Payment Express EFTPOS

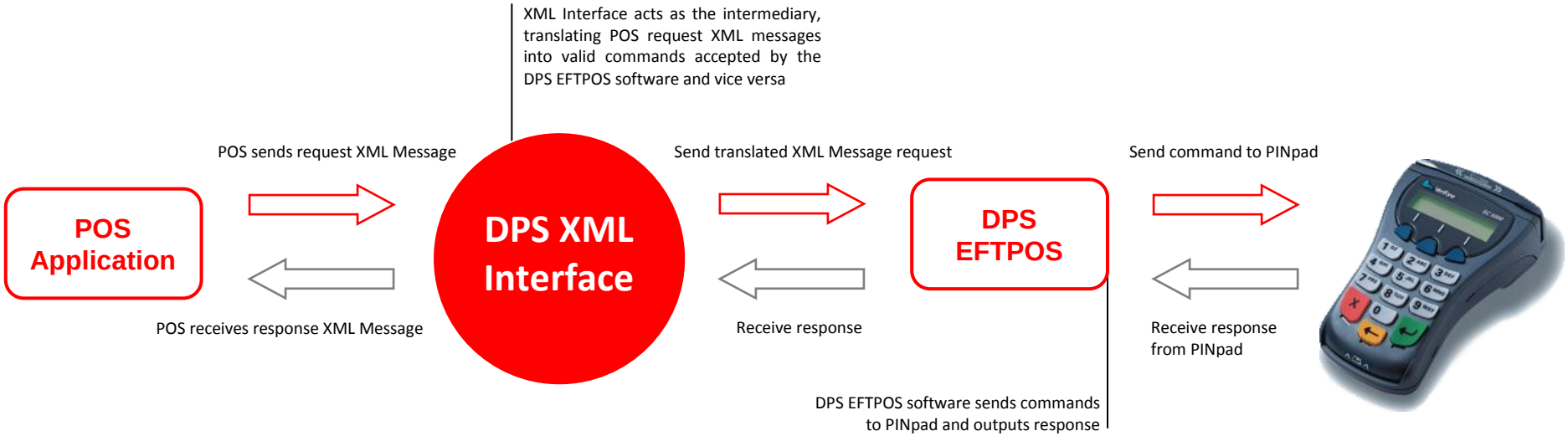
XML Socket Interface

dps | paymentexpress

CONTENTS

OVERVIEW

The Payment Express® EFTPOS TCP/IP XML socket interface allows Point of Sale (POS) applications to communicate with Payment Express® EFTPOS terminals across a network. By default the XML interface will be installed and configured to listen locally i.e. 127.0.0.1 on port 65.



Using XML as the communication medium allows for interoperability between Payment Express® EFTPOS, POS applications and the Operating System (OS) environment. The XML interface acts as the intermediary, translating POS request XML messages into valid commands accepted by the DPS EFTPOS software (or vice versa). Therefore, as a developer looking to integrate with the XML interface, you need only be concerned with the left-most section of the diagram above.

PREPARATION

To begin integration testing with Payment Express EFTPOS, you will need the following:

- **Payment Express® EFTPOS development account** – Contact our EFTPOS sales team to request a dev account. Call us on 0800 PAYMENT (729 6368) or +64 9 309 4693, apply online at https://www.paymentexpress.com/pxmi/apply, or email us at eftpos@paymentexpress.com.
- **PINpad hardware** – Once your application has been processed, you will be issued the hardware for development and testing. Available hardware options can be found at www.paymentexpress.com/products/integrated_eftpos/integrated_eftpos_hardware.html
- **Test cards** – Cards used for card present tests will be included with your PINpad hardware package. Test card details can be found at www.paymentexpress.com/technical_resources/integrated_eftpos/eftpos_test_cards.html
- **DPS EFTPOS software** – This is the software your POS application will be integrating with And will be downloaded
- **XML Socket Interface Integration Guide -** www.paymentexpress.com/technical_resources/integrated_eftpos/eftpos_xml_socket_interface.html
- **DPS Test Scripts** - www.paymentexpress.com/technical_resources/integrated_eftpos/eftpos_test_scripts.html
- **XML Socket Interface Technical Specification** - www.paymentexpress.com/technical_resources/integrated_eftpos/eftpos_xml_socket_interface.html
- **Sample C# EFTPOS Application/Code** - www.paymentexpress.com/technical_resources/integrated_eftpos/eftpos_xml_socket_interface.html

Before you begin, consider the following:

User Interface Management

The EFTPOS operator dialogs can be managed by DPS or POS application. The preferred method is to have the POS handle all dialog boxes. If UI is to be managed by DPS, a standalone XML interface service needs to be installed (note: this may not be possible in some systems e.g. Linux).

If the POS application is to manage the EFTPOS operator dialogs (e.g. customized UI) the XML interface will provide display messages for the POS application to interpret. An additional section in the test scripts caters for POS-managed UI ("Part Six"). *POS providers must inform DPS QA which method is selected when submitting the POS certification request.*

Example "Transaction Accepted" dialog box (DPS)

Example "Transaction Accepted" dialog box (POS)

Receipt Printing

Printing of EFTPOS receipts can be managed by DPS or POS application. When submitting a request, "ReceiptAutoPrint" can be set to "1" for DPS controlled printing, or "0" (default) for POS controlled printing. For POS controlled printing, the Receipt response can be used, as it contains a digital copy of the receipt data. The receipt data must be printed out as supplied. DPS controlled printing allows you to add header and footer text, however more complicated features such as embedding EFTPOS receipt within a larger POS receipt will require the POS to handle the printing. *Please specify your preferred method when requesting for your test account and PINpad. This ensures that your PINpad is setup accordingly*

An additional section in the test scripts caters for POS managed receipt printing ("Part Five").

Receipt Printer
A minimum of one receipt printer is required to process EFTPOS transactions. The majority of EFTPOS setups consist of one receipt printer per PINpad/POS system; however EFTPOS can also be configured for shared network printing also. Before proceeding, please make sure a test page can be successfully printed. More information on receipt printing found on page 10.

DPS stock and supply the Tysso PRP-085 (USB/Serial) Receipt printers. To enquire about these printers, you can email our EFTPOS Sales team at eftpos@paymentexpress.com, or call us on 0800 PAYMENT (729 6368) or +64 9 309 4693.

Deployment Environment
A local setup running Windows is the most common scenario. Supported platforms are currently.

Windows XP
Windows 2000
Windows 2003
Windows Vista
Windows 7

Payment Express® EFTPOS PINpads can also be used in a Citrix or Terminal Services environment, where PINpads are connected locally to the thin client and are mapped through using Citrix's ICA and Terminal Service's RDP protocols to a server with Citrix MetaFrame or Terminal Services installed. Supported Terminal Services platforms are currently:

Windows 2000 Server
Windows 2003 Server
Windows 2008 R2 Server

For more information regarding Citrix or Terminal Services support, please refer to the “Citrix and Terminal Services” section on page 43.

USB or Serial port availability
One free USB or Serial port is required for the PINpad. When requesting a PINpad please ensure that the appropriate unit and cables are requested for your desired connection type.

Privileges
To execute and install DPS EFTPOS software administrative privileges are required during the setup phase. Permanent write privileges are also required for the user on the system (install volume) for temporary storage of transaction information.

Firewall considerations
DPS EFTPOS testing requires access to our QA2 server.

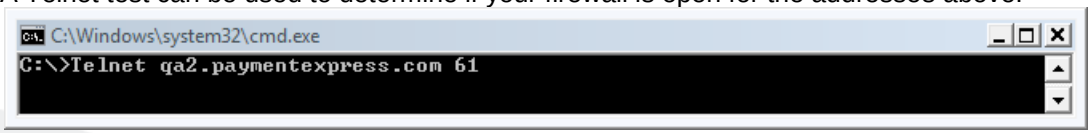
Address: qa2.paymentexpress.com
IP: 125.236.50.181 (*note: we do not recommend using IP address. We recommend using DNS address above as it will prevent any issues in the unlikely event the IP address needs changing*).
Port: 61

Address changes to eft6.paymentexpress.com (primary), eft7.paymentexpress.com and eft2.paymentexpress.com on port 61 during production.

Other firewall considerations:
Outbound access on port 80 is required to browse to our website (www.paymentexpress.com) and download the installer (.exe). The installer needs to connect to port 443 as it uses SSL encryption to connect to our server and download application and configuration files. In summary, outbound connectivity to www.paymentexpress.com and qa2.paymentexpress.com on ports 80 and 443 is required.

DPS also troubleshoot remotely using 'GoToAssist Express'. Your network will need to allow connection to www.fastsupport.com on ports 8200, 80 and 443. Further detailed documentation regarding GoToAssist Express can be supplied upon request.

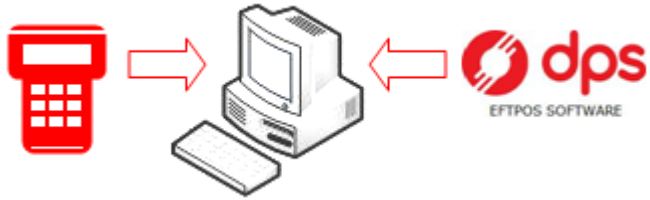
A Telnet test can be used to determine if your firewall is open for the addresses above.



A successful result will look like the following.

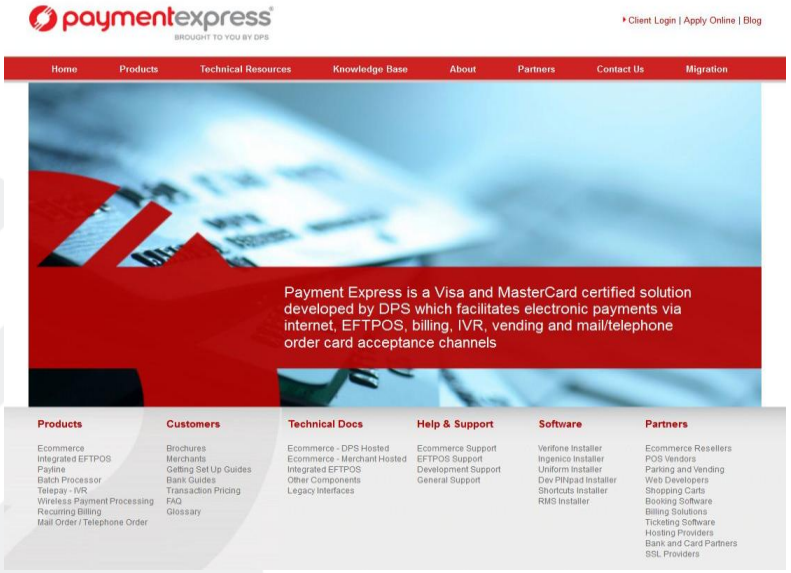
GETTING STARTED

To get started browse to www.paymentexpress.com to download the EFTPOS installer for development PINpads. Unless otherwise specified, the installation executable will be “pxinstall_qa.exe”. Run the installer and plug the PINpad into the PC when prompted. Once initialized, the installer will pick up the PINpad and the installation will continue downloading all required components. Detailed installation process documented below (For Citrix or Terminal Services installation refer to page 43).



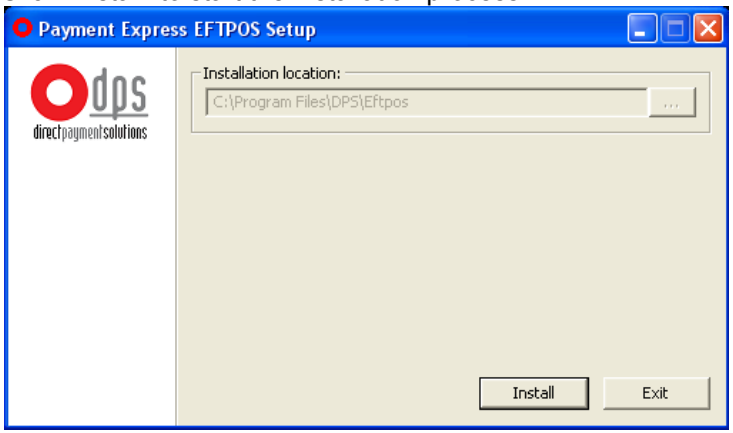
DPS EFTPOS Software Installation

1. Open a web browser and go to www.paymentexpress.com



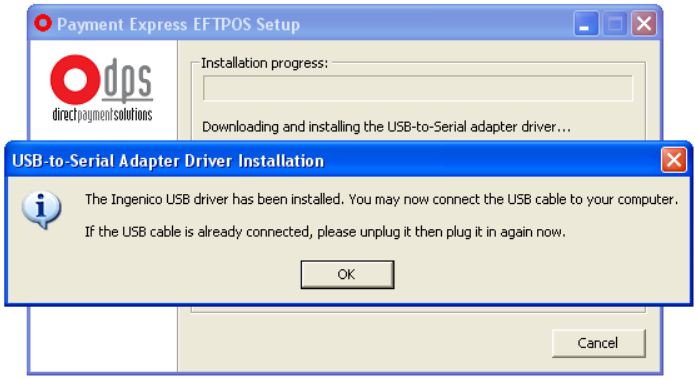
2. Find and run the appropriate installer - located on the bottom of the main page under the “Software” category.

3. Click “Install” to start the installation process

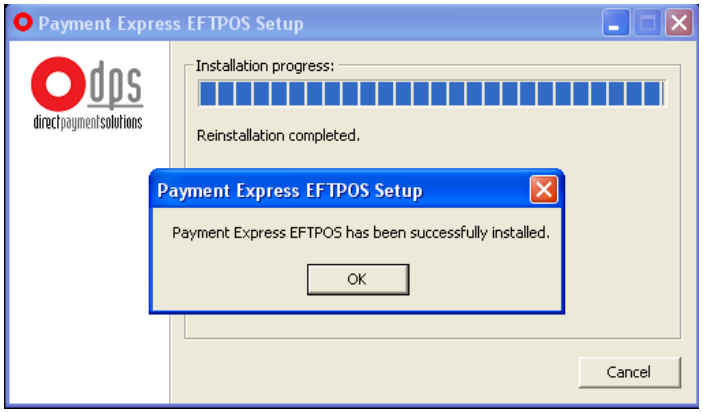


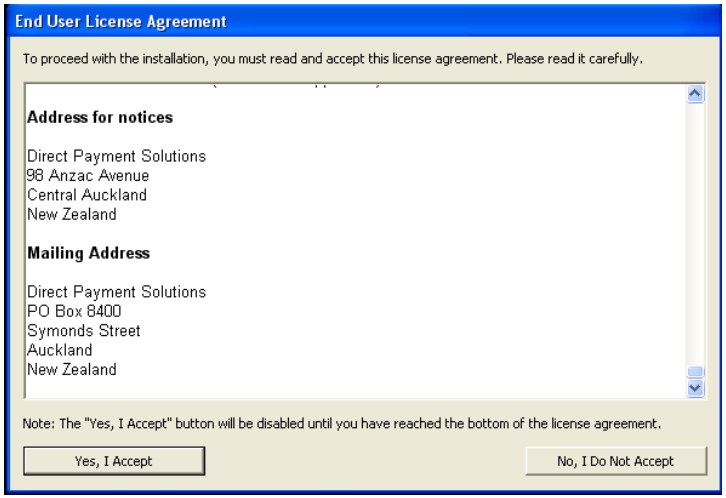
4. Please read the End User License Agreement carefully. Once you have scrolled all the way to the bottom, the option to accept DPS license terms will be presented
(Please note: these need to be accepted; if any questions, please contact sales@paymentexpress.com).

5. Plug in your EFTPOS PINpad and Click “OK”. Please note that this message will vary depending on the installer used. The installer used for the screenshot below is the Ingenico installer.



6. Your DPS EFTPOS software is now successfully installed. Click “OK” to complete installation.





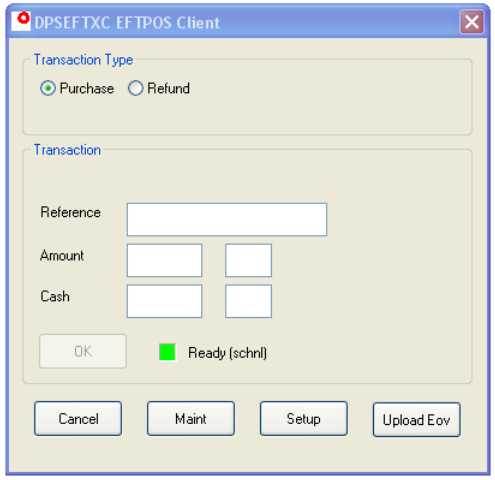
Software Components

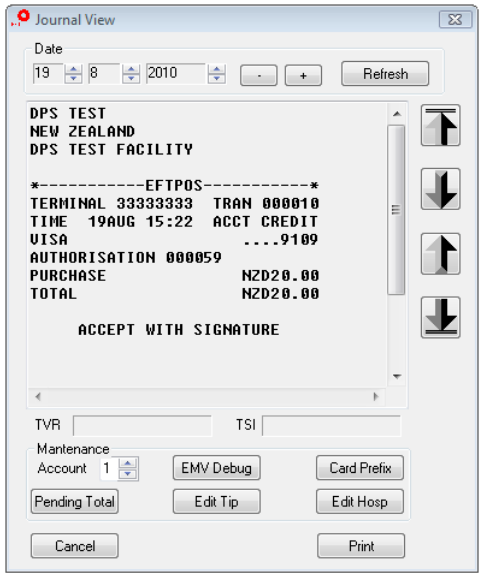
These are the main files that are installed (and updated) during installation. The files downloaded and installed during installation may differ depending upon the configuration loaded at DPS for the particular PINpad being installed. The default installation directory is “C:\Program Files\DPS\EFTPOS”.

Filename	Description
dpseftx.ocx	ActiveX Control Object. Will be registered by the installer.
efttray.exe	Tray application to handle capturing of input files. Also allows access to open the Config and Maintenance control panels. Optional install.
pxpp.exe	PINpad Controller. NT service that is installed and started with the installer.
pxeftp.exe	Communication Server. NT service that is installed and started with the installer.
pxjview.exe	Journal Viewer. Standalone application to view past transaction history.
dpseftxc.exe	Standalone EFTPOS client for non-integrated POS transactions.
uninst.exe	Starts the uninstaller process for removing the software.

EFTPOS Client (dpseftxc.exe)

Journal Viewer (pxjview.exe)



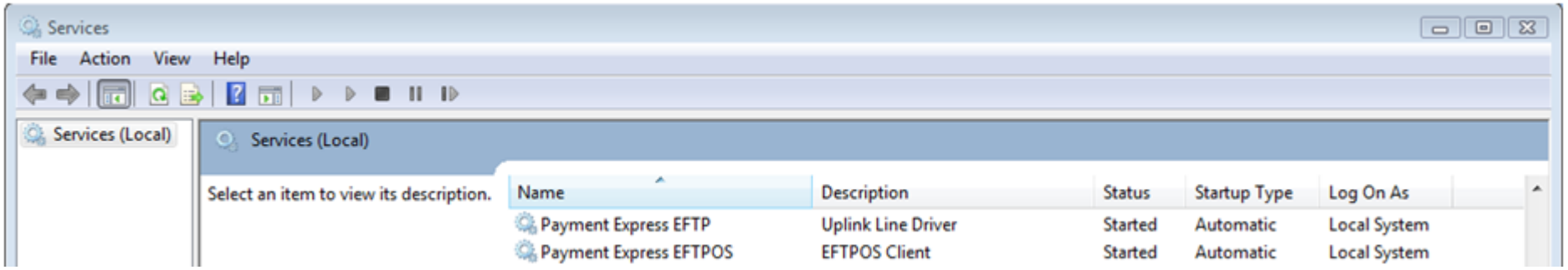


Services

These are the services installed during installation and required to be started for Payment Express® EFTPOS to function. The main services are the Payment Express EFTPOS (PXPP) and Payment Express EFTP (PXEFTP). Depending on your setup, additional services may be installed.

Service	Description
PaymentExpressPxpp	PINpad Controller
PaymentExpressPxEftp	Communication Server

Windows Services (services.msc)



RECEIPT PRINTING

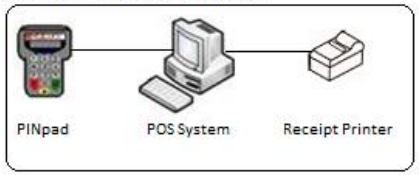
A minimum of one receipt printer is required to process EFTPOS transactions. It is mandatory to print receipts for all EFTPOS transactions.

Four common setups:

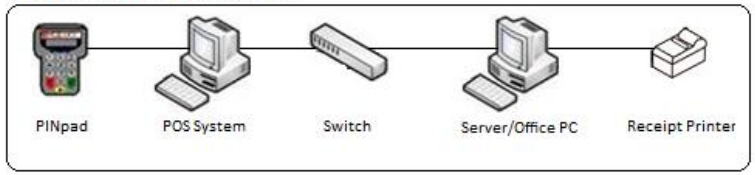
- Single Lane setup with 1 local printer
- Single Lane setup with 1 network printer
- Multi-Lane setup with 1 printer per POS/PINpad
- Multi-Lane setup with 1 printer shared

Examples:

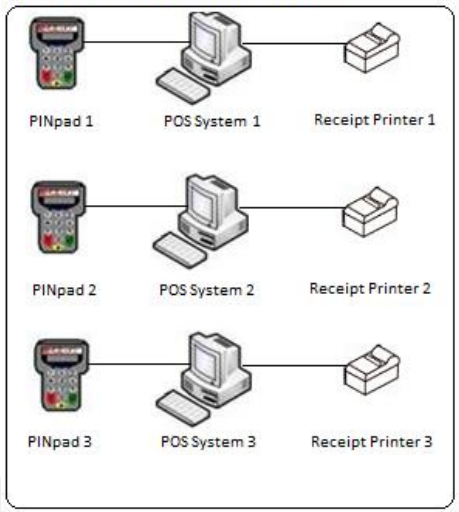
Single Lane setup with 1 local printer



Single Lane setup with 1 network printer

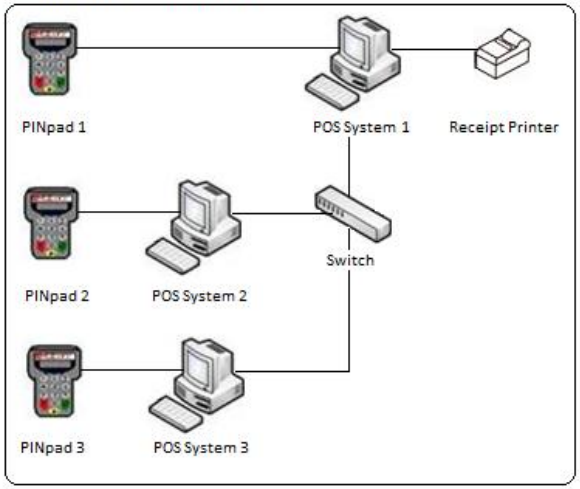


Multi-Lane setup with 1 printer per POS/PINpad

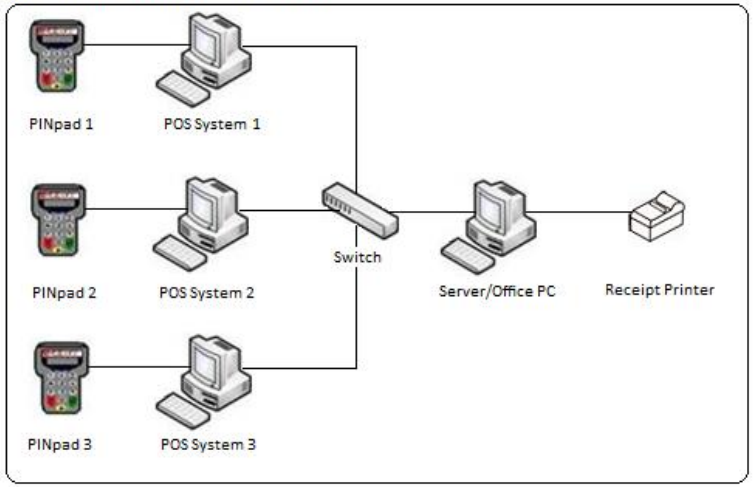


Multi-Lane setup with 1 printer shared

Receipt printer connected to POS PC



Receipt printer connected to server/office PC



If DPS manage the printing, the PrinterName and PrinterName2 fields in pxpp_cfg.txt must contain the exact name of the printer setup. This is stands true for both windows queue driver and OPOS drivers. For shared/network printers, the DNS or IP address of the printer must also be supplied to the other PC's sharing the printer. This is usually the name or address of the PC which has the printer locally installed. The PrinterName and PrinterName2 fields will contain the name or address along with the printer name.

Local Setup example:	Shared/Network (IP) example:	Shared/Network (DNS) example:
<PrinterName>Receipt</PrinterName> <PrinterName2>Receipt</PrinterName2>	<PrinterName>\\192.168.1.200\Receipt</PrinterName> <PrinterName2>\\192.168.1.200\Receipt</PrinterName2>	<PrinterName>\\POS1\Receipt</PrinterName> <PrinterName2>\\POS1\Receipt </PrinterName2>

PrinterName handles the printing of Transaction receipts, while PrinterName2 handles the printing of Merchant receipts logons, settlements, enquiries, pending receipts etc. If the POS manages the printing of receipts then the settings above are ignored.

When a receipt is to be printed a Receipt (Response) message is sent from the EFTPOS terminal to notify the POS. This message contains a digital copy of the data that would be printed on the receipt. The EFTPOS terminal will send this message every time a receipt is to be printed, even if the terminal is responsible for printing the receipts. More information on the Receipt (Response) message can be found on page 28.

The card holder is entitled, and in many regions required, to receive a receipt at the conclusion of a financial transaction. Because of this, the POS application needs to ensure that a transaction cannot begin unless the receipt printer is fully operational e.g. a transaction cannot start if the printer is out of paper.

If the POS does not support printer availability checking, EFTPOS software can handle the checking instead. The setting EnableCheckPrinter (found in pxpp_cfg.txt) If enabled, will check the printer status before attempting to print a receipt.

Customer Copy

As above, all EFTPOS transactions require an EFTPOS receipt. The customer copy is printed for all transactions. This is the receipt that is given to the customer.

Example:

```
DIRECT PAYMENT SOLUTIONS
96-98 ANZAC AVE
AUCKLAND

*----------------EFTPOS----------------*
TERMINAL 00905302  TRAN 000321
TIME  10JUN 15:17  ACCT CHEQUE
                   503871....7361
PURCHASE                  NZ$1.00
TOTAL                     NZ$1.00
              ACCEPTED
*----------------------------------------*
          CUSTOMER COPY
```

Merchant Copy

The merchant copy is a duplicate receipt of the customer copy printed for the merchant to keep. These receipts are used by merchants as hard copies for signature verified transactions, EOV transactions and general reconciliation purposes.

Example:

```
DIRECT PAYMENT SOLUTIONS
96-98 ANZAC AVE
AUCKLAND

*----------------EFTPOS----------------*
TERMINAL 00905302  TRAN 000321
TIME  10JUN 15:17  ACCT CHEQUE
                   503871....7361
PURCHASE                  NZ$1.00
TOTAL                     NZ$1.00
              ACCEPTED
*----------------------------------------*
          MERCHANT COPY
```

If DPS handle the printing, a merchant copy can be obtained by setting EnablePrintDuplicateReceipt to ‘1’ in pxpp_cfg.txt. EnablePrintDuplicateReceipt is set to ‘0’ by default. Please note: To apply changes made in pxpp_cfg.txt, the PaymentExpressPxPp service needs to be restarted.

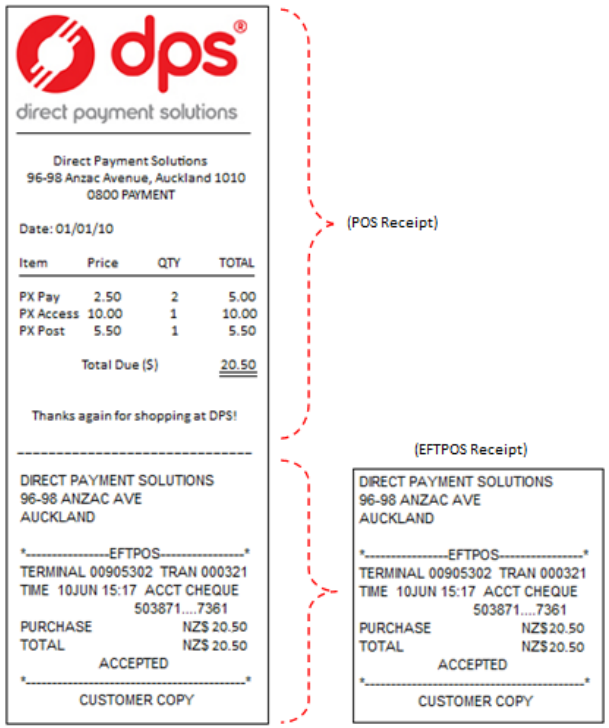
I.e. Change <EnablePrintDuplicateReceipt>0</EnablePrintDuplicateReceipt> to <EnablePrintDuplicateReceipt>1</EnablePrintDuplicateReceipt>

If POS manages the printing, a merchant copy can be obtained by using the LastRequest request message. The response contains receipt data with “MERCHANT COPY” instead. This can then be printed by POS.

Embedded Receipts

EFTPOS Receipts can be combined with the POS receipt to print one receipt instead of two. Each Receipt response message contains the <IsSeparate> element which indicates whether the EFTPOS receipt should be combined with the POS receipt or is a merchant copy. If the returned value is '0' then the POS can print the POS receipt data and EFTPOS receipt data as one receipt. The EFTPOS receipt data must not be changed. IsSeparate will be set to '0' for the customer copy only. All merchant copies will have IsSeparate set to '1' to tell the POS that it needs to be printed on its own. Please note that this functionality requires the POS to manage the printing.

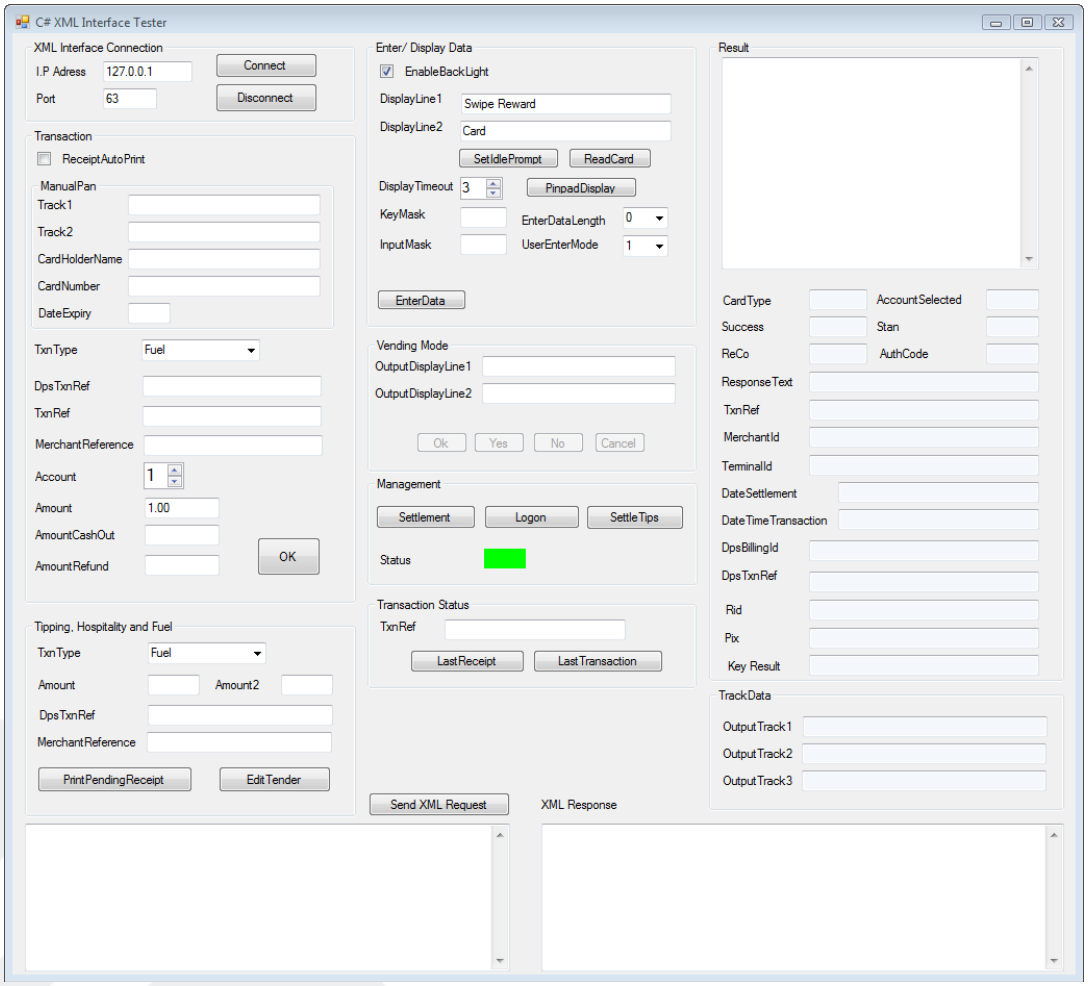
Example:



SAMPLE APPLICATION/CODE

Sample application and code can be downloaded from our website at
www.paymentexpress.com/technical_resources/integrated_eftpos/eftpos_xml_socket_interface.html

Sample is currently available in C#.



TEST CARDS

EFTPOS Test cards will be included with your development PINpad. The following are the available test cards that can be processed through the Payment Express 997 Host Id test network. You are also able to use the track data for transactions and card numbers for manual pan transactions.

Visa Credit with optional PIN	
CardNumber	4999999999999109

Visa Credit and Local Debit (Dual Card) with optional PIN on credit

Track2	;4999999999999109=09081014764094900000?
Pin Enabled	Yes. Valid pin is 1234.
Accounts	Credit

CardNumber	4999999999999108
Track2	;4999999999999108=09081014764094900000?
Pin Enabled	Yes. Valid pin is 1234.
Accounts	Cheque & Credit

MasterCard Credit only	
CardNumber	5999999999999108
Track2	;5999999999999108=09081014764094900000?
Pin Enabled	No
Accounts	Credit

Debit Card CHQ	
CardNumber	9999999999999108
Track2	;9999999999999108=09081014764094900000?
Pin Enabled	Yes. Valid pin is 1234.
Accounts	Savings

CONNECTING TO THE PINPAD

Now that we have the DPS software, PINpad and receipt printer sorted we need to connect to the PINpad. Since we are documenting the XML interface, connection will be handled via the XML Listener. Port 65 is the default configured TCP/IP port that the XML Listener opens for connection and communication with your POS application. For Flash/AS 2.0 the port will need to be greater than or equal to 1024.

This setting can be found in PXPP configuration file “pxpp_cfg.txt”.

```
<XmlInterface>
        <XifAddress>0.0.0.0</XifAddress>
        <XifPort>65</XifPort>
</XmlInterface>
```

Note: For integrations using the standalone XML interface service, the relevant settings will be found in dpseftxifxp_cfg.txt. The default port will be “63”.

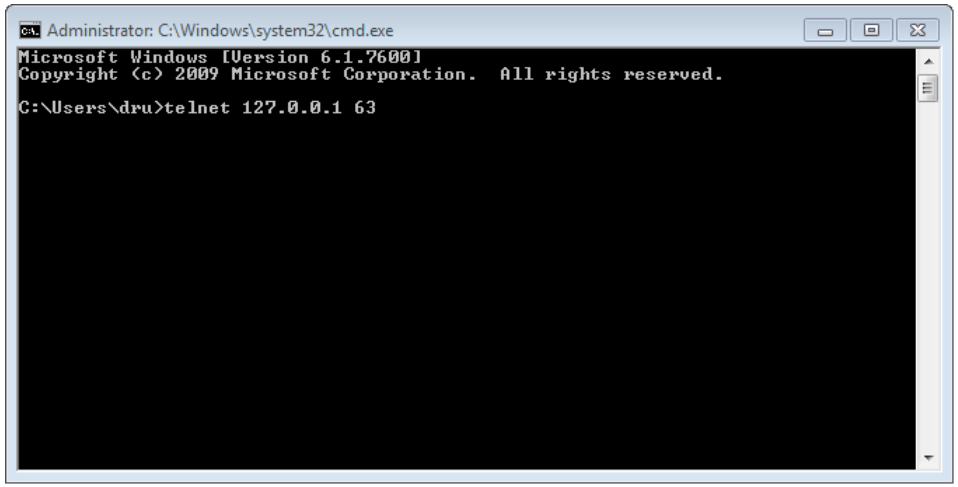
Connecting to the PINpad

To connect to the PINpad, you can either use your POS application, sample/test applications provided by DPS (available in C#), or use Command Prompt (CMD). We will be using CMD prompt for this guide.

- Open a command prompt window (Windows Start Menu > Run > Type in “CMD” and click OK).

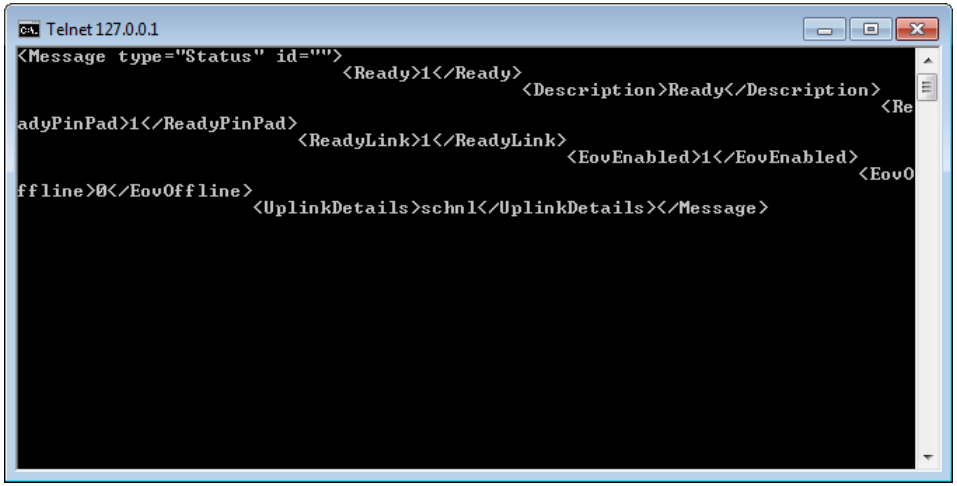
Telnet to the localhost on port 65.

- Telnet to the localhost on port 65 i.e. Type “telnet 127.0.0.1 65”.



Connection established!

Connection established! You have now received your first XML response message. This message contains details regarding the PINpads current status.




```
<Message type="Status" id="">
<Ready>1</Ready>
<Description>Ready</Description>
<ReadyPinPad>1</ReadyPinPad>
<ReadyLink>1</ReadyLink>
<EovEnabled>1</EovEnabled>
<EovOffline>0</EovOffline>
<UplinkDetails>schnl</UplinkDetails></Message>
```

Now that you are connected you can now start sending XML messages through to the EFTPOS terminal.

In the following sections we outline the basic XML message format used by DPS and then walk you through the two main categories: Normal messages and unsolicited messages.

BASIC COMMUNICATION

Normal Messages

Each individual message sent to or received from the terminal will look like:

```
<Message type="message type" id="application-specified ID">
   ... message-specific data ...
</Message>
```

The same message format is used for messages in both directions. Each Message item has two attributes.

- The first attribute, type, specifies the type of data that is present in the message. The EFTPOS terminal and the POS software must examine the type of each message and take the appropriate action.
- The second attribute, id, is used to track associated messages. This attribute is optional but strongly recommended. When the POS software specifies an id in a Message sent to the EFTPOS terminal, all corresponding Messages sent in return will contain the same id value. This is particularly useful when an operation causes multiple Messages to be sent back to the POS software.

Any line breaks inside the message data will be encoded to single-byte format, i.e. the line break is encoded as '0x0A'. All messages will be encoded into ASCII format.

An XML Schema document (.XSD) can be downloaded from our website.
Link: http://www.paymentexpress.com

Unsolicited Messages
These messages are usually sent from EFTPOS terminal to POS system. An unsolicited message may or may not be associated with a request message. It is usually used by EFTPOS terminal to display information on the POS system's screen or prompt for user input from POS system. It will also be used for notifying the status changes of EFTPOS terminal. Depending on the type of unsolicited message the POS system may need to take actions upon receipt of the message. This is important in situations where the POS system is shutdown or has crashed in the middle of a transaction, and the EFTPOS terminal is awaiting feedback from the POS. The POS application must be able to reconnect immediately to the XML listener upon restart to resume the process.

COMMON FUNCTIONS

These messages are usually request/response pairs. The request message is usually initiated by POS system to make a request to EFTPOS terminal, and then being replied by the EFTPOS terminal in the format of response message. The response message uses the same format as the request message and includes a common field "Success" to indicate whether the request is successfully being processed by the EFTPOS terminal.

The acknowledge message will include the "success" element set to "1" and other message specific fields. It will look like the following:

```
< Message type="command type" id="application-specified ID"
>
< Success>1< /Success>
... message-specific fields ...
< /Message>
```

The error (NACK) message will include the "success" element set to "0" and two other elements "Reco" and "ResponseText" to specify the error details. It will look like the following:

```
< Message type="command type" id="application-specified ID">
< Success>0</Success>
<Reco>I2</Reco>
<ResponseText>Unknown message type</ResponseText>
... other message-specific fields ...
</Message>
```

The EFTPOS terminal can only perform one operation at a time. Consequent request messages will be rejected if the previous request is still pending processing. An ACK or NACK response message indicates that the pending request has been processed completely.

Now that we have a basic understanding of the message format and what a normal message is, we can look at sending some of these messages to the EFTPOS terminal. By doing these examples, you should be able to gain a good understanding on how DPS EFTPOS works via the XML Interface.

Logon (Request/Response)
In order to prepare the terminal for processing transactions, the Logon message is used. A Logon uses the assigned merchant number and terminal ID to login to the banking switch. *Note: Logon functionality is mandatory and required for POS certification.*

A Logon is also used as a manual trigger to restart online processing after a period of offline processing.

Logon (Request)

The Logon request notifies the EFTPOS terminal that the POS wished to perform a manual logon to the EFTPOS network.

Example:

```
<Message type="Logon" id="1234">
<Account>1</Account>
</Message>
```

Logon (Response)

The Logon response notifies the POS that the EFTPOS terminal has completed its logon.

Example:

```
<Message type="Logon" id="1234">
<ReCo>00</ReCo>
<ResponseText>ACCEPTED</ResponseText>
<Account>1</Account>
</Message>
```

In the above example we are sending a request message to initiate a merchant “Logon”. The message tells the EFTPOS terminal to logon to Account 1 (which is also the default account). A standard setup will have one Account which refers to a merchant number (CAID) and terminal ID (CATID). In the testing environment, dummy CAID/CATID numbers are issued. If there is only one Account, you are not required to submit the message with the account value.

The response returned shows that the Logon has been processed successfully i.e. ResponseText “ACCEPTED”, and ReCo “00”. When this message has been received, the logon operation is complete, terminal is ready to process transactions and the POS system may continue with other operations.

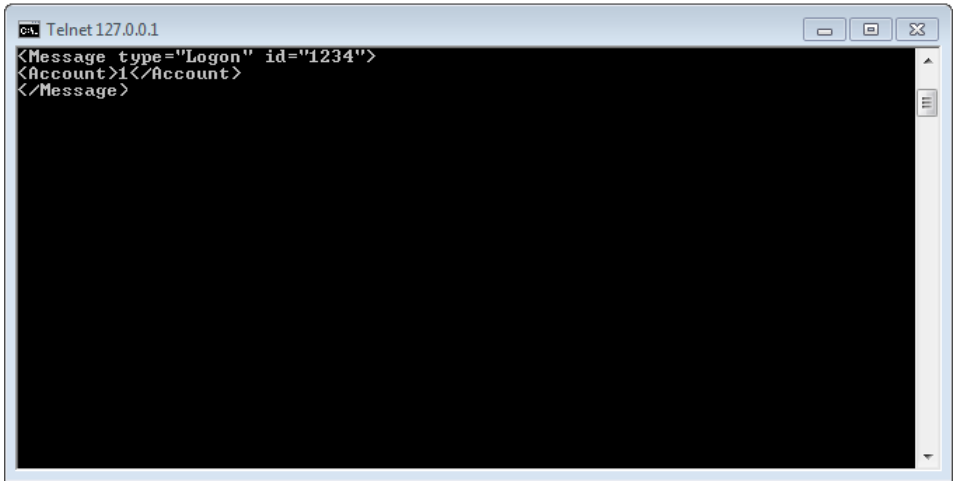
How do to do a manual Logon (via Command Prompt).

- Connect to PINpad via CMD prompt.

- Once PINpad is connected and “Ready”, we can send the Logon request message.

- Copy and paste the Logon example into CMD prompt to initiate a Logon.

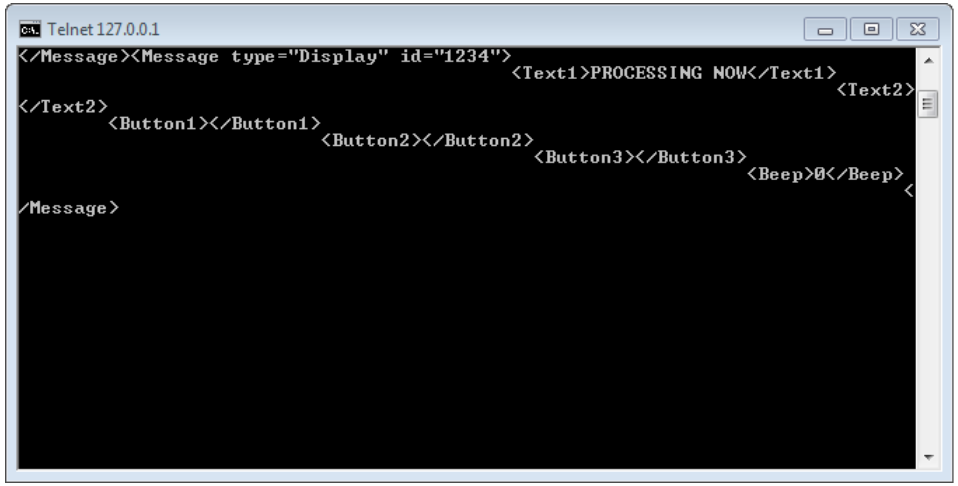
  <Message type="Logon" id="1234">
  <Account>1</Account>
  </Message



- If EFTPOS config is setup for POS managed dialog boxes, the following Display (Response) message will be returned.

  Message type="Display" id="1234">
  <Text1>PROCESSING NOW</Text1>
  <Text2>
  </Text2>
  <Button1></Button1>
  <Button2></Button2>
  <Button3></Button3>
  <Beep>0</Beep>
  </Message>

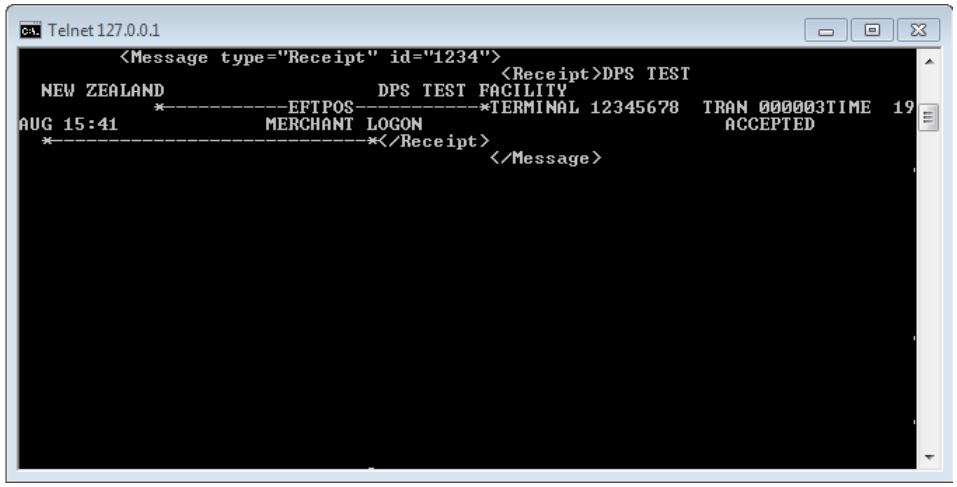
  This message is sent from the EFTPOS terminal telling the POS to display “PROCESSING NOW”. If your EFTPOS config is setup for DPS managed dialog boxes, you will not receive this message.



- Once the request has been processed a Receipt (Response) message is returned.

<Message type="Receipt" id="1234">
<Receipt>DPS TEST
NEW ZEALAND DPS TEST FACILITY
*----------EFTPOS----------*TERMINAL 33333333 TRAN
000003TIME 19
AUG 15:41 MERCHANT LOGON
ACCEPTED
----------</Receipt>
</Message>

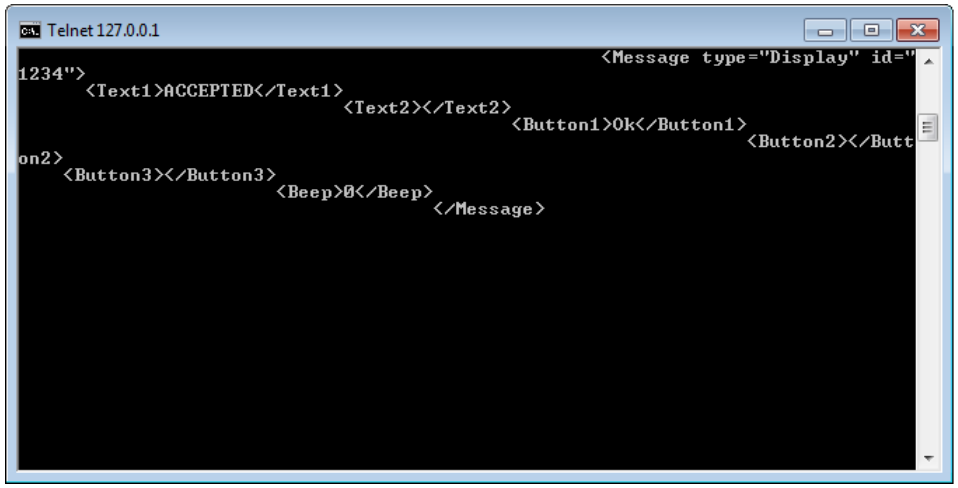
This message notifies the POS system that a receipt is to be printed. Depending on who controls the printing, the EFTPOS config needs to be setup accordingly. If POS controls printing, the Receipt data returned can be used by POS. *More information on Receipt (Response) messages on page 18.*



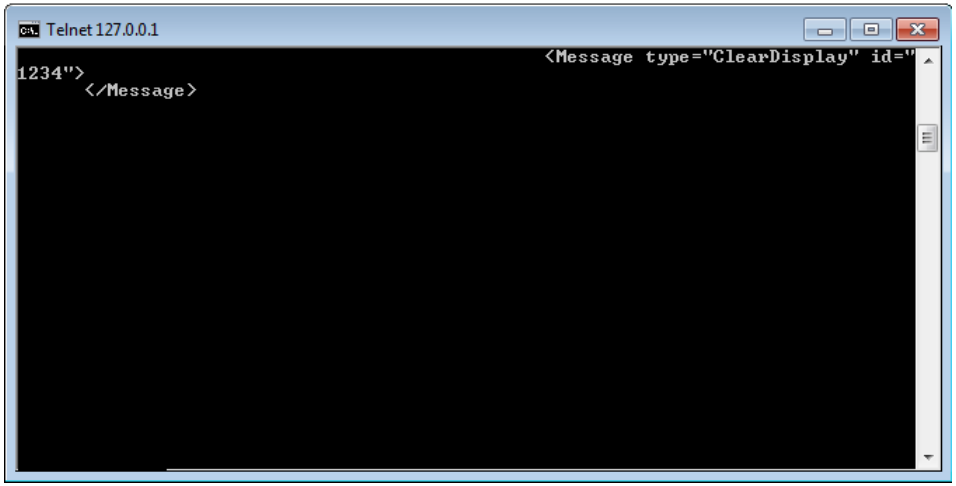
- Again if EFTPOS config is setup for POS managed dialog boxes, the following Display (Response) message will be returned.

<Message type="Display" id="1234">
<Text1>ACCEPTED</Text1>
<Text2></Text2>
<Button1>Ok</Button1>
<Button2></Button2>
<Button3></Button3>
<Beep>0</Beep>
</Message>

This message is sent from the EFTPOS terminal telling the POS to display “ACCEPTED” If your EFTPOS config is setup for DPS managed dialog boxes; you will not receive this message.



- A ClearDisplay (Response) message will be sent. This message notifies the POS system that it should remove any message that it was asked to display. *More information on ClearDisplay (Response) message on page 23.*

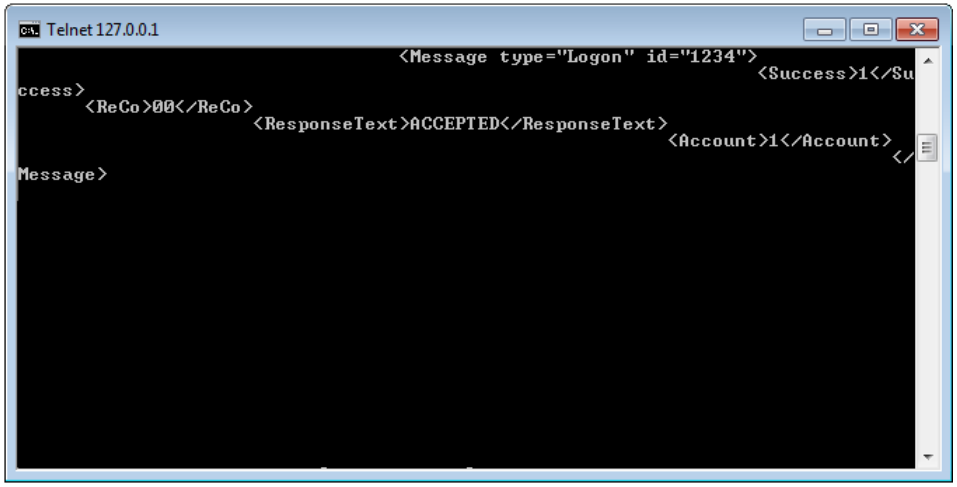


- Finally, a Logon response will be returned.

  The response should look like the following:

  <Message type="Logon" id="1234">
  <Success>1</Success>
  <ReCo>00</ReCo>
  <ResponseText>ACCEPTED</ResponseText>
  <Account>1</Account>
  </Message>

  You have now processed your first successful manual Logon.



Purchase Transaction (Request/Response)

Another common message you will be using is the Purchase Transaction (Request/Response) message. This is how your POS will be telling the EFTPOS terminal to charge a customer's card.

Purchase Transaction (Request)

This transaction request message notifies the EFTPOS terminal that the POS system wishes to initiate an EFTPOS purchase transaction. The message data specifies the type of transaction and the amount to be processed.

Example:

```
<Message type="Transaction" id="1234">
<TxnType>Purchase</TxnType>
<TxnRef>TXN12345</TxnRef>
<AmountPurchase>1.00</AmountPurchase>
</Message>
```


Purchase Transaction (Response)

Once the transaction has been processed (or cancelled) a transaction message will be sent back to the POS application, indicating the outcome of the transaction.

Example:

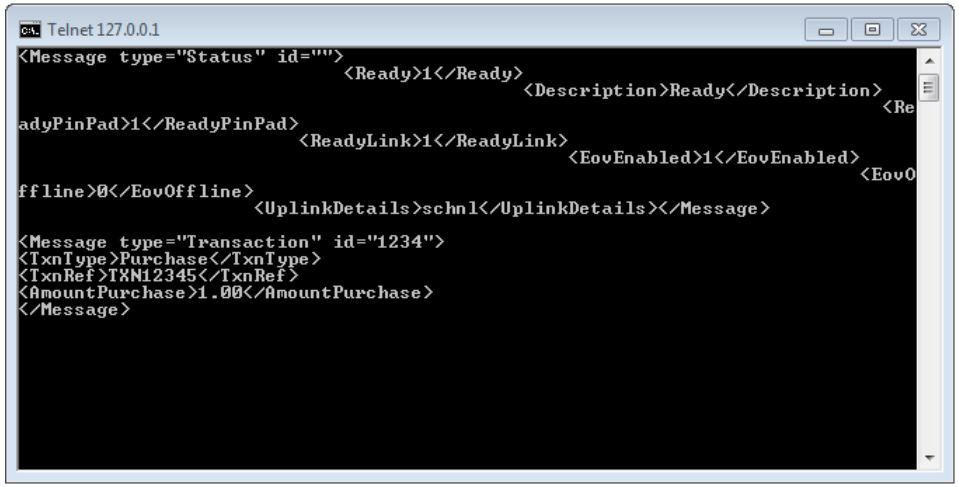
```
<Message type="Transaction" id="1234">
<ReCo>00</ReCo>
<ResponseText>ACCEPTED</ResponseText>
<Success>1</Success>
<Account>1</Account>
<TxnRef>TXN12345</TxnRef>
<TxnType>Purchase</TxnType>
<TxnDate>20060101</TxnDate>
<SettleDate>20060102</SettleDate>
<CardType>Visa</CardType>
<AccountType>Credit</AccountType>
<AmountPurchase>1.00</AmountPurchase>
<AmountCashOut>0.00</AmountCashOut>
</Message>
```

In the example above, a customer wants to purchase an item valued at 1.00. The POS initiates a "Purchase" type transaction and requests that the purchase amount to be "1.00". Please note that this is a very basic example and does not use all available elements applicable for a transaction request.

A full list of applicable elements for the transaction request message can be found on our technical specifications page at:
http://www.paymentexpress.com/technical_resources/integrated_eftpos/eftpos_xml_socket_interface.html

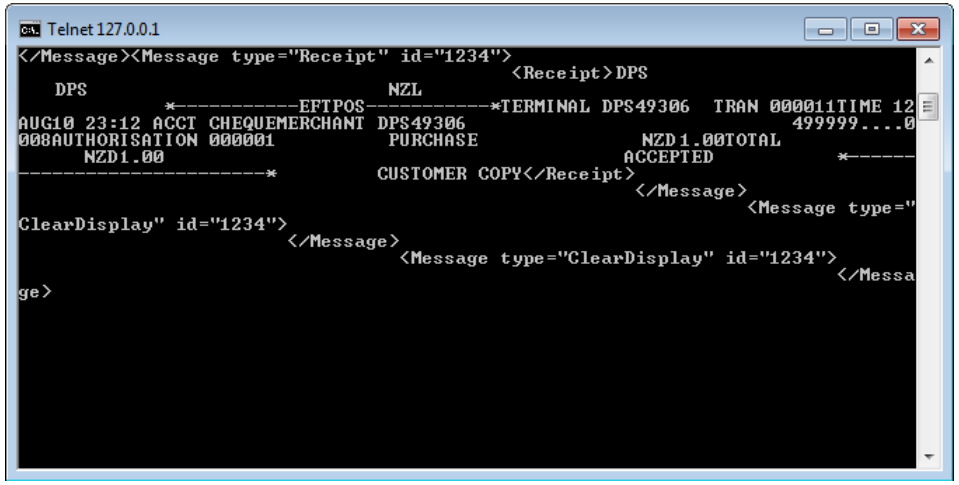
The response returned shows that the transaction has been processed successfully i.e. Success “1”, ResponseText “ACCEPTED”, and ReCo “00”. Response data can then be interpreted by the POS application. To replicate the example on the previous page, we need to send the transaction request message to the EFTPOS terminal. For this example, we will process this purchase using the VISA dual test card that supports PIN.

- Connect to PINpad via CMD prompt.
- Once PINpad is connected and “Ready”, we can now send the transaction request message.
- Copy and paste the example message into CMD prompt.
- Your EFTPOS PINpad should now display “PLS SWIPE CARD" on its screen.
- Using the Visa test card (dual card supporting PIN), swipe card through the magstripe reader.
- PINpad will now request for Account i.e. CHQ, SAV, or CRED. Please use the relevant one for the EFTPOS test card i.e. CHQ.
- PINpad will now display “ENTER PIN”. Enter the PIN to proceed.
- The transaction will now be processed and PINpad will display “PROCESSING NOW” on its screen.



- Once the transaction has been processed a Receipt (Response) message is returned. This notifies the POS system that a receipt is to be printed. For the scope of this example we will ignore it for now. Receipt (Response) messages will be discussed on page 18.

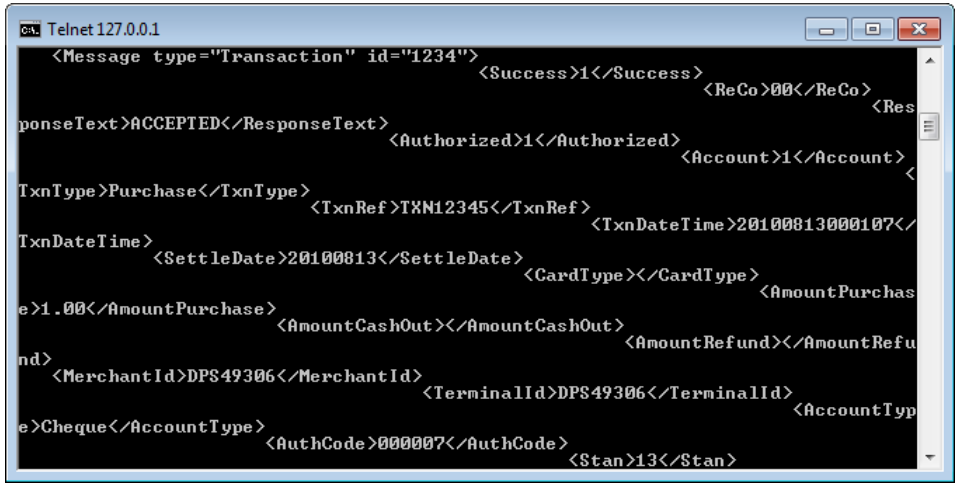
- Also at this stage a ClearDisplay (Response) message will be sent. This message notifies the POS system that it should remove any message that it was asked to display. Again for the scope of this example, we will ignore it for now. ClearDisplay (Response) message will be discussed on page 23.



- A transaction response will be returned (The PINpad screen will display “ACCEPTED”).

  The response should look like the following:

  ```
  <Message type="Transaction" id="1234">
  <ReCo>00</ReCo>
  <ResponseText>ACCEPTED</ResponseText>
  <Success>1</Success>
  <Account>1</Account>
  <TxnRef>TXN12345</TxnRef>
  <TxnType>Purchase</TxnType>
  <TxnDate>20060101</TxnDate>
  <TxnTime>120000</TxnTime>
  <SettleDate>20060102</SettleDate>
  <CardType>Visa</CardType>
  <AmountPurchase>1.00</AmountPurchase>
  <AmountCashOut>0.00</AmountCashOut>
  </Message>
  ```

  Again please note that this example does not use all available elements applicable for a transaction request. *A full list of applicable elements for the transaction request message can be found on our technical specifications page at:* *http://www.paymentexpress.com/technical_resources/integrated_eftpos/eftpos_xml_socket_interface.html*

- You have now successfully processed your first purchase transaction.

In the following sections you will find other common functions (e.g. Refund Transaction and Settlement Cutover) followed by more advanced functions (e.g. Tipping and Hospitality). Like above, all examples can be tested via CMD Prompt.

COMMON FUNCTIONS CONTINUED

Refund Transaction (Request/Response)

When the user wants to credit a customer for a previous transaction, a Refund message can be used.

DPS does not provide any authentication method for the refunds; however it is common practice for EFTPOS switches to provide refund/merchant cards for refund authentication. DPS is configured to support these refund/merchant cards by default. Any other authentication method will need to be managed by the POS.

The setting “EnableRefundMerchantCard” in pxpp_cfg must be set to “1” for the EFTPOS terminal to request for refund/merchant card during a refund transaction. If the setting is set to “0” the EFTPOS terminal will not request for the refund/merchant card.

Refund Transaction (Request)

This transaction request message notifies the EFTPOS terminal that the POS system wishes to initiate an EFTPOS refund transaction. The message data specifies the type of transaction and the amount to be refunded.

Example:

```
<Message type="Transaction" id="1234">
<TxnType>Refund</TxnType>
<TxnRef>TXN12345</TxnRef>
<AmountRefund>1.00</ AmountRefund>
</Message>
```

Refund Transaction (Response)

Once the refund transaction has been processed (or cancelled) a transaction message will be sent back to the POS application, indicating the outcome of the transaction.

Example:

```
<Message type="Transaction" id="1234">
<ReCo>00</ReCo>
<ResponseText>ACCEPTED</ResponseText>
<Success>1</Success>
<Account>1</Account>
<TxnRef>TXN12345</TxnRef>
<TxnType>Refund</TxnType>
<TxnDate>20060101</TxnDate>
<TxnTime>120000</TxnTime>
<SettleDate>20060102</SettleDate>
<CardType>Visa</CardType>
<AmountRefund>1.00</AmountRefund>
<AmountCashOut>0.00</AmountCashOut>
</Message>
```

In the example above, we are sending a request message to initiate a “Refund” transaction with the refund amount to be “1.00”. This is the amount that we wish to credit back to the customers card.

The response returned shows that the transaction has been processed successfully e.g. Success “1”, ResponseText “ACCEPTED”, and ReCo “00”. Response data can then be interpreted by the POS application.

CashOut Transaction (Request/Response)

Cash out can be used in conjunction with a standard Purchase transaction. In addition to the elements used in the purchase transaction example used earlier, the AmountCashOut element is also used.

Submit the desired cash out value in AmountCashOut. AmountPurchase will still be required, however this value can be submitted as “0.00” if the user wishes to do a cash out only i.e. without charging the customer.

Purchase Transaction (Request)

This transaction request message notifies the EFTPOS terminal that the POS system wishes to initiate an EFTPOS purchase transaction. The message data specifies the type of transaction and the amount to be processed.

Example:

```
<Message type="Transaction" id="1234">
<TxnType>Purchase</TxnType>
<TxnRef>TXN12345</TxnRef>
<AmountPurchase>20.00</AmountPurchase>
<AmountCashOut>5.00</AmountCashOut>
</Message>
```


Purchase Transaction (Response)

Once the transaction has been processed (or cancelled) a transaction message will be sent back to the POS application, indicating the outcome of the transaction.

Example:

```
<Message type="Transaction" id="1234">
<ReCo>00</ReCo>
<ResponseText>ACCEPTED</ResponseText>
<Success>1</Success>
<Account>1</Account>
<TxnRef>TXN12345</TxnRef>
<TxnType>Purchase</TxnType>
<TxnDate>20060101</TxnDate>
<TxnTime>120000</TxnTime>
<SettleDate>20060102</SettleDate>
<CardType>Visa</CardType>
<AmountPurchase>20.00</AmountPurchase>
<AmountCashOut>5.00</AmountCashOut>
</Message>
```

In the example above, we are sending a request message to initiate a “Purchase” transaction with cash out. We have requested the purchase amount to be “20.00” and cash out amount to be “5.00”.

The response returned shows that the transaction has been processed successfully i.e. Success “1”, ResponseText “ACCEPTED”, and ReCo “00”. Response data can then be interpreted by the POS application. The card holder’s card will be charged a total of 25.00, where 5.00 cash is given to the card holder from the till.

Settlement Cutover (Request/Response)

When the user wants to process an end-of-day settlement, a Cutover message can be used. This function is used to cut over the day's financial takings and print the expected settlement totals. These totals can then be used to reconcile the day's processing against the POS totals, merchant bank statements, and DPS Payline reports (more information on Payline reporting on page 9).

Please note that this process can only be done once a day, even on the test server. In the LIVE environment, the time/period where the user is able to process a settlement cutover will depend on the setting at their merchant bank. This will normally match the businesses closing time.

Settlement Cutover (Request)

Notifies the EFTPOS terminal that an end-of-day settlement should be performed. In multi-merchant setups, the merchant number must be supplied.

When the settlement cutover has completed, the POS system will receive a Settle message from the EFTPOS terminal.

Example:

```
<Message type="Cutover" id="1234">
<Account>1</Account>
</Message>
```

Settlement Cutover (Response)

Notifies the POS system that a settlement operation has completed. The message data specifies whether the cutover was successful or not.

When this message has been received, the settlement operation is complete and the POS system may continue with other operations.

Example:

```
<Message type="Cutover" id="1234">
<Success>1</Success>
<ReCo>97</ReCo>
<ResponseText>Transaction Completed</ResponseText>
<Account>1</Account>
</Message>
```

In the example above, we are sending a request message to initiate a "Cutover" on Account "1". Please note that if the account element is not present, the default merchant account will be used.

The response returned shows that the settlement cutover has been processed successfully e.g. Success "1", ResponseText "Transaction Completed", and ReCo "97". Response data can then be interpreted by the POS application.

If a settlement cutover has already been processed for the day, a declined message will be returned. An example of the ResponseText returned would be "This Transaction may only be performed once a day between 00:00 and 22:00".

Settlement Enquiry (Request/Response)
When the user wants to obtain a breakdown of their current financial takings, an Enquiry message can be used.

Settlement Enquiry (Request)

Notifies the EFTPOS terminal that a settlement enquiry operation should be performed. A receipt will be printed containing a breakdown of transactions that have been or can be settled for the given date. In multi-merchant setups the merchant number must be supplied.

When the settlement enquiry has completed the POS system will receive an Enquiry message from the EFTPOS terminal.

Example:

```
<Message type="Enquiry" id="1234">
<Account>1</Account>
<SettleDate>20100315</SettleDate>
</Message>
```


Settlement Enquiry (Response)

Notifies the POS system that an enquiry operation has completed. The message data specifies whether the enquiry was successful or not.

When this message has been received, the enquiry operation is complete and the POS system may continue with other operations.

Example:

```
<Message type="Enquiry" id="1234">
<Success>1</Success>
<ReCo>97</ReCo>
<ResponseText>Transaction Completed</ResponseText>
<Account>1</Account>
</Message>
```

In the example above, we are sending a request message to initiate an “Enquiry” on Account “1” for the 15th of March 2010. Please note that if the account element is not present, the default merchant account will be used.

The response returned shows that the Enquiry has been processed successfully e.g. Success “1”, ResponseText “Transaction Completed”, and ReCo “97”. Response data can then be interpreted by the POS application.

Receipt (Response)
This message is sent from the EFTPOS terminal every time a receipt is to be printed, even if the terminal is responsible for printing the receipts.

Receipt (Response)

Notifies the POS system that a receipt is to be printed. This message contains a digital copy of the data that would be printed on the receipt.

The EFTPOS terminal will send this message every time a receipt is to be printed, even if the terminal is responsible for printing the receipts.

Example:

```
<Message type="Receipt" id="1234">
<Receipt>THE VENDOR 1 MERCHANT LANE THE CITY *----------EFTPOS----------* TERMINAL 00000001 TRAN 000001 TIME 01JAN 12:00 ACCT CREDIT VISA 411111....1111 AUTHORISATION 000001 EXP 12/10 PURCHASE NZ$1.00 TOTAL NZ$1.00 ACCEPTED *----------*</Receipt>
</Message>
```

In the example above, the message sent provides the POS with the receipt data. If POS handles the printing, this receipt data can then be formatted and sent to printer. Please note that you will need to format, so that each line only has 30 characters.

LastReceipt (Request/Response)

When the user wants a duplicate copy of their receipts, the LastReceipt message can be used. A LastReceipt response is also sent every time a receipt is to be printed.

LastReceipt (Request)

Notifies the EFTPOS terminal that the POS system wishes to retrieve the last receipt.

When the last receipt retrieve has completed, the POS system will receive a LastReceipt message from the EFTPOS terminal.

Example:

```
<Message type="LastReceipt" id="1234">
</Message>
```


LastReceipt (Response)

Received in response to a LastReceipt message request.

The EFTPOS terminal will send this message every time a receipt is to be printed, even if the terminal is responsible for printing the receipts.

Example:

```
<Message type="LastReceipt" id="1234">
<Success>1</Success>
<ReCo>97</ReCo>
<ResponseText>Transaction Completed</ResponseText>
<Receipt>THE VENDOR 1 MERCHANT LANE THE CITY *----------EFTPOS----------* TERMINAL 00000001 TRAN 000001 TIME 01JAN 12:00 ACCT CREDIT VISA 411111....1111 AUTHORISATION 000001 EXP 12/10 PURCHASE NZ$1.00 TOTAL NZ$1.00 ACCEPTED *---------- *</Receipt>
</Message>
```

In the example above, we are sending a request message to initiate a “LastReceipt”.

The response returned shows that the LastReceipt has been processed successfully e.g. Success “1”, ResponseText “Transaction Completed”, and ReCo “97”. Receipt data can then be interpreted by the POS application.

LastTransaction (Request/Response)

When the user wants to obtain the transaction outcome details from the previous transaction, the LastTransaction message can be used.

LastTransaction (Request)

Notifies the EFTPOS terminal that the POS system wishes to retrieve the last transaction details.

Example:

```
<Message type="LastTransaction" id="1234">
</Message>
```


LastTransaction (Response)

Received in response to a LastTransaction request.

When this message has been received, the last transaction retrieve operation is complete and the POS system may continue with other operations.

Example:

```
<Message type="LastTransaction" id="1234">
<ReCo>00</ReCo>
<ResponseText>ACCEPTED</ResponseText>
<Success>1</Success>
<Account>1</Account>
<TxnRef>TXN12345</TxnRef>
<TxnType>Purchase</TxnType>
<TxnDate>20060101</TxnDate>
<TxnTime>120000</TxnTime>
<SettleDate>20060102</SettleDate>
<CardType>Visa</CardType>
<AmountPurchase>1.00</AmountPurchase>
<AmountCashOut>0.00</AmountCashOut>
</Message>
```

In the example above, we are sending a request message to initiate a “LastTransaction”.

The response returned shows that the LastTransaction has been processed successfully e.g. Success “1”, ResponseText “ACCEPTED”, and ReCo “00”. Response data can then be interpreted by the POS application.

Button (Request)

When the user has selected a button in response to a display message, the Button request message can be used to inform the EFTPOS terminal of the user's decision. The POS system will not receive any response to this message.

Button (Request)

Notifies the EFTPOS terminal that the POS user has hit a button in response to a Display message.

This message takes only a single parameter, which specifies which button was pressed.

The id attribute for this message MUST match the id attribute of the Display message that caused the button to be shown.

Example:

```
<Message type="Button" id="1234">
<Button>Yes</Button>
</Message>
```

In the example above, we are sending a Button request message to tell the EFTPOS terminal that a user has pressed the "Yes" button. A likely scenario for this would be when the user verifies a cardholder's signature and selects "Yes" on the dialog box to process the transaction by signature.

Display (Response)
When the EFTPOS terminal needs a message displayed on the POS for the user to view, a Display response message is used.

Display (Response)

Notifies the POS system that a message needs to be shown to the POS user.

The data inside this message indicates the message to be displayed, and whether there are any buttons that should also be displayed.

If buttons are to be displayed, the POS system must allow for the buttons to be pressed. When any such button is pressed, the POS should send a Button message to the EFTPOS terminal. In general, these messages are used to convey information that would normally be displayed on the operator's screen on a standalone EFTPOS terminal.

Example:

```
<Message type="Display" id="1234">
<Text1>SIGNATURE OK Y/N</Text1>
<Button1>Yes</Button>
<Button2>No</Button>
</Message>
```

In the example above, the EFTPOS terminal is telling the POS to display "SIGNITURE OK Y/N" to the user. It also specifies that there are 2 available buttons. Button 1 being "Yes" and Button 2 being "No". This is prompted to ask the user to verify the cardholder's signature when attempting to process a transaction using signature instead of PIN.

When the user selects one of the buttons above, the POS should be sending a Button request message to the EFTPOS Terminal. Refer to the previous page for details on the Button request message.

ClearDisplay (Response)
When the EFTPOS terminal wishes to clear the message it previous asked the POS to display, a ClearDisplay response message is used.

ClearDisplay (Response)

Notifies the POS system that it should remove any message that it was asked to display.

When the POS receives this message it should remove any message that the EFTPOS terminal has asked it to display. If such messages are displayed in a separate pop-up window, this window may be hidden.

Example:

```
<Message type="ClearDisplay" id="1234">
</Message>
```

In the example above, the EFTPOS terminal is telling the POS to remove the message it was asked to display in message sent earlier with ID “1234”.

SetIdlePrompt (Request)
When the PINpad is idle, the screen will display “EFTPOS”. This is the default display line for all setups. If you would like customize this, the SetIdlePrompt request message can be used.

SetIdlePrompt (Request)

This call is used to display the idle message of the Pinpad. The default is "EFTPOS", which can be changed by using the DisplayLine1 & DisplayLine2 input properties.

Example:

```
<Message type="IdlePrompt" id="1234">
<DisplayLine1>Swipe Staff</DisplayLine1>
<DisplayLine2>Card</DisplayLine2>
</Message>
```

In the example above, we are sending a SetIdlePrompt request message to tell the EFTPOS terminal to display “Swipe Staff Card” on the EFTPOS terminal’s screen when it is idle.

PinpadDisplay (Request)
When the POS would like a temporary message displayed on the EFTPOS terminal screen, the SetIdlePrompt request message can be used. Available preconfigured messages and their corresponding codes can be found on page 69.

PinpadDisplay (Request)
This call is used to display temporary messages on the PINpad for a certain amount of time.

Example:

```
<Message type="PinpadDisplay" id="1234">
<DisplayLine1>52</DisplayLine1>
<DisplayLine2> </DisplayLine2>
<EnableBackLight>1</EnableBackLight>
<DisplayTimeout>5</DisplayTimeout>
</Message>
```

In the example above, we are sending a PinpadDisplay request message to tell the EFTPOS terminal to display a message corresponding to 52 i.e. “TRAN UNAVAILABLE”. This will be displayed on the EFTPOS terminal’s screen for 5 seconds (default time is 3 seconds) with the screens backlight turned on.

Status (Response)
When the EFTPOS terminal status is changed, a Status response message is automatically sent to inform the POS.

Status (Response)
Notifies the POS system that the status of terminal has changed. This message contains the information of the EFTPOS terminal status.

Example:

```
<Message type="Status" id="1234">
<Ready>1</Ready>
<Description>Ready</Description>
</Message>
```

In the example above, the EFTPOS terminal is reporting its status as “Ready”.

ADVANCED FUNCTIONS

In addition to the common functions already discussed, more advanced functions can be used.

Tip Transaction - EditTender (Request/Response)

If a merchant would like to accept tips from its customers, the EditTender message can be used. Please note that Tipping is not enabled by default. If this functionality is not requested prior to receiving your PINpads, you will need to enable this in your EFTPOS configuration file i.e. pxpp_cfg. (Note: PXPP service must be restarted for changes to apply). Change the following in pxpp_cfg to enable Tipping.

…
<EnableCreditCardTipping>**0**</EnableCreditCardTipping>
<EnableHospitality>**0**</EnableHospitality>
…

…
<EnableCreditCardTipping>**1**</EnableCreditCardTipping>
<EnableHospitality>**1**</EnableHospitality>
…

Both EnableCreditCardTipping and EnableHospitality must be enabled. Also depending on the CAID/CATID assigned, you may need to contact the DPS Support team to make some changes.

EditTender (Request)

Notifies the EFTPOS terminal to edit amount for the transaction that matches the DpsTxnRef supplied.

Example:

```
<Message type="EditTender" id="1234">
<Account>1</Account>
<DpsTxnRef>0000000001</DpsTxnRef>
<TxnType>Tip</TxnType>
<Amount>10.00</Amount>
<Amount2>12.00</Amount2>
</Message>
```


EditTender (Response)

The response notifies the POS that the EFTPOS terminal has completed adding the Tip transaction.

Example:

```
<Message type="EditTender" id="1234">
<Success>1</Success>
<ReCo>00</ReCo>
<ResponseText>Accepted</ResponseText>
<Account>1</Account>
</Message>
```

In the above example we are sending a request message to initiate a “Tip” transaction for the initial transaction (DpsTxnRef “0000000001”). Along with the original DpsTxnRef, we also include the original amount of 10.00, and provide the new amount to be charged 12.00. This means that we have added a 2.00 tip onto the original transaction.

The response returned shows that the tip of 2.00 has been successfully added to the initial transaction (DpsTxnRef “0000000001”).

Hospitality Transaction - EditTender (Request/Response)

If a merchant would like to accept Hospitality transaction from its customers, the EditTender message can be used. Please note that Hospitality is not enabled by default. If this functionality is not requested prior to receiving your PINpads, you will need to enable this in your EFTPOS configuration file i.e. pxpp_cfg.txt. (Note: PXPP service must be restarted for changes to apply). Change the following in pxpp_cfg to enable Hospitality.

…
<EnableHospitality>**0**</EnableHospitality>
<EnableCreditCardTipping>**0**</EnableCreditCardTipping>
…

…
<EnableHospitality>1</EnableHospitality>
<EnableCreditCardTipping>**1**</EnableCreditCardTipping>
…

Both EnableHospitality and EnableCreditCardTipping must be enabled. Also depending on the CAID/CATID assigned, you may need to contact the DPS Support team to make some changes.

EditTender (Request)

Notifies the EFTPOS terminal to edit amount for the transaction that matches the DpsTxnRef supplied.

Example:

```
<Message type="EditTender" id="1234">
<Account>1</Account>
<DpsTxnRef>0000000001</DpsTxnRef>
<TxnType>Hospitality</TxnType>
<Amount>10.00</Amount>
<Amount2>12.00</Amount2>
</Message>
```


EditTender (Response)

The response notifies the POS that the EFTPOS terminal has completed adding the Hospitality transaction.

Example:

```
<Message type="EditTender" id="1234">
<Success>1</Success>
<ReCo>00</ReCo>
<ResponseText>Accepted</ResponseText>
<Account>1</Account>
</Message>
```

In the above example we are sending a request message to initiate a “Hospitality” transaction for the initial transaction (DpsTxnRef “0000000001”). Along with the original DpsTxnRef, we also include the original amount of 10.00, and provide the new amount to be charged 12.00. This means that we have added a 2.00 hospitality total onto the original transaction.

The response returned shows that 2.00 is successfully added to the initial transaction (DpsTxnRef “0000000001”).

TipSettlement (Request/Reponse)
When the user wishes to finalize and upload pending Tip or Hospitality transactions, TipSettlement can be used.

TipSettlement (Request)

Notifies the EFTPOS terminal to finalize the pending tipping transaction.

Example:

```
<Message type="TipSettlement" id="1234">
<Account>1</Account>
</Message>
```


TipSettlement (Response)

Notifies the POS system that a tip settlement operation has completed

Example:

```
<Message type="TipSettlement" id="1234">
<Success>1</Success>
<ReCo>00</ReCo>
<ResponseText>Accepted</ResponseText>
<Account>1</Account>
</Message>
```

In the above example we are sending a request message to tell the EFTPOS terminal to finalize and upload any pending tip transactions stored against Account 1.

The response returned shows that the TipSettlement has been processed successfully i.e. ResponseText “Accepted”, and ReCo “00”.

The ResponseText “NO OFFLINE TRANS TO UPLOAD” will be returned if there are no pending EOV transactions.

PrintPendingReceipt (Request/Reponse)

When the user wishes to print a copy (receipt) of current pending Tip or Hospitality transactions, the PrintPendingReceipt message can be used.

PrintPendingReceipt (Request)

Notifies the EFTPOS terminal to prints the pending Tip or Hospitality receipt.

Example:

```
<Message type="PendingReceipt" id="1234">
<Account>1</Account>
<ReceiptAutoPrint>1</ReceiptAutoPrint>
<DpsTxnRef>0000000001</DpsTxnRef>
<TxnType>Tip</TxnType>
</Message>
```


PrintPendingReceipt (Response)

Notifies the POS system that a PrintPendingReceipt operation has completed

Example:

```
<Message type="PendingReceipt" id="1234">
<Success>1</Success>
<ReCo>97</ReCo>
<ResponseText>Transaction Completed</ResponseText>
<Receipt>
…receipt data…
</Receipt>
</Message>
```

In the above example we are sending a request message to tell the EFTPOS terminal to print all pending tip transactions stored against Account 1.

The response returned shows that the PrintPendingReceipt has been processed successfully i.e. ResponseText “Accepted”, and ReCo “97”, and the receipt data is available for printing.

Auth/Complete Transaction

The Auth/Complete transaction model is a two-stage process. The first stage involves processing an Auth type transaction (a.k.a Pre-Auth) to reserve the funds and the second stage involves processing an EditTender to finalize the funds transfer.

Stage 1: Transaction (TxnType = Auth)

An Auth transaction type verifies that funds are available for the requested card and reserves the specified amount. The amount is reserved for a period of 5-10 days (depending on merchant bank) before it clears. No funds are transferred. If you do not wish to take money from the customer you do nothing, and the reservation on the money will expire after 5-10 days. The message type used for this stage is “Transaction”.

Example:

Transaction (Request)

```
<Message type="Transaction" id="1">
<TxnType>Auth</TxnType>
<TxnRef>987654321</TxnRef>
<AmountPurchase>1.00</AmountPurchase>
</Message>
```


Transaction (Response)

```
<Message type="Transaction" id="1">
<Success>1</Success>
<ReCo>00</ReCo>
<ResponseText>ACCEPTED</ResponseText>
<Authorized>1</Authorized>
<TxnType>Auth</TxnType>
<TxnRef>987654321</TxnRef>
<TxnDateTime>20110209155713</TxnDateTime>
<SettleDate>20110209</SettleDate>
<CardType>Visa</CardType>
<AmountPurchase>1.00</AmountPurchase>
<MerchantId>18833333333</MerchantId>
<TerminalId>33333333</TerminalId>
<AccountType>Credit</AccountType>
<CardHolderName>TEST1 DPS</CardHolderName>
<ExpiryDate>0910</ExpiryDate>
<CardNumber>4999999900000009</CardNumber>
<DpsTxnRef>33333333000004</DpsTxnRef>
</Message>
```

In the above example we are sending a request message to initiate an “Auth” transaction for the amount of 1.00.

The response returned shows that the Auth transaction was successfully processed and the amount of 1.00 has been reserved on the customer’s credit card. The POS application should store the DpsTxnRef as it will be required to finalize the transaction.

Stage 2: EditTender (TxnType = AuthComplete)

An AuthComplete transaction type is used to finalize the funds transfer process. Referencing to the original Auth transaction is done using the DpsTxnRef and the final amount is submitted in “Amount2”. The message type used is “EditTender”.

Example:

EditTender (Request)
Notifies the EFTPOS terminal to edit amount for the transaction that matches the DpsTxnRef supplied.

```
<Message type="EditTender" id="2">
<Account>1</Account>
<DpsTxnRef>33333333000004</DpsTxnRef>
<TxnType>AuthComplete</TxnType>
<Amount2>1.00</Amount2>
</Message>
```


EditTender (Response)
The response notifies the POS that the EFTPOS terminal has completed adding the Hospitality transaction.

```
<Message type="EditTender" id="2">
<Success>1</Success>
<ReCo>00</ReCo>
<ResponseText>ACCEPTED</ResponseText>
<Account>1</Account>
</Message>
```

In the example above we send an EditTender request message to initiate an AuthComplete transaction for the amount of 1.00. We reference the original Auth transaction by using the DpsTxnRef.

The Amount2 sent in the AuthComplete transaction can be less, more, or identical to the original Amount specified in the original Auth transaction.

The response returned shows that our request was successful. The software will then send the transaction to the banking switch to finalize the funds transfer.

Notes:

- If the AuthComplete transaction is sent when the funds are still reserved (i.e. within the reservation period) and Amount2 is identical or less than the original Auth amount, then the funds are guaranteed to be available for the merchant to charge.
- If Amount2 is more than the original Auth amount then the amount up to the original Auth amount is guaranteed, however the additional amount is not.
- If the AuthComplete transaction is sent when funds are not reserved (i.e. after the reservation period) the transaction can still be processed, however the funds are not guaranteed.
- One AuthComplete transaction can be done per Auth transaction. If a Complete transaction is processed for a lesser amount than the original Auth amount, the remaining balance is no longer reserved.

EovSettlement (Request/Reponse)
When there are pending EOV transactions awaiting to be uploaded, the EovSettlement message can be used.

EovSettlement (Request)

Notifies the EFTPOS terminal to upload and settle any pending EOV transactions.

Example:

```
<Message type="EovSettlement" id="1234">
<Account>1</Account>
</Message>
```

EovSettlement (Response)

Notifies the POS system that an EOV settlement operation has completed

Example:

```
<Message type="EovSettlement" id="1234">
<Success>1</Success>
<ReCo>00</ReCo>
<ResponseText>Accepted</ResponseText>
<Account>1</Account>
</Message>
```

In the above example we are sending a request message to tell the EFTPOS terminal to upload all pending EOV transactions stored against Account 1.

The response returned shows that the EovSettlement has been processed successfully i.e. ResponseText “Accepted”, and ReCo “00”, meaning all previously pending EOV transactions have now been uploaded.

The ResponseText “NO OFFLINE TRANS TO UPLOAD” will be returned if there are no pending EOV transactions.

SubTotalEnquiry (Request/Reponse)
This feature can be used to keep track of shift totals without effecting the financial settlement.

A running total of all purchase and refund transactions are kept by the EFTPOS client. This running total can be accessed by using the message type 'SubTotalEnquiry'. A receipt will be printed which includes the purchase and refund totals since the last 'SubTotal' or 'Cutover' was performed. The totals will not be cleared i.e. totals are accumulated.

Example: Three staff members each work a different shift in a store i.e. morning shift, afternoon shift, night shift. Once each staff member finishes their shift, a SubTotalEnquiry is processed.

	Morning shift	Afternoon shift	Night shift
Sales Total (per shift)	$80	$100	$50
SubTotalEnquiry prints	$80	$180	$230

Usually at the end of day, users will process a settlement “Cutover”. This will cut over all of the day’s takings and reset the total for the next day. Therefore the next time a SubTotalEnquiry is used the total will be calculated from the settlement cutover.

SubTotalEnquiry (Request)

Notifies the EFTPOS terminal to print the total of all purchase and refund transactions since the last 'SubTotal' or 'Cutover'.

Example:

```
<Message type="SubTotalEnquiry" id="1234">
<Account>1</Account>
</Message>
```

SubTotalEnquiry (Response)

Notifies the POS system that a SubTotalEnquiry operation has completed. When this message has been received, the enquiry operation is complete and the POS system may continue with other operations.

Example:

```
<Message type="SubTotalEnquiry" id="1234">
<Success>1</Success>
<ReCo>90</ReCo>
<ResponseText>READY TO SETTLE</ResponseText>
<Account>1</Account>
</Message>
```

In the above example we are sending a request message to tell the EFTPOS terminal to print a receipt with the totals of all purchase and refund transactions stored against Account 1 since the last 'SubTotal' or 'Cutover'.

The response returned shows that the SubTotalEnquiry has been processed successfully i.e. ResponseText “READY TO SETTLE”, and ReCo “90”.

SubTotal (Request/Reponse)
Like the SubTotalEnquiry, this function can be used to keep track of shift totals without effecting the financial settlement.

A running total of all purchase and refund transactions are kept by the EFTPOS client. This running total can be accessed by using the message type 'SubTotal'. A receipt will be printed which includes the purchase and refund totals since the last 'SubTotal' or 'Cutover' was performed. However unlike the SubTotalEnquiry, the shift totals will be cleared.

Example: Three staff members each work a different shift in a store i.e. morning shift, afternoon shift, night shift. Once each staff member finishes their shift, a SubTotal is processed.

	Morning shift	Afternoon shift	Night shift
Sales Total (per shift)	$80	$100	$50
SubTotal prints	$80	$100	$50

SubTotal (Request)

Notifies the EFTPOS terminal to print the total of all purchase and refund transactions since the last 'SubTotal' or 'Cutover'.

Example:

```
<Message type=" SubTotal" id="1234">
<Account>1</Account>
</Message>
```

SubTotal (Response)

Notifies the POS system that a SubTotalEnquiry operation has completed. When this message has been received, the enquiry operation is complete and the POS system may continue with other operations.

Example:

```
<Message type="SubTotal" id="1234">
<Success>1</Success>
<ReCo>90</ReCo>
<ResponseText>READY TO SETTLE</ResponseText>
<Account>1</Account>
</Message>
```

In the above example we are sending a request message to tell the EFTPOS terminal to print a receipt with the totals of all purchase and refund transactions stored against Account 1 since the last 'SubTotal' or 'Cutover'.

The response returned shows that the SubTotal has been processed successfully i.e. ResponseText “READY TO SETTLE”, and ReCo “90”.

Gift/Loyalty Cards – ReadCard (Request/Response)

Gift/Loyalty cards can be read via the EFTPOS terminal using ReadCard. ReadCard should only be used to retrieve the track data from loyalty cards and other non-payment cards.

The information returned in this message contains the raw data from the ISO 7813 magnetic strip on the card. This data includes the start sentinel and end sentinel characters for each track. This data can then be interpreted by the POS.

A message from our message prompt lookup table can be displayed in Text1 and Text2. Available messages can be found on page 69.

ReadCard (Request)

Notifies the EFTPOS terminal that the POS system wishes to read the track data from a card.

Example:

```
<Message type="ReadCard" id="1234">
<Text1>155</Text1>
<Text2></Text2>
<EnableBackLight>1</EnableBackLight>
</Message>
```

ReadCard (Response)

The response notifies the POS that Read/Card operation has completed.

Example:

```
<Message type="ReadCard" id="1234">
<Success>1</Success>
<ReCo>97</ReCo>
<ResponseText>Transaction Completed</ResponseText>
<Track1>%B4111111111111111^PUBLIC/JOHN Q./MR^1012201?</Track1>
<Track2>;4111111111111111=1012201?</Track2>
<Track3></Track3>
</Message>
```

In the example above, we are sending a ReadCard request message to tell the EFTPOS terminal to begin the ReadCard function. The message also specifies the PINpad to display message 155 (“Swipe Gift Card”) on the screen with the backlight turned on.

The response returned shows that the ReadCard was a success i.e. Success “1”, ResponseText “Transaction Completed”, and ReCo “97”. The track data can now be used by the POS.

Fuel Transaction (Request/Response)

Fuel transactions include authorisation and completion (financial advice) type transactions. This is because we may not always know how much fuel the vehicle may require to fill up, or to prevent drivers from filling up their vehicles and driving away without paying.

Please note that Fuel capability is not enabled by default. If this functionality is not requested prior to receiving your PINpads, you will need to enable this in your EFTPOS configuration file i.e. pxpp_cfg.txt. (Note: PXPP service must be restarted for changes to apply). Change the following in pxpp_cfg to enable Fuel capabilities.

…
<EnableFuelDispensing>**0**</EnableFuelDispensing>
…

→

…
<EnableFuelDispensing>**1**</EnableFuelDispensing>
…

Authorisation Phase: To start your Fuel transaction you can use the Transaction request message with the TxnType set to Fuel.

A prompt will allow you to Select Pump Number and then display the amount with PUMP X READY TO DISPENSE UP TO $50.00 or PUMP X READY FOR DELIVERY if the authorisation amount is greater than $50.00. Normal account selection and PIN entry will follow.

Fuel Transaction (Request)

Notifies the EFTPOS terminal that the POS system wishes to initiate a Fuel transaction.

Example:

```
<Message type="Transaction" id="1234">
<Account>1</Account>
<TxnType>Fuel</TxnType>
<TxnRef>TXN12345</TxnRef>
<AmountPurchase>50.00</AmountPurchase>
</Message>
```

Fuel Transaction (Response)

The response will return a DpsTxnRef to be used for the corresponding completion.

Example:

```
<Message type="Transaction" id="1234">
<ReCo>00</ReCo>
<ResponseText>ACCEPTED</ResponseText>
<Success>1</Success>
<Account>1</Account>
<TxnRef>TXN12345</TxnRef>
<TxnType>Fuel</TxnType>
<DpsTxnRef>00000053000014</DpsTxnRef>
<CardType>Visa</CardType>
<AmountPurchase>50.00</AmountPurchase>
</Message>
```

In the above example, we are sending a Fuel transaction request of 50.00 to the EFTPOS terminal. The response returned shows that the transaction was a success i.e. Success “1”, ResponseText “ACCEPTED”, and ReCo “00”. It also provides the DpsTxnRef required for the completion phase.

Completion Phase: When the Fuel transaction needs to be completed, EditTender is used. Set TxnType to Fuel, provide the DpsTxnRef returned from the first transaction and include the original auth amount and the actual/final amount. Please note that each fuel authorisation transaction can only be completed once.

From the example on the previous page, the DpsTxnRef returned was “00000053000014”.

EditTender (Request)

Notifies the EFTPOS terminal to edit amount for the transaction that matches the DpsTxnRef supplied.

Example:

```
<Message type="EditTender" id="1234">
<Account>1</Account>
<DpsTxnRef>00000053000014</DpsTxnRef>
<TxnType>Tip</TxnType>
<Amount>50.00</Amount>
<Amount2>50.00</Amount2>
</Message>
```

EditTender (Response)

The response notifies the POS that the EFTPOS terminal has processed the completion.

Example:

```
<Message type="EditTender" id="1234">
<Success>1</Success>
<ReCo>00</ReCo>
<ResponseText>Accepted</ResponseText>
<Account>1</Account>
</Message>
```

In the above example, we are sending the EditTender message to the EFTPOS terminal to tell it to complete the fuel authorization transaction done earlier. We include the original amount of 50.00 (Amount) and the final/actual amount of 50.00 (Amount2). If the value of the fuel brought was greater or less than the original amount of 50.00, then the value in Amount2 would be different.

The response returned shows that the transaction was a success i.e. Success “1”, ResponseText “Accepted”, and ReCo “00”. The fuel transaction is now complete.

EnterData (Request/Response)

EnterData is mainly used with vending and fuel POS. This function can be used to retrieve additional data from the user. EnterData tells the EFTPOS terminal to display a message, which requests for user input. The user will then enter the data using the EFTPOS terminal, and data is sent back to POS for interpretation. Plain Text can be masked as well to be displayed on the PINpad e.g. input masked as user is keying in the data.

EnterData (Request)

Notifies the EFTPOS terminal that the POS system wishes to display a message from the message prompt table, which requires user input.

Example:

```
<Message type="EnterDataEvent" id="1234">
<EnableBackLight >1</EnableBackLight>
<InputMask>*</InputMask >
<KeyMask>FFF819</KeyMask >
<EnterMode></EnterMode>
<MaxLength></MaxLength>
<DisplayLine1>87</DisplayLine1>
<DisplayLine1></DisplayLine1>
</Message>
```


EnterData (Response)

The response notifies the POS that EnterData operation has completed.

Example:

```
<Message type="EnterData" id="1234">
<Account>1</Account>
<Success>1</Success>
<ReCo>00</ReCo>
<ResponseText>1</ResponseText>
<Result>1000</Result>
</Message>
```

In the example above, we are sending an EnterData request message to tell the EFTPOS terminal to display message corresponding to code 87 i.e. “PUMP OFF LINE”.

The response returned shows that the EnterData was a success i.e. Success “1”, ResponseText “1”, and ReCo “00”. The data in Result informs what keys were pressed during this process. Result will include character or string of characters if multiple buttons were pushed. This can then be interpreted by POS.

GetStatus (Request/Response)

When the POS would like a list of available and current EFTPOS settings, GetStatus request can be used. This may be useful for advanced exception handling and troubleshooting.

GetStatus (Request)

Requests a list of EFTPOS settings.

Example:

```
<Message type="GetStatus" id="1234">
</Message>
```

GetStatus (Response)

Returns current EFTPOS settings to the POS.

Example:

```
<Message type="GetStatus" id="1234">
<Ready>1</Ready>
<Description>Ready</Description>
<ReadyPinPad>1</ReadyPinPad>
<ReadyLink>1</ReadyLink>
<EovEnabled>1</EovEnabled>
<EovOffline>0</EovOffline>
<UplinkDetails>
<UplinkDetails>
<Interface>
<Name>EFT6</Name>
<Address>eft6.paymentexpress.com</Address>
<Port>61</Port>
<Enabled>1</Enabled>
<Ready>1</Ready>
<Gprs>0</Gprs>
<RTTExceeded>0</RTTExceeded>
</Interface>
```

In the example above, we are sending a GetStatus request message to gather all current settings from the EFTPOS terminal.

The response will contain a long list of data (*example above is only a portion of actual response*). The example response above shows that the EFTPOS terminal is currently ready and connected to EFT6 (Primary server) via port 61. Other useful information returned by this request is: EOV status, PINpad Info, Merchant Number, Terminal ID, Printer Name, daily refund processed total, currency symbol, and manpan status etc. *A full list of applicable elements for the GetStatus request message can be found on our technical specifications page at:*
http://www.paymentexpress.com/technical_resources/integrated_eftpos/eftpos_xml_socket_interface.html

Enable & Disabling Account Selection Options
When the POS would like to prevent a user from selecting a particular account on the PINpad (i.e. CHQ/SAV/CRED) the following elements can be used.

EnableCreditAccount | *Datatype: BOOL (Max 1 Bytes)*
EnableCheckAccount | *Datatype: BOOL (Max 1 Bytes)*
EnableSavingAccount | *Datatype: BOOL (Max 1 Bytes)*

To disable an account option, the POS will set the required element to “0”. The default value is “1” i.e. enabled.

Purchase Transaction (Request)

Example:

```
<Message type="Transaction" id="1234">
<TxnType>Purchase</TxnType>
<TxnRef>TXN12345</TxnRef>
<EnableCreditAccount>0</EnableCreditAccount>
<AmountPurchase>1.00</AmountPurchase>
</Message>
```

The example above disables the Credit account option. The user will be prompted with “ACCT NOT ALLOWED” if the CRD option is selected.

MANUAL PAN

Whether the customer is providing their details over the phone, over the internet, or simply because the card cannot be read, certain scenarios require the manual entry of the full credit card number. The Manual PAN functionality fills this requirement and is available on both Payment Express® ActiveX and XML Socket Interfaces.

The DPS EFTPOS software is integrated with the PINpad to allow sensitive card data to be entered directly onto the PINpad itself. This relieves the POS application from having to deal with sensitive data. The following section provides instructions for setup and integration.

Requirements:
PINpad Hardware: VeriFone SC5000 or Ingenico i3070
PINpad BIOS: 1228 (minimum)
PXPP software version: v2.9.4.2 (minimum)

Setup:
The following must be setup in pxpp_cfg.txt.

```
<EnablePinPadEnterPAN>1</EnablePinPadEnterPAN>
<EnableManPANEciPrompt>0</EnableManPANEciPrompt>
<ManPANEci>67</ManPANEci>
<ManPANCscIndicator>0</ManPANCscIndicator>
```

HOW IT WORKS

The steps below provide an overview of how the Manual PAN functionality works. To progress from step to step, action from the POS or the User is required. Even through the message prompts are displayed on the PINpad screen, the XML interface will still send a *Display* message for the POS to interpret. How the POS wishes to handle these display messages is up to the vendor; however DPS recommend that all messages from the PINpad are interpreted and not simply ignored *(note: display messages are shown as "DisplayMessage #" below, full description of each message can be found on page 52).*

1. POS submits a standard transaction request.

XML Example:

```
<Message type="Transaction" id="1234">
<TxnType>Purchase</TxnType>
<TxnRef>1234567890</TxnRef>
<AmountPurchase>1.00</AmountPurchase>
</Message>
```

---

2. PINpad screen shows "SWIPE CARD - OR INSERT CARD"

POS UI to display "SWIPE CARD OR INSERT CARD" with two available buttons (Cancel, Manual). (DisplayMessage #1)

Screen display:

```
SWIPE CARD
OR INSERT CARD
```

---

3. POS submits "Manual" button.

Alternatively the user can press the "Back" button (yellow arrow) on the PINpad instead.

XML Example:

```
<Message type="Button" id="1234">
<Button>Manual</Button>
</Message>
```

---

4. PINpad displays "ENTR CARD NUMBER"

POS UI to display "ENTER CARD NUMBER ON PINPAD" with one available button (Cancel). (DisplayMessage #2)

Screen display:

```
ENTR CARD NUMBER
```

---

5. User enters full credit card number using keypad and presses the enter key.

e.g. 4111111111111111 -> enter Key

---

6. PINpad displays "ENTR EXPIRY DATE - _ _/ _ _ (MM/YY)"

POS UI should still display "DisplayMessage #2"

Screen display:

```
ENTR EXPIRY DATE
_ _/_ _    (MM/YY)
```

7.	User enters credit card expiry date using keypad and presses the enter key.	e.g. 1012 -> enter key
8.	PINpad displays "CSC ON CARD – NO YES". POS UI to display “CSC ON CARD Y/N?” (DisplayMessage #3)	Screen display: CSC ON CARD NO YES
9.	User selects “YES”. If User selects “NO” skip to step 12.	
10.	PINpad displays “ENTER CARD CVC” POS UI to display “PLEASE ENTER CVC” with one available button (Cancel). (DisplayMessage #4)	Screen display: ENTER CARD CVC
11.	User enters CVC using keypad and presses the enter key	e.g. 123 -> enter key

12. PINpad displays “ACCOUNT $X.XX CHQ SAV CRD” (Account selection screen). Transaction can now be processed following the standard transaction process i.e. select appropriate account and process using SIG or PIN.

POS UI#1

<Message type="Display" id="1234"> <Text1>SWIPE CARD</Text1> <Text2>OR INSERT CARD</Text2> <Button1>Cancel</Button1> <Button2>Manual</Button2> <Button3></Button3> <Beep>0</Beep> </Message>	Description: This is your standard display “SWIPE CARD OR INSERT CARD” message. Buttons: Cancel – Cancels the transaction process. Manual – Starts Manual PAN.

POS UI#2

<Message type="Display" id="1234"> <Text1>PLEASE ENTER CARD NUMBER</Text1> <Text2>ON PINPAD</Text2> <Button1>Cancel</Button1> <Button2></Button2> <Button3></Button3> <Beep>0</Beep> </Message>	Description: This display message is sent when the PINpad is waiting for the user to enter the credit card details into the PINpad. The POS should display this message in their UI to let the user know that they need to enter using the PINpad. Button: Cancel – Cancels the transaction process.

POS UI#3

<Message type="Display" id="1234"> <Text1>CSC ON CARD Y/N?</Text1> <Text2></Text2> <Button1>Yes</Button1> <Button2>No</Button2> <Button3></Button3> <Beep>1</Beep> </Message>	Description: This display message is sent when the PINpad is waiting for the user to confirm if they have a CSC/CVC or available or not. The POS should display this message in their UI to let the user know that they need to enter using the PINpad. Buttons: Yes – Selected if CSC is available No – Selected if CSC is not available

POS UI#4

<Message type="Display" id="1234"> <Text1>PLEASE ENTER CVC</Text1> <Text2></Text2> <Button1>Cancel</Button1> <Button2></Button2> <Button3></Button3> <Beep>0</Beep> </Message>	Description: This display message is sent when the PINpad is waiting for the user to enter the CSC/CVC. The POS should display this message in their UI to let the user know that they need to enter using the PINpad. Button: Cancel – Cancels the transaction process.

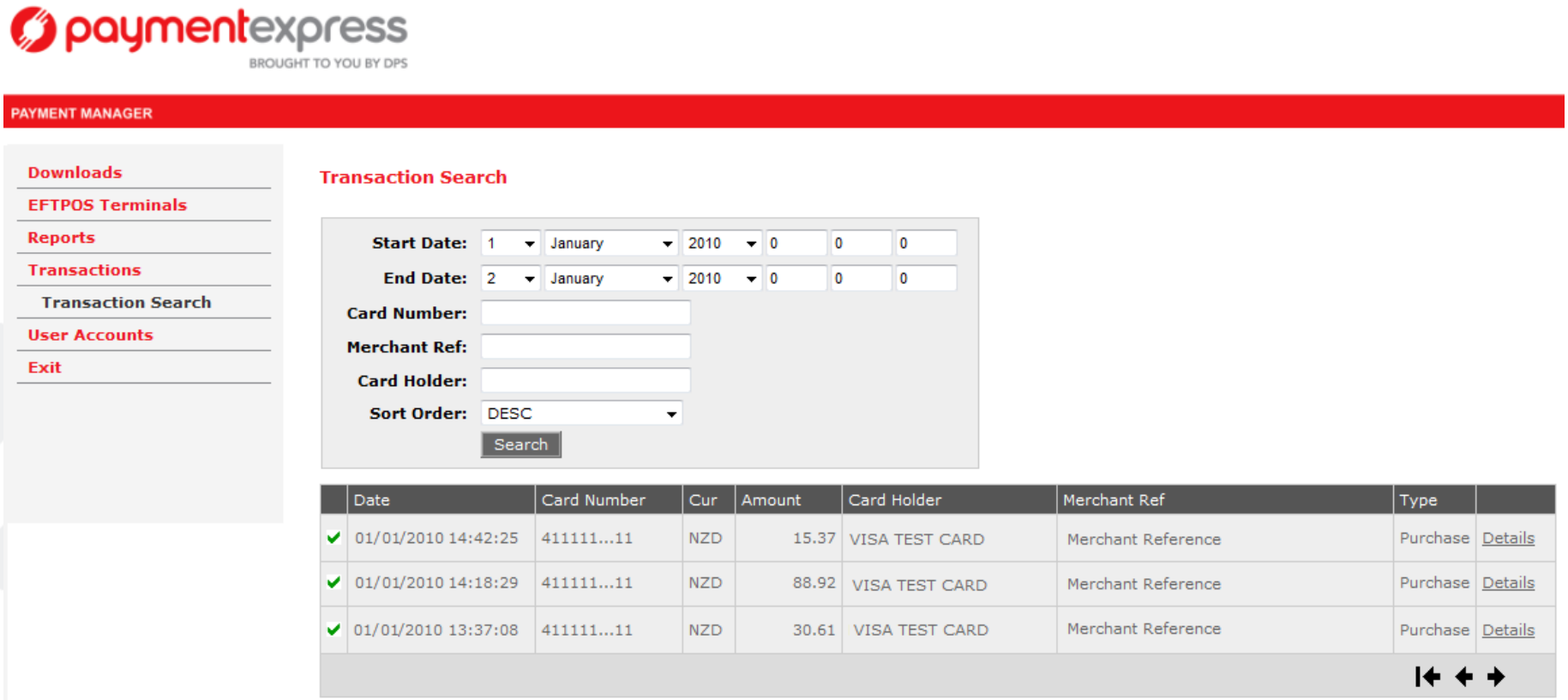
PAYLINE®

All development accounts are setup with a Payline® account. Payline is a web-based management facility which allows the user to monitor transactions and PINpad status in real-time and generate detailed EFTPOS reports. This service can be accessed from anywhere with internet access.

To access your Payline account, go to https://www.paymentexpress.com/pxmi/logon and enter the username and password issued in the activation email.

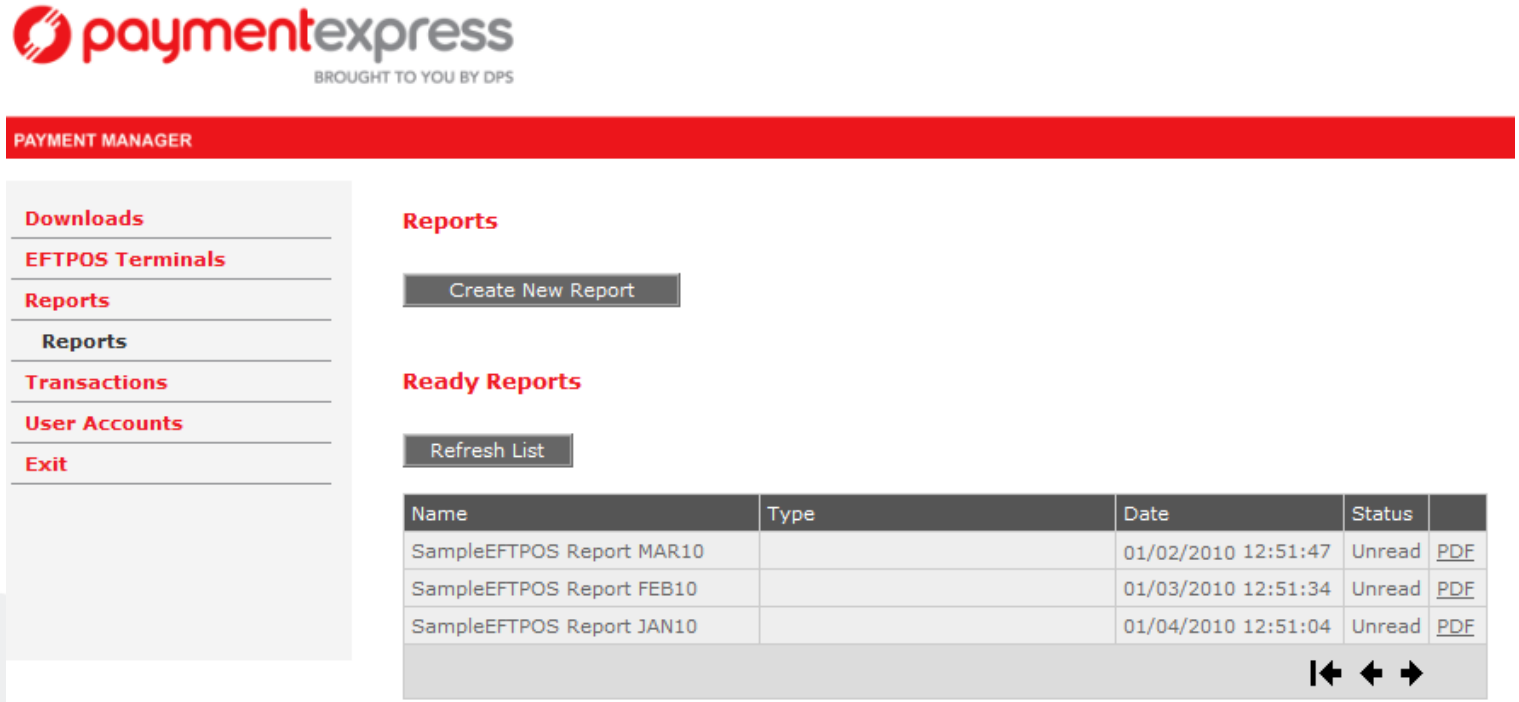
Transaction Search screen

This screen allows the user to actively monitor EFTPOS transactions in real-time. The user can also search for past transactions using the date, full card number, card holder name and merchant reference. TIP: Sending unique reference values in the “MerchantReference” free text field can be used for tracking transactions in Payline and generated DPS reports.



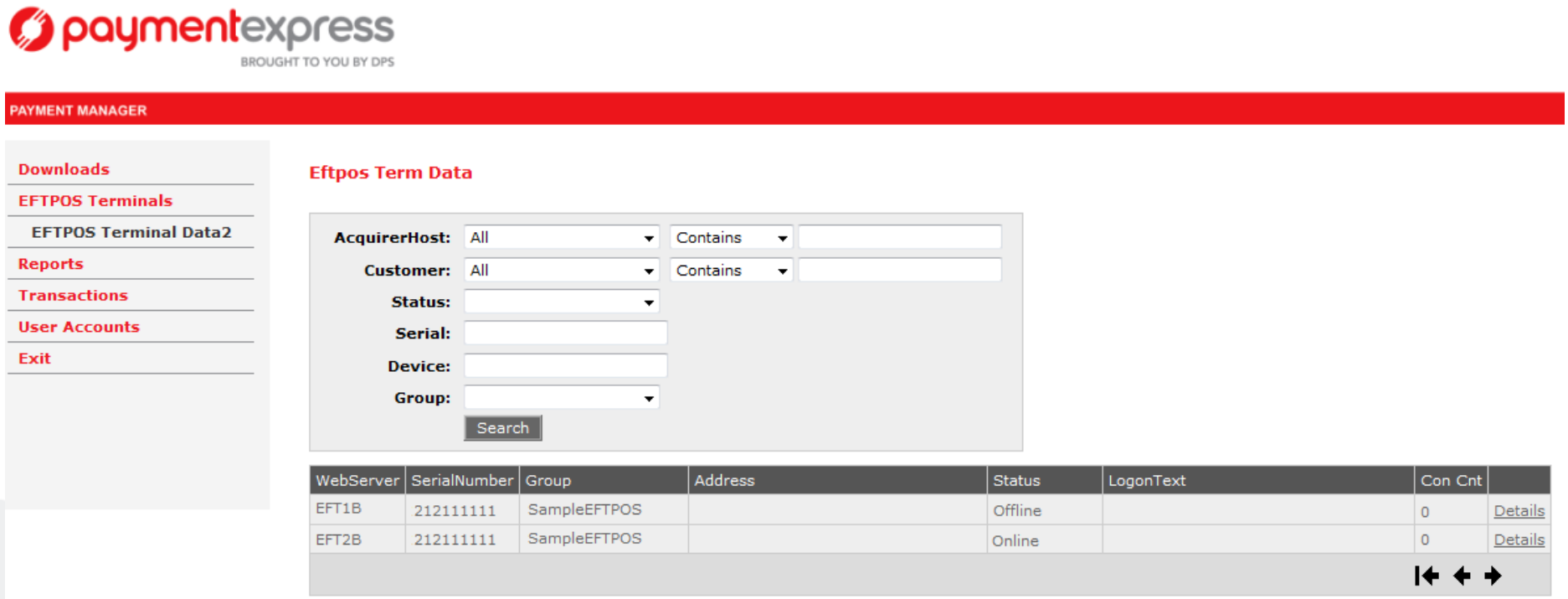
Reports screen

Users can use the Reports screen to generate standard and settlement EFTPOS reports. Reports can be generated in CSV or PDF formats. All reports are stored in the “Ready Reports” list for easy access.



EFTPOS Terminal Data screen

This screen allows the user to monitor all PINpads allocated to their group. The screen provides key information on the PINpads current status (i.e. online/offline), serial number, and the server connected to.



More information on Payline® can be found in the Payline user found at: http://www.paymentexpress.com/downloads/PaymentManager.pdf

ELECTRONIC OFFLINE VOUCHER (EOV)

Electronic Offline Voucher (EOV) is a failover system which allows merchants to store transactions securely in offline mode should the connection to the banking network become temporarily unavailable. These transactions are processed once the connection to the banking network has been restored.

Once in EOV mode the transaction details will be stored securely until the connection to the banking network has been restored. Once the connection has been restored the transaction information will be automatically uploaded to the banking network or uploaded manually by using the upload EOV function on the Payment Express client software. When the transactions are uploaded a receipt will be printed for each EOV transaction which was not completed.

- The terminal will only accept one EOV transaction per card.
- The terminal will prompt merchants to ensure customers sign the merchant copy of the receipt for every transaction.
- The merchant may set a floor limit for EOV transactions at their bank, and the terminal will prompt staff when a transaction exceeds this limit. No limit set on test PINpads by default.
- The terminal will allow only purchase transactions to be processed in EOV mode.

EOV support is enabled on all PINpads by default.

How to test EOV

1) Stop DPS communication service i.e. “Payment Express EFTP" - Uplink Line driver. This will simulate an internet connectivity issue.
2) Process 2 manual login attempts (e.g. DPS EFTPOS Client > Maint > Logon). Note: these will fail and timeout after about 45 seconds each. After the 2nd failed logon attempt the client should jump into “EFTPOS OFFLINE”.
3) You can now process EOV transaction tests.
4) To get the client back online start the DPS communication service i.e. “Payment Express EFTP" - Uplink Line driver.
5) Process a manual login to confirm that the client is back online.

Notes

- The same test cards used for online testing can be used in EOV testing.
- There are two ways to get back Online from EOV.
  - Manual log on i.e. via DPS EFTPOS Client.
  - Auto log on processed by the DPS software every 10 minutes.
- Pending EOV transactions will be automatically uploaded when EFTPOS software comes back online (regardless of Manual or Auto log on).
- Maximum amount of EOV transactions allowed is 99 (default). Please be aware that for LIVE terminals, this value is determined by the merchant bank.
- For live terminals, Debit, Visa, MasterCard and Diners are all supported in EOV mode. This will also need to be setup with the merchant bank. AMEX is not currently supported.

An overview of our EOV service can be found here: http://www.paymentexpress.com/downloads/EOV%20Guide.pdf

EXCEPTION HANDLING

The POS application must be designed to handle the occurrence of exceptions and special conditions that can alter the flow of normal EFTPOS processing.

Financial integrity goes beyond ensuring that the POS can correctly interpret the transaction result returned by the EFTPOS terminal. The POS also needs to take suitable steps to determine the result of a transaction which had no response. Every transaction has a point of no return where payment has been authorised with the card issuer, but the result has not yet been returned to the POS application.

The POS application also needs manage irregularities with messages resulting from the transport layer. A POS which is unable to deal with these irregularities can fail to correctly act upon the result message.

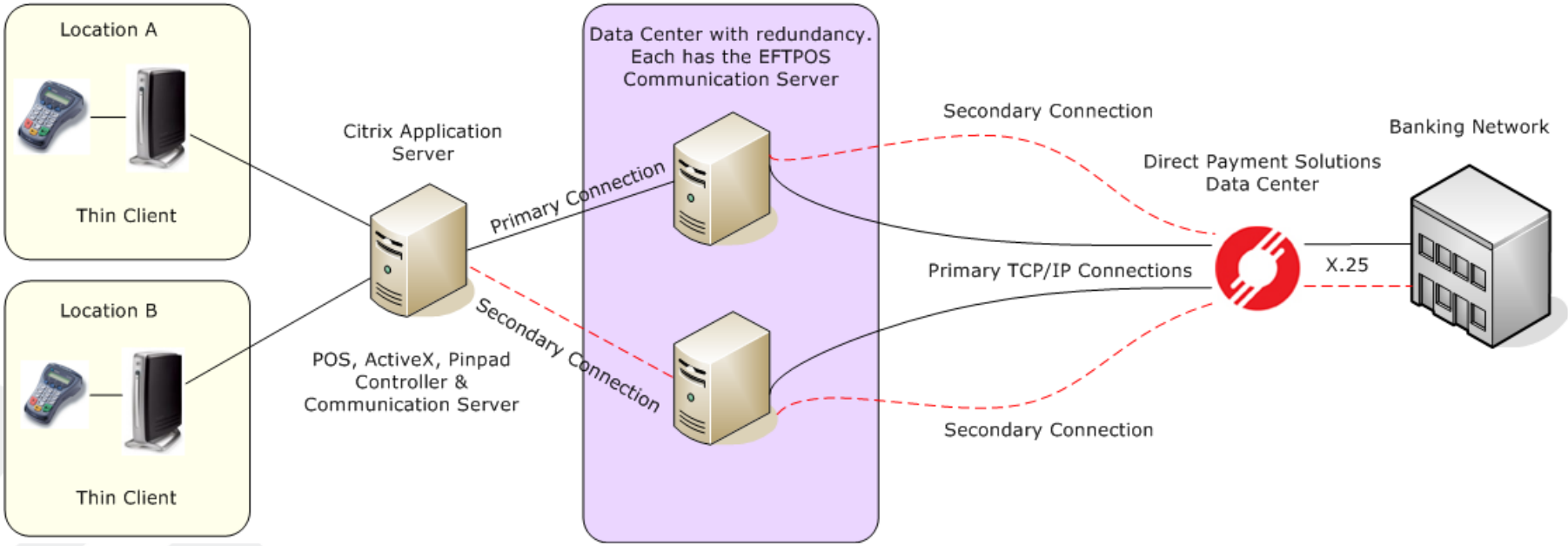
The POS application must be able to handle the following situations:

- Ensure the POS resolves a transaction halted after the financial point of no return.
- Ensure the POS resolves a transaction halted before the financial point of no return.
- The POS can parse irregular XML messages.
- Ensure the POS can correctly parse an xml message fragmented over two TCP sends.
- Ensure the POS can read two messages contained within a single TCP send.

For detailed information on how to test the above, please refer to the testing scripts included in the DPS Certification Request document.

CITRIX & TERMINAL SERVICES

Payment Express® EFTPOS support the two major terminal server offerings, Citrix and Terminal Services. Payment Express® EFTPOS PINpads are connected locally to the thin client and are mapped through using Citrix's ICA and Terminal Service's RDP protocols to a server with Citrix MetaFrame or Terminal Services installed. Reliability is directly related to the ability of the Citrix/Terminal Services and internet connection to maintain a stable COM port mapping (Port redirection) between the Thin Client and Citrix/Terminal Server.



Installation
Download the “Terminal Services/Citrix EFTPOS package” from http://www.paymentexpress.com/eftpos/terminalserver.zip

Extract terminalserver.zip. This file contains two folders “Terminal Server” and “Thin Client”.

Terminal Server folder contents
- pxeftp.exe (Communication Server)
- pxeftp_cfg.txt (Communication server configuration file)
- dpseftx.ocx (ActiveX component)
- LOG folder (Storage for tracing & log files)
- Readme text document
- Install.bat (batch install file)

Thin Client folder contents
- pxpp.exe (PINpad client application)
- pxpp_cfg.txt (PINpad Client configuration file)
- dpseftxc.exe (DPS EFTPOS Client)
- pxjview.exe (Journal Viewer)
- LOG (Storage for tracing & log files)
- Readme text document

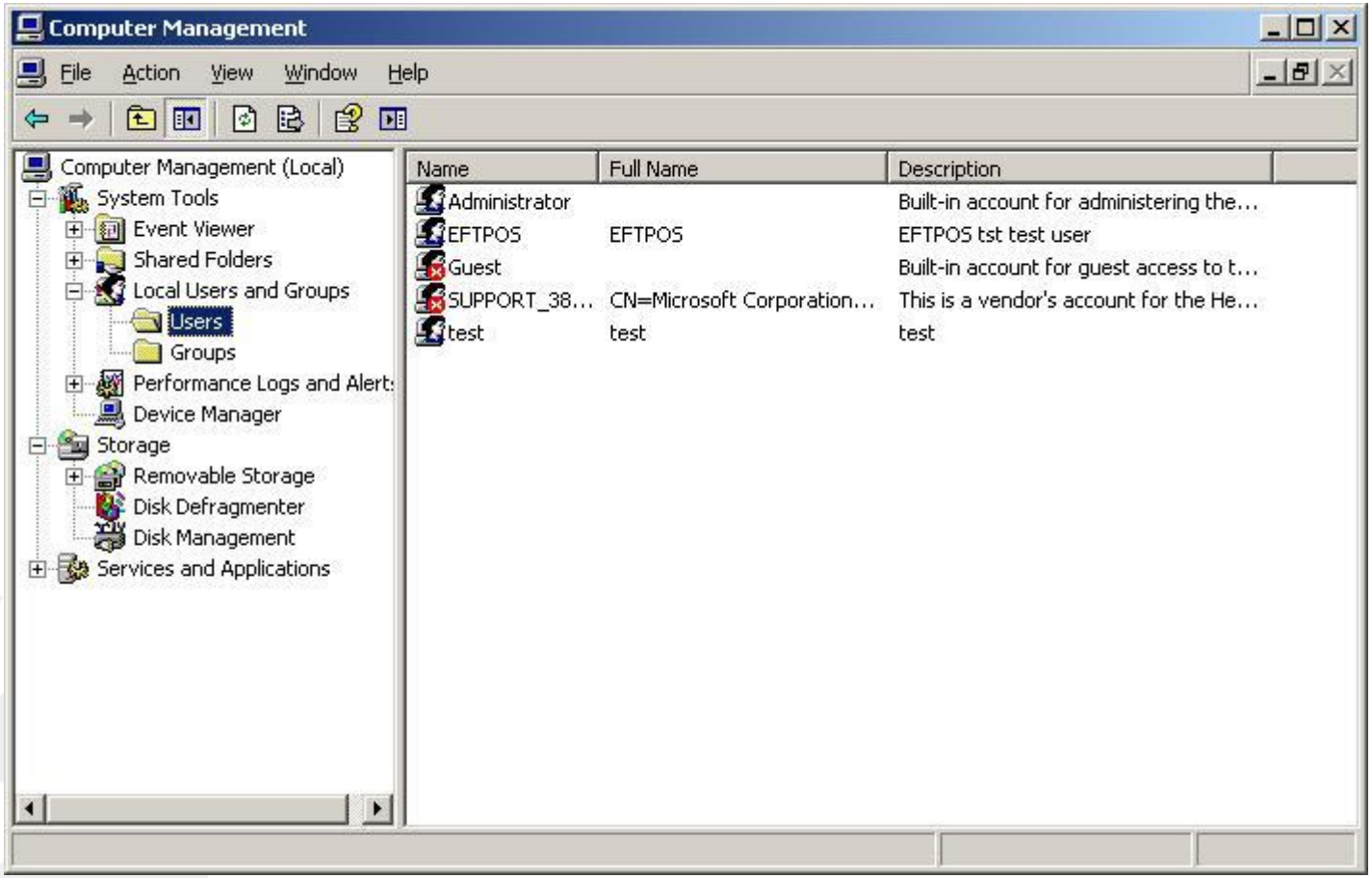
Terminal Server Installation
1. Copy all files from the “Terminal Server” folder to your terminal server. Location = {default hard drive location:}\Program Files\DPS\EFTPOS\ (create this folder if it does not exist).
2. Open pxeftp_cfg.txt.
3. Find line <Address>x.x.x.x</Address> and replace x.x.x.x with the IP address of the server hosting the service.
4. Save pxeftp_cfg.txt
5. Run install.bat. Batch file automatically installs “pxeftp.exe” as a service, starts the service, and registers the ActiveX component. Please note that the install.bat may need editing if files are not installed on c: drive. Installation can also be done via cmd or run box if preferred.

Thin Client (User Profile) Installation
1. On thin client profile copy all files from the “Thin Client” folder to the Thin Client %HomePath%
2. Open pxeftp_cfg.txt.
3. Find line <SvrAddress>x.x.x.x</SvrAddress> and replace x.x.x.x with the IP address of the terminal server hosting the pxeftp service.
4. Save pxeftp_cfg.txt.

DPS EFTPOS software is now installed; the next step is to configure the users and connection.

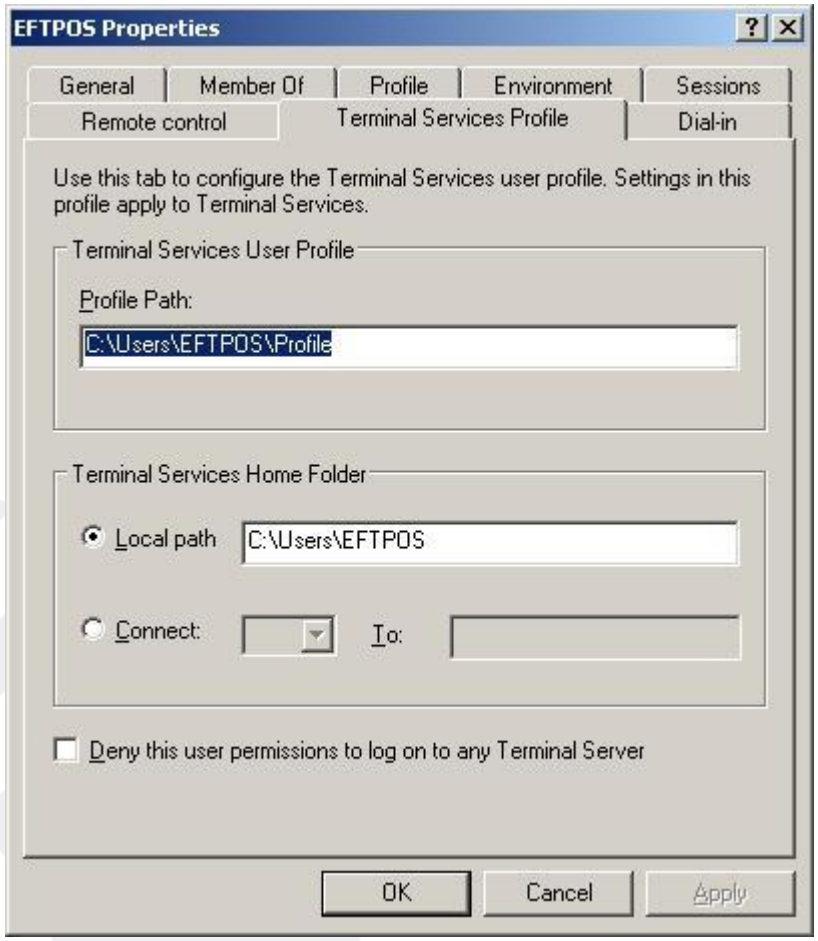
User Configuration

1. Users are setup using Windows Computer Management. The user “EFTPOS” has been setup in the example below.

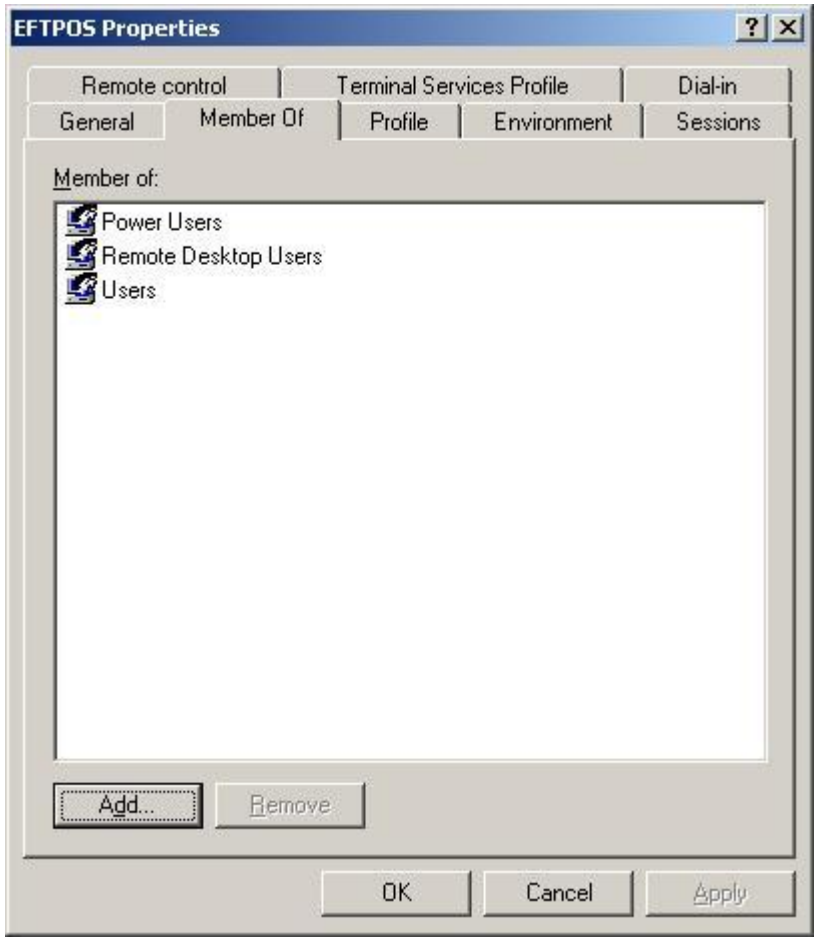


2. To begin configuring the user, right click the user and select properties.

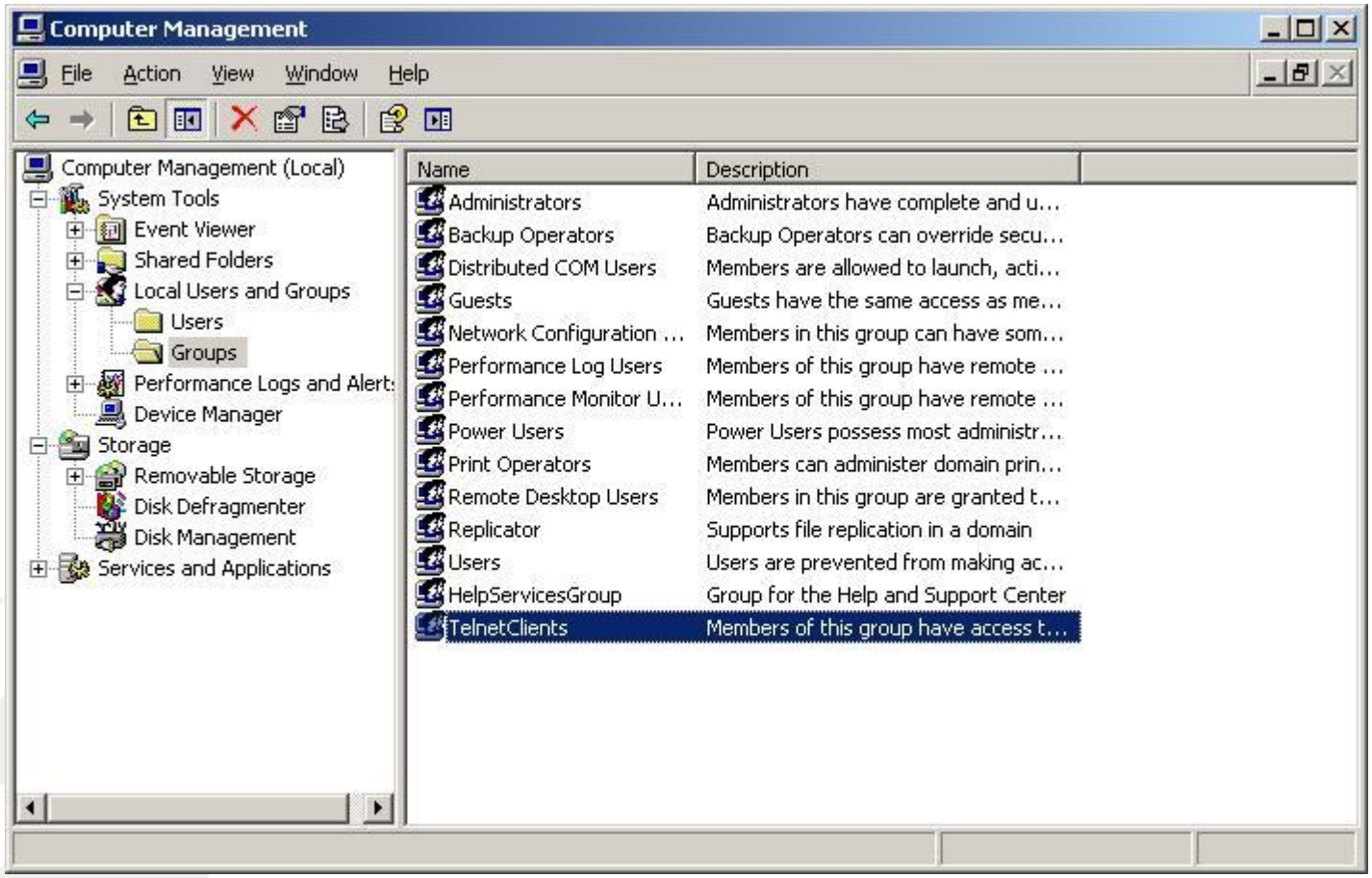
3. You will see the properties of the user. Select the Terminal Services Profile tab. Here you can set the directory where the thin client files are stored (Local Path) and the actual profile settings (Profile path).



4. You can also check what group they are a member of by selecting the Member Of tab. You can add them as part of a group if they require those permissions, for example add the EFTPOS user to the Remote Desktop Users group.

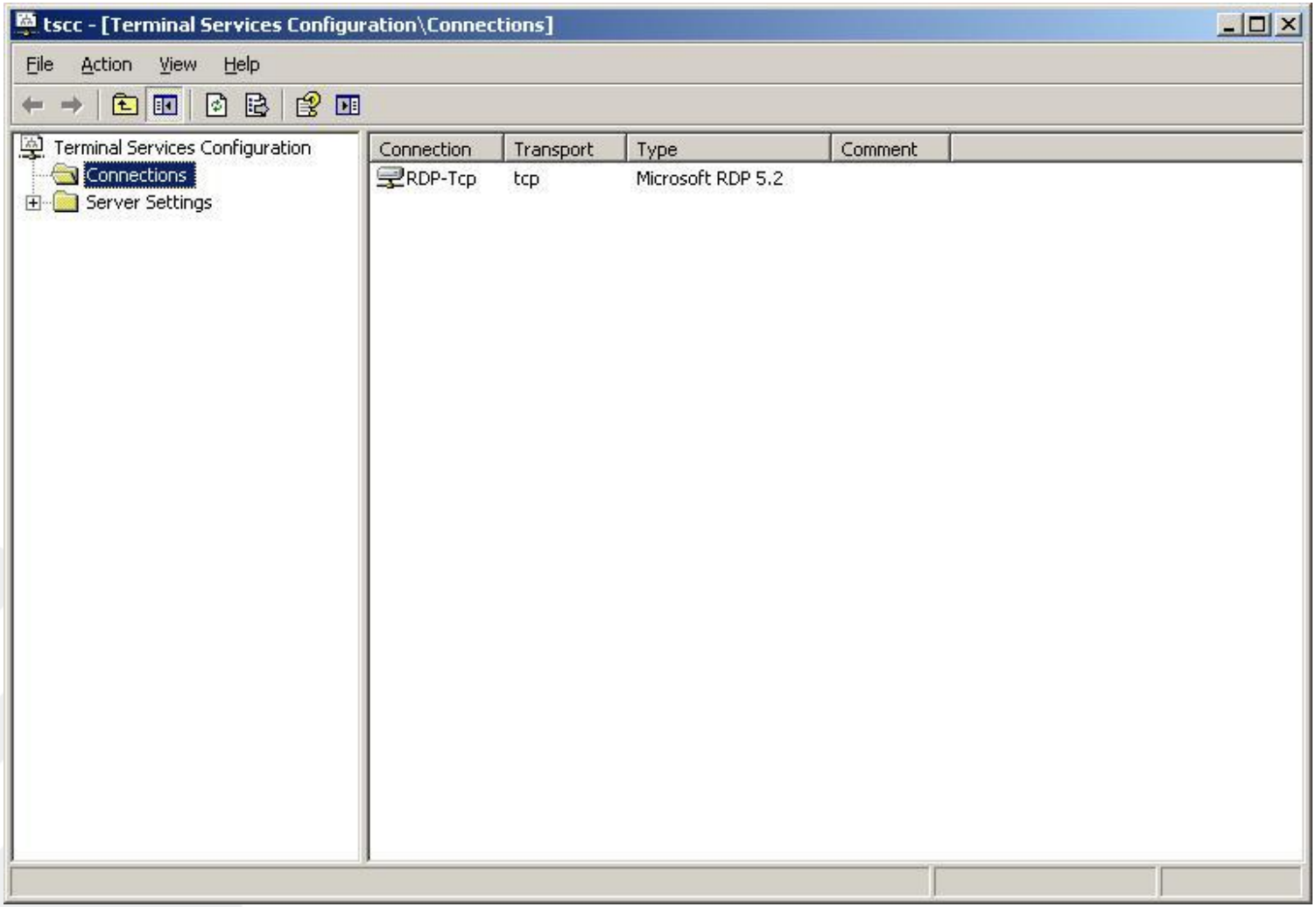


5. You can check what groups exist and their permissions in Computer management under Groups shown below.



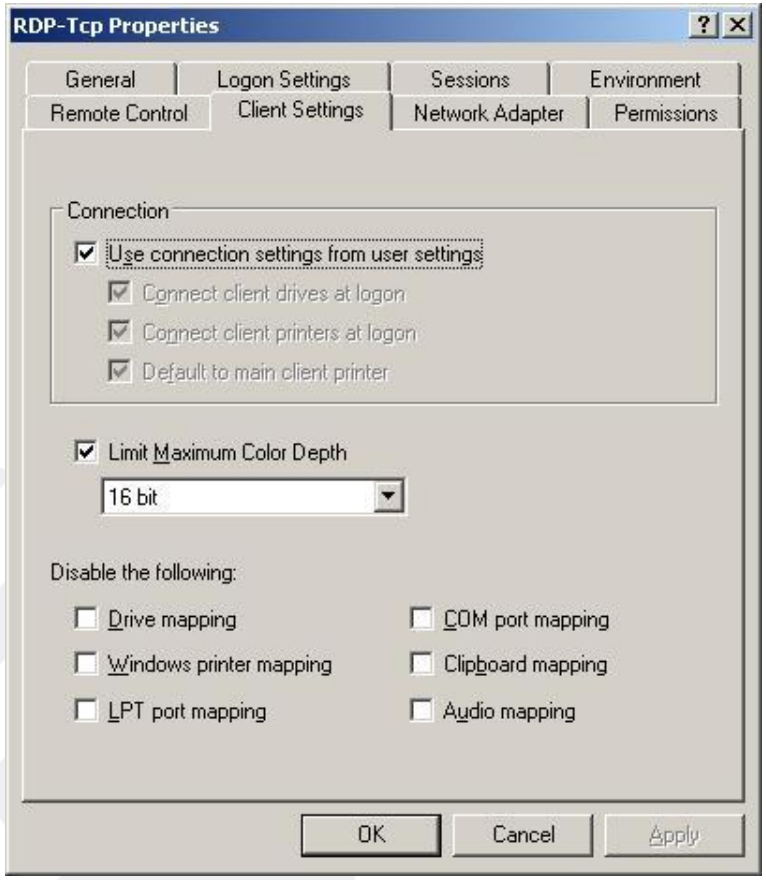
Connection Configuration

1. The connection is also set up using Windows Configuration Manager. There is a connection for each Ethernet card.

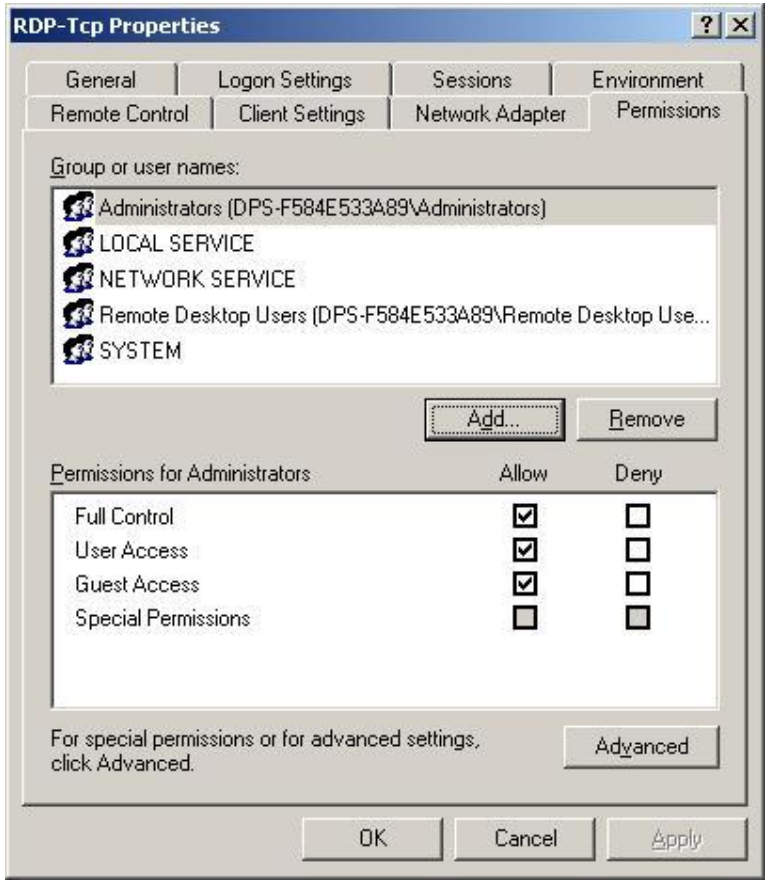


1. Right click the connection and select properties.

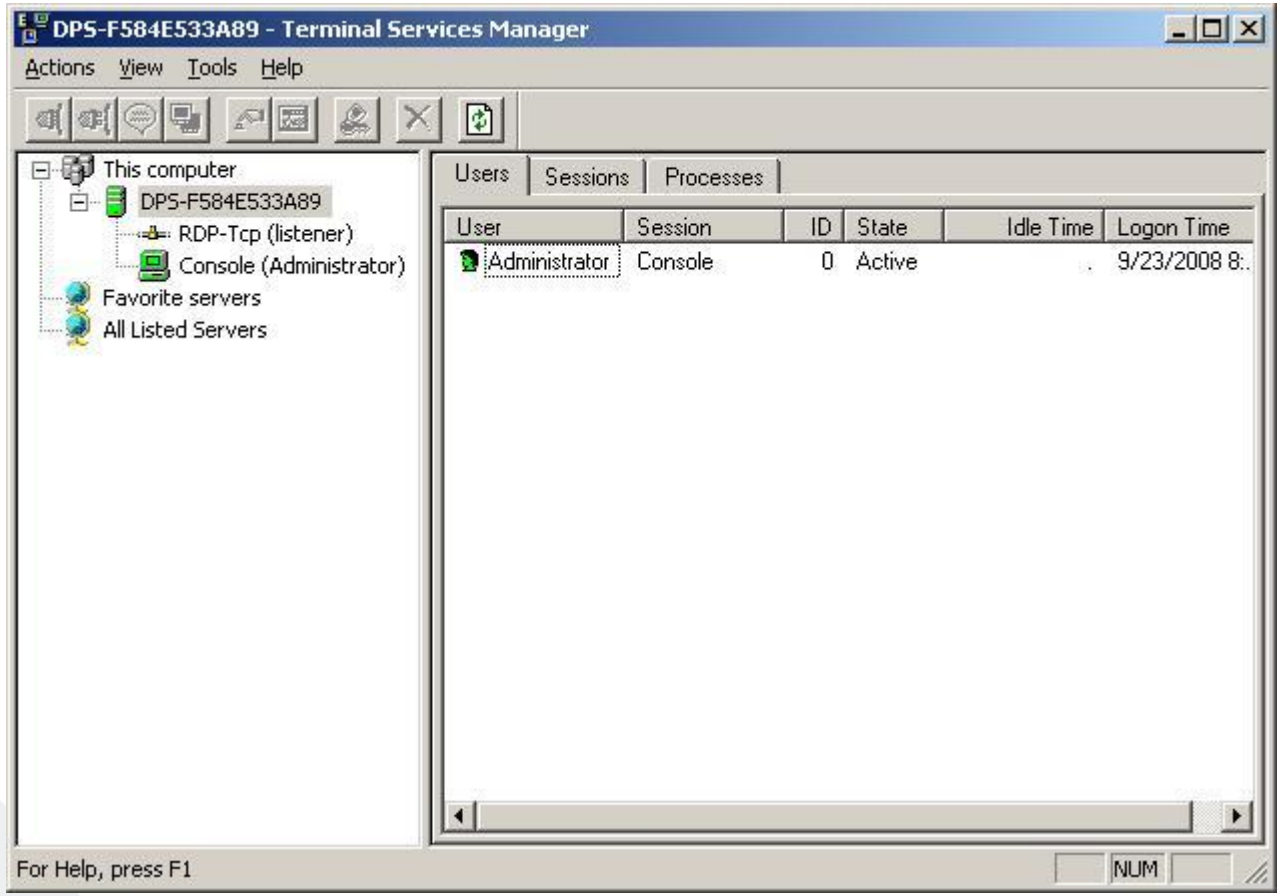
2. Go to the Client Settings tab. The mapping settings should be unticked to enable mapping. Most importantly “COM port mapping” should be enabled. This setting is used to forward the COM port of the PINpad in the thin client hardware to the pxpp.exe sitting on the Terminal Server. LPT forwarding may be required for a printer.



3. Go to the Permissions tab. Here you can add users and groups to the connection and adjust their permissions.



4. The connection is administered using the Terminal Services Manager. This screen can be used to monitor status of connections.



Troubleshooting

Link failure

This can be caused by failing to set the server address in the pxeftp_cfg.txt config, or by general network problems between the server and thin client. A telnet test can be used to verify the connection between the server and thin client.

To test the Client-to-Server connection, telnet from the Thin Client to the Terminal Server on the IP set in SvrAddress above on port 62 (default port) e.g. “telnet 192.169.1.200 62”.

To test the Server-to-DPS connection, telnet from the Terminal Server to the DPS host eft6.Paymentexpress.com on port 61 e.g. “telnet eft6.paymentexpress.com 61”.

PINpad Offline

The port mapping may not be correctly set up. Verify that the tick box for “COM port mapping” is not ticked. Check that no other devices are using the COM port. Connect to the COM port with HyperTerminal. If you cannot connect, the COM port is being used. If this is setup correctly, this can be caused by network latency (with no noticeable effect on the remote client connection). The NETBIOS HELO/EHLO traffic can be tuned to reduce network traffic.

Cross-Talk

If there are issues such as separate PINpads sharing the same details, e.g. serial number, CAID etc. there may be cross talk.

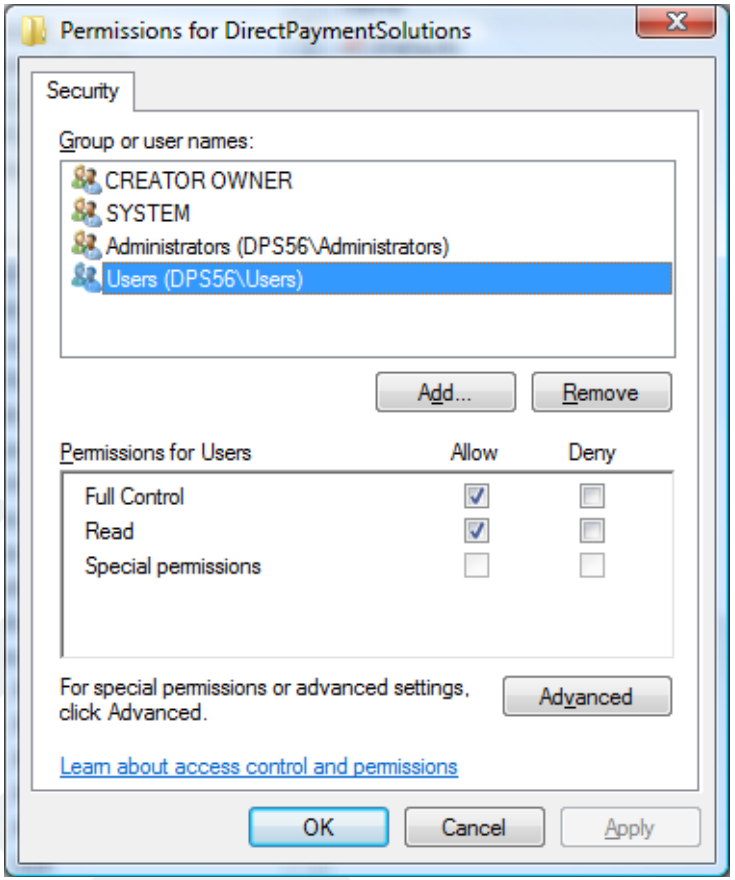
Check pxpp_cfg.txt “Port” setting (not com port) is set to 0. This sets the pxpp to auto-assign a port, avoiding conflicts with other services using the same port.

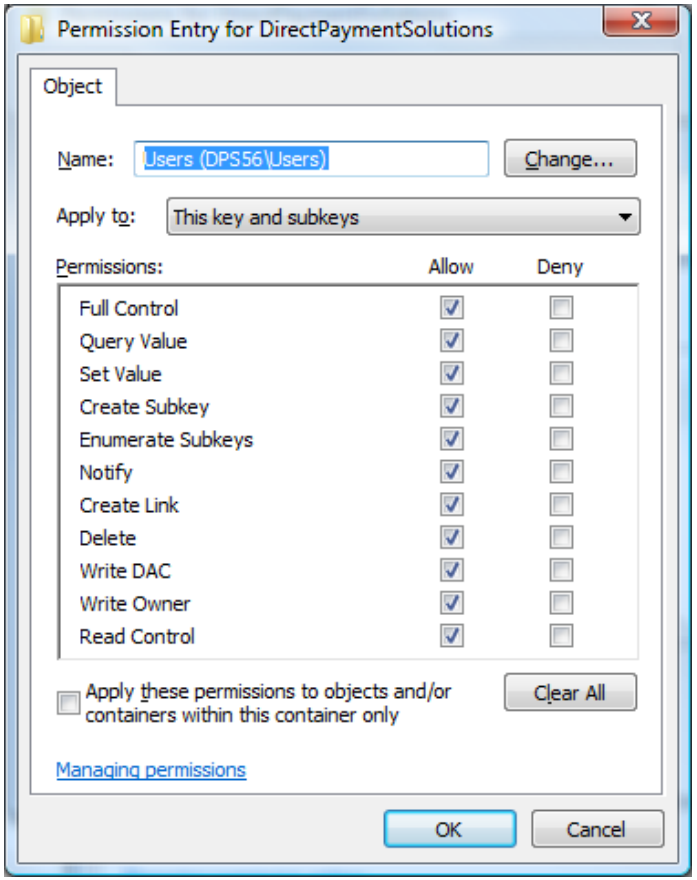
If it is a new install done by copying another user’s directory i.e. same configuration used, removing the .DAT files should resolve the issue.

PX Offline

Pxpp.exe is not running. Verify that pxpp is running in the Processes tab in Windows Task Manager. Pxpp can be set to start during startup. This is easily achieved by creating a shortcut to pxpp.exe in the Startup folder.

If PXPP does not run it may be that it does not have adequate privileges required in order to maintain the port assignment and other settings required. Full control should be given to the user at HKEY_LOCAL_MACHINE\SOFTWARE\DirectPaymentSolutions. Click the 'advanced' button and check the settings here to see that the user has the appropriate privileges.





GETTING CERTIFIED

Now that you are familiar with the DPS EFTPOS software and our XML Interface, the next step is to get your POS certified by DPS.

Before submitting your POS for certification, you will need to complete the standard test scripts included with the DPS Certification Request document.

Once your POS has passed the test scripts and you believe your POS is ready for certification please contact our QA department. It is very important that you book your POS in for certification. DPS requires at least 2 weeks' notice to ensure that there are no delays.

Please use the booking form at the beginning of the DPS Certification Request document.

Please send the completed form to qa@paymentexpress.com and our QA team will be in touch ASAP.

QUESTIONS?

If you have any questions, please feel free to contact our dev support team at devsupport@paymentexpress.com

APPENDIX

Elements - Descriptions of elements used in our XML messages.

Account (input)
Datatype: INT (Max 1 Bytes)
The merchant account which this transaction is to be process under. This parameter is only valid if multi-merchant capability is enabled. If this parameter is not present, the default merchant account will be used.

AmountPurchase (input)
Datatype: BSTR (Max 13 Bytes)
Set the amount to be charged or refunded (depending on the TxnType). Format is d.cc (d=dollars, c=cents). Max amount is 99999.99

AmountCashOut (input)
Datatype: BSTR (Max 13 Bytes)
Set the "Cashout" amount. Format is d.cc (d=dollars, c=cents). If no cash out is to be made, this property must be empty (blank).

AmountRefund (input)
Datatype: BSTR (Max 13 Bytes)
The amount, in dollars and cents, to be refunded. This parameter is only valid for refund transactions otherwise this property must be empty (blank).

Button (input)
Datatype: BSTR (Max 6 Bytes)
Value will be the dialog button that needs selecting. Valid parameters: Yes, No, OK, Cancel. This parameter is case-insensitive.

Button1 (input)
Datatype: BSTR (Max 6 Bytes)
The text for first button.

Button2 (input)
Datatype: BSTR (Max 6 Bytes)
The text for the second button.

Button3 (input)
Datatype: BSTR (Max 6 Bytes)
The text for the third button.

Beep (input)
Datatype: BOOL (Max 1 Bytes)
If present and set to 1, the POS should beep to indicate that it requires user input.

CardType (output)
Datatype: BSTR
The card type used for the transaction. Note that the list may be expanded as support for new cards is added. The CardName format is to capitalize the first letter with remaining letters in lowercase.

CardType	Description
Eftpos	Eftpos Cards
VISA	VISA Cards
DINERS	DINERS Cards
AMEX	AMEX Cards
MCARD	MasterCard

CurrencyId (input)
Datatype: LONG
The currency to use for Dynamic Currency Conversion (DCC).

Currency Id	Currency Name
36	Australian Dollar
242	Fijian Dollar
250	French Franc
280	Deutschemark
344	Hong Kong Dollars
548	Vanuatu Vatu
554	New Zealand Dollar
598	PNG Kina
702	Singapore Dollar
776	Tonga Pa'anga
840	US Dollar
882	Samoa Tala
826	Pound Sterling
756	Swiss Franc
710	South African Rand
978	Euro
458	Malaysian Ringgit
414	Kuwaiti Dinar
392	Japanese Yen
124	Canadian Dollar
90	Solomon Islands Dollar

CurrencyRate (input)
Datatype: BSTR
The rate of the foreign currency against the home currency.

DateTimeTransaction (output)
Datatype: BSTR
Indicates when the transaction was processed. HHMMSS format.

DpsTxnRef (output)
Datatype: BSTR (Max 16 bytes)
A unique identifier returned for every transaction.

EftStatus (output)
Datatype: LONG
Reserved

EnableBackLight (output)
Datatype: BOOL
If present and set to 1, the EFTPOS terminal should turn back light on. DPS recommend Backlight enabled when any messages are to be displayed on the PINpad screen.

Ready (output)
Datatype: BOOL
Indicates whether the terminal is ready or not. If the value is 1, the terminal is ready for processing request. If the value is 0, the terminal is in an error status and may require manual intervention.

ReceiptAutoPrint (output)
Datatype: BOOL
If this value is 1, the receipt will be printed by the EFTPOS terminal. If this value is absent, or any value other than 1, the POS system will be responsible for printing the receipt.

Receipt (output)
Datatype: BSTR
Receipt of the transaction that can be printed out for the customer.

```
DIRECT PAYMENT SOLUTIONS
DIRECT PAYMENT
SOLUTIONS - TEST

*----------EFTPOS----------*
TERMINAL 00905302  TRAN 000321
TIME  10JUN 15:17  ACCT CHEQUE
                    503871....7361
PURCHASE                 NZ$1.00
TOTAL                    NZ$1.00
      INCORRECT PIN
        DECLINED
*------------------------------*
```

Reco (output)
Datatype: BSTR (Max 2 Bytes)
The client application should not interpret the Response Code property contents - it is provided as informational only. The AuthorizedIf this value is 1, the transaction has processed successfully. If this value is 0, the transaction was declined or cancelled.

ResponseText (output)
Datatype: BSTR (Max 20 Bytes)
The Response Text is associated with ResponseCode. For successful transactions this is usually Approved and for unsuccessful transactions this can be a number of texts depending on why the transaction declined. For example it could be Card Expired, Declined, Invalid Card, REFER TO CARD ISSUER, DO NOT HONOUR. All acquirers have their own response texts and should be displayed for better understanding of why the transaction got declined.

SettleDate (output)
Datatype: BSTR (Max 8 bytes)
Indicates Date of settlement (when money will be deposited in Merchant bank account) if this is supported by the Acquirer, otherwise contains the date the transaction was processed in YYYYMMDD format.

Success (output)
Datatype: Boolean true/false
Indicates success or failure of a method call.

Text1 (input)
Datatype: BSTR
The first line of text to be displayed on the EFTPOS terminal.

Text2 (input)
Datatype: BSTR
The second line of text to be displayed on the EFTPOS terminal.

Track1 (input)
Datatype: BSTR
Capture track 1 data with the magnetic strip

Track2 (input)
Datatype: BSTR
Capture track 2 data with the magnetic strip

Track2 (input)
Datatype: BSTR
Capture track 3 data with the magnetic strip

TxnDateTime (output)
Datatype: BSTR (Max 14 bytes)
Date of the transaction, as reported by the EFTPOS network. The format is "YYYYMMDDhhmmss"

TxnRef (input/output)
Datatype: BSTR (Max 16 Bytes)
Unique transaction reference 1-16 alphanumeric character.

TxnType (input)
Datatype: BSTR

Value	Description
Purchase	Purchase - Funds are transferred immediately.
Refund	Refund - Funds transferred immediately.

DPS EFTPOS Response Codes

The client application should not interpret the ReCo property contents - it is provided as information only.

Error Code	Explanation
A0	CANCELLED
A1	Cannot read Card
A2	Invalid Slot
A3	AutoLogon Failed TX
A4	AutoLogon Failed Timout
A5	AutoLogon Failed MAC
A6	AutoLogon Comms Error
A7	Cannot save terminal state
A8	Invalid ReceiptLineTerminator from POS
AC	Invalid ReceiptLineTerminator from SysParam
AD	Invalid ReceiptREset from SysParam
AE	Invalid ReceiptSeparator from POS
AF	Invalid ReceiptSeparator from SysParam
AG	Invalid ReceiptEject from SysParam
AH	Invalid ReceiptEject from SysParam
AI	Select Account timeout
AJ	GetPin timeout
AK	Invalid cash out Amount

Error Code	Explanation
AL	Invalid Amount
AM	Invalid Amount - Too High
AN	Zero amount transaction not allowed
AO	Operator Cancel During Enter card
AP	Operator Cancel During Manual PAN entry
AQ	Operator Cancel During Select Account entry
AR	Operator Cancel During PIN Entry
AS	Invalid Amount - Cash out for purchase only
AT	Terminal parameters not set
AU	Card Read error
AV	MAC VAlidation Error
AX	Message Validation Error
AY	Invalid Response Code
AZ	Invalid field48
B0	AmountPurchase or AmountCashOut not zero for Balance txn
B1	AmountCreditLimit Invalid
B2	Invalid Card - is loyalty card, not purchase card
BA	Operator Cancel During Select Currency (DCC only) entry

Error Code	Explanation
BB	SelectCurrency timeout
BC	Invalid CurrencyRate
BD	CurrencyId
BE	CurrencyId Not Valid
BF	Init terminal 2
BG	EnableCurrencyConversion Not Allowed
BH	CurrencyRate longer than 7 digits including decimal place
BI	Invalid Date
E0	EMV Download completed
E1	App Blocked
E2	TxnRef Error / Trnasaction not found
E3	No offline Pin support
E4	DCC CurrencyId invalid
E5	DCC AcquirerId Invalid
E6	DCC CurrencyRate Invalid
E7	EMV card removed
E8	ICC Declined
F0	Please log on - offline transactions not allowed
F1	Offline exceeded - either num transactions or offline duration
F2	Invalid offline transaction (not purchase only)
F3	Declined offer to process offline
F4	Offline over purchase limit / Tip limit exceeded
F5	Offline transaction already stored for this card
F6	Only signature supported
F7	Timeout waiting for merchant to allow offline transaction
F8	Timeout waiting for merchant to allow excess offline amount
F9	Timeout waiting for PIN Pad MAC Generate
U9	Timeout for Transaction

Error Code	Explanation
UA	Power Fail
UB	Auto Logon timeout during transaction
UC	Timeout for reversal (CANNOT PROCESS)
UD	Auto Logon Failed
Z1	Link Down
Z2	Busy
Z3	MacGen Error
Z4	Printer Offline or error
Z5	Busy - PINPad Offline
Z6	Busy - Auto logon in progress
Z7	Busy - EMV file in progress
Z9	Signature Declined
ZA	Sign Receipt (PINPad)
ZB	PINPad not initialized
ZC	Card Reader Error
ZD	PINPad offline
ZE	PINPad Back Online
ZF	No unit selected
ZG	Unknown command
ZH	Service Shutdown
ZI	PC Shutdown
ZJ	System Startup
ZK	System Startup but PIN Pad Offline
ZL	PIN Pad serial Swapout - serial number changed

Message Prompt Lookup Table

ID	Message Prompt
0	EMPTY
1	ACCEPT WITH SIG
2	ACCEPTED
3	ACCOUNT ERROR
4	CANNOT COMPLETE
5	CANNOT READ CARD
6	CANNOT ROUTE
7	CARD BLOCKED
8	CARD ERROR REFER
9	CARD EXPIRED
10	CARD NO ERROR
11	CARD REMOVED
12	CARD TYPE ERROR
13	DAILY LMT EXCESS
14	DECLINED
15	EMV COMPLETE
16	FORMAT ERROR
17	ICC CARD ERROR
18	ICC DECLINED
19	INCORRECT PIN
20	INVALID AMOUNT
21	INVALID CARD
22	INVALID DATE
23	INVALID MERCHANT
24	INVALID TRANS
25	MERCH CARD ERROR

ID	Message Prompt
26	MERCHANT PIN ERR
27	MTHLY LMT EXCESS
28	NO OFFLINE PIN
29	ODO REQUIRED
30	OVER CASH LIMIT
31	OVER SYS LIMIT
32	PHONE FOR AUTH
33	PHONE HELP DESK
34	PIN REQUIRED
35	PIN TRY EXCESS
36	PLEASE TRY AGAIN
37	PLEASE WAIT
38	PROCESSING ERROR
39	PROCESSING NOW
40	PROD LMT EXCESS
41	PROD NOT ALLOWED
42	READY TO SETTLE
43	RE ENTER TRANS
44	REFER ISSUER
45	REMOVE CARD
46	SIG ACCEPTED
47	SIG DECLINED
48	SIG REQUIRED
49	SIGN RECEIPT
50	SWIPE CARD
51	SYSTEMS FAULT

ID	Message Prompt
52	TRAN UNAVAILABLE
53	TRANS CANCELLED
54	TRANS COMPLETE
55	SELECT PUMP AND
56	PUSH ENTER
57	REQUIRE RECEIPT
58	YES OR NO
59	LOAD USER ID
60	AND ENTER
61	LOAD CONTRACT NO
62	LOAD FLEET NO
63	LOAD VEHICLE NO
64	LOAD DRIVER NO
65	LOAD EMPLOYEE NO
66	LOAD PLANT NO
67	LOAD ODOMETER
68	LOAD ENGINE HRS
69	LOAD ODO ENG HRS
70	LOAD PRESET
71	GO TO PUMP
72	AND TAKE FUEL
73	WRONG SYSTEM
74	WRONG NETWORK
75	SYSTEM NOT READY
76	EXPIRED CARD

ID	Message Prompt
77	BAD CARD
78	CARD NOT ALLOWED
79	INVALID
80	IDENTIFIER
81	CARD ALREADY
82	IN USE
83	DOLLAR LIMIT
84	REACHED
85	INVALID PUMP
86	NUMBER
87	PUMP OFF LINE
88	PUMP IN USE
89	PUMP IN ERROR
90	PUMP UNAVAILABLE
91	RESTRICTED TIME
92	RESTRICTED FUEL
93	INVALID ENTRY
94	ODO ENTRY
95	OUT OF RANGE
96	PRINTER NOT
97	WORKING
98	PRINTER OUT OF
99	PAPER
100	NO RECEIPT TO
101	PRINT
102	NOT ACCEPTED

ID	Message Prompt
103	EFTPOS
104	INSERT CARD
105	PRINTING
106	SWIPE CARD FUEL
107	PLEASE WAIT FUEL
108	MAN PAN
109	NOT ALLOWED
110	TRANS LMT EXCESS
111	ACCT NOT ALLOWED
112	PIN BYPASS
113	LAST PIN TRY
114	ENTER AUTH AMNT
115	TAKE FUEL
116	HANG UP NOZZLE
117	PUMP X READY
118	FOR DELIVERY
119	MIN AUTH 50 DOLLARS
120	INSERT AND
121	EMV TRANS
122	PUMP X IS
123	PROD X OK
124	ODOMETER X
125	CORRECT
126	ORDER NUMBER
127	REGISTRATION
128	CARD NOT YET

ID	Message Prompt
129	VALID
130	RECEIPT
131	UNAVAILABLE OK
132	P X Y
133	READY TO DELIVER
134	AVAIL
135	BAL
136	SWIPE
137	GLENGARRY CARD
138	LOYALTY CARD
139	DPS EFTPOS
140	PIN ERROR
141	SWIPE FLYBUYS
142	ENTER PKT VOUCHR
143	CODE
144	PKT VOUCHR CODE
145	USED
146	EXPIRED
147	INSUFFICIENT
148	FUNDS
149	MAX REDEMPTIONS
150	INVALID PRODUCT
151	DISCOUNT ITEM
152	MOBILE NUMBER
153	ENTER MOBILE NUM
154	XMAS CLUB CARD

ID	Message Prompt
155	SWIPE GIFT CARD
156	SWIPE PHONE CARD
157	EZI PAY CODE
158	XMAS CLUB CODE
159	GIFT CARD CODE
160	ENTER
161	DOLLAR AMOUNT
162	ENTER TIP AMOUNT
163	ADD TIP AMOUNT
164	YES NO
165	CVC ON CARD