

**271604**

Aptean Ltd  
Copyright © 2011-2025.

# Contents

- 1 271604.....1
- 2 271604 - PA-7XVL6X/ EFX Download Additions.....2
- 3 FUNCTIONAL OVERVIEW.....3
  - 3.1 Client Requirement.....3
  - 3.2 Solution.....3
  - 3.3 Scope.....4
  - 3.4 Data.....4
  - 3.5 System Registries.....5
- 4 FUNCTIONAL DESCRIPTION.....6
  - 4.1 Message Processing.....6
  - 4.2 Trip Forms EFX Tab.....8
- 5 REFERENCES.....9
- 6 DOCUMENT HISTORY.....10
- 7 AUTHORISED BY.....11

1 271604



## 2 271604 - PA-7XVL6X/ EFX Download Additions

Copyright OBS Logistics © 2010

The information contained herein is the property of OBS Logistics and is supplied without liability for errors or omissions. No part may be reproduced or used except as authorised by contract or other written permission. The copyright and foregoing restriction on reproduction and use extend to all media in which the information may be embodied



## 3 FUNCTIONAL OVERVIEW

### 3.1 Client Requirement

Create a new interface from EFX into MTS to update previously posted trips (EFX reference as the key).

The trigger in EFX would be a Site 'Accepting' a load, which would generate a message to be sent to the posting site if MTS enabled.

The Site EFX ID would be applied to the existing Decode table in MTS to identify the accepting site.

This message would amend the Subcontractor in MTS from EFX to the nominated subbie site and amend the payment from EFX to the accepting subbie site.

The status of the trip would finally be changed to 'ACCEPTED'.

### 3.2 Solution

This solution will handle the following messages from EFX

CONFREQ (EFX request has been accepted by EFX) REJREQ (EFX request has been rejected by EFX) ACCEPTED (TRIP has been accepted by someone else) EXPIRED (TRIP has expired on EFX) REJECTED (TRIP is rejected on EFX) CONFCANC (Cancellation has been confirmed by EFX) REJCANC (Cancellation has been rejected by EFX) COMPLETED (Additional Charges entered in EFX)

The READ\_EFX\_TRIP\_INBOUND procedure in INT\_MSG will be amended to handle all of the messages which can be received from EFX.

The procedure will read the records in the files received from EFX, extracting the following information:

EFX\_STATUS TRIP\_ID ACCEPTING\_DEPOT EFX\_REF EFX\_MESSAGE EFX\_ADD\_CHARGE\_1 EFX\_ADD\_DESC\_1  
EFX\_ADD\_AMOUNT\_1 EFX\_ADD\_CHARGE\_2 EFX\_ADD\_DESC\_2 EFX\_ADD\_AMOUNT\_2 EFX\_ADD\_CHARGE\_3  
EFX\_ADD\_DESC\_3 EFX\_ADD\_AMOUNT\_3

For each record that is successfully extracted from the file a record will be inserted into the table INT\_EFX\_TRIP\_INBOUND with a status of Success. If a record is not successfully extracted, a record will be inserted into the same table with a status FAILURE. Depending on the status that is received with the record, different actions will be carried out including updating the trip and creating messages. The table on the next page documents the actions carried out for each status.

As part of the validation, before any action is taken we will check the following information

The trip exists on the system The trip is at the correct status for the information we have received from EFX The file has not been loaded before.



EFX STATUS	ACTION
CONFREQ	Validation Set SCH_TRIP.EFX_REF
REJREQ	Validation Set SCH_TRIP.EFX_MESSAGE
ACCEPTED	Validation SCH_TRIP.STATUS = 'ACCEPTED' SCH_TRIP.CARRIER_ID = Accepting Carrier/Depot (Reverse Decoded) Blank SCH_TRIP.EFX_MESSAGE Validate Trip ACC_PAYMENT.CREDIT_ACCOUNT = CARRIER_ID (where event_ref like TRIP_ID and credit_account = 'EFX') Add ACC_PAYMENTS (for new Payment Types EFX1, EFX2 and EFX3 if provided)
EXPIRED	Validation SCH_TRIP.CARRIER_ID := NULL SCH_TRIP.STATUS := PLANNED Blank SCH_TRIP.EFX_REF Set SCH_TRIP.EFX_MESSAGE (or 'Expired on EFX' if not provided) Validate Trip DELETE FROM ACC_PAYMENT where ( where event_ref like TRIP_ID and credit_account = 'EFX')
REJECTED	Validation SCH_TRIP.CARRIER_ID := NULL DELETE FROM ACC_PAYMENT where ( where event_ref like TRIP_ID and credit_account = 'EFX') SCH_TRIP.STATUS := PLANNED Blank SCH_TRIP.EFX_REF Set SCH_TRIP.EFX_MESSAGE (or 'Rejected on EFX' if not provided)
CONFCANC	Validation SCH_TRIP.CARRIER_ID := NULL DELETE FROM ACC_PAYMENT where ( where event_ref like TRIP_ID and credit_account = 'EFX') SCH_TRIP.STATUS := PLANNED Validate Trip
REJCANC	Validation Tick EFX_FLAG
COMPLETED	Validation Add ACC_PAYMENTS (for new Payment Types EFX1, EFX2 and EFX3 if provided)

### EFX Changes Required

EFX will be required to send additional status notification when :-

The trip has been successfully posted on EFX, The trip has not been successfully posted on EFX The trip time has EXPIRED A cancellation has been successful on EFX A cancellation has not been successful on EFX Extra payments added in EFX (completed status)

It is assumed that the failure messages will be passed back to MTS where appropriate so the user has visibility of issues..

## 3.3 Scope

This change will be applied to system version 10.5

## 3.4 Data

Add new field EFX\_MESSAGE varchar2(1000) to SCH\_TRIP.

Add the following new fields to INT\_EFX\_TRIP\_INBOUND :-

EFX\_MESSAGE VARCHAR2(1000) EFX\_ADD\_CHARGE\_1 VARCHAR2(20) EFX\_ADD\_DESC\_1 VARCHAR2(100)  
EFX\_ADD\_AMOUNT\_1 NUMBER(20,2) EFX\_ADD\_CHARGE\_2 VARCHAR2(20) EFX\_ADD\_DESC\_2 VARCHAR2(100)  
EFX\_ADD\_AMOUNT\_2 NUMBER(20,2) EFX\_ADD\_CHARGE\_3 VARCHAR2(20) EFX\_ADD\_DESC\_3 VARCHAR2(100)  
EFX\_ADD\_AMOUNT\_3 NUMBER(20,2)



NB) These additional fields will also be added to the end of the existing interface layout (see appendix A for current layout - Is Owning Depot actually Accepting Depot ? If not then also add.).

### 3.5 System Registries

Existing system registries required are :-

'INT\_READ\_EFX\_COMMENTED\_OUT' set to ?Y? so only updates EFX Ref

'INT\_READ\_EFX\_CURSOR' set to ?Trip? so uses Load Ref as Trip\_ID not Bill\_of\_Lading.



## 4 FUNCTIONAL DESCRIPTION

### 4.1 Message Processing

A process already exists (INT\_MSG.READ\_EFX\_TRIP\_INBOUND) which is called by a database job running the procedure INT\_MSG.FILE\_HANDLING at regular intervals.

This process (for consumer the system registry 'INT\_READ\_EFX\_COMMENTED\_OUT' is currently set to ?Y? and 'INT\_READ\_EFX\_CURSOR' is set to ?TRIP?) so currently only updates the EFX\_REF on the associated trip and orders using the provided load reference as the Trip ID.

The process currently only caters for a message type of ?ACCEPTED? or ?REJECTED? but this will be extended to also cover all of the new possibilities which ESI will now have to provide :-

Message Type	Description
CONFREQ	EFX request has been accepted by EFX
REJREQ	EFX request has been rejected by EFX
ACCEPTED	TRIP has been accepted by someone else
EXPIRED	TRIP has expired on EFX
REJECTED	TRIP is rejected on EFX
CONFCANC	Cancellation has been confirmed by EFX
REJCANC	Cancellation has been rejected by EFX
COMPLETED	Further charges have been added in EFX and is now complete

Each of these message types will have the same validation as now and the additional validation on the carrier, current trip status (i.e. TENDERED) in MTS etc. but each process will be handled slightly differently and will do the following :-





Message Type	Processing
CONFREQ	Set EFX_REF on related SCH_TRIP (ST) record to value provided
REJREQ	Set EFX_MESSAGE on ST to value provided
ACCEPTED	Update Carrier_ID on ST to provided valid carrier (decoded) Set trip status to 'ACCEPTED' Validate Trip Update related ACC_PAYMENT setting CREDIT_ACCOUNT from 'EFX' to decoded carrier provided (event_ref is TRIP_ID) Add ACC_PAYMENT records for new (EFX1,EFX2 and EFX3) payment types (if provided)
EXPIRED	Update Carrier_ID on ST to Null Set trip status to 'PLANNED' Blank EFX_REF on ST Set EFX_MESSAGE on ST to value provided else 'Expired on EFX' Validate Trip Delete ACC_PAYMENT for Trip where CREDIT_ACCOUNT = 'EFX'
REJECTED	Update Carrier_ID on ST to Null Set trip status to 'PLANNED' Blank EFX_REF on ST Set EFX_MESSAGE on ST to value provided else 'Rejected on EFX' Validate Trip Delete ACC_PAYMENT for Trip where CREDIT_ACCOUNT = 'EFX'
CONFCANC	Update Carrier_ID on SCH_TRIP to Null Set trip status to 'PLANNED' Delete ACC_PAYMENT for Trip where CREDIT_ACCOUNT = 'EFX'
REJCANC	Set EFX_Send_Flag on ST back to 'Y'.Set EFX_MESSAGE on ST to value provided else 'Cancellation Request Rejected by EFX'
COMPLETED	Add ACC_PAYMENT records for new(EFX1,EFX2 and EFX3) payment types (if provided)

If any of these updates fail then the INT\_EFX\_TRIP\_INBOUND record will be updated with a RECORD\_STATUS of ?FAILURE? and an appropriate value written to the VALIDATION\_ERROR field.

NB) To data will be written to MSG\_EVENT table as now with the EVENT\_TYPE of ?MSG\_INT\_SUB? as they are not required.

The payments generated for the additional charge entered in EFX will have the following fields set from the values passed in:-

ID will go into Narrative 1 Desc will go into Narrative 2 Amount will go into Amount

Other hard coded values will be :-

Type will be ?EFX1?, ?EFX2? or ?EFX3? Status will be ?A? (for Actual) Credit Account will be Trips Cost\_Centre Debit Account will be the decoded Carrier ID Event Ref will be the Trip ID

Further possible enhancements that need additional discussions include keeping an EFX Status field against the trip so that the user can see if any actions are required by them (i.e. chase up potential carriers to encourage them to accept the trip on EFX).

Also to colour the EFX\_Ref depending on its current status to give the user a visual interpretation of the status of the trip with regards to EFX (see appendix B for DHL's suggested colours).



## 4.2 Trip Forms EFX Tab

A new Tab will be added to the Trip forms which will display all EFX related data such as status (including colouring), any message text passed back from EFX, last message sent/received types and dates.

This tab will also show information entered when posting EFX jobs and will allow Cancellations to be requested.



## 5 REFERENCES

Not Available



## 6 DOCUMENT HISTORY

Version	Date	Status	Reason	Initials
0.1	02/03/10	Draft	Initial version	DRM
0.2	18/03/10	Draft	Changes following an internal meeting	DRM
1.0	19/03/10	Issued	Reviewed and Issued	MJC



## 7 AUTHORISED BY

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

