



INGENICO

---

***Online Credit Verification  
Server  
Developers  
Specification***  
*Version 1.16*

---

## REVISION RECORD

File Creation Date: 18 November 1997

Revision Number	Date	Author	File Name
1.00	21 Nov 97	Craig Godden	
1.01	8 Sept 98	Craig Godden	
1.02	Sept 98	Craig Godden	
1.03	Sept 98	Craig Godden	
1.04	October 98	Craig Godden	
1.05	October 98	Craig Godden	
1.06	January 99	Craig Godden	
1.07	January 99	Craig Godden	
1.08	May 99	Craig Godden	
1.09	May 00	Craig Godden	
1.10	Sep 00	Craig Godden	
1.11	Nov 01	Craig Godden	
1.12	March 02	Craig Godden	
1.13	February 03	Craig Godden	
1.14	March 03	Craig Godden	
*1.15	June 03	Craig Godden	
1.16	July 03	Craig Godden	

\* ANZ AMENDED

This document was prepared by:

Craig Godden  
Ingenico International (Pacific) Pty Ltd  
9 Apollo St  
Warriewood NSW 2102 AUSTRALIA

Phone: +61 2 9979 9333  
Fax: +61 2 9979 9751  
Email: [cgodden@ingenico.com.au](mailto:cgodden@ingenico.com.au)

*If you have any comments or suggestions relating to this guide  
please contact Ingenico International (Pacific) Pty Ltd directly  
or send a copy of the fax sheet included at the end of the guide*

## TABLE OF CONTENTS

Introduction.....	1
System Overview .....	2
Development Tools @ Website .....	3
Development Suite Setup .....	5
Client Application Development.....	7
Virtual Pinpad Details.....	11
OCV Server Application .....	12
EFT Server.....	13
EFT Application .....	14
Hardware .....	16
Appendix A - Client to Server interface.....	17
Appendix B – Response Codes.....	36
Appendix C – Electronic Journal Format .....	37
Appendix D – ActiveX Control.....	39
Appendix E - OCV Test Procedures .....	40
Appendix F - Development Procedures.....	41
Appendix G - Development Suite Differences.....	43
Appendix H – Internet Transaction Data.....	45
Ingenico Contact Details.....	46

# ***Introduction***

The purpose of this document is to outline the functional requirements of the Ingenico Online Credit Verification (OCV ) system.

- The OCV Server is designed to allow unattended processing of credit card transactions for multiple merchants using a single server and bank dial-up or leased line interface via the standard EFTPOS network.
- The credit card verification requests can originate from customer workstations or from an automated application. A telemarketing operation could use the OCV Server to accept authorisation requests from Win95 or Windows NT phone operator workstations while an Internet Service Provider (ISP) could use the product to validate credit card details originating from an Internet Web Site.
- A bank approved encryption device is used to provide reliable transaction recovery and maintain security of bank terminal master keys used to generate Message Authentication Codes (MACs) for AS2805 transactions. A standard PC running Windows 95 or Windows NT is used to host the server software and the bank communications interface.
- The OCV server is a hardware/Software solution. The hardware consists of a security device, and a modem. The Development Suite version does not require any hardware.

This document will outline the entire system including the client application, the server application and also the EFT application.

The latest versions of all software and documentation is available through Ingenico or at the support Web page [www.pceftpos.com](http://www.pceftpos.com)

## **Accreditation:**

Ingenico provides the POS vendor with support during system integration and when integration is complete, Ingenico 'accredits' the POS system against a test script. If the POS system passes the Ingenico accreditation, Ingenico will notify the Bank that the POS vendor has been accredited by Ingenico and a pilot with the ePOS system may proceed.

VbV accreditation is a separate process between the POS vendor and the Bank.

## **Disclaimer:**

*This system is designed to merely clear transactions to the bank using the existing EFTPOS network. Ingenico accepts no responsibility for the delivery of the transactions to the OCV server. These transactions may come from a variety of sources (e.g. Web, phone operator, database etc) It is the responsibility of the client application to ensure that these details are correct (e.g. correct card number , amount)and securely passed to the OCV server. The OCV server will only validate the data.*

*Any implementation or use of Visa 3D secure or the use of the Internet Data field is the responsibility of the client application.*

# System Overview

The requirements of this system are to provide an electronic method of gaining online credit card approvals using the existing EFTPOS network.

## Server Specifications

- Up to 200 Merchants Per Secure Encryption Device (Ingenico PX318)
- Up to 4 Secure Encryption devices may be supported per server.(custom system only)
- Up to 9999 client-selected accounts configurable.
- Generic TCP/IP interface clients.
- Drop-in Windows ActiveX control for Windows 95 and Windows NT clients.
- No special requirements for Host Acquirer (bank) - utilises existing EFTPOS network standards and infrastructure.
- Supported AS2805 transactions:
  - 0200 Purchase
  - 0200 Refund
  - 0100 Pre-Auth (supported by some banks)
  - 0220 Completion (supported by some banks)
  - 0300 CPAT messages (if supported by the bank)
  - 04xx Reversal
  - 0800 Logon
- Up to 64 active transactions (depends upon line speed).
- Live system requires 2 COM ports on the server PC.
- Supports use of Visa 3D Secure and CAVV fields

## Network

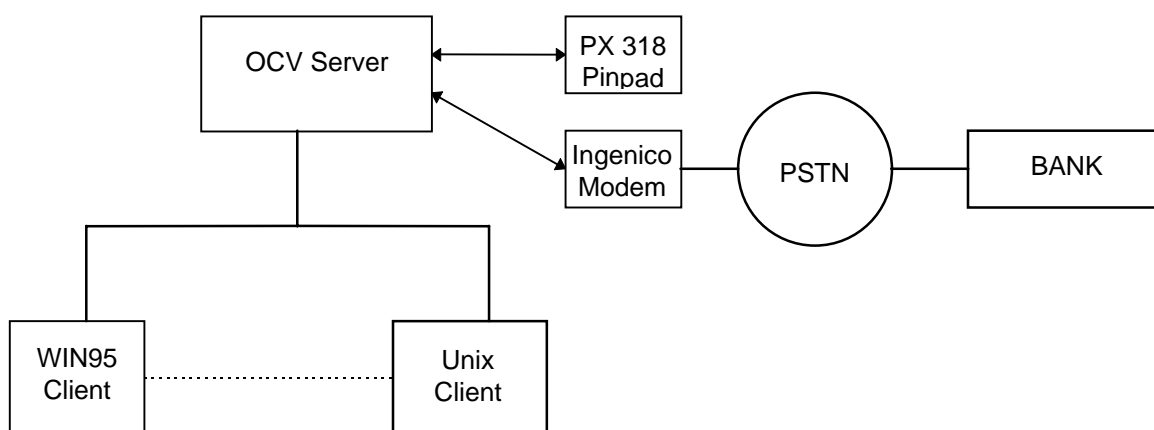


Figure 1 - Network

## *Development Tools @ Website*

The website [www.pceftpos.com](http://www.pceftpos.com) has been setup to provide you with the latest information and development tools that are available for the OCV server. The website has the following:

### *Online Documentation:*

- The following documentation is available from the "Download: page at the website:
  1. The OCV Developers Spec. (this document)
  2. The OCV Overview. A brief word document on some of the functionality of the OCV server
  3. The OCV Server ActiveX control Specification. The outline of how to use the ActiveX control for the OCV server.
- The OCV Server TCP/IP interface and ActiveX specification are also available in HTML format at the website. These can be found on the "*Reference*" page
- For Verified by Visa documentation refer to Bank web site for details eg <http://www.anz.com/>.

### *Development Tools*

- There is also a range of products available at the site for OCV development and maintenance. The following are on the "*Download*" page
  1. **The ActiveX control.** This is a standard Microsoft ActiveX control which allows for quick PC application development. The full source code is available on the internet, and the product is not supported by Ingenico.
  2. **The Borland component versions of the ActiveX control.** These are the same as the ActiveX control, but are geared specifically for the Borland range of products.
  3. **The Internet DLL** version of the ActiveX control.
  4. **The Software Development Kit (SDK).** This provides the developer with the ability to fully develop there client application without the need for a connection to the bank.
  5. **Journal Viewer.** This utility will allow you to view the journal files created by the OCV server in a more meaningful way.
  6. **Simple Interface.** This utility provides a simple graphical user interface to the OCV server. This utility may be used by client applications as a quick interface to the actual OCV server.
  7. **Batch Processor.** This utility program allows you to use the OCV server without any client development at all. Just pass it a file (in a pre-defined format), press GO, and this utility will generate a new file with the results of every transaction.

### *Source Code*

- We even give away a lot of our source code. The "Download" page contains the following:
  1. Delphi, Builder C++, and Visual Basic simple demos.
  2. Delphi source code for the OCVTEST.exe file included in the SDK.
  3. ActiveX and DLL source code is available on the website.
  4. Simple Interface Source code (Delphi)

## ***Case Studies and Reference sites***

- We have provided some case studies of a few of our customers to give you a few ideas on how some people have implemented the OCV server into their environments. These case studies can be accessed from the main page by clicking on the Case Studies link.
- We hope to add further case studies, and reference links to this page as we progress. If you would like to be added to this page, please contact Craig Godden at Ingenico ([cgodden@ingenico.com.au](mailto:cgodden@ingenico.com.au))

## ***Development Procedures***

- The main page of the website has a "Development Procedures" link which provides you with a step-by-step guide to what is needed along the development path.

## ***Mailing List***

- The website offers the chance to subscribe to the Ingenico OCV mailing list. This list is used to update people on OCV server developments.

## ***OCV Discussion Area***

- We hope to have a discussion area for OCV developers on the website in the near future. Keep checking the website.

## *Development Suite Setup*

The OCV development suite Software Development Kit (SDK) is a software only method for integrating your client applications to the Ingenico OCV Server application. The SDK will perform all of the functions of the real OCV server except communicate with the bank (It will simulate this). There are a few differences between the SDK and the live OCV server. Please read Appendix G to find out more.

When you purchase the live system, you will install the live system, and your client will begin communicating with the live system instead of the SDK. There will be no change to your application. The SDK has been developed to be as realistic as possible.

### *Installation*

The Development Suite is installed by clicking on the setup.exe program in the "development suite" directory of the CD-ROM development kit. Follow the setup wizard to install the development kit onto you hard drive.

After the installation is complete you will have the following files in the directory that you selected to install the Development Suite:

- **OcvDev.exe** This is the Development OCV Server (OCVDEV). It will emulate the real OCV server and provide you with the ability to develop your client application without the need for any special hardware OR bank intervention.
- **OCVTest.exe** This application is an example client application which provide the ability to send every type of message to the OCVDEV server. The complete source code for this application is available on the CD-ROM in the source directory. It is written in Delphi.

You are now ready to begin client application development.

Please read the following documents regarding the OCVDEV application:

- **Appendix G** This outlines the differences between the OCVDEV server and the 'real' OCV server. It is important that you are aware of the differences.

### *Accounts*

Each Virtual PINpad MUST be configured with an Account Number. (Note The same account number can be issued to multiple Virtual Pinpads. This allows multiple outstanding transactions for that particular account.)

The accounts are merely numbers to the server (0000 to 9999). A textual representation of the account number should be set up on the OCV server using either of the following methods:

- Use the Account Modification Request message. The accounts will be stored in the registry of the machine which is running the OCV server.
- Use the Ingenico OCV Maintenance program to update the account list. Details of this program are at the Ingenico Web site ( [www.pceftpos.com](http://www.pceftpos.com) )
- Place the file ocvacc.dat file in the installed directory area. The OCV server will read this file in and update the registry. NOTE. This method is for backward compatibility with Version 1 of the OCV server. It is recommended that after you read the file in, that you remove the file from the install directory and use the previous methods for updating the account list.

### ***Account File Format***

The format of the OCVACC.dat file is as follows:

Account Name      1  
Account Name2     2  
etc.

NB You can not use account 0. It is always <Default>

The textual representation is ONLY for when you view the actual OCV server. The OCV server will show the textual name instead of the number on the totals screen and the Virtual PINpad configuration screen.

When a server is installed, the OCV Maintenance program (on the CD-ROM) should be used to set the accounts up as they are on the client.

### ***Example Account Setup.***

The OCV server has 3 Virtual Pinpads:

- Virtual PINpad 1 is linked to the Water Board account at the bank. This VPP is predefined by the client to have account 0001
- Virtual PINpad 2 is linked to the Electricity commission account at the bank. This VPP is predefined by the client to have account 0002
- Virtual PINpad 3 is also linked to the Electricity commission account at the bank. This VPP is predefined by the client to have account 0002

When a client requests a purchase authorisation to the server, the client must specify the account number to use. If the client wishes to credit the money to the Electricity commission, the client sends the account field with 0002 in it. The OCV server will then look up the first available VPP which has an account type of 0002. The OCV server will use that VPP to perform the transaction.

If the water board was to be credited, the client would send account 0001

If 2 people wanted to use the Electricity commission, the client would send 0002 for both transactions, and the OCV server would choose the VPP 2 for the first transaction and VPP 3 for the second transaction.

Only one transaction would be possible (at a time) for the Water Board as it only has one VPP. If a second transaction were to be sent, the second would fail with (A6) SERVER BUSY. The client should then wait for the first transaction to complete, and retry the second.

## *Client Application Development*

- The client application is responsible for initiating the transaction.
- The client application can be any application that is capable of transmitting to the server using TCP/IP.

### *Interface Options*

#### **1. Generic Interface (Recommended implementation Method)**

- The Generic interface is the message formats that the clients use to access the OCV server. The ActiveX control uses these generic message formats to access the OCV server. They are provided so that application programmers can use any development platform they wish.
- Error handling must be handled by the client application.
- Ingenico recommend that developers use this interface as the client application is then not dependant upon Ingenico for part of the application. This narrows the points of failure for an application.

#### **2. Windows ActiveX control**

- Ingenico offers a Windows ActiveX control (or Delphi Component, or C++ Builder Component) which wraps the generic interface. It does not provide any error handling. This MUST be implemented by the developer. Contact Ingenico for details on this control. (NOTE. This ActiveX control is not the same control which was initially released. Support for the original OCX control has been discontinued.)
- Ingenico offers the source code for a Delphi, Visual Basic, or Builder C++ applications which uses the generic interface.
- Error handling must be handled by the client application.

#### **3. Windows DLL**

- This is the same as the ActiveX control, but it behaves like a DLL

It is anticipated that customers of this system will integrate either the ActiveX control or generic interface into there own applications.

A simple Window 95/NT client application is also available for testing the ActiveX control. The source code to this application is also available upon request.

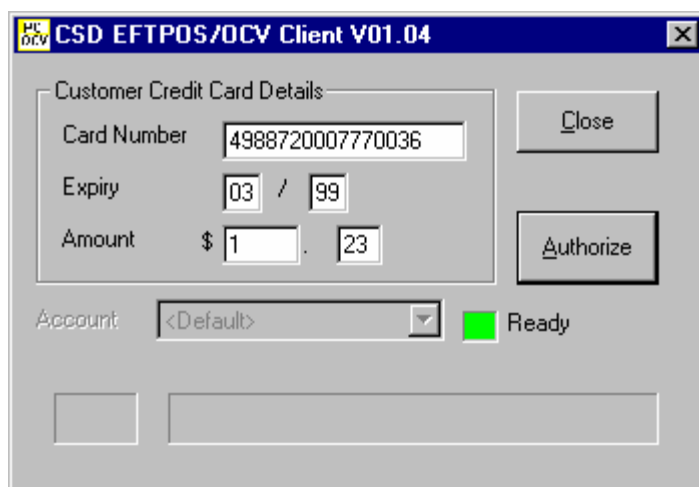


Figure 2. Simple win95 client application

The following commands will be available from the client to the server:

- Financial Request (Purchase, Refund, Pre-Auth, Completion ) Pre-Auth and Completion are bank dependant, and may not be supported by all banks.
- Transaction status
- Totals request
- Logon Request ( This does not need to be used by the client)
- OCV server Statistics Request
- Virtual Pinpad Configuration Request
- Virtual Pinpad Status Request
- Accounts List Request
- Account Modification Request
- OCV server configuration request

## ***Client Error Recovery***

- The client programmer **MUST** perform some basic client to server error recovery.
- Error recovery sequence of a transaction should be attempted if:
  1. The client has lost communication with the server,
  2. A response to a transaction request has not been received within 2 minutes of the transaction being sent
- Error recovery for the generic interface **MUST** perform a transaction status request to the server to find out the result of the transaction. The error recovery should continue to establish communications with the server until the client to server interface is recovered.
- ***When the link is re-established, the server will report the result of the transaction which was performed. It is the responsibility of the client to maintain the financial integrity of the transaction in regard to the clients customer.***

## *User Interface Guidelines*

- The user interface for the client is under the total control of the application developer and hence the user input can be through any means available to the programmer. This may include user keyboard entry, database entry, or via electronic methods such as IVR's (Interactive Voice recognition systems)
- The refund function should be password protected. This will ensure that only the appropriate people have access to the refund capability.
- Ingenico recommends that the unique transaction reference number that is sent to the server from the client is presented to the customer as a means of tracing a transaction.
- The OCV server journals all transactions to disk. Because of this, there is no need for the client to journal the data returned from the OCV server. It is up to the client application to journal this if it is needed.
- The client application should include all configuration of the OCV server that is possible. The OCV server provides TCP/IP API calls to remotely configure each virtual pinpad, retrieve statistics, modify accounts, and retrieve individual Virtual pinpad statuses.
- Once the OCV Server has been initially installed and setup (PINpad COM port), it should not be necessary to ever view the OCV server application again. All configuration should be done by the client application. This allows the client application to fully define the user interface, and maintain information backups.
- It is however possible for the client to implement only the minimum commands, and use the OCV server to configure the system.
- The OCV server does maintain totals for each account performed on the system for the last 7 days. This information is NOT generated by the bank. It is merely accumulated by the OCV server. It is recommend that the client application maintain some sort of totals based upon the settlement date returned in the transaction response, as the server totals reflect the totals for a particular account, NOT the client that requests the totals.
- As this system may be used by clients which may not be operated by humans, it is up to the client application to build a receipt if necessary. All of the information that is required for a receipt will be returned to the client from the server in the response packet.
- It is also up to the client application to backup any of the setup information so that the server may be reinstalled if necessary. This information includes:
  - Account List.
  - Virtual Pinpad Details (see Virtual Pinpad Details section.)
- The server will maintain a log of all transactions which have been performed.
- If the Client application supports Visa 3D secure or CCV processing, please refer to the bank specific documentation on Internet Data Field Processing.

## ***Comms Methods***

- The main communications methods for this product are:
  - Dial-Up. The dial-up connection uses a standard telephone line.
  - Leased-Line. The leased-line is a direct permanent phone line between the merchant site, and the bank. This leased-line is arranged by your bank.
- Dialup transactions will incur a time delay to dial the bank. This delay is approximately 12 seconds. This would not be a problem for batch systems as only the first transaction will take the extra 12 seconds.
- The OCV server will not hang up the phone line until all outstanding transactions have been returned by the bank.
- Leased Line has no dial time at all. Please contact the bank to see if you are eligible to use leased line.

## ***Transaction Turnaround***

- A single transaction which has to dial the bank will take dial time (12 seconds) PLUS time at the bank (6-20 seconds) PLUS the time in local network (approx. 1 second). E.g. 2 transactions sent at the same time. The first transaction would take approx. 19 seconds. The second would return 2 seconds later because it did not have to dial (as the dial delay was in the first transaction).
- An example of transaction throughput is a batch system which maintains 8 outstanding transactions. (This requires at least 8 virtual Pinpads). If the client maintains 8 outstanding transactions, the OCV server can process in excess of 1500 transactions per hour to the bank development system on a single dial-up modem.
- A 1 Virtual pinpad solution would be able to process approx 200-300 per hour if continually used.

## *Virtual Pinpad Details*

Each virtual pinpad (VPP) within the OCV server is in fact a complete EFTPOS device. Because of this each virtual pinpad must have unique configuration details which are provided by the bank.

The following information is required for each virtual pinpad:

- **Merchant ID:** This number is provided by the bank. (15 characters)
- **Terminals ID:** This number is provided by the bank. (15 characters)
- **Network Type:** This will be supplied by Ingenico. (15 characters)
- **Network ID:** This will be supplied by Ingenico. (15 characters)
- **Accounts:** These are to be set up at installation time. (15 characters)

Each VPP can be in a number of different states including Ready, Not Ready, Busy. These states indicate what the VPP is currently doing. They are explained in the client-server message interface. **The state of one VPP does not affect any other VPP.**

The OCV server is responsible for choosing which VPP the purchase request uses. The client application purchase request selects an Account ID which the OCV server uses to look up the next available VPP to use. Because you can group more than 1 VPP to a single account (this will allow multiple outstanding transactions for that account), you cannot guarantee which VPP the OCV server will choose for each transaction.

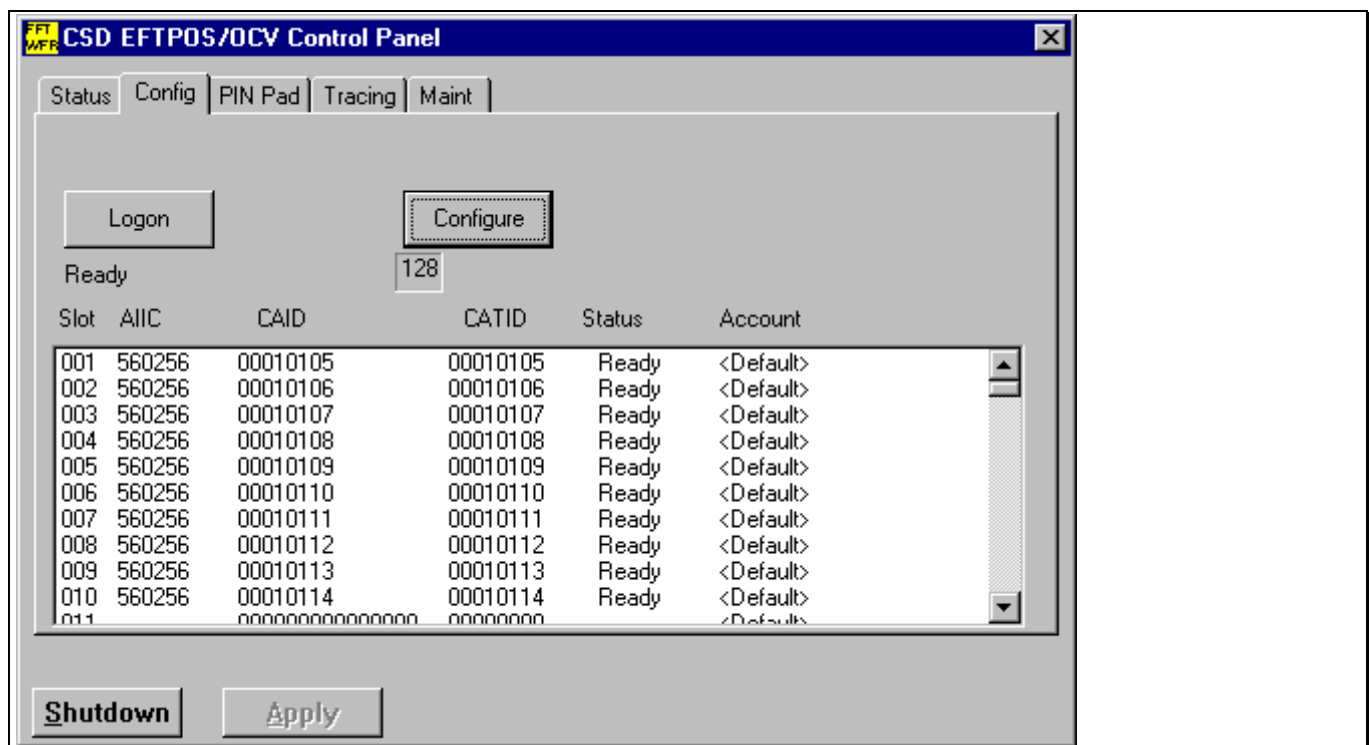
The transaction response will return the Merchant ID of the VPP that was used for the transaction, but this information is not necessarily needed or useful to the client application.

These properties are also configurable for the OCV server.

- **EFT Server Address and EFT Server Port:** This is usually the same as the OCV server TCP/IP address, however it is possible to run the EFT Server (Modem link to bank) on a different machine.

## OCV Server Application

- This server will be responsible for the routing of all credit transaction requests from the clients through the PX-318 PINpad and then out to the bank via the Modem.
- **This OCV server software will run on the Windows NT or Windows 95/98 operating system. The server software will be able to multi-task on the platform provided. The hardware requirements needed for the server are 2 serial ports (one for the PX-318 PINpad, and the other for the modem)**
- All messaging will be via the TCP/IP protocol.
- The server will be capable of accepting up to 128 clients.
- The maximum number of transactions that can be handled by the system at any one time is limited by the line speed of the modem interface to the bank network.
- A server timeout for each transaction can be programmed at the server. This time will typically be around the 45 second mark.
- The server will validate the card number(luhn check), expiry date, and amount fields before sending them to the bank.
- The OCV server will maintain the local totals for each transaction based upon the settlement date returned by the host.



Live OCV server.

## EFT Server

- This software/hardware component is responsible for the communications to the bank.
- The software handles all message header building and caters for CLNP, TPDU or no header.
- The transport medium to connect to the bank can be any of the following:
  1. Leased Line (CLNP or TPDU)
  2. Dial –up (CLNP or TPDU)
  3. X.25
- The EFT server is totally configurable to all of these transport mediums, and the transport mediums are totally transparent to the client and OCV server applications.
- The OCV Installation wizard will configure the EFT Server for the transport medium selected.

The screenshot shows a window titled "CSD EFTPOS/Server Control Panel" with a menu bar containing "Status", "Server", "Routing", "Links", "Lines", and "Trace". The "Status" tab is active. The main area contains several data fields and buttons:

	Current	Avg	Peak	
Clients	0		0	Tx Queued 0
Tx TPS	0	0	0	
Transmitted	0			Discarded Rx 0
Received	0			Tx 0
Start Time	09-May-1998 20:17:04			
Elapsed	0 Days 00:00:13			

Buttons: Reset, Version, Shutdown, Apply

## ***EFT Application***

- The EFT application will use the existing Bank EFTPOS network.
- The application will simulate up to 128 different merchants within the one Ingenico PX-318 PINpad.
- The PINpad is responsible entirely for the EFTPOS transaction. All key handling, message building, and reversal logic handling will be performed within the PX-318 PINpad.
- The EFT application will have the following functionality:
  1. Perform a Credit card transaction (Purchase or Refund) online to the Bank host.
  2. Perform an online Initialisation to the Bank host (either individually or the PINpad will log all of them on at once to establish the session keys).
  3. Provide the client with the status of any of the merchants.
- Every virtual pinpad will enter a logged off state every 7 days since the last logon. If a logon is required, it will be performed automatically when a financial transaction is requested for the virtual pinpad.

### ***Totals***

- The pinpad does not maintain any settlement or transaction totals at all. The pinpad will inform the OCV server of the result of each transaction and the OCV server will maintain a local settlement totals based upon the settlement date returned by the bank in the financial transaction.
- ***It is recommend that the client use the settlement date returned in each transaction response to maintain its own totals.***
- See Appendix C for details on the Electronic Journal File, and how to use this file to obtain totals.

## *Pinpad Display*

- This is not available in the development kit as there is NO pinpad for the development kit. The 'live' kit will contain a PX318 pinpad.
- The pinpad display is only used to provide some feedback on the status of the virtual pinpads. The pinpad display should mimic the OCV server configuration page. The OCV server and the pinpad should never be out of synchronisation.
- The pinpad display will look like:

RDY BSY VER # 1
003 001 501 *--*

- This display indicates that there are 3 virtual terminals that are initialised to the bank. There is currently 1 virtual terminal performing a transaction to the bank. (The transaction could be a logon, reversal, CPAT, or financial). The version number of the pinpad is v5.01. The '\*--\*' is a heartbeat message to indicate that the pinpad is still going.
- The number in the top right-hand corner indicates the pinpad number (up to 4 per server). The other possible values for this field are
  1. LERR which indicates a Link error exists between pinpad and server.
  2. SERR which indicates a problem between OCV server and EFT Server.

# ***Hardware***

## ***Ingenico PX-318 PINpad***

- This is the standard PX318 pinpad as approved by the banks for use on the EFTPOS network.
- The PINpad will be a multi-merchant PINpad. To the bank, it will look as though there are up to 128 different merchants sending messages to the Bank.
- The PINpad will be used for all message construction and encryption. It will also handle all reversal logic.
- NOT required for the Development kit.

## ***Modem***

- The modem will be either leased line or dial-up. The server will control either type.
- It will be a synchronous 1200bps modem which will connect to Bank via the current dial-up network or via Tran\$end ( leased line).
- The currently supported modems for this system are:
  1. Ingenico P66
  2. Intralink Modem
  3. X.25
- NOT required for the Development kit.

## ***PC Hardware Requirements***

- Pentium 100Mhz or better (running Windows 95/98 or NT)
- 16 Megabyte Ram (Win 95/98), 32Megabyte (Windows NT)
- 2 COM Ports (1 for modem, 1 for security device)
- Windows (95/98 or NT 4.0 (service pack 3)
- 1 Ethernet interface controller configured for TCP/IP

## ***Appendix A – Client to Server interface***

- This section describes the message interface between the clients and the server for a simple, generic TCP/IP interface to the OCV server.
- The socket for the TCP/IP interface is **3005**
- All of these messages are passed to and from the server using the TCP/IP protocol.
- All of the bytes in the following messages are in ASCII.
- **NOTE:** Virtual Pinpad #1 is not allowed to be used for transactions. This pinpad is used to identify the system to the bank. The server will return an error if you attempt a transaction on this Virtual Pinpad.
- The ActiveX control is merely a wrapper of these message formats. Anything possible with these message formats is possible with the ActiveX control.
- The following formats define the contents of the TCP/IP packets.

## Transaction Request

The transaction Request will instruct the OCV server to perform a purchase, Status, refund, Pre-Auth or Completion transaction.

	Field	Length	Contents
1.	Start Flag	1	'#'
2.	Length	4	Length of entire message in bytes including start and length fields
3.	Transaction Type	1	'1' indicates Purchase '2' Status Request '4' indicates Refund '7' Pre-Auth-request (Bank specific) '8' Completion Request
4.	Client ID	8	Unique client ID.(leading and trailing spaces are ignored)
5.	Txn reference	16	See note 1
6.	Polled Mode	1	See note 2
7.	Account Number	4	See note 3
8.	Card data	20	A valid card number See Note 4
9.	Card Expiry Date	4	MMYY
10.	Total Amount	12	Leading zero padded implied decimal place
11.	Client Type	1	See Note 6
12.	Authorisation Number	12	The Authorisation number is returned in the Pre-auth. This number connects the original pre-auth with a completion. It should be blank for all other requests.
13.	CVV Field	6	This field is for the Card Verification Value which is required for some credit card schemes. See Note 8
14.	Internet Indicator Field	Var	See Note 7

**Note 1** Each transaction request should contain a unique transaction number. This number can be used to request the status of that particular transaction. Ingenico recommends that this number be presented to the customer as a means of tracing the transaction.

**Note 2** There are 2 modes for the client to talk to the server. If polled mode is set to '1', the server will return the current status immediately. The client must then continually poll the server to obtain the status of the transaction.(It is recommended that a delay of 1 second between polls is maintained).

If the polled mode flag is '0' or space character, the server will return the result when the transaction is complete. The non-polled method is recommended as it is more efficient.

**Note 3** The account number is the pre-defined merchant that this client wishes to use.(these must be set up at installation time).

**Note 4** The card number must be in the following format:  
<Card Number> Space Filled to 20 characters, Left justified

**Note 5** The Status request only requires up to (and including) field 6.

**Note 6** This field MUST be completed by the client. It communicates the origin of the transaction to the bank. Possible values are:

'0' = Secure eCommerce transaction (SSL) or Downgraded – No ACS 'single'

'1' = Telephone Order 'single'

'2' = Mail Order 'single'

'3' = Customer present (Bank specific)

'4' = Mail Order 'recurring'

'5' = Mail Order 'installment'

'6' = 3D secure or SET authenticated transaction – Full ACS 'single'

'7' = Non-authenticated transaction, merchant 3D secure enabled – With or No attempt ACS 'single'

'8' = Non Secure txn (Bank specific)

'9' = Non-authenticated security txn (Bank specific)

'A' = Telephone Order 'recurring'

'B' = Telephone Order 'installment'

'C' = eCommerce 'recurring'

'D' = eCommerce 'installment'

**Note 7** This field is optional. It is required for;

Internet based transactions that use an issuer based authentication scheme eg Visa 3D secure 'VbV' and identified via Client Type 6 or 7 for transmission of the CAVV TAGs and data. Capture of 'Card Check Value' for transmission of the CCV TAG and data.

The contents of this field are potentially bank and card scheme specific.

Refer to Appendix H for further details.

**Note 8.** This field is for backward compatibility only and Bank specific. If not applicable should be space filled.

## Transaction Response [Index](#)

The transaction Response is sent by the OCV server when a transaction result has been obtained from the bank OR a timeout has occurred for a transaction request. The client should match the transaction response with the client ID, and transaction reference number of the transaction request sent by the client.

	Field	Length	Contents
1.	Start Flag	1	'#'
2.	Length	4	Length of entire message in bytes including start and length fields
3.	Transaction Type	1	'3' indicates server response
4.	Client ID	8	Unique client ID.(leading and trailing spaces are ignored)
5.	Txn reference	16	Echoed from original request
6.	Result	1	See note 1
7.	Response Code	2	See appendix B
8.	Response text	20	Text describing the result.
9.	Account	4	The account that was used
10.	STAN	6	See note 2
11.	Authorisation Code	6	Auth Code.
12.	Terminal ID	8	Terminal ID used (CATID)
13.	Merchant ID	15	Merchant ID used (CAID)
14.	Retry	1	Indicates if the transaction should be retried. ('1' = retry, '0' = fatal error)
15.	Settlement Date	8	The settlement date of the txn. (depends upon the bank. This field may not be present in some systems)
16.	Card Bin Number	2	Card Type used
17.	Card Description	20	Card Description.
18.	Pre-Auth Number	12	Number to be used for a completion to match this pre-auth. See Note 3
19.	Acquirer ID	1	The Acquirers ID (for multi-bank systems only)

**Note 1** The result indicates the current status of the transaction. Possible values are:

'0' = Transaction was approved.

'1' = Transaction was declined.

'2' = Transaction is in progress.

'3' = Transaction NOT FOUND.

The response code indicates the reason for the result.

### **Note 2**

The STAN is the unique transaction number returned by the bank for this financial transaction. It is a reference for the bank for the complete details about this transaction.

### **Note 3**

The Pre-auth transaction will return a number for the completion transaction to use to complete the sale.

### **Please Note**

The original OCV server will only accept upto and including fields 17



### ***Logon Request*** [Index](#)

The logon request is sent to the OCV server to invoke a logon for a particular Virtual PINpad. This request should NOT be used by client applications. It is to be used internally by Ingenico applications ONLY.

	<b>Field</b>	<b>Length</b>	<b>Contents</b>
1.	Start Flag	1	'#'
2.	Length	4	Length of entire message in bytes including start and length fields
3.	Transaction Type	1	'9' indicates a logon request
4.	Client ID	8	Unique client ID.(leading and trailing spaces are ignored)
5.	Pinpad Unit ID	1	The number of the physical pinpad which the virtual pinpad is on.
6.	Virtual Pinpad Number	3	The number of the virtual pinpad to attempt the logon.

### ***Logon Response*** [Index](#)

The Logon response will be returned by the OCV server when a logon has been completed by the virtual pinpad or a timeout has occurred.

	<b>Field</b>	<b>Length</b>	<b>Contents</b>
1.	Start Flag	1	'#'
2.	Length	4	Length of entire message in bytes including start and length fields
3.	Transaction Type	1	'9' indicates logon response
4.	Client ID	8	Unique client ID.(leading and trailing spaces are ignored)
5.	Unit ID	1	always '1' for a logon
6.	Virtual Pinpad Number	3	The number of the virtual pinpad the logon was attempted on.
7.	Response Code	2	See Note 1
8.	Response text	20	Text describing the result.

**Note 1** The response code indicates the current status of the transaction. Possible values are:

'00' = Transaction was approved.

'C0' = Logon failed because the slot was not enabled

'C1' = Logon failed because there was no pinpad.

'C2' = The virtual Pinpad was not in READY state.

**Note 2** There is only NON-POLLED mode for the logon request.

**Note 3** The logon request should not be used in a client application. It is for testing, and installation use only. The transaction requests will log the pinpads on if necessary.

### **Bank Settlement Request** [Index](#)

The bank settlement request is sent to the OCV server to ask the bank for a settlement on a particular Virtual PINpad.

This request should NOT be used by client applications. All banks do not support it.

	<b>Field</b>	<b>Length</b>	<b>Contents</b>
1.	Start Flag	1	'#'
2.	Length	4	Length of entire message in bytes including start and length fields
3.	Transaction Type	1	'0' indicates a bank settlement request
4.	Client ID	8	Unique client ID.(leading and trailing spaces are ignored)
5.	Pinpad Unit ID	1	The number of the physical pinpad which the virtual pinpad is on.
6.	Virtual Pinpad Number	3	The number of the virtual pinpad to attempt the bank settlement.
7.	Settlement Date	4	MMDD

### **Bank Settlement Response** [Index](#)

The OCV server will return the Bank Settlement response when a settlement has been completed by the virtual pinpad or a timeout has occurred.

	<b>Field</b>	<b>Length</b>	<b>Contents</b>
1.	Start Flag	1	'#'
2.	Length	4	Length of entire message in bytes including start and length fields
3.	Transaction Type	1	'0' indicates bank settlement response
4.	Client ID	8	Unique client ID.(leading and trailing spaces are ignored)
5.	Unit ID	1	always '1'
6.	Virtual Pinpad Number	3	The number of the virtual pinpad the settlement was attempted on.
7.	Response Code	2	See Note 1
8.	Response text	20	Text describing the result.
9.	Bank Settlement Date	4	MMDD
10.	Debit Count	6	
11.	Debit Amount	12	\$\$\$\$\$\$cc
12.	Credit Count	6	
13.	Credit Amount	12	\$\$\$\$\$\$cc

**Note 1** The response code indicates the current status of the transaction. Possible values are:

'00' = Transaction was approved.

'C0' = Settlement failed because the slot was not enabled

'C1' = Settlement failed because there was no pinpad.

'C2' = the virtual Pinpad was not in READY state.

**Note 2** There is only NON-POLLED mode for this request

### **OCV Server Statistics Request [Index](#)**

This request will retrieve the current OCV server statistics from the OCV server. The sub-code indicates whether the current or permanent statistics are returned.

	<b>Field</b>	<b>Length</b>	<b>Contents</b>
1.	Start Flag	1	'#'
2.	Length	4	Length of entire message in bytes including start and length fields
3.	Transaction Type	1	'A' indicates a Statistics request
4.	Client ID	8	Unique client ID.(leading and trailing spaces are ignored)
5.	Reset Statistics	1	'1' will return stats and then reset them all to 0 Anything else will NOT.
6.	Sub-Code	1	'0' = Return the Current Stats response '1' = Return the Permanent Stats response

### **OCV Server Statistics Response (Current) [Index](#)**

This response is the Current OCV server statistics. It is a list of all the statistics maintained by the OCV server.

	<b>Field</b>	<b>Length</b>	<b>Contents</b>
1.	Start Flag	1	'#'
2.	Length	4	Length of entire message in bytes including start and length fields
3.	Transaction Type	1	'A' indicates Statistics response
4.	Client ID	8	Unique client ID.(leading and trailing spaces are ignored)
5.	Sub-Code	1	'0' = Current Stats. '1' = Permanent Stats
6.	OCV Link Status	1	0= GOOD, 1 = NO LINK , 2= NO SERVER
7.	Link Response Text	20	Textual description of link status (note 2)
8.	OCV Pinpad Status	1	0= GOOD, 1 = LOADING, 2 = NO PINPAD
9.	# Actual Pinpads	1	The Number of physical pinpads connected to the OCV server
10.	# Virtual Pinpads	3	The total number of virtual pinpads connected to the entire OCV server
11.	# txns Transmitted	8	Number of messages transmitted to the bank.
12.	# txns received	8	Number of messages received from the bank.
13.	# current clients	3	The current number of clients connected to the OCV server
14.	Peak # clients	3	The Maximum number of clients ever connected to the OCV server
15.	OCV Start Time	20	The start time of the OCV server with format DD-MMM-YYYY HH:MM:SS
16.	Elapsed Time	20	The Time since last Statistics reset with

			format " xx Days HH:MM:SS"
17.	Qtimeout reject	4	The number of transactions that timed out waiting for a VPP slot to be ready.(logging on)
18.	Parameter reject	4	The number of transactions that were rejected because of bad parameters in the request
19.	Offline Reject	4	The number of transactions that were rejected because of the pinpad being offline.
20.	Busy Reject	4	The number of transactions that were rejected because of no VPP's available at all.
21.	Link Reject	4	The number of transactions that were rejected because of a link error to the bank.
22.	# Status requests	4	The number of status requests received from the clients
23.	EFT-Server Address	15	nnn.nnn.nnn.nnn
24.	EFT-Server Port	4	
25.	Pinpad Comms Ports	12	Each byte represents the COM port of that pinpad. (Max 4 per server). e.g. 001000000000 represent s pinpad on COM1 only
26.	Pinpad Message t/o	3	Value the pinpad timeout value is set to.
27.	Bank Network Name	20	The name of the bank.
28.	Peak VPP Use	3	Peak number of VPP in use
29.	Current VPP Use	3	Current number of VPP in use
30.	Current Txns Per Min	3	
31.	Peak Txns Per Min	3	

## **OCV Server Statistics Response (Permanent) [Index](#)**

This response is a list of all the permanent statistics that are maintained by the OCV server. The permanent statistics are maintained in the registry of the OCV server machine.

	<b>Field</b>	<b>Length</b>	<b>Contents</b>
1.	Start Flag	1	'#'
2.	Length	4	Length of entire message in bytes including start and length fields
3.	Transaction Type	1	'A' indicates Statistics response
4.	Client Id	8	Unique client ID.(leading and trailing spaces are ignored)
5.	Sub-Code	1	'0' = Current Stats. '1' = Permanent Stats
6.	Number of Start times	3	The number of times that the server has been started.
7.	Peak Txns per Minute	30	nnnn Date and time of occurrence
8.	Peak Virtual Pinpads in use	30	nnnn Date and time of occurrence
9.	Peak Pinpad Queue	30	nnnn Date and time of occurrence
10.	Peak Number Clients	30	Nnnn Date and time of occurrence

### ***Virtual Pinpad Configuration Request*** [Index](#)

This command will configure an individual Virtual Pinpad. See the Virtual Pinpad Details section for more details on Virtual Pinpads.

	<b>Field</b>	<b>Length</b>	<b>Contents</b>
1.	Start Flag	1	'#'
2.	Length	4	Length of entire message in bytes including start and length fields
3.	Transaction Type	1	'B' indicates a virtual pinpad config
4.	Client ID	8	Unique client ID.(leading and trailing spaces are ignored)
5.	Pinpad Unit ID	1	The number of the physical pinpad which the virtual pinpad is on.
6.	Virtual Pinpad Number	3	
7.	Network Type	1	'0'=AllC, '1'=NII
8.	Network ID	11	See Note 1
9.	Merchant ID	15	Bank Merchant representation
10.	Terminal ID	8	Bank terminal representation
11.	Account	4	The account number to associate with this VPP
12.	Client Type	1	Not Used
13.	Enable Flag	1	'1' = Enable the pinpad, '0'= Disable.

### ***Virtual Pinpad Configuration Response*** [Index](#)

	<b>Field</b>	<b>Length</b>	<b>Contents</b>
1.	Start Flag	1	'#'
2.	Length	4	Length of entire message in bytes including start and length fields
3.	Transaction Type	1	'B' indicates virtual pinpad config
4.	Client ID	8	Unique client ID.(leading and trailing spaces are ignored)
5.	Unit ID	1	The number of the physical pinpad which the virtual pinpad is on.
6.	Virtual Pinpad Number	3	
7.	Response Code	2	See Note 2
8.	Response text	20	Text describing the result.

**Note 1** The Network type is bank dependent. For AllC, this field must be 6 in length with trailing spaces. For NII, it must be 3 in length with trailing spaces.

**Note 2** The Response code is as follows:

00 = SUCCESS

01 = Slot is Busy

02 = No Pinpad.

03 = Slot Not defined.

04 = Account Not defined.

**Note 3** A VPP Status should be performed after each configuration to ensure the details are correct. There is a possibility that the Security device will reject the request . This may be because it already contains the terminal ID that is being loaded.

### ***Virtual Pinpad Status Request*** [Index](#)

This command will return the current status of the virtual pinpad requested, along with the parameters associated with that VPP. The list of states the virtual Pinpad can be in is listed below.

	<b>Field</b>	<b>Length</b>	<b>Contents</b>
1.	Start Flag	1	'#'
2.	Length	4	Length of entire message in bytes including start and length fields
3.	Transaction Type	1	'C' indicates a virtual pinpad status
4.	Client ID	8	Unique client ID.(leading and trailing spaces are ignored)
5.	Pinpad Unit Number	1	The number of the physical pinpad which the virtual pinpad is on.
6.	Virtual Pinpad Number	3	

### ***Virtual Pinpad Status Response*** [Index](#)

	<b>Field</b>	<b>Length</b>	<b>Contents</b>
1.	Start Flag	1	'#'
2.	Length	4	Length of entire message in bytes including start and length fields
3.	Transaction Type	1	'C' indicates Virtual pinpad status
4.	Client ID	8	Unique client ID.(leading and trailing spaces are ignored)
5.	Unit ID	1	The number of the physical pinpad which the virtual pinpad is on.
6.	Virtual Pinpad ID	3	
7.	Current Status	1	See Note 1
8.	Network Type	1	'0' = AIIIC, '1' = NII
9.	Network ID	11	The number for the network.
10.	Merchant ID	15	
11.	Terminal ID	8	
12.	Client Type	1	Not Used
13.	Enabled	1	'1' = currently enabled, '0'= disabled
14.	Account	4	The account that the VPP is associated with

**Note 1** The current status is of the following type:

- |               |  |
|---------------|--|
| 0 – Busy      | The VPP is busy performing a transaction.    |
| 1 – Not Ready | The VPP has not been logged onto the bank.   |
| 2 – Ready     | The VPP is Ready to perform a transaction.   |
| 3 – Reversal  | The VPP has a reversal to clear to the bank. |
| 4 – Unknown   | Unknown state                                |
| 5 – Logon     | The VPP is currently performing a logon      |

- 6 – Settle                                    The VPP is currently performing a settlement (Not supported)
- 7 – NO Keys                                The VPP has lost its keys. It is unusable.
- X – No Such Virtual PINpad. There is no such virtual pinpad.

### ***OCV Accounts List Request*** [Index](#)

This command retrieves the list of Accounts which are currently setup within the OCV server. See the section on Accounts for an understanding of accounts.

	<b>Field</b>	<b>Length</b>	<b>Contents</b>
1.	Start Flag	1	'#'
2.	Length	4	Length of entire message in bytes including start and length fields
3.	Transaction Type	1	'D' indicates a virtual pinpad status
4.	Client ID	8	Unique client ID.(leading and trailing spaces are ignored)

### ***OCV Accounts List Response*** [Index](#)

	<b>Field</b>	<b>Length</b>	<b>Contents</b>
1.	Start Flag	1	'#'
2.	Length	4	Length of entire message in bytes including start and length fields
3.	Transaction Type	1	'D' indicates Virtual pinpad status
4.	Client ID	8	Unique client ID.(leading and trailing spaces are ignored)
5.	Number of accounts	4	Number of the following to expect. See Note 1
6.	Account Number	4	
7.	Account Name	20	

**Note 1** The OCV server will return a number of these structures.  
The size of this structure will be 18 + (Number of accounts \* 24)

### ***OCV Server Configuration Request*** [Index](#)

This command will allow the configuration of a limited number of OCV server parameters. Most parameters in the OCV server require the server to be reset after alteration, so the list is quite limited.

	<b>Field</b>	<b>Length</b>	<b>Contents</b>
1.	Start Flag	1	'#'
2.	Length	4	Length of entire message in bytes including start and length fields
3.	Transaction Type	1	'E' indicates a virtual pinpad status
4.	Client ID	8	Unique client ID.(leading and trailing spaces are ignored)
5.	EFT Server address	15	the address of the EFT Server in format nnn.nnn.nnn.nnn
6.	EFT-Server Port	4	
7.	Pinpad Timeout	3	May not be supported by all banks

### ***OCV Server Configuration Response*** [Index](#)

	<b>Field</b>	<b>Length</b>	<b>Contents</b>
1.	Start Flag	1	'#'
2.	Length	4	Length of entire message in bytes including start and length fields
3.	Transaction Type	1	'E' indicates Virtual pinpad status
4.	Client ID	8	Unique client ID.(leading and trailing spaces are ignored)
5.	Response	1	'1' = Successful, '0' = Failed

#### Note 1

This command cannot be used to change a remote EFTSRV address to a local address.

### **Account Modification Request** [Index](#)

This command allow for the addition/modification of an account on the OCV server. An account cannot be deleted. If an account has no VPP connected to it, then it is not used.(therefore a delete command is not needed).

	<b>Field</b>	<b>Length</b>	<b>Contents</b>
1.	Start Flag	1	'#'
2.	Length	4	Length of entire message in bytes including start and length fields
3.	Transaction Type	1	'F' indicates an account modification request.
4.	Client ID	8	Unique client ID.(leading and trailing spaces are ignored)
5.	Account Number	4	
6.	Account Name	20	

### **Account Modification Response** [Index](#)

	<b>Field</b>	<b>Length</b>	<b>Contents</b>
1.	Start Flag	1	'#'
2.	Length	4	Length of entire message in bytes including start and length fields
3.	Transaction Type	1	'F' indicates a Account modification response
4.	Client ID	8	Unique client ID.(leading and trailing spaces are ignored)
5.	Response	1	'1' = Successful, '0' = Failed

**NOTE 1:** You can not modify account 0000. This is the DEFAULT account.

### **Totals Request** [Index](#)

The totals request obtains the currently stored totals from the OCV server. These totals are maintained by the OCV server in the registry, and only represent the totals as maintained by the server. They are in NO way a representation from the bank of the settlement totals.

Field	Contents	Length	Description
1.	Start Flag	1	'#'
2.	Length	4	Length of entire message in bytes including start flag and length field.
3.	Transaction Type	1	'5' or '6'
4.	Client ID	8	Unique Client ID. Leading and trailing spaces are ignored.
5.	Polled Mode	1	Poll Mode/ Immediate Response
6.	Day	2	Day Offset see Note 1
7.	Account Number	4	Account

### **Totals Server Response** [Index](#)

Field	Contents	Length	Description
1.	Start Flag		Always '#'
2.	Length	4	Length of entire message in bytes including start flag and length field.
3.	Transaction Type	1	'6' indicates response
4.	Client ID	8	Unique Client ID
5.	ReCo	2	Response Code "00" = Success "XX" = Failure.
6.	Date	8	Date (YYYYMMDD)
7.	Account	4	

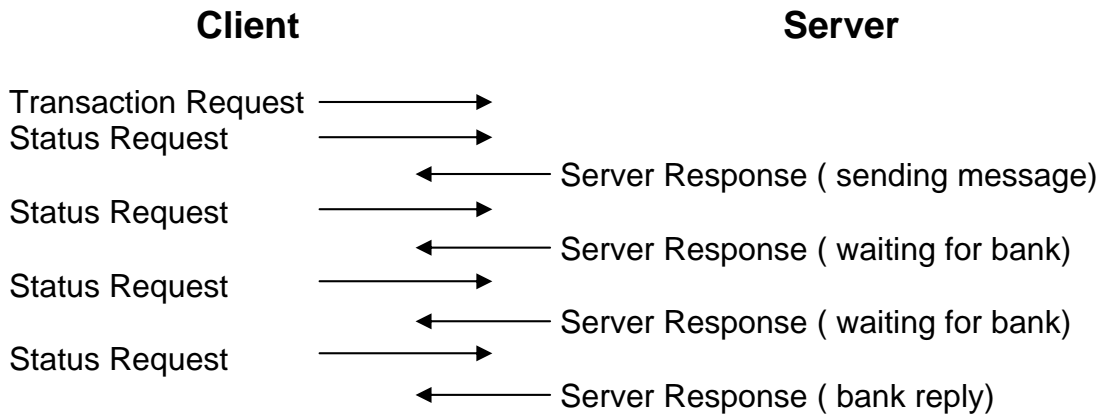
Followed by 16 Card Totals Structures of following format

1.	CardName	20	Card Name
2.	Amt Purchase	12	Leading zero padded implied decimal place
3.	Total Purchase Count	8	Leading zero padded
4.	Amt Refund	12	Leading zero padded implied decimal place
5.	Total Refund Count	8	Leading zero padded

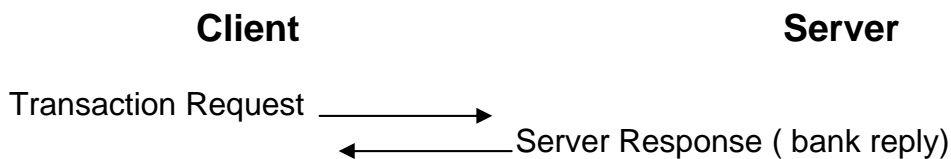
**NOTE1** The Day offset indicates which days totals you wish to obtain. Valid ranges are 0 through to 6 (0 = today, 6 = 6 days ago).

## *Client - Server Message Flow*

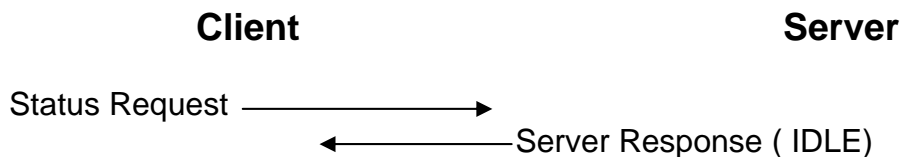
### 1. Client requests financial transaction with server polling enabled.



### 2. Client requests financial transaction with server polling disabled.



### 3. Client requests status of the server.



## Appendix B – Response Codes

Refer to the bank documentation for each particular response code description.

Code	Text	Explanation
00..99	Defined by the bank	These codes are returned by the bank.
A4	LINK ERR	A link error has occurred between the bank and the modem.
A5	PINPAD OFFLINE	The secure PIN Pad unit is not responding.
A6	SERVER BUSY	No free PIN Pad slots were available to service the transaction request.
A7	INVALID MODE	A generic interface request specified an illegal value in "Polled" field.
A8	INVALID AMOUNT	An invalid amount was specified.
A9	INVALID CARD NUMBER	An invalid card number was specified.
AA	INVALID ACCOUNT	An account invalid value for account was specified.
AB	INVALID EXPIRY	An invalid month or year was specified for expiry date.
AC	CARD EXPIRED	A past date was specified for expiry
AD	ACCOUNT ERROR	The specified account is not available on the server.
AE	TIMEOUT	A queued Authorisation timed-out.
AF	RECORD NOT FOUND	A journal lookup did not find the requested transaction.
B1	INVALID REQ TYPE	An invalid request was received
	ERROR 03	There was a STAN mismatch
	ERROR 05	There was an incoming MAC error
T1	CARD UNSUPPORTED	The card was not in the banks CPAT
	MESSAGE TYPE ERROR	The message received was not expected
	MAXREQ EXCEEDED	The maximum number of transactions from this OCX has been exceeded
X1	LINK FAIL NO DIAL TONE	There is no phone line
	NO ANSWER	The NAC did not answer the call
	NO CARRIER	There was a lost connection or an incorrect call establishment
	CNP ERR xxyy	A Tran\$end CNP error was detected. xxyy represents the cause of the fault.
	DIAL ERR	A failed dial has occurred. This may be an incorrect setting.
	PORT ERR	The EFT-server cant talk to the modem
	NO PHONE NUMBER	No configured phone number
U9	NO RESPONSE	A valid response was not received in time from the Bank Host.
V9	TXN NOT ACTIVE	A transaction was returned from the bank late (or unexpected). The transaction had already been declined by the OCV server.
W6	NOT SUPPORTED	The function requested is not supported by the OCV servers bank.

## ***Appendix C – Electronic Journal Format***

- The electronic Journal is the servers record of all transactions performed by the system.
- Logons, Financial transactions, reversals, and error exceptions are all recorded to an electronic journal by the OCV server.
- The journal may be read into a database as a flat file.
- This file is an accurate record of exactly what has occurred on the OCV server. If it is not in the journal files, the transaction did not occur.
  
- The journal will create the following files according to the following rules:
  - There will always be a file called ocv.ejl This file contains the current days transaction information.
  - When the bank settlement date changes, the OCV server will copy this file to a file named ocvprev.ejl. The OCV server will then record all new transactions to the now blank ocv.ejl file.
  - If the file ocvprev.ejl DID exist, the OCV server will copy the original ocvprev.ejl file to a directory called .\jnl The file will be called <date>.ejl where the date is the settlement date of the transactions in the file in the format YYYYMMDD.
  
- The electronic journal files can be used by applications to gather statistics about the transactions on the server, or to reconcile with the client totals. (Note to get accurate totals, it is recommend that you search both ocv.ejl, and ocvprev.ejl for all transactions with the current settlement date. This is because it is possible to get a few transaction from the previous settlement period in the current ocv.ejl file.
  
- Ingenico will be offering a maintenance program to interrogate the EJL files to produce statistics, and reconciliation totals.

### ***Reversals***

- Reversals are transactions that the OCV server has asked the host to reverse. This process is automatic in the OCV server and does not concern the client application.
- The transaction may have been reversed because the bank did not respond to the original transaction, or there is some doubt in the OCV server about the original transaction getting to the bank.
- For the purposes of journalling, ignore all reversals

The format of the journal is as follows:

```
char IdNumber[8];           // ID Number for the journal
char PCDateTime[14];       // PC Date&Time DDMMYYYYHHMMSS
char AccountID[4];        // The account ID selected by the client
char TxnType;             // 'L'= Logon, 'R'= Reversal,
                          // 'E'= Transaction, 'X' = Exception Error

char MsgType[4];          // 0810 = Logon , 0210 = Financial
char Merchant[4];        // Merchant that was used
char Stan[6];            // STAN of the transaction
char Date[6];            // Bank Date (if returned )
char Time[6];            // Bank Time (if returned )
char ReCo[2];            // Response Code for this transaction.
char Success;            // '1' = Successful txn, '0' = failure
char ResponseText[20];   // Text describing the response code
char Catid[8];           // Field 41 identifying the merchant
char Caid[15];           // Field 42 identifying the merchant
char PrCo[6];            // Processing Code of the txn
char Nii[3];             // Nii
char Aiic[11];           // Bank AIIC
char Acquirer;           // Bank type (eg 'X' = ANZ.)
char Elapsed[5];         // Turnaround Time in 10ms increments
char Status;             // New merchant Status
```

A Financial Transaction will also have the following:

```
char Pan[32];             // Card # First 6 digits ... last 3 digits
char AmtTotal[CSD_SIZ_AMT]; // Total Amount of the transaction
char AuthCode[6];        // Auth code if returned by the bank
char Posc[2];            // POSC used in the txn
char Pose[2];            // POSE used in the txn
char Rrn[12];            // RRN returned by the bank
char DateExpiry[4];     // Card Expiry Date
char DateSettlement[8]; // Settlement date returned by bank
char TxnRef[16];        // Unique Number generated by Client
char TxnRefPrivate[20]; // Number sent from OCX client to server
char ClientId[8];       // Client Name
char CardName[20];      // Card Description
char EftTxnTypeStr[16]; // Not Used
char EftTxnType;        // 'P' for Purchase. 'R' for refund
                          //
char AccType[6];        // Not Used
char CardBin Number[2]; // Card BIN number
char CardNameLocal[20]; // Card Name Local
char PreAuthNumber[12]; // The Authorisation Number for a completion
char Data[1];           // The end.
```

An initialisation will have the following:

```
char SwVersion[32];      // Software Version
char PhoneNumber[12];    // Not used in Australia
```

## ***Appendix D – ActiveX Control***

- The ActiveX control merely wraps the Generic interface (as defined in Appendix A).
- This means that the programmer does not have to worry about TCP/IP programming for Windows. The control will take the properties from the user application, and passes them to the server using the generic interface.
- This control will only work for Windows 95/98 and NT development languages (e.g. Delphi, Visual Basic, Visual C++, Builder etc)
- There will also be a Delphi 3 and Builder C++ 3 component made available as well. The interface to this component is the same as the ActiveX control

**Note:**

***Ingenico does not support the ActiveX control.***

***The source and instructions for the ActiveX control are available at***

***[www.pceftpos.com](http://www.pceftpos.com)***

## ***Appendix E – OCV Test Procedures***

When the client interface has been developed, there are a number of important items which must be checked prior to system integration in the EFTPOS network. The bank which is deploying the system may require the developer to have their system accredited by Ingenico. The following is a guide to some of the items which should be checked and tested. After these have been performed, the developer should contact Ingenico (email [cgodden@ingenico.com.au](mailto:cgodden@ingenico.com.au)) and ask for the Ingenico OCV server Test script. We will email out a test script which must be completed and returned to Ingenico. Ingenico will then issue you with an accreditation letter for the OCV server product.

### ***OCV Test Checklist***

1. The complete system requires 2 communications ports. One is for the security device, the other is for the modem.
2. The real system will not return transactions as quickly as the development system. The client **MUST** wait for up to 80 seconds for a reply to an authorisation from the OCV server. If no reply is received within the 80 second period, the client **MUST** perform error recovery as defined in test 3.
3. If the client to server communications has failed for any reason (e.g. Ethernet break, OCV server has been shutdown etc), the client **MUST** send a status request to the OCV server to attempt to find out the exact result of the transaction. **NOTE** The OCV server does not know if the result did not get back to the client successfully. Because of this, the OCV server will be able to answer a status query on a transaction for up to 2 settlement periods.
4. The transaction reference number field in the generic interface **MUST** be incremented for each transaction. The bank may use this as a means of tracing the transaction.
5. The client application **MUST** check the result field in the response message. A '0' will indicate an approved transaction. A '1' will indicate a failed transaction. The Response code field will give a more detailed explanation of what has occurred.
6. The client application is responsible for retrying a transaction. The OCV server may return BUSY which indicates that there are no virtual pinpad slots available for the transaction at the moment. The transaction has **NOT** been declined by the bank and may be retried.
7. The Refund option should be protected by the client.
8. The release system does **NOT** install and register the ActiveX control (if you are using the control). This must be included in the client application.

## **Appendix F – Development Procedures**

If you have any queries or problems please visit the Support web-page first. The address is [www.pceftpos.com](http://www.pceftpos.com) If you can not find what you need there, please drop us an email at Ingenico. Please send it to [cgodden@ingenico.com.au](mailto:cgodden@ingenico.com.au)

The following steps should help you with what to do at the varying steps during your development.

### **1. Initial Development Process and Assessment**

- You should contact a bank which is involved with the system. These banks are listed on the web-site ([www.pceftpos.com.au](http://www.pceftpos.com.au)) They will inform you of the availability of the system for you on there network. (Note. It is possible for you to develop the interface, and then have the bank refuse you connection to there network. Because of this it is always best to check with the bank first)
- Email Ingenico with Company name, address, email address, phone number, contact name, web-site address, and an overview of what the system will be used for.
- Ingenico will email the development OCV server for your use.
- Download the Developers guide documentation from the web-site.
- The documentation will provide you with an overview of how the system works, and how it can be incorporated into you system.
- Email Ingenico with any questions you have on the system before you begin any development. We will answer them quickly, and provide you with any important development advise before you begin.

### **2. Development of the interface.**

- Ingenico will provide you with phone, and email support during the development of the interface. The following web site ( [www.pceftpos.com.au](http://www.pceftpos.com.au) ) will provide you with all the latest developments with the product as well as development assistance.
- There is also a mailing list that is setup to keep developers and customers in touch with the latest developments for the OCV server. To get on the mailing list please send an email to [cgodden@ingenico.com.au](mailto:cgodden@ingenico.com.au) Put the word SUBSCRIBE in the subject and you will be on the list.

### **3. Development Completion**

- When you are completed development, email Ingenico ([cgodden@ingenico.com.au](mailto:cgodden@ingenico.com.au) )and ask for the OCV server Test script.
- Email the completed test script which is generated by the development OCV server along with your details to Ingenico.
- Ingenico will inform you of the acceptance of your system, and issue an accreditation letter.

### **4. Purchasing the OCV server**

- Contact your bank sales representative, and arrange for the purchase of the system you require. You will need to fill in some forms that the bank will supply you.
- You will also need to send a purchase order (on your letter head) through to the bank. The bank will forward this to Ingenico for processing. Ingenico will

invoice you for the purchase of the OCV server. Payment for the system can be via cheque or credit card. Please forward the payment details along with the purchase order to the bank. The system will not be dispatched until payment has been cleared.

- The bank will forward a purchase order through to Ingenico, and we will dispatch the system as soon as the payment has been cleared. The bank will inform you of any lead times due to bank issues (e.g. phone lines, leased lines, merchant details etc) or Ingenico delivery delays.

## **5. Installation of the Full system**

- When you receive the system, follow the installation instructions to install the system. The documentation in the full system will help you through any installation problems.

## Appendix G – Development Suite Differences

The OCVDEV server is an emulation of the full OCV server. Because it is an offline version of the full system there are a few differences which should be noted:

- The Real OCV server needs 2 COM ports. (One for the security device, and one for the modem)
- The Real OCV server DOES NOT visually look like the OCVDEV Server. It will function the same from the client viewpoint though.
- The following statistics are NOT maintained on OCVDEV :
  1. Q Timeout
  2. All the parameter Reject statistics
  3. Pinpad Timeout
  4. PP Com Ports
  5. Peak VPP Used
  6. Current VPP Used
  7. Txns Per Minute
  8. Peak Txns per minute
  9. All Permanent Statistics.

You can set these statistics up on OCVDEV to return whatever you like.

- The Transaction time for the development system will be the same for each transaction.(It is programmable on OCVDEV. The default is set to 4 seconds for response times). Each transaction to the real bank will vary in the amount of time it takes to respond.
- OCVDEV has 50 virtual pinpads. The real system has from 1 to 128 (depending upon which system you purchase)
- The totals for OCVDEV are predefined. The fields are correct however the values in those field will not change. The real system will return the totals as stored by the system.
- OCVDEV will return the cents value of the transaction as the Response Code. (e.g. \$1.00 will return Approved as 00 is the Approved Response Code, \$1.01 will return Decline Error 01 etc...
- There are a couple of special Response codes to help you test real world scenarios. They are:
  1. **Response Code 80** (e.g. \$1.80) will not return a response at all. This is used to simulate the OCV server not responding at all. This should never happen in the real system unless the connection between the client and the server is lost. The client should send a status request after 90 seconds, and the server will then respond with a transaction result.
  2. **Response Code 81** (e.g. \$1.81) will return the Link Failure ( A4) response code. This means that there is a comms problem to the bank.
  3. **Response Code 82** (e.g. \$1.82) will return Pinpad OFFLINE (A5). This means that the security device on the OCV server is missing.
  4. **Response Code 83** (e.g. \$1.83) will return the NO RESPONSE error (U9). This means that the OCV server is running correctly, but the bank did not respond to the transaction that was sent to them. The transaction integrity (reversal) will be handled automatically by the OCV server.

5. **Response Code 84** (e.g. \$1.84) will return Server BUSY (A6). This means that the server could not find an available VPP to use for the transaction. Please retry the transaction shortly.

6. **Response Code 85** (e.g. \$1.85) will return LINK FAIL (X1). This means that the communications to the bank has failed (e.g. no dial tone etc).

- The response texts from OCVDEV may not exactly match the bank response texts. (The bank response texts will vary for each bank anyway)
- The settlement date field in OCVDEV is based upon the actual date in the PC. Some banks will have the settlement date field in this way, and others will base it upon the settlement period.

## ***Appendix H – Internet Transaction Data***

Internet transactions can pass additional data to the acquirer to indicate a pre-approval of the transaction by the issuing bank.

An example of these types of schemes are Visa 3D secure.

Each acquirer that supports the OCV system may require different data sent in this field.

Please consult with Ingenico on the exact data required for each acquirer.

## ***Ingenico Contact Details***

**Phone:** +61 2 9979 9333

**Fax:** +61 2 9979 9751

### ***Internet Access***

Ingenico World Wide Web Home Page:

<http://www.ingenico.com.au>

Support Home page

<http://www.pceftpos.com>

### ***Technical Questions***

Technical questions regarding any part of our systems should be directed to:

OCV Support

Email: [ocv@ingenico.com.au](mailto:ocv@ingenico.com.au)

Craig Godden – Ecommerce Manager

Email: [cgodden@ingenico.com.au](mailto:cgodden@ingenico.com.au)

### ***Sales Enquiries***

Sales enquiries should be directed to:

OCV Support

Email: [ocv@ingenico.com.au](mailto:ocv@ingenico.com.au)

### ***Equipment Maintenance Enquiries***

Any enquiries regarding equipment maintenance should be directed to:

OCV Support

Email: [ocv@ingenico.com.au](mailto:ocv@ingenico.com.au)