Process flow, if a new product type is found, rather than generating an error, a new record will be added to the PRD_PRODUCT_TYPE. The following fields will be populated.

PROD_TYPE_ID   <PRODUCT_TYPE>
PROD_TYPE_NAME <PRODUCT_TYPE>
CREATED_BY   MTS_OWNER
CREATED_DATE   SYSDATE

Ready for **What's Next, Now**™

4 Table Updates Required

References

Ref No	Document Title & ID	Version	Date
	{Detail attachments to RIO here}		

Document History

Version	Date	Status	Reason	Initials
0.1	20/03/11	Draft	Initial version	SEW
1.0	24/03/11	Issued	Reviewed and Issued	MJC

Ready for **What's Next, Now**™

5 AUTHORISED BY

Matt Crisford	Development Manager
Peter Greer	TMSCC MTS Product Manager

Ready for **What's Next, Now**™