

ECR – Terminal Integration

Market Pay

V1-9

Author of the document

Name	Function	Creation Date
Joël Coutayar	Market Pay Product Dept	07-03-2024

Review

Name	Function	Date

Validation

Nam	Function	Date

Following

Version	Update	Author	Date
01	Creation	Joël Coutayar	07-03-2024
1-1	Issuer Options specification	Joël Coutayar	19-03-2024
1-2	Time-out information	Joël Coutayar	05-04-2024
1-3	Merchant-Option clarification	Joël Coutayar	24-04-2024
1-4	Pin Change function updated	Joël Coutayar	26-04-2024
1-5	Partial Status Response	Joël Coutayar	15-05-2024
1-6	Partial Status response updated	Joël Coutayar	16-05-2024
1-7	Contado & Credito option in Refund and Cash Advance	Joël Coutayar	24-06-2024
1-7-1	Contado & Credito option in Refund and Cash Ad. detail	Joël Coutayar	02-07-2024
1-8	DNS Menu configuration + BankID	Joël Coutayar	04-07-2024
1-9	To adapt the document by removing the customer specifics	Joël Coutayar	30-07-2024

1. Introduction.....	5
2. Documentation	5
3. Global architecture and definitions	5
4. Definitions and responsibilities.....	6
5. Integration mode.....	6
USB connection	6
USB – IP Terminal connection.....	10
Ethernet connection.....	10
6. Network configuration.....	11
7. Library integration.....	11
8. Message to implement	11
9. Messages description.....	12
9-1 Message Open	12
9-2 Process transaction	14
9-3 CallBack	17
9-4 TestConnection	17
9-5 Close ()	18
9-6 UptadeTerminal ()	19
9-7 getLastTransaction()	19
9-8 Reversal	20
9-9 Terminal Activation	21
9-10 Capture Card	22
9-11 Abort Processing	22
9-12 Pin Change	23
9-13 Payment by Cheque	24
10. Integration test.....	25
11. Fields description	25
12. Parameters management	27
13. Main Functionalities description.....	27
Transaction Id	27
Update Terminal	27
Terminal Reboot Timer.....	27
Time Out management.....	28
Specific characters	30
Difference between Reversal (Cancelation) and Refund.....	30

Logs	30
14. Annexes	31
15.1 Error status	31
15.2 Basics tests required	31
15.3 Recurring questions	33

1. Introduction

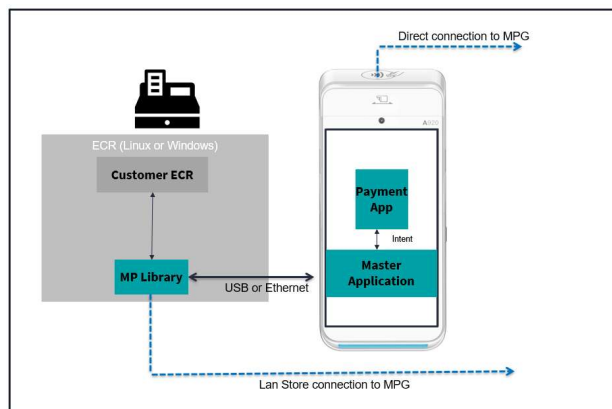
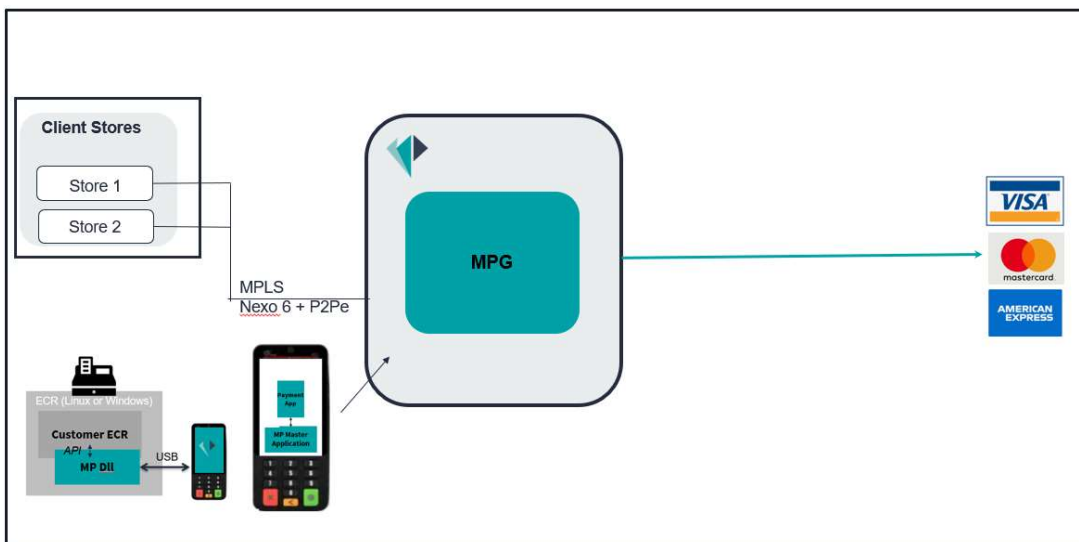
This documentation is a guide that allows ECR integrators to know the different ECR integration modes with the Terminal, the messages and logic necessary for integrated the ECR with Market Pay terminal payment.

This document includes basic tests requested by Market Pay.

2. Documentation

Documents	
ECR-Terminal Integration	Global guide document
https://knowledge-marketpay.stonly.com/kb/en	Technical documentation. This document contains development code examples
Xxx	Basic tests integration

3. Global architecture and definitions



4. Definitions and responsibilities

Term	Definition
MPG	Market Pay Gateway – Central Authorization server
PinPad	Terminal payment device
nTMS	Market Pay Terminal Management
ECR	Cash register
POI	Customer point of sale
Terminal	Terminal payment device
USB	Universal Serial Bus connection

Responsibilities

Market Pay

- Providing the support integration
- Providing the Library to implement in the ECR merchant application
- Managing the Central authorization server connecting to banks and processors
- Managing the nTMS server
- Developing the terminal Application

Merchant

- Integrating the Market-Pay library in the ECR application
- Managing the network connecting to MPG (central server) and TMS
- End-to-end testing

5. Integration mode

Market Pay offers three integration modes

- **USB**
The terminal uses the ECR connection to connect to MPG (Authorization Central Server)
- **USB – Terminal IP**
The terminal is connected in USB with the ECR (where it receives payment request) and it have a direct IP(Internet) connection to MPG for payment request authorization.
- **Ethernet connection**
The Terminal is connected with the ECR by IP (internet) and the Terminal has a direct IP connection to MPG
- **Host**
To completed

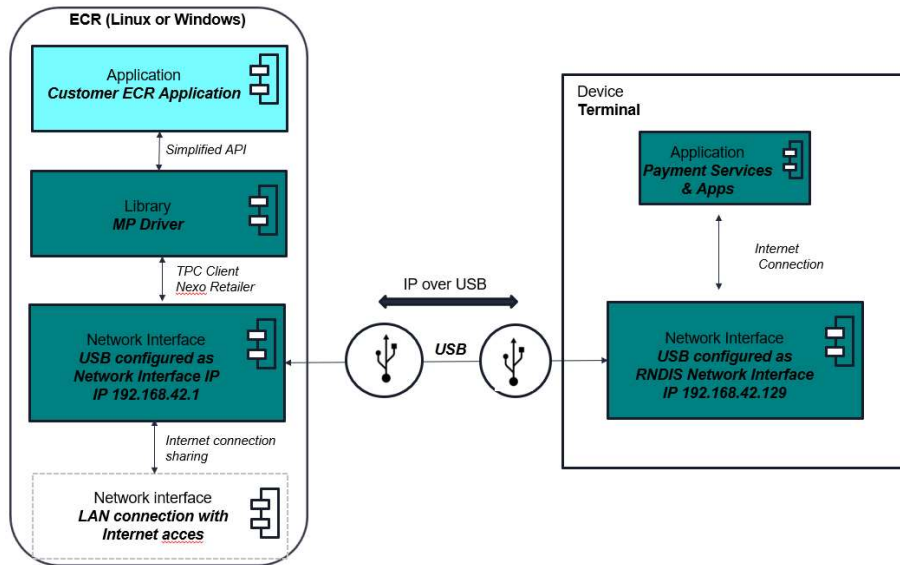
USB connection

Description

The terminal doesn't have a direct connection to MPG server; it connects using the USB ECR connection connected to an Ethernet Network.

The terminal and Library communicate using TCP-IP over USB

Architecture



OS version disponible:

The MP Library is available for different OS version (Linux and Windows)

Deliverable

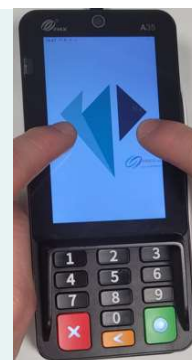
Market Pay provides the ECR integrator the Library compatible with the Merchant ECR OS.

Terminal configuration

For working in USB connection mode, the terminal needs a specific configuration in the terminal configuration described below:

- 1- Inter in the Terminal Menu (password: pax9876@@@)

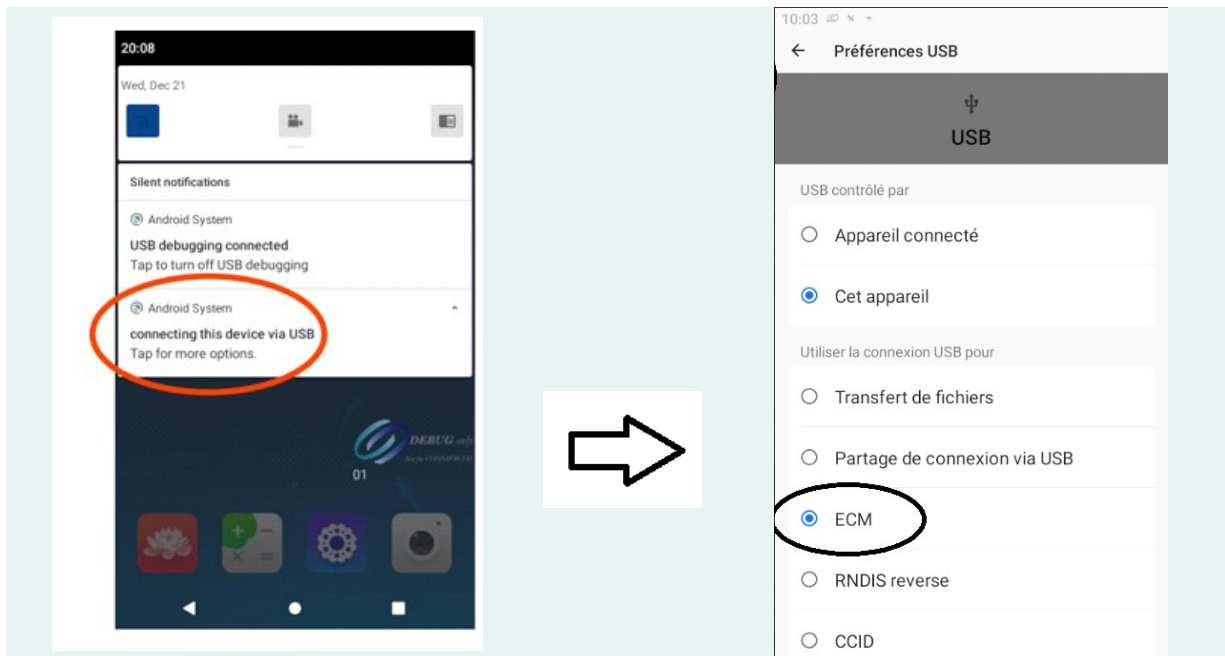
To enter the Menu it is necessary to press the screen with two fingers as shown in the picture.



The terminal presents a keyboard where the technician can enter the password



- 2- Connect the terminal to the ECR USB port.
- 3- On the payment terminal drop-down menu, tap the USB connection options and next select:
 - Windows: RNDIS reverse option
 - Linux: ECM



4- ECR configuration

On the Windows computer:

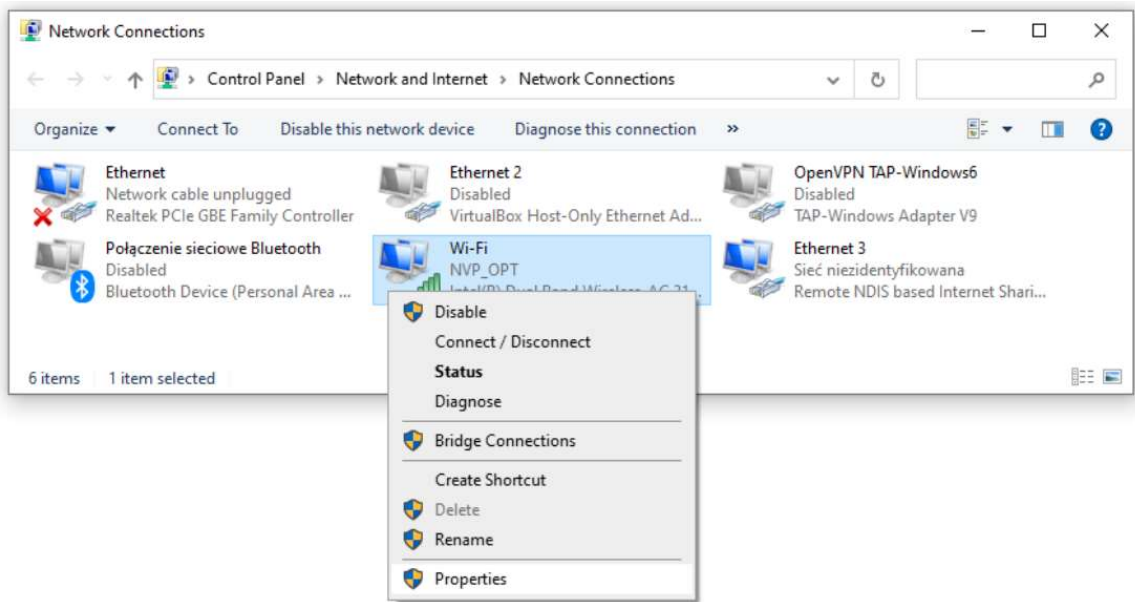
open Network Connections window:

- Click the Start menu and type View network connections
- Alternatively, open Control Panel → Network and Internet → Network and Sharing Center and click Change adapter settings

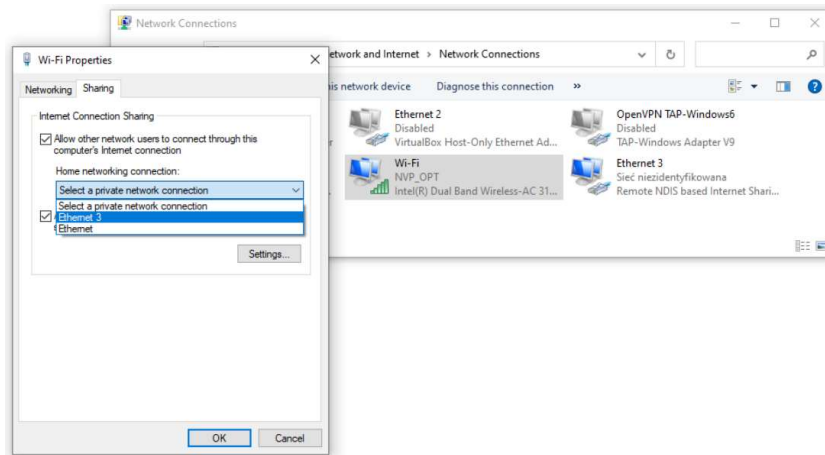
A new network connection should be available; in this case, Ethernet 3 is the new one:



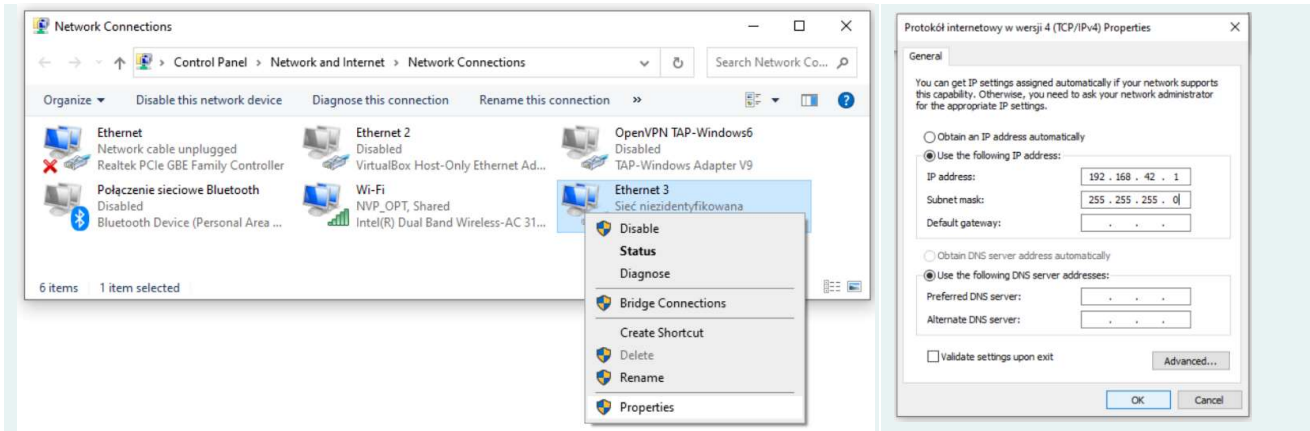
Right click the connection with internet access and share it to the payment terminal connection. In this case, NVP_OPT is the connection with the internet access:



Select the payment terminal RNDIS connection



Configure the payment terminal connection. Static IP 192.168.42.1, netmask 255.255.255.0



The payment terminal should now have internet access.

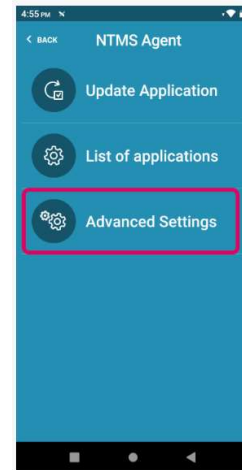
DNS menu configuration on the terminal

In case that is necessary to configure the nTMS DNS in the Terminal, this one disposes a specific menu where the maintainer or technical can configures it manually.

How to configure DNS IP addresses

To configure DNS IP addresses, you have to:

- 1- Enter NTMS Agent application
- 2- Click on "Advanced Settings" & Password is 4315
- 3- Click on "ETH over USB Settings"
- 4- Update with your information



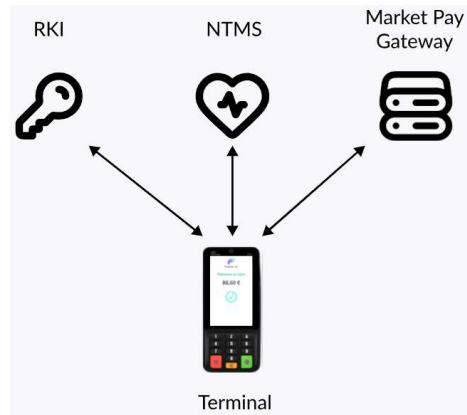
USB – IP Terminal connection

to be defined

Ethernet connection

to be defined

6. Network configuration



The Merchant needs to set up their firewall to whitelist outgoing HTTPS traffic from the IP addresses associated with the ECR applications and terminals, directing to:

nTMS: Nexo Terminal Management System
Managing all terminal settings and application updates
 Test: tntms.novelpay.pl
 Prod: ntms.novelpay.pl
 ports: 443, 8453, 4747 and 14747

MPG: Market Pay Gateway (central authorization server)
Processing and routing all transactions to processors, acquirers and banks
 Test: preprod.mpg.market-pay.com
 Prod: prod.mpg.market-pay.com
 ports: 50456 & 50460

RKI: Remote Key Injection
Enable remote key injections to encrypt sensitive information
rki.paxitalia.com

7. Library integration

MP library files necessary to implement on ECR side are:

- The marketpay_pos_api.dll (API definitions)
- Marketpay-pos-driver.dll

both have to be included into project as DLL library reference in VisualStudio.
 And marketpay-driver-cpp.dll

8. Message to implement

The MP Library includes the following functions to implement in the ECR:

Functions		
Open	Open function establishes the connection between the ECR and the Terminal	M
Process transaction	This function generates a Purchase, refund, cashadvance)	M
CallBack	Call back function: inform the ECR events on the terminal.	M

Close	Closes the payment terminal session. After this call the ECR cannot call other operations. In this case it is necessary to generate another Open message	M
TestConnection()	Checks whether the terminal connection is correct or not. Informs about the status of terminal, including the number of offline transactions.	O
updateTerminal()	Connection to TMS for getting parameters and update application	M
getLastTransaction()	Tries to find last transaction result. Useful when last transaction failed due to connection error.	M
Reversal (cancellation)	A payment reversal refers to a situation where funds from a transaction are returned automatically to the cardholder's bank account	M
Terminal Activation	Allows the ECR to activate a terminal at the time of installation.	M
Capture Card	Allows to recuperated the PAN	O
Abort Processing	Allows the ECR to cancel the transaction in progress	O
Pin Change	Allows the User to change the Pin	O

Intermediate status message:

During the terminal process, this sends information to the ECR in order to inform which process the terminal is in.

The messages are:

	Device to ECR
Waiting for card in the terminal	WaitingForCard (= "insert card")
User Pin required	PinRequired (= "Pin entry in progress")
Remove card	Remove card
Option choice customer	To complete
The terminal is processing an authorization with the bank	BankAuthorization (= "Authorisation in progress")

9. Messages description

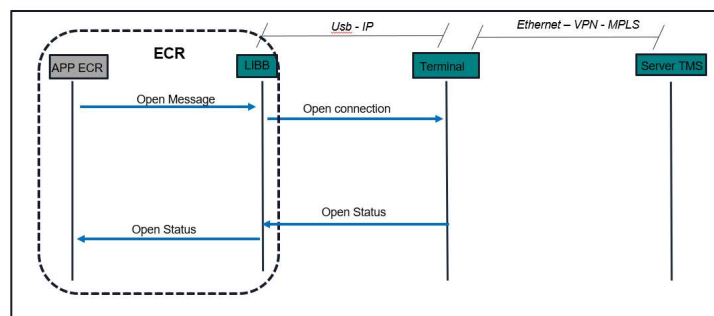
9-1 Message Open

Main Function

Open function is in charge to establish the connection between the ECR and the Terminal.

Library Command: EcrRdv_Open

Functional diagram



Message usage recommendations:

The Open message needs to be implemented

- At the beginning of each day (before first payment)
- after each connection loss
- After each cashier change

The Open Message returns a status of the connection:

EcrDrv_error-None: 0 => OK

EcrDrv_error-None: not 0 => error

The Error status are described in Annex 1 of the document

In case that the function returns error, is necessary to close before to send to request Open again.

Fields required in the function:

The Open message needs that the ECR completes the following structures:

EcrConfiguration()

PoiConfiguration()

Function	Structure	Field		Description
Open()	ecrConfiguration		M	
		EclId	M	
		CashierId	O	
		ProviderId	M	
		appName	M	Example: SMARTPOS
		Version	M	
		ComponentDescription	O	
		OperatorLanguage	M	
		PrinterAvailable	M	
	poiConfiguration		M	
		Address	M	
		maxMessageSize	M	By default: 32768
		TC1	M	By default: 1000
		TC2	M	By default: 2000

Open Example integration

```

ecrConfiguration.ecrId = "11234001"
ecrConfiguration.cashierId = "456"
ecrConfiguration.providerId = "MarketPay"
ecrConfiguration.appName = "ECR ESP Client XXX"
ecrConfiguration.version = "0.0.5"
ecrConfiguration.componentDescription = ""
ecrConfiguration.operatorLanguage = "ES"
ecrConfiguration.printerAvailable = True
poiConfiguration.address = "192.168.42.129:7778"
poiConfiguration.maxMessageSize = 32768
poiConfiguration.tc1 = 1000
poiConfiguration.tc2 = 2000
    
```

9-2 Process transaction

Main Function

The process transaction allows the ECR system to generate different authorization requests to bank:

- Purchase
- Refund
- Pre-Auth

Function: EcrDrv_processTransaction()

Two transaction Modes are available:

- **Direct Mode**

When the ECR process in direct Mode, after getting all cardholder information, the terminal goes directly to authorisation with MPG. The terminal doesn't wait for any confirmation from the ECR.

The ECR receives intermediate Status messages, and next directly the response and ticket from the terminal.

- **Dual Mode**

In Dual Mode, 2 steps are required to complete the payment.

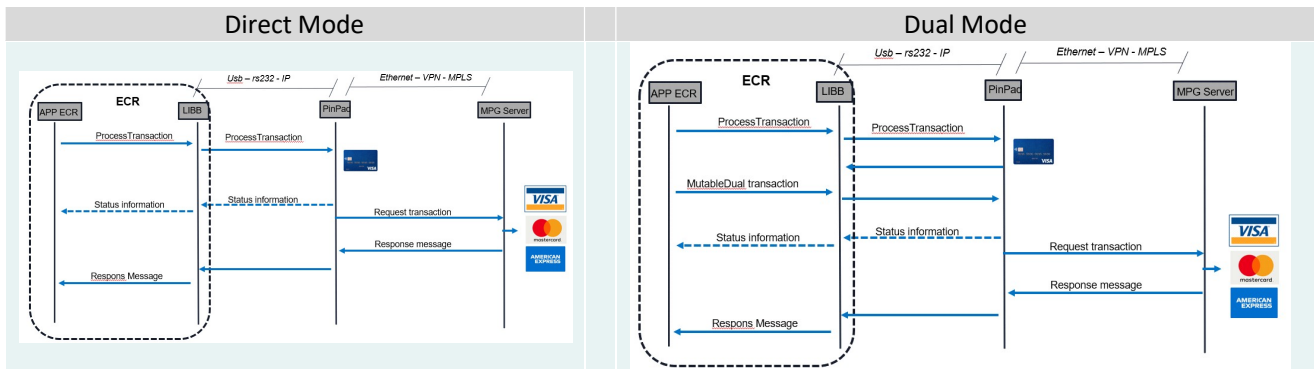
The ECR sends the first request message to the terminal, and when the terminal has captured the cardholders card, it will return to the ECR for validation, providing the PAN (Bin + 4 last digits).

If the ECR accepts to continue, it will send a confirmation to the terminal. When the terminal receives the confirmation it will go to MPG for authorization.

Before the Confirmation message, the ECR can change the amount.

Library Command: EcrDrv_processTransaction

Functional diagram



Fields required in Direct Mode

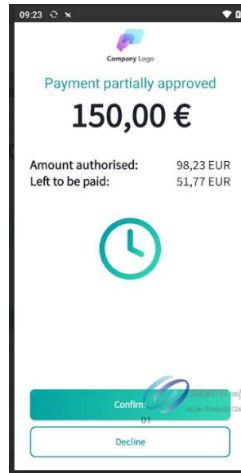
Message Request	Structure	Field	Description
	TransactionParams		M
		TransactionType	M Options: Purchase Refund CashAdvance Pre-authorization
		TransactionMode	M EcrDrv_TransactionMode_Direct o EcrDrv_TransactionMode_Dual
		Amount	M

processTransaction	amountTip	O	
	amountCashback	O	
	transactionId	M	One value per transaction and day
	currency	M	978 (Euro)
	CashierId	O	
	MerchantOption	O	
	TransactionResult	M	
	Status	M	EcrDrv_Status_OK EcrDrv_Status_NOK EcrDrv_Status_Partial (*)
	ResponseCode	M	
	AuthorizationCode	O	
	CashierReceipt	M	
	CustomerReceipt	M	
	SignatureRequired	M	
	ForcedOffline	M	
	DCCDetails	O	
	PrivateData	O	
	IssuerOption	O	
	ParBank	O	
MerchandId	M		
Terminall	M		
CardData:			
- Cardcapture			
- ExtractedPan			
- LoyaltyId			
- Schema			
- ApplicationID			
- PSN			
- BankID			
finalTransactionParams	O	Amount , amountTip, amountCashback	

(*) Partial transaction accepted

the Library send a response Status_Partial to the ECR when the transaction is accepted but not for all the amount. In this situation, the ECR is in charge to check with the issuer to complete the remaining payment. This one can be do in cash or another card.

When the terminal communicates a Partial status, it displays the missing amount to be paid:



And the ECR can know the total amount accepted. This information is present in the structure `FinalTransactionParams` – Amount.

This structure, `EcrDrv_finalTransactionParams`, has the same data as in the initial request transaction informed in the `TransactionParams`, expect the field Amount.

Example Purchase:

```
transactionParams.type = EcrDrv_TransactionType_Purchase;
transactionParams.mode = EcrDrv_TransactionMode_Direct;
transactionParams.amount = 1234; //amount without point
transactionParams.amountTip = 0;
transactionParams.amountCashback = 0;
transactionParams.transactionId = ("12343432");
transactionParams.currency = 978; //euro
transactionParams.cashierId = (87654);
transactionParams.merchantOption = " ";
```

Fields required in Dual Message Mode

The advantage of the double message is that it allows the Merchant to change, next `TransactionResult`, :

- Amount
- Merchant Option

The fields are the same as in Direct Mode the only difference is concerning the Transaction Mode. The value to complete in the ECR is: `EcrDrv_TransactionMode_Dual`

Once the result has been received, using `transactionResult`, the ECR applies the function:

`EcrDrv_MutableDualTransactionParam`, whether or not the customer has changed the amount field or the Merchant Options field.

DualMessage processing:

When *driver's* dual transaction is triggered by the ECR (by the user of the driver) following steps are performed, (below only steps related to the `EcrDrv_MutableDualTransactionParams` data structure):

- *driver* internally allocates the `EcrDrv_MutableDualTransactionParams` structure (technically: inside the `EcrDrv_processTransaction()` function implementation stack).
- *driver* internally calls `EcrDrv_MutableDualTransactionParams_init()` function to init the `EcrDrv_MutableDualTransactionParams` - so the *client* never calls this function directly.

- *driver* calls the **transactionHandler->onDualProcessing()** function provided by the *ECR* with The **EcrDrv_MutableDualTransactionParams** already initialized by the *driver* in previous step. The *ECR* cannot call **EcrDrv_MutableDualTransactionParams_init()** or **..._fini()** on this **params** structure. The *ECR* just only updates the amount if needed - of already allocated structure by the *driver* - so it is a memory managed by the *driver*.
- *driver* reads the (same or modified) amount - *driver* checks if the *client* has modified the amount during the **transactionHandler->onDualProcessing()** callback.
- *driver* internally updates the *PaymentRequest's* amount and internally call **EcrDrv_MutableDualTransactionParams_fini()**.

9-3 Callback

This message is used to notify the *ECR* about the current transaction status.

During the transaction process, the terminal reports on the status of the transaction to the Merchant Application.

The terminal can notify the following information:

- Waiting for card on the terminal
- User PIN required
- The terminal is processing an Authorization with the bank

The functions

- EcrDrv_TransactionStatus_WaitingForCard
- EcrDrv_TransactionStatus_PinRequired
- EcrDrv_TransactionStatus_BankAuthorization
- EcrDrv_TransactionStatus_Undefined

9-4 TestConnection

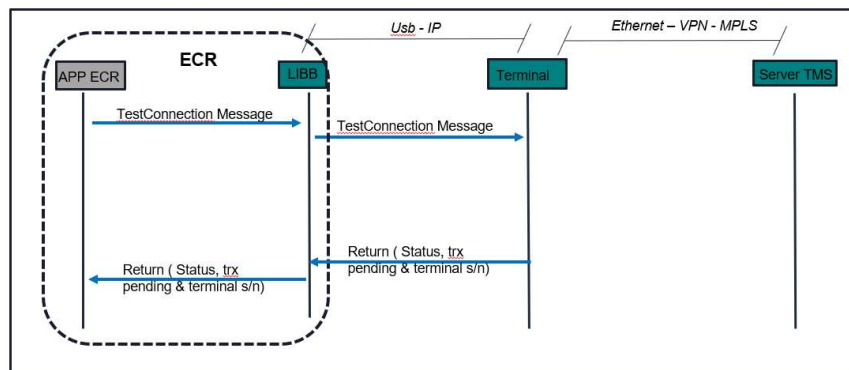
Main Function

The main function of this function is:

- To test the connection between the *ECR* and the Terminal
- To get the Terminal Serial Number
- To get the total pending transactions (offLine)

Library function: EcrDrv_testConnection()

Functional diagram



Message usage recommendations:

This function is recommended to use in the following scenarios:

- The ECR need to know if the Terminal is present or before each transaction request.
- To get the Terminal Serial Number for any control required.
- To know how many offline transactions are pending in the terminal in case that the merchant supports the offline functionality

The Test connection Message returns a status of the connection:

EcrDrv_error-None: 0 => OK

EcrDrv_error-None: not 0 => error

The Error status are described in Annex 1 of the document

This function doesn't inform if the Terminal is correctly connected with central server, nTMS.

Fields required in the function:

Not specific field is required.

Function	Structure	Field		Description
TestConnection	Testconnection()		M	

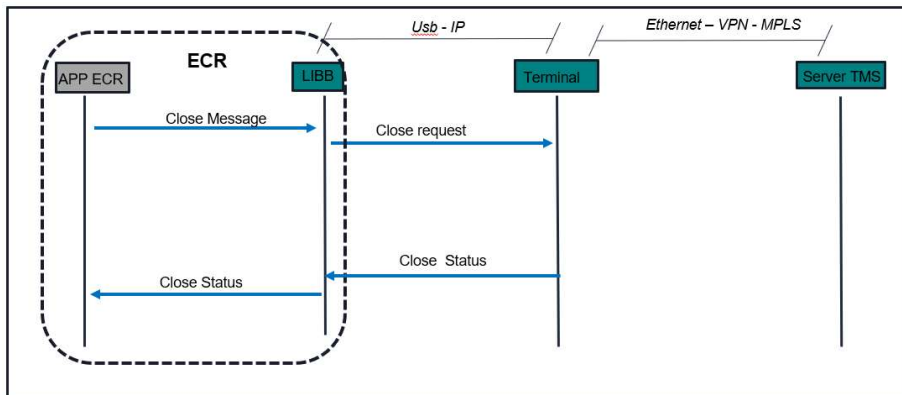
9-5 Close ()

Main Function

This function closes the connection between the ECR and the Terminal. After Close, the ECR cannot do another transaction.

Library function: EcrDrv_close()

Functional diagram



Message usage recommendations:

It is recommended to send a Close request:

- After a loss of connection between ECR and Terminal
- Before all Open requirement
- At the end off day

The Close Message returns a status of the connection:

EcrDrv_error-None: 0 => OK
 EcrDrv_error-None: not 0 => error
 The Error status are described in Annex 1 of the document

Fields required in the function:
 No specific field is required.

Function	Structure	Field	Description
Close()	EcrDvr_close		M

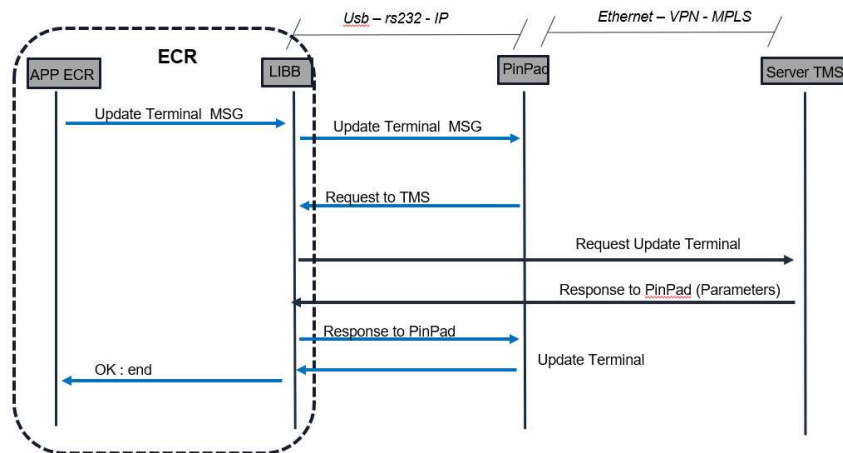
9-6 UptadeTerminal ()

Main Function

This message is intended to request an update of the terminal.
 When the terminal receives this command it automatically connects to nTMS server for getting parameters and new application if it is required by Market Pay

Library function : EcrDrv_updateTerminal()

Functional diagram



The Message response returns a status of the connection:
 EcrDrv_error-None: 0 => OK
 EcrDrv_error-None: not 0 => error
 The Error status are described in Annex 1 of the document

9-7 getLastTransaction()

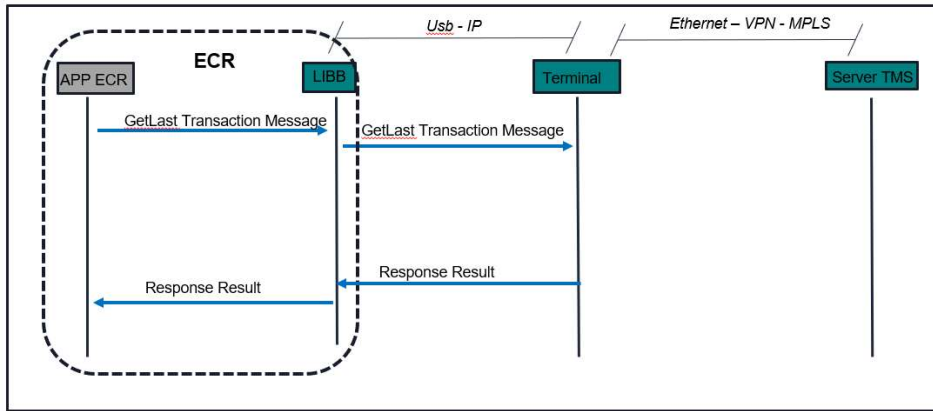
Main Function

This function allows the ECR to recover the response of the last transaction.
 This function is recommended in case the ECR has lost the connection with the terminal or any other problem that prevents knowing the result of the transaction.

The terminal stores only the last transaction result.
 The ECR receives all the information for a completed the transaction

Library function : EcrDrv_getLastTransaction()

Functional diagram



Fields send to ECR

Message Request	Structure	Field		Description	
GetLastTransaction	TransactionResult	Status	M	Ok or Not OK	
		ResponseCode	M		
		AuthorizationCode	O		
		CashierReceipt	M		
		CustomerReceipt	M		
		SignatureRequired	M		
		ForcedOffline	M		
		DCCDetails	O		
		PrivateData	O		
		IssuerOption	O		
		ParBank	O		
		MerchandId	M		
		TerminalId	M		
		CardData:			
		- Cardcapture			
- ExtractedPan					
- LoyaltyId					
- Schema					
- ApplicationId					
- PSN					

9-8 Reversal

Main Function

This function performs a Reversal.

A reversal can be requested from the ECR to reverse a transaction and return the funds back to the client.

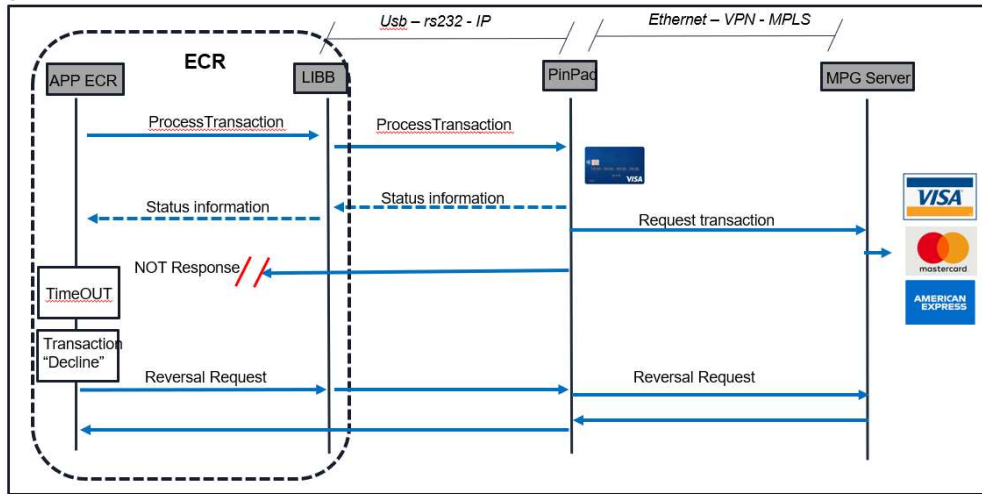
This request may come from a technical problem (time out on ECR or terminal side) if case where there is no response from MPG or the bank.

The system supports different levels of Reversal. This document focuses only on the Reversal between ECR and Terminal.

Kinds off Reversal	
From ECR	TimeOut at ECR level. Not response from the Library / Terminal
From Terminal	Not response from MPG / Bank. Transaction decline to ECR

Library function : EcrDrv_transactionCancellation(...)

Functional diagram



Fields

Function	Structure	Field		Description	
Reversal	TransactionCancelation		M		
		Address	M		
		maxMessageSize	M	By default: 32768	
		TC1	M	By default: 1000	
		TC2	M	By default: 2000	
			Amount	M	Same value that original transaction
			amountTip	O	Same value that original transaction
			amountCashback	O	Same value that original transaction
			transactionId	M	Same value that original transaction
		currency	M	978 (Euro)	

9-9 Terminal Activation

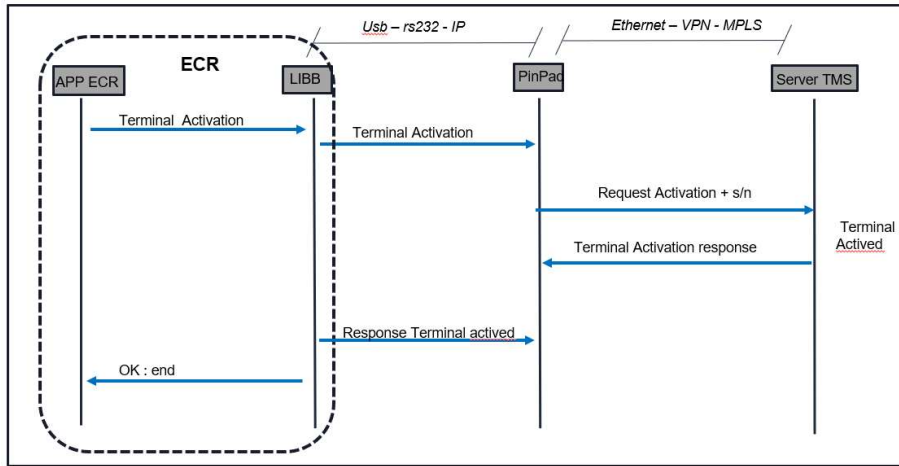
Main Function

Performs terminal activation before first use.

When installing a terminal to an ECR, it is necessary to declare to MP this terminal and activate it to be operational. The Terminal is not operative until it has been declared.

Library function : EcrDrv_activateTerminal()

Functional diagram



Fields

Function	Structure	Field	Description
Activation	EcrDrv_activateTerminal()		M
	EcrDrv_TerminalActivationResult		M
		merchantId	M
		terminalId	M
		merchantName	M
		location	M
		serialNumber	M
		errorMessage	M

All fields present in TerminalActivatinResult come from the nTMS configuration

When the terminal has been successfully activated. All fields except errorMessage should be initialized.

9-10 Capture Card

Not used at the moment

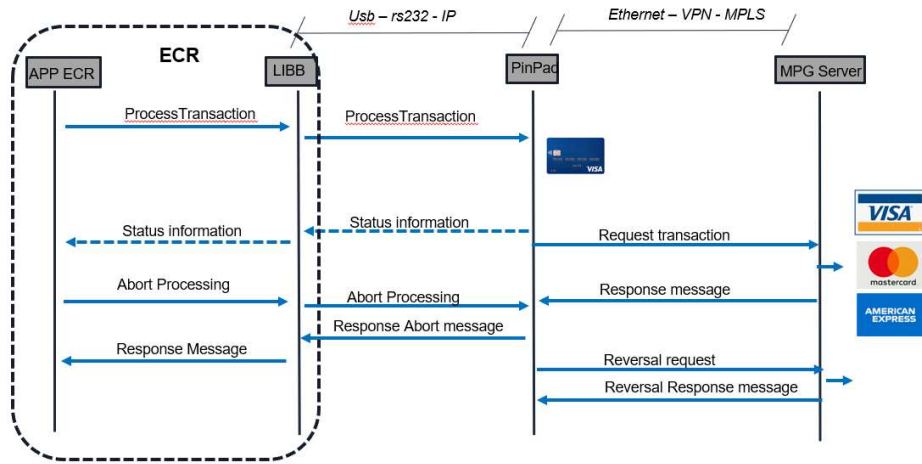
9-11 Abort Processing

Main Function

The Abort message allows the ECR to stop and terminate prematurely the processing of a transaction. Most of the time a message is aborted, because the processing is too long or the Sale System is resolving error situation.

Library function : EcrDrv_abortProcessingTransaction()

Functional diagram



The AbortProcessing Message returns a status of the connection:

- EcrDrv_error-None: 0 => OK
- EcrDrv_error-None: not 0 => error

The Error status are described in Annex 1 of the document

Fields required in the function:

Not specific field is required.

Function	Structure	Field	Description
Abort	EcrDvr_abortProcessing		M

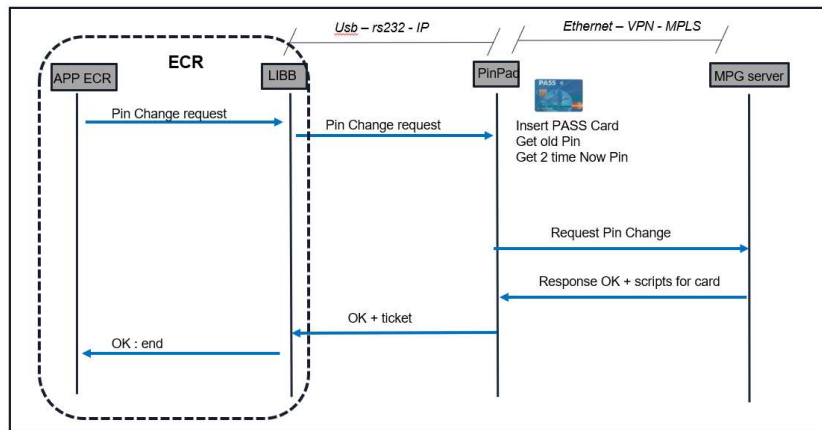
9-12 Pin Change

Main Function

Cardholder can ask to change the PIN of the card directly in the store. This function is only available for Pass card Spain. The ECR sends to the Pinpad a specific command to initiate the operation.

Library function : EcrDrv_PinChange

Functional diagram



Fields required in the function:

Not specific field is required.

Function	Structure	Field	Description
----------	-----------	-------	-------------

Pin Change	EcrDvr_PinChange		O	
	EcrDrv_PinChangeResult			"0" on success, other value on error.

9-13 Payment by Cheque

Main function

Allows the Cashier to make a payment by cheque if the ECR has a cheque reader and supports these operations.

Library function: EcrDrv_processCheckTransaction()

This function is used only in Direct Mode

Message Request	Structure	Field		Description	
EcrDrv_processCheckTransaction	TransactionParams		M		
		TransactionType	M	Options: Purchase	
		TransactionMode	M	EcrDrv_TransactionMode_Direct	
		Amount	M		
		amountTip	O		
		amountCashback	O		
		transactionId	M	One value per transaction and day	
		currency	M	978 (Euro)	
		CashierId	O		
		MerchantOption	O		
	TransactionResult		M		
		Status	M		Ok or Not OK
		ResponseCode	M		
		AuthorizationCode	O		
		CashierReceipt	M		
		CustomerReceipt	M		
		SignatureRequired	M		
		ForcedOffline	M		
		DCCDetails	O		
		PrivateData	O		
		IssuerOption	O		
		ParBank	O		
		MerchantId	M		
		TerminalId	M		
	CardData:				
	- Cardcapture				
	- ExtractedPan				
	- LoyaltyId				
	- Schema				
	- ApplicationID				
	- PSN				

10. Integration test

Market Pay asks every integrator to perform integration tests.

The basic tests are presented in Annex 2 "Basic Tests Required" and the ECR integrator will need to present to MP the detailed results, specified in the doc: *ECR integration - Checklist - autovalidation*

11. Fields description

	Field	Config		Description
A	appName		M	ECR Application identifier. example SMARTPOS
	Address	Char	M	Address and port of ECR device : example "192.168.0.1:7778"
	Amount	?	M	amount without point: 12,45€ = 1245
	amountTip	?	O	Additional amount
	amountCashback	?	O	Cashback amount
	ApplicationID	Char	M	Return the AID card used for the transaction
	AuthorizationCode	Char	O	Only present if Response code is "00" (accepted) Is a transaction response ID code that the authorizing institution assigns
B	BankID	Char	M	Card BIN issuer informed in the response (Result)
C	cashierId	Char	M	Customer Cashier identificatory defined by the Merchant
	componentDescription		O	Description of the ECR type
	Currency	Char	M	Local Merchant Currency
	CashierReceipt	Char	M	Contains the lines of text to print cashier receipt.
	CustomerReceipt	Char	M	Contains the lines of text to print customer receipt.
	CardCapture	Char	M	Describes how the card has been read. <ul style="list-style-type: none"> - EcrDrv_CardCapture_Chip - EcrDrv_CardCapture_Contactless - EcrDrv_CardCapture_MagneticStripe - EcrDrv_CardCapture_TapPhone - EcrDrv_CardCapture_Other - EcrDrv_CardCapture_Undefined
D	DccDetail	Struct	O	EcrDrv_Boolean : dccOffered EcrDrv_Boolean : dccUsed EcrDrv_Amount : dccAmount int dccCurrency
E	ECR client device identificatory	Char	O	allows the customer to identify the point of sale (POI)
	EcrLd	?	M	ECR device identificatory defined by the Merchant
	ExtractedPan	Char	M	Contains partial PAN number and 4 last digits. example 5311110000001234
F	ForcedOffline	Boolean	M	Indicates if the transaction has been accepted in offline mode.
I	IssuerOption	Char	O	Card Option selected by issuer on the terminal
L	LoyaltyId	Char	O	Loyalty card account identification

	Location	Char	M	The store location address
M	maxMessageSize			Maximum size of message in bytes
	MerchantOption	Char	O	Custom merchant options (n months) : for specific use.
	MerchantId	Char	M	Merchant identification
	MerchantName	Char	M	The merchant name defined in the nTMS Server
O	OperatorLanguage	Char	M	Customer language based on ISO 639-1 (standardized nomenclature used to classify languages)
P	ProviderId		M	Provider identifier of the ECR software. Example "Super North"
	PrinterAvailable		M	Informs if the ECR has a Printer available
	PrivateDate	Char	O	Text Buffer for private information
	PerBank	Char	O	PAR generated by the bank.
	PendingTransaction	Int		The terminal informs how many transactions are not synchronized with Acquirer (offline transaction)
	PSN	Char	O	Inform the Option selected by user
R	ResponseCode	Char		Response code obtained from acquirer during the authorization
S	Status	Char	M	EcrDrv_Status_Ok, EcrDrv_Status_Nok
	SignatureRequired	Boolean	M	"0": No "1": Yes
	SerialNumber	Char	M	Terminal serial number
	Schema	Char	M	Indicate the card Scheme VISA: Visa MASTERCARD: Mastercard MAESTRO: Maestro AMEX: Amex
T	TC 1	Int 2	M	Transport connection establishment timeout (milliseconds) After sending a transport connection request, the Requestor arms the TC1 timer to watch for the response from the Responder. TC1 is reset on the transport connection confirmation reception.
	TC 2	Int 2	M	Application message reception timeout (milliseconds) This timer arms the reception of the application message prefix to supervise the reception of the complete application message. It allows the detection of an incomplete application message reception, avoiding a deadlock of the transport connection.
	TransactionType		M	Define the kind of transaction required for the Merchant Values: - EcrDrv_TransactionType_Purchase - EcrDrv_TransactionType_Refund, - EcrDrv_TransactionType_CashAdvance, - EcrDrv_TransactionType_Preauthorization
	TransactionMode		M	Determine if the transaction is in Direct or Dual Mode. Values: - EcrDrv_TransactionMode_Direct - EcrDrv_TransactionMode_Dual
	transactionId	?	M	Only one value per transaction and day. The ECR sends the same value only in case it has not received response from the terminal. All next request for the same basket have same TransactionId.

TerminalId	Char	M	Terminal identification
V	Version		Provider identifier of the ECR software

12. Parameters management

Market Pay manages the terminal parameters.

These parameters are managed in the nTMS server and nTMS sends parameters to the terminal.

Parameters	Descriptions
Terminal Logo	
Time Out Terminal disconnection	Is the time from the moment that the terminal has generated the response to the ECR until it returns to the idle state. By default: 30 sec
Card types	The Merchant informs MP about the type of card brands accepted. (Visa, MC, Amex ..)
Acquirer parameters	For example, Terminal Capabilities, Contacless terminal floor limit, TAC's ..

Is under the Merchant / ECR to manage the following parameters

Parameters	Descriptions
Time Out	Is the time that the ECR waits a response from the Terminal (recommended: 20 seg)

13. Main Functionalities description

Transaction Id

The value is defined by the Merchant and need to be the same for all the messages related to the same transaction / basket.

For example in case that the ECR lose the connection with the Terminal and decides to generate a new payment request for the same basket , it is recommended to send the same TransactionId.

This allow Market Pay and the Merchant to identify possible double transactions.

Update Terminal

Market Pay recommends to push this request once per day, at the beginning of the day, before starting card payment.

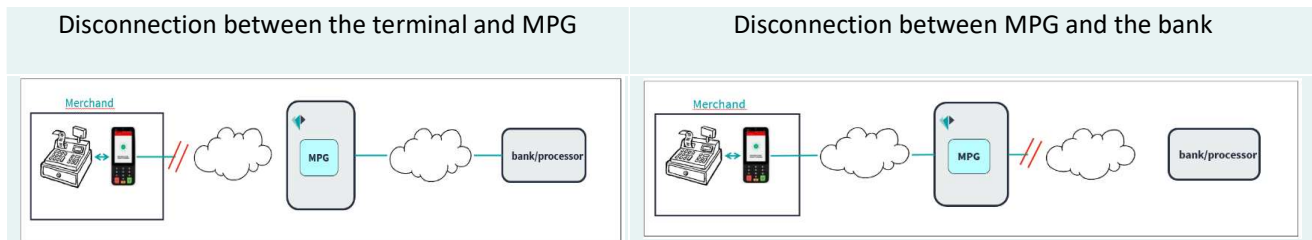
Terminal Reboot Timer

regardless of whether the ECR has performed an Update Terminal, the Terminal has an internal Reboot timer, managed by the nTMS.

Once a day, the terminal must be reset in order to be compliant with the scheme's security requirements.

Offline Most cards require systematic authorization (request to the issuing bank). In the event of a communication breakdown between the card and its bank, these are automatically refused.

In order to provide continuity of service, Market Pay proposes an “Forced Offline” called OffLine mode, which consists of forcing an authorization based on parameters in the terminal in the case that the terminal cannot reach the bank.



In each response the ECR receive the information if the transaction was accepted in Offline Mode:

TransactionResult – OfflineForced

The TerminalConnection message allows the ECR to know how many offline transactions are pending.

The recommendation for the ECR is to send a request alert to the responsible in the store for checking the terminal connection.

When the connection is established again, one minute later the terminal sends automatically all offline transactions archived to the central server by lots (Only 5 transaction per lot).

Time Out management

When the ECR system sends a request message and wait for a response to this request, the application must watch the response with a timer in order to:

- Limit the waiting time for the requestor during a real-time process,
- Take the appropriate action in absence of a response to the request,
- Not wait forever, leaving resources busy.

The best timer depends off the ProcessTransaction Mode defined by the Merchant (Direct or Dual message).

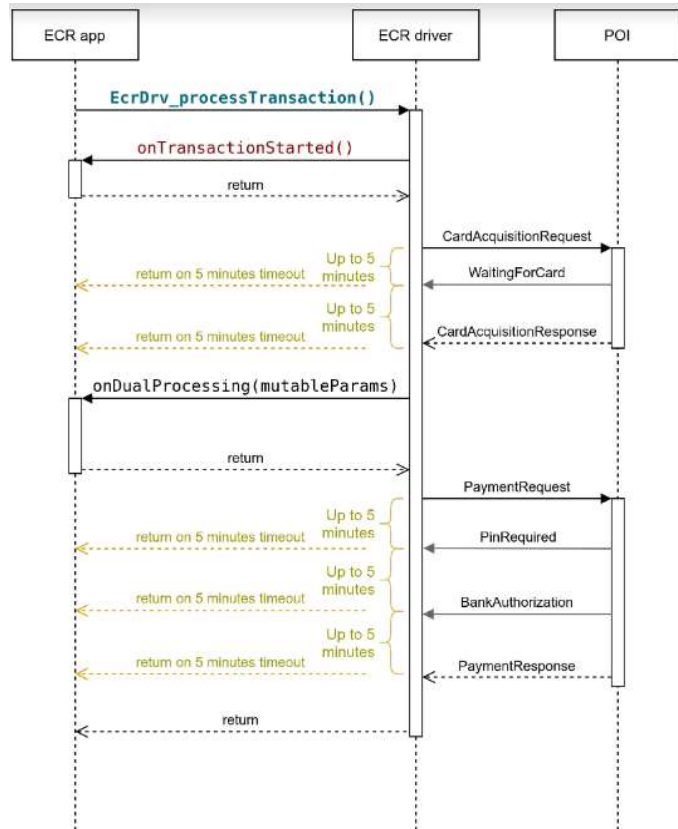
On direct request, the ECR activates the timer but must leave enough time to allow the customer to insert his card, enter the PIN, for authorisation towards the bank.

In Dual Mode, the ECR arms the timer just after having updated the MutableDualTransaction.

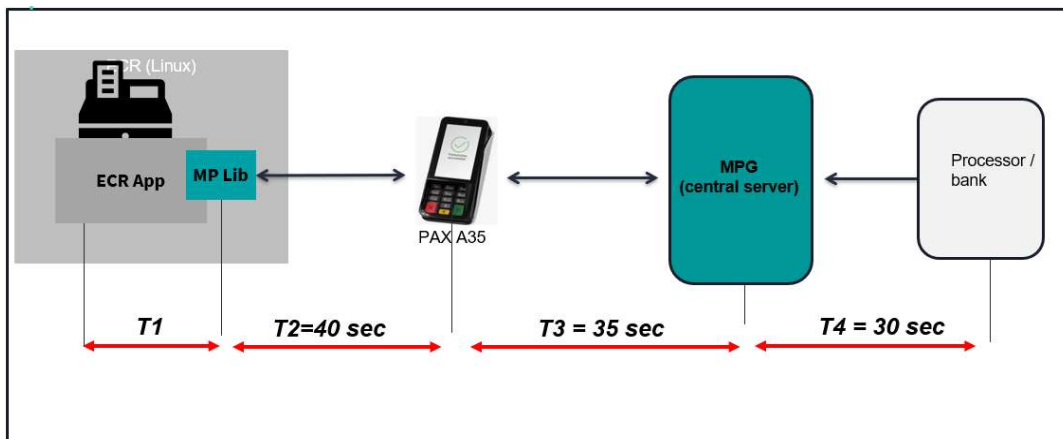
The system supports different Timeout:

- [ECR - Terminal] Heartbeat between ECR and Terminal. The timeout is configurable and usually set around few seconds.
- [ECR - Terminal] Display request-response timeout between Terminal and ECR. DisplayRequest is the message sent from Terminal to the ECR during Terminal Activation. Max activation time in seconds after which this function will return.
- [Terminal - MPG] Connection timeout. Configured by nTMS and usually set to few seconds.
- [Terminal - MPG] MPG response timeout. Configured by nTMS and usually set between several seconds.
- [Terminal - MPG] Loyalty generation timeout. Configured by nTMS and usually set between several seconds.

When Terminal can handle another request from ECR, there is no timeout. The Terminal can handle new request immediately after displaying (and sending the response to ECR) the transaction result of the last transaction.



Time-Out recommendation



T3: 2 options

T3-A: the terminal tries to connect to MPG , if not connection possible in 15 sec the terminal can switch in Offline mode

T3-B: The terminal is connected to MPG but not response: the terminal waits 35 sec and next it can switch to offline

Specific characters

On the ECR system side, it is recommended to install the UTF-8 encoding to support possible international characters.

Difference between Reversal (Cancellation) and Refund

In a refund, the merchant returns the money to the customer's account, and the transaction is considered completed. In a reversal transaction, the bank or payment processor cancels the transaction, and the funds are not transferred from the customer's account to the merchant's account.

To do a Refund it is necessary for the cardholder to present the card again. It is a new transaction.

The Reversal or cancellation doesn't need the cardholder's card again. It is generated by the ECR to the terminal when:

- The ECR have not received response during the delay defined
- The client push "Cancel" key during the issuer process on the PinPad (get Pin, application validation ..)
- The Cashier decide to cancel the transaction.
- After any problem on the ECR side during the payment process.

Logs

The Library has the capability to generate Log where it returns details information of the transaction to ECR.

A specific function is proposed and recommended by Market Pay

EcrDrv_LogLevel which describes importance of occurring logged message.

Example:

```
#include "marketpay/driver/api_c/Typedefs.h"
```

```
typedef enum
```

```
{  
    EcrDrv_LogLevel_Verbose = 0,  
    EcrDrv_LogLevel_Debug,  
    EcrDrv_LogLevel_Info,  
    EcrDrv_LogLevel_Warning,  
    EcrDrv_LogLevel_Error  
} EcrDrv_LogLevel;
```

```
typedef void (*EcrDrv_logCallbackFn)(EcrDrv_LogCtx* logCtx, const EcrDrv_LogLevel level, const char* const message);
```

```
#ifdef __cplusplus
```

```
  
}  
#endif
```

```
#endif /* MARKETPAY_DRIVER_API_C_LOG_H_ */
```

The Library generates a log with the name LGXXXXXX.pay (where xxxxxx is the year/month/day)

End of the Document

14. Annexes

15.1 Error status

`#define EcrDrv_ERROR_NONE 0` : Successful operation result, no error. More...

`#define EcrDrv_ERROR_INTERNAL 1` : Any error which is not provided with details. When this error occurred, driver has to be reopen (close and open again) More...

`#define EcrDrv_ERROR_PARAM_IS_NULL 100` : Any parameter is mandatory but nullptr passed. More...

`#define EcrDrv_ERROR_NOT_IMPLEMENTED 101`: Called functionality is not implemented. More...

`#define EcrDrv_ERROR_TRANSPORT_CONNECT_FAILURE 102` : Failed to connect transport layer. When this error occurred, driver has to be reopen (close and open again) More...

`#define EcrDrv_ERROR_TRANSPORT_DISCONNECT_FAILURE 103` : Failed to disconnect transport layer. It is possible that it was already disconnected. More...

`#define EcrDrv_ERROR_TRANSPORT_CONNECTION_BROKEN 104`
Transport layer IO operation failed due to broken connection. When this error occurred, driver has to be reopen (close and open again) More...

`#define EcrDrv_ERROR_TRANSPORT_RECEIVE_TIMEOUT 105`
Failed to receive message due to timeout. When this error occurred, driver has to be reopen (close and open again) More...

`#define EcrDrv_ERROR_TRANSPORT_RECEIVE_MALFORMED_DATA 106`
Failed to receive message because incoming data is malformed. When this error occurred, driver has to be reopen (close and open again) More...

`#define EcrDrv_ERROR_POI_IS_BUSY 107`
Payment terminal is busy and cannot perform requested action. More...

`#define EcrDrv_ERROR_DRIVER_NOT_OPEN 108`
Cannot perform operation because driver is not open. More...

`#define EcrDrv_ERROR_MESSAGE_FORMAT 109`
Failed to parse message from the payment terminal. Some XML fields may be missing. More...

`#define EcrDrv_ERROR_INVALID_CALL 110`
Driver function has been called incorrectly or at the wrong time. More...

15.2 Basics tests required

Below the Market Pay tests recommendation to check before certification & Pilot.

Test	O/ M	Test Step	Test Result
------	---------	-----------	-------------

1	Terminal Activation	M	To connect the Terminal at the ECR. Send Open Message Send Activation message	The terminal is OK for working. To check information received in the response Terminal status should be OK without any errors.
2	Start ECR application & Terminal connection	M	Start Open message	To check that the connection with the terminal is established Terminal status should be OK without any errors.
3	Terminal Update request	M	To push Update request	Terminal status should be OK without any errors.
4	Transaction after opening ECR connection	M	Perform transaction	Transaction should be approved without any errors in ECR
5	New transaction after Close-Open ECR session	M	To check the close and Open message	Transaction should be approved without any errors in ECR
6	Fast transaction after Close-Open ECR session	M		Transaction should be approved without any errors in ECR
7	2nd transaction triggered very quickly after 1st one	M	1.Perform transaction 2.Tap the card on Present card screen 3.Change amount and quickly perform next transaction (when application present final approved screen of 1st trx)	Terminal should show screen with new amount and after transaction smoothly returns to main screen
8	Abort transaction by pressing Terminal Red button		1.Establish connection with ECR and perform transaction 2.Press Red abort button on terminal	Transaction should be aborted and application should returns to main screen
9	Abort transaction by pressing key on ECR		1.Establish connection with ECR and perform transaction 2.Press abort button on ECR	Transaction should be aborted and application should returns to main screen
10	Timeout on Present card screen		Perform transaction and wait timeout on Present card screen	After timeout transaction should be aborted and finally Terminal should return to main screen
11	Timeout on PIN entry screen		Perform transaction and wait timeout on Pin entry screen	After timeout transaction should be aborted and finally Terminal should and return to main screen
12	Abort on PIN entry screen by pressing Red button		1.Establish connection with ECR and perform transaction 2.On Pin entry screen press Red abort button on terminal	After timeout transaction should be aborted and finally Terminal should and return to main screen
13	Abort on PIN entry screen by pressing key on ECR		1.Establish connection with ECR and perform transaction 2.On Pin entry screen press F4 abort button on ECR	Transaction should be aborted and application should returns to main screen
14	Changing transaction type		1.Establish connection with ECR and perform approved transaction type Purchase 2.Change transaction type to Refund and perform new transaction	After approved purchase transaction terminal should display Refund transaction screen which should be also approved after using the card.

15	Changing transaction mode	O	<ol style="list-style-type: none"> 1.Establish connection with ECR and perform approved Purchase transaction with Direct mode. 2.Change transaction mode on Dual and perform transaction with other amount . Confirm amount by OK on displayed by ECR MutableDualtransaction parameters screen 	After approved purchase transaction in Direct mode, the Terminal should display screen with new amount performed on Dual mode which should be also approved after using the card
16	LAN removal during Transaction	M	Perform a transaction and remove LAN cable (during card reading)	Transaction should be declined, Terminal should returns to main screen and ECR should generate error Transport layer broken signal received.
17	Transaction after LAN reconnecting	M	<ol style="list-style-type: none"> 1.Connect LAN to device and close ECR connection 2.Open new ECR connection and perform new transaction 	After establish connection new transaction should be approved without any errors on ECR
18	Connection after terminal restart	M	<ol style="list-style-type: none"> 1.Restart terminal by holding Green+Red button. In the same time close ECR connection and establish new one. 2.After restart perform new transaction 	Transaction performed after device restart should be approved without any errors in ECR
19	Test specific character	M	Some country have specifics letter “ ñ, Ñ, Ö ;;”	Not error on Terminal side should be expected
20	Broken communication between ECR and Terminal	M	<ol style="list-style-type: none"> 1. Perform transaction and break connection between ECR and Terminal (unplug USB, block communication...) 2. ECR detects broken connection 3. Restestbalish connection: close and Open 4. Perform get last transaction 	

15.3 Recurring questions

1	When to apply Cancelation and when apply Refund?
2	<p>Timeout</p> <p>The recommendation is to have a timeout of 10 sec after the transaction request in Dual Mode and 20 sec in Direct Mode.</p> <p>When the TimeOut expire, it is recommended:</p> <ul style="list-style-type: none"> To send a request GetLastTransaction If the transaction is OK, the ECR complete the transaction (printer) If the transaction is Not OK (the ECR can send a new request payment If the card holder has decided to pay with other method or card, the ECR need to send a Reversal message (cancelation)

3	<p>The ECR have lost the connection with the terminal</p> <p>It is recommended to send first a close request before sending a new Open message request.</p> <p>if the problem persists, check cables connection and network status</p>
4	<p>Can I remove a terminal and connect directly in other store ?</p> <p>A terminal can be removed and connected again in the same store.</p> <p>In case that the Merchant need to move the terminal to another store, it is necessary to:</p> <ol style="list-style-type: none">1) Remove the terminal on nTMS2) Declare the terminal (installation) under the new Merchant- store3) To active the terminal
5	<p>Amount Tip</p> <p>In case that the ECR provides the AmountTip, the total transaction amount, used in the authorization to the bank, is Amount plus AmountTip. The terminal does not have a specific authorization for Tip only.</p>