

PC EFTPOS ActiveX Reference



REVISION HISTORY

Version	Comment	Date
1.0	First version for PC EFTPOS v5.1	01/11/2002
2.0	Added recommendation for calling methods on worker threads	13/04/2005
3.0	Removed reference to obsolete Appendix A	07/09/2009
3.1	Clarification on results of DoGetLastTransaction in event of power failure.	14/04/2010
3.2	Clarification on MessageType values.	29/04/2010
3.3	Clarification on legacy Success property value for DoGetLastReceipt.	29/03/2011
3.4	Updated Operating Systems supported Removed unused Methods, Properties and Events	31/05/2012
4.0	New InfoEvent to advise POS of truncated PAN on entry Include Cardholder name in transaction result	11/11/2014
4.1	Updated Card Types	18/01/2017
4.2	Removed reference to obsolete method	14/03/2017
4.3	Documented CreditOnly, EnableInfoEvent, InfoEvent and DoCancel	31/07/2017

CONTENTS

1	PRODUCT OVERVIEW.....	6
1.1	INTRODUCTION.....	6
1.2	DESIGN CONSIDERATIONS.....	6
1.3	POS PLATFORM REQUIREMENTS.....	6
2	INTEGRATION OF THE OCX.....	8
2.1	VISUAL BASIC INTEGRATION.....	8
2.2	BORLAND DELPHI INTEGRATION (VERSION 2 OR ABOVE REQUIRED) 8	
2.2.1	Delphi 2:.....	8
2.2.2	Delphi 6:.....	8
2.3	BORLAND C++ BUILDER INTEGRATION.....	9
2.4	VISUAL C++ INTEGRATION.....	9
2.5	DOS AND UNIX APPLICATION INTEGRATION.....	9
3	USING THE ENZEFT ACTIVEX CONTROL.....	10
3.1	OVERVIEW.....	10
3.2	METHODS WORKER THREAD RECOMMENDATION.....	10
3.3	TERMINAL LOGON.....	11
3.4	TRANSACTIONS.....	11
3.5	CHEQUE VERIFICATION.....	12
3.6	TERMINAL SETTLEMENT ENQUIRY & CUTOVER.....	12
3.7	RECEIPT PRINTING.....	12
3.8	POWER-FAILURE RECOVERY.....	13
3.9	AUTO-LOGONS & REVERSALS.....	13
3.10	EFTCLIENT CONTROL PANEL.....	14
3.11	INCORRECT ENZEFT.OCX VERSIONS.....	14
3.12	POS USING EXTERNAL CARD READERS.....	14
3.13	VERSION CONTROL.....	14
4	EXCEPTION HANDLING.....	16
4.1	PROCESS FOR POWER-FAILURE RECOVERY.....	16
4.1.1	Initiate the Transaction:.....	16

4.1.2	Power fail the PC:.....	17
4.1.3	Power on the PC:	17
4.2	TXNREF PROPERTY - PURPOSE	18
4.3	VERSION CONTROL	18
4.3.1	POSVersion	18
4.3.2	ENZEFT.OCX Version.....	18
5	ENZEFT PROPERTIES.....	20
5.1	ACCESSING PROPERTIES.....	20
5.1.1	Visual C++	20
5.1.2	Visual Basic	20
5.2	PROPERTY TYPES.....	20
5.2.1	CURRENCY Types	21
6	ENZEFT METHODS.....	34
6.1	DOCANCEL.....	34
6.2	DOCHEQUEVERIFY	34
6.3	DOCONTROLPANEL	34
6.4	DOGETLASTRECEIPT	35
6.5	DOGETLASTTRANSACTION	35
6.6	DOGETMERCHANTS	35
6.7	DOJOURNALVIEWER	36
6.8	DOLOGON.....	36
6.9	DOQUERYCARD.....	36
6.10	DOREPRINTRECEIPT	37
6.11	DORESET	38
6.12	DOSETTLEMENTCUTOVER	38
6.13	DOSETTLEMENTENQUIRY	38
6.14	DOSTATUS	39
6.15	DOTRANSACTION	39
7	ENZEFT EVENTS	41
7.1	CHEQUEVERIFYEVENT	41
7.2	CONTROLPANELEVENT	41

7.3	GETLASTRECEIPTEVENT	41
7.4	GETLASTTRANSACTIONEVENT	41
7.5	GETMERCHANTSEVENT	42
7.6	INFOEVENT (BSTR INFOTYPE).....	42
7.7	JOURNALVIEWEREVENT.....	42
7.8	INPROGRESS (BOOL INPROGRESS).....	42
7.9	LOGON EVENT	42
7.10	PRINTRECEIPTEVENT.....	43
7.11	QUERYCARDEVENT.....	43
7.12	REPRINTRECEIPTEVENT.....	44
7.13	SETTLEMENT CUTOVER EVENT.....	44
7.14	SETTLEMENT ENQUIRY EVENT.....	44
7.15	STATUS EVENT.....	44
7.16	TRANSACTIONEVENT	45

1 PRODUCT OVERVIEW

1.1 INTRODUCTION

ENZEFT.OCX is the programmer interface to the Integrated PC EFTPOS product for Windows XP (Service Pack3), XP Embedded, Windows 7 (32 & 64 bit) Workstations. ENZEFT.OCX is an OLE-2 compliant ActiveX control that may be utilized by developers with 32-bit development tools such as Borland Delphi and Microsoft Visual Studio. The OLE interface has been chosen for the following reasons:

- It is environment independent - the same control supports different development tools from more than one vendor.
- It follows the event-driven, OO model preferred for Windows development.
- It is an emerging standard for client-server application development on Microsoft platform.

The ENZEFT.OCX ActiveX interface is distributed as part of the ENZ PC EFTPOS kit. The Point of Sale (POS) should utilise the OCX which has been installed with PC EFTPOS, but be still able to run if the OCX is not available.

Note: Do not include ENZEFT.OCX with the POS installation.

1.2 DESIGN CONSIDERATIONS

ENZEFT is not a visible GUI control. EFTPOS user interaction is handled by the PC EFTPOS EFTClient process which must be running to process ENZEFT requests. ENZEFT is designed solely as a standard interface between a POS or other application and the EFTPOS subsystem. The EFTPOS subsystem runs as a completely separate Win32 process and will automatically manage any required operator interaction. The visible EFTPOS controls are part of the EFTPOS Acquirer certification process and are subject to rigorous testing to ensure conformance with the latest EFTPOS specification, therefore it is not desirable for the POS application to directly interact with the EFTPOS subsystem, except via ENZEFT.OCX.

1.3 POS PLATFORM REQUIREMENTS

OPERATING SYSTEM

PROCESSOR

RANDOM ACCESS MEMORY (RAM)

HARD DISK DRIVE (HDD)

CLIENT PC(s)

- PC EFTPOS runs under Microsoft Windows Operating Systems, including Windows XP (Service Pack3), XP Embedded, Windows 7 (32 & 64 bit).
- Pentium 100MHz and above (PII 266MHz and above recommended).
- For machines running Windows XP & Windows 7, at least 64MB.
- At least 30MB free.
- One free DB9 serial COM port or USB port for the PIN pad.

SERVER PC(s)

CLIENT/SERVER PC

- One Ethernet Network Interface Card (NIC) per client PC.
- TCP/IP installed.
- One Ethernet Network Interface Card (NIC) per server PC.
- TCP/IP installed.
- One free DB9 serial COM port (if external dialup modem is to be used).
- PC EFTPOS can run both client and server standalone on the same PC. This requires one free DB9 Serial COM Port (or USB port) for the PIN pad and one free DB9 serial COM port if an external dialup modem is to be used.
- TCP/IP installed.

2 INTEGRATION OF THE OCX

2.1 VISUAL BASIC INTEGRATION

ENZEFT.OCX is a fully event driven design and may be freely used with Windows User Interface designs. Only one instance of the ENZEFT control may be active per workstation at any time. To insert the ENZEFT control into a Visual Basic project, use the "Custom Controls" option under the Tools menu item to display the "Custom Controls" list. Check the box next to ENZEFT OLE Control Module to insert the control into the project toolbox. Then, click on the project toolbox ENZEFT control (an EFTPOS New Zealand 'e' logo) and place the control on the form. Visual Basic 4.0 (32-bit) or higher is required. The ENZEFT developer kit includes a copy of EftDemo, a Visual Basic 4.0 project including source code which demonstrates integration of ENZEFT.OCX with Visual Basic. Download a ZIP file containing source and program EXE of EFTDEMO from the Download Zone.

2.2 BORLAND DELPHI INTEGRATION (VERSION 2 OR ABOVE REQUIRED)

2.2.1 Delphi 2:

- From the Delphi II desktop select the item 'Install' under the 'Components' menu.
- Select the 'OCX' Button.
- Select the 'Register' Button.
- Locate the 'ENZEFT.OCX' file and press 'Open'.
- Select 'ENZEFT OLE Control Module' from the 'Registered Controls' list box.
- Select 'OK'
- Select 'OK' again.
- Save your work if necessary
- Delphi will then recompile its library, this can take some time depending on the speed of the computer. Once finished Delphi will display the component palette again, the control, displayed as an EFTPOS New Zealand 'e' logo should be listed under the 'OCX' tab in the palette. Download a ZIP file containing source and program EXE for EFTDELPH from the Download Zone.

2.2.2 Delphi 6:

- As a minimum, install PC EFTPOS Client
- From Delphi 6, select the 'Import ActiveX Control...' option from the 'Component' menu
- Select 'ENZEFT OLE Control module' from the list of registered components

- Set the other options here if desired, then click 'Install...'
- Choose the destination package and press OK
- In the Package editor, press the 'Compile' button followed by the 'Install' button
- You will now have a new non-visual component which you can place on a form and use just like any other Delphi component.

2.3 BORLAND C++ BUILDER INTEGRATION

Refer to EFTBDR.ZIP, a sample Borland C++ Builder V3.0 project including all source code. Download a self-extracting ZIP file containing source and program EXE for EFTBDR from the Download Zone.

2.4 VISUAL C++ INTEGRATION

Refer to EFTVCC.ZIP, a sample Visual C++ V5.0 project including all source code. Download a ZIP file containing source and program EXE for EFTVCC from the Download Zone.

2.5 DOS AND UNIX APPLICATION INTEGRATION

DOS and UNIX are **not** supported in this version.

3 USING THE ENZEFT ACTIVEX CONTROL

3.1 OVERVIEW

POS applications supporting EFTPOS capability need to support the following features:

- Transactions (Purchase, Cash & Refund)
- Receipt Printing
- Power failure Recovery

Optional:

- Terminal Logon
- Cheque Verification
- Terminal Settlement Enquiry & Cutover

PC EFTPOS does not print receipts directly – please see the section on Receipt Printing for more details.

The following sections describe the tasks that need to be implemented by the POS developer.

All EFTPOS functions are performed by the POS application doing the following steps for each task:

1. Population of required OCX properties (e.g. setting value of **AmtPurchase**)
2. Calling of required OCX method (e.g. calling **DoTransaction**)
3. Waiting for appropriate response event (e.g. **TransactionEvent**)
4. Retrieval of results from required OCX properties (e.g. **ResponseCode**)

PC EFTPOS supports multiple merchants. Most methods therefore require a merchant to be specified prior to calling the method.



NOTE: Most methods require a corresponding event to be received before properties are updated and used by the POS.

Ensure that the POS application receives the associated event **before** using the properties. E.g. When calling **DoGetLastTransaction()**, wait for a **GetLastTransactionEvent()** and then use the properties.

Do not watch or poll properties for value changes.

3.2 METHODS WORKER THREAD RECOMMENDATION

Experience has shown that some POS applications exhibit the POS blocking (hanging) during a PC EFTPOS operation until the resulting event has been received. This is noticeable when grey/blank areas are left on the POS screen after a PC EFTPOS dialog window has been

displayed (usually the card swipe dialog). This occurs when the POS GUI thread is not able to process Windows repaint messages.

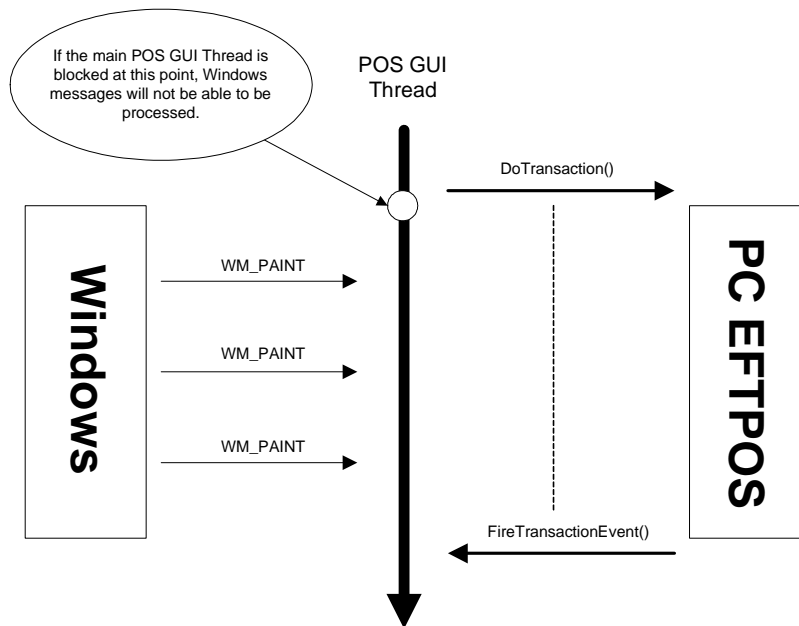
ENZ recommend that POS applications use a separate worker thread to call the OCX in these cases.

POS applications should not implement tight loops that wait for OCX events to be returned. Generally such loops are not required if true event driven behaviour is implemented in the POS.

ENZ discourage POS applications from waiting for OCX property values to change.

Typically the PC EFTPOS OCX methods themselves all return immediately. OCX methods post requests to the EFTClient so that control is returned back to the POS as soon as possible.

However, calling some methods such as DoTransaction will result in other events (e.g. **PrintReceiptEvents**) being received before the matching Fire **TransactionEvent**.



3.3 TERMINAL LOGON

The **DoLogon** method needs to be called to initiate a terminal logon.

Consult the **ENZEFT Methods** section for further details.

3.4 TRANSACTIONS

The following types of EFTPOS transactions are supported by PC EFTPOS:

- Purchase

- Cashout
- Purchase & Cashout
- Refund

The **DoTransaction** method should be called to perform one of the above transaction types. If the terminal is not logged on, this method will attempt a terminal logon as part of the initial transaction process.

The **DoTransaction** method usually causes a dialog box to be displayed that requests a card to be swiped on the PIN pad.

Consult the **ENZEFT Methods** section for further details.

3.5 CHEQUE VERIFICATION

POS applications can verify cheque numbers using the ANZ switch (subject to ANZ cheque verification being enabled at the ANZ Tandem switch).

The **DoChequeVerify** method initiates this function. The POS application is responsible for acquiring the required data such as the cheque serial, branch and account numbers and should populate the appropriate properties prior to calling this method.

Consult the ENZEFT Methods section for further details.

3.6 TERMINAL SETTLEMENT ENQUIRY & CUTOVER

The **DoSettlementEnquiry** or **DoSettlementCutover** methods need to be called to initiate a terminal settlement. **DoSettlementEnquiry** requests the current totals for the merchant since the last cutover. **DoSettlementCutover** requests the current totals for the merchant since the last cutover and also closes off the current settlement if the terminal is allowed to do so.

Consult the **ENZEFT Methods** section for further details.

3.7 RECEIPT PRINTING

The EFTPOS Certification process requires customer and merchant receipts to be produced for all EFTPOS transactions. A merchant receipt is used to capture the customer's signature and is retained by the merchant for signature verified transactions. When a PIN is entered, the merchant receipt is not printed, but is stored in the electronic transaction journal within PC EFTPOS. A customer receipt is always printed for every transaction.

Receipt printing **must** be implemented by the POS application. **PrintReceiptEvent** events will be sent to the POS application whenever a receipt print is required. PC EFTPOS pre-formats required receipts and sends them to the POS application expecting the POS application to print receipts on demand. It is assumed that the receipt text will be printed using a fixed width font. The receipt data is available in the **Receipt** property.

The POS can optionally add headers, footers, indenting etc., but the content of the EFTPOS Receipt should be left intact and printed in full.

Note: The POS application should be prepared to receive a **PrintReceiptEvent** at **any** time. The developer should not assume that **PrintReceiptEvent** events are only generated at the end of requested tasks such as logons and transactions.

The ENZEFT.OCX, like all OCX entities is actually a DLL. DLLs are loaded into the address space of a process (i.e. POS application) when:

1. The DLL is actually loaded (e.g. by explicitly calling the Windows API LoadLibrary() function)
2. A DLL method is called (e.g. this is what happens with Visual Basic applications). This will be the case for most POS applications – i.e. the DLL (OCX) is loaded behind the scenes.

Under power failure circumstances the PC EFTPOS EFTClient may attempt to send a **PrintReceiptEvent** to the ENZEFT.OCX before the POS application has started. The PC EFTPOS EFTClient will queue the print request until the ENZEFT.OCX is loaded into the POS application address space.



NOTE: The **DoGetMerchants** method ***MUST*** be called first when the POS application is started. This will load the OCX and allow any queued **PrintReceiptEvents** to be received.

The POS application must be prepared to process these **PrintReceiptEvents** or any other **PrintReceiptEvents** at any other time, whether expected or not.

Consult the **ENZEFT Methods** section for further details.

3.8 POWER-FAILURE RECOVERY

PC EFTPOS is designed to prevent the customer (or merchant) from being incorrectly debited of funds in the event of a power failure. Special steps need to be taken by the POS application to assure that it agrees with the conclusion of the transaction from the perspective of PC EFTPOS.

Consult the **Exception Handling** section for more information.

3.9 AUTO-LOGONS & REVERSALS

PC EFTPOS may perform an automatic terminal logon or reversal at any time.

Reversals will occur in the event of a power failure or communications failure. A reversal reverses the financial transaction that potentially took place on the ANZ switch.

Reversals and auto-logons are handled in the background by PC EFTPOS. If the POS application calls an ENZEFT.OCX method during auto-logon or reversal processing the following will occur:

- The “PROCESSING NOW” pop-up window will appear indicating that PC EFTPOS is currently busy.
- The requested action will be queued and executed once the background processing is complete.

3.10 EFTCLIENT CONTROL PANEL

The ENZEFT.OCX control can cause PC EFTPOS to display the EFTClient Control Panel using the **DoControlPanel** method. The Control Panel allows the operator to perform supervisory EFTPOS-related tasks.

While all of the functionality implemented by the Control Panel could be implemented by the Point Of Sale application, it is convenient to utilize the **DoControlPanel** method as it provides a full operator interface dialog that has already been certified by the relevant EFTPOS authority.

Alt-F9 usually has the same effect. The **Information** button (found on the **Diagnostics** menu) on the EFTClient Control Panel is also a useful diagnostic tool which will display the versions of EFTClient, ENZEFT.OCX being used by PC EFTPOS.

3.11 INCORRECT ENZEFT.OCX VERSIONS

Whenever the POS developer experiences problems such as PC EFTPOS methods having no effect, the **Information** screen should be consulted. There should be no red/pink lines in the list – these indicate invalid versions.

Developers may experience incorrect ENZEFT.OCX version errors if previous versions of PC EFTPOS have been installed – this can result in multiple copies (different versions) of ENZEFT.OCX residing on a PC. It is important that the correct ENZEFT.OCX is registered. This can be done from the command prompt in Windows by calling the following command in the directory in which the ENZEFT.OCX resides: **regsvr32 enzeft.ocx**

3.12 POS USING EXTERNAL CARD READERS

Card scheme mandates state that card swipe & insertion is to be the responsibility of the cardholder, not the merchant. Removal of this feature avoids ‘double-dipping’ of cards to store card track data. As a result, only PC EFTPOS certified hardware may be used for card reading.

3.13 VERSION CONTROL

The PC EFTPOS system supports its own version control checking where necessary.



NOTE: It is imperative that POS developers do not restrict POS applications to work with particular PC EFTPOS component versions. Failure to do so will make it impossible for ENZ to upgrade/support PC EFTPOS without requiring changes to the POS application software.

ENZ may change component software versions at any time. POS applications must not check the versions of PC EFTPOS components.

4 EXCEPTION HANDLING

A crucial aspect of EFTPOS implementations is correct handling of exception events such as equipment failure during EFTPOS processing. The EFTPOS PIN pad provides for checkpoints and full power fail recovery in accordance with EFTPOS requirements.

The Point Of Sale application must implement power failure recovery to cope with the possibility that an EFTPOS transaction was accepted by the PIN pad. That is, the "point of no return" has been passed - funds have been transferred from customer account to merchant. The ENZEFT.OCX control provides an easy means for the programmer to handle this scenario.

The Point Of Sale application should maintain a non-volatile flag to indicate whether an EFTPOS transaction has been started or not. This flag should be reset after the ENZEFT control has fired the **TransactionEvent** and the POS has determined the success or failure of the transaction. In addition to maintaining this flag, the POS must be able to "tag" each EFTPOS transaction submitted for processing with a POS-unique transaction reference passed in the ENZEFT **TxnRef** property. Upon restarting, the POS should check the flag. If it is set, it indicates that an EFTPOS transaction may have been in progress. The POS may not assume that the EFTPOS transaction necessarily failed in this case. Instead, call **DoGetLastTransaction** to retrieve details of the last EFTPOS transaction processed by the EFTPOS subsystem. **DoGetLastTransaction** will restore the state of the ENZEFT OCX control to the results of the last processed transaction. If the **TxnRef** property matches the **TxnRef** of the last POS transaction and the **LastTxnSuccess** property indicates a successful transaction (set to True), then the POS application must proceed on the basis that the transaction that was in progress during system failure did succeed.



Note: If the **LastTxnSuccess** value is zero (False), the values of all other properties should be assumed invalid. The **TxnRef** property will also be invalid in this case so no comparison should be attempted. The POS is to assume that the transaction was not completed.

A Visual Basic demo application including source code is available to demonstrate how power fail recovery can be implemented in this manner. Request EFTDEMO5 application from the vendor.

4.1 PROCESS FOR POWER-FAILURE RECOVERY

The sequence below shows how to replicate a power-failure scenario.

4.1.1 Initiate the Transaction:

- Once all the transaction details are gathered on the POS application (finalisation of the sale), call **DoTransaction**
- Enter all customer card details e.g. card number, account and PIN

4.1.2 Power fail the PC:

- Switch off the power to the PC. (Switch off the mains to the PC running the POS application and PC EFTPOS – this may seem somewhat brutal but the results under the event of a PC reset are different from a true power-failure.)

4.1.3 Power on the PC:

- On start-up, PC EFTPOS will start as a service. A **PrintReceiptEvent** will be queued and sent to the ENZEFT.OCX (then POS application) as soon as the POS application loads the ENZEFT.OCX. This event will be the power-failure receipt.



Note: The **PrintReceiptEvent** will *not* be sent to the POS if the EFTClient managed to send the usual **PrintReceiptEvent** event to the ENZEFT.OCX prior to power failure. Even though the ENZEFT.OCX may have received the event, this does not necessarily mean that the POS will have received the event prior to power failure.

- PC EFTPOS will then process the pending reversal from the power-failure.



Note: A reversal will only be processed if the EFTClient did not manage to complete the transaction prior to power failure.

- On startup of the POS, call **DoGetLastTransaction**
- Check the status of the **LastTxnSuccess** property:
 - If set to “TRUE” (1)
 - The transaction was completed successfully and the **TxnRef** will be set. (The **TxnRef** number can be checked with the one in the POS registry)
 - Else
 - The **TxnRef** was not set. (You could display an exception error here accordingly)
- Call **DoGetLastReceipt**
- Check the **TxnRef** number with the one in the POS registry.
 - If they are the same:
 - Print the Receipt. (For current transaction)
 - Else
 - Discard the receipt. (From previous transaction)
- To minimise the chance of a system failure, it is recommended that the system be protected by a UPS (Uninterruptible Power Supply). A UPS with an adequate sized battery will allow

the PC, monitor, modem, PIN pad, and printer to be protected from power outages. This will prevent the current sale from being interrupted, and allow transactions for several minutes. If the power-cut is expected to last longer, the system can be gracefully shut down before the UPS battery runs out.

4.2 TXNREF PROPERTY - PURPOSE

The POS must maintain a transaction reference that is **unique** for consecutive EFTPOS transactions. The POS must store this transaction reference in non-volatile storage.

There isn't really a true "non-volatile storage" means on typical hardware, but by simply creating a file to store information, file contents are often lost at the time of a power failure because Windows buffers/caches file storage within memory. Data typically isn't written immediately to hard disk storage. Steps can be taken to request that Windows does flush data back to disk if possible but this varies depending upon the development environment being used.

The Windows registry is a convenient storage medium to use as it is easy to store values and have them flushed back to disk using the RegFlushKey() API. Care must be taken to flush the registry after writing the transaction reference to ensure it is written to disk before an EFT transaction is started. The stored transaction reference can then be used for power failure recovery processing.

4.3 VERSION CONTROL

The EFTPOS Certification process requires EFTPOS subsystem and related POS software to be frozen at a specified release.

4.3.1 POSVersion

The ENZEFT.OCX should also supply the POSVersion information to the EFTPOS acquirer. The POS application must set the **POSVendor**, **POSProductId** (POS application name) and **POSProductVersion** properties as part of its initialization.

4.3.2 ENZEFT.OCX Version

- The PC EFTPOS system supports its own version control checking where necessary.



NOTE: It is imperative that POS developers do not restrict POS applications to work with particular PC EFTPOS component versions. Failure to do so will make it impossible for ENZ to upgrade/support PC EFTPOS without requiring changes to the POS application software.

ENZ may change component software versions at any time. POS applications must not check the versions of PC EFTPOS components.

The **VersionMajor** and **VersionMinor** properties are provided for informational purposes only.

5 ENZEFT PROPERTIES

Relevant properties should be set before calling the control methods. Results are returned in certain properties when control events are fired. The following documentation provides details of all ENZEFT.OCX properties and how to use them. Refer also to the documentation in methods for details of which properties to set before calling a given method. Refer to the documentation in events for details of which properties to check when a given event fires.

Properties are member variables that belong to the ENZEFT.OCX. Depending upon the development environment being used, the property values may be read or set in different ways. Usually the development environment being used will support the IDispatch interface of the ENZEFT.OCX control.

5.1 ACCESSING PROPERTIES

5.1.1 Visual C++

Visual C++ automatically generates a wrapper class for ENZEFT.OCX. For example, consider the **AccountType** property. Visual C++ will automatically generate two methods for accessing **AccountType** – the **GetAccountType()** and **SetAccountType()** methods.

E.g. Assignment in Visual C++

```
ENZEFT.SetAccountType( "CHQ" );
```

5.1.2 Visual Basic

Visual Basic allows the ENZEFT.OCX properties to be accessed as if they were ordinary BASIC variables. e.g. Assignment in Visual Basic:

```
ENZEFT.AccountType = "CHQ"
```

Whichever development environment is used, this document refers simply to the property name – i.e. **AccountType** in the above examples.

5.2 PROPERTY TYPES

The ENZEFT.ODL (Object Definition Language) file defines the interface for the ENZEFT.OCX control, including the property types. The ODL types are not used by the developer. The development environment substitutes its own types for the ODL types as shown in the table below.

ENZEFT.ODL Types	Visual C++ Types
CURRENCY	CY / CURRENCY / COleCurrency

BSTR	CString
Boolean	BOOL
Long	long

Consult the **Win32 SDK OLE Programmer's Reference** for more information on the above types.

This document will refer to the ODL types for consistency across all development environments.

5.2.1 CURRENCY Types

Using CURRENCY types within Visual Basic/C++ is straightforward. Other development platforms may manipulate CURRENCY values in different ways e.g. certain Java platforms treat CURRENCY types as strings.

The OLE defined CURRENCY value consists of two 32 bit (long) values. The first value stores dollar values; the second value stores cents as 'hundredths of cents'.

```
COleCurrency cur;
```

```
cur.ParseCurrency("$135.95", 0);
```

```
ASSERT(cur == COleCurrency(135, 9500));
```

Above is a Visual C++ example that illustrates how a string is converted into a COleCurrency type and what the resulting two 'long' components become.



Note: This information is only provided to assist developers who are experiencing incorrect currency values in their developments.

5.2.1.1 AccountType [BSTR (string)]

Format: Up to 6 characters.

Customer **AccountType** selected for the transaction - can be: "CHQ", "SAV", or "CRD" which equates to the Cheque, Savings, or Credit accounts respectively.

Note: If this property is used by the POS, the POS should also be able to handle "CHEQUE", "SAVING" and "CREDIT" as these may be returned instead of the 3 character values.

APPLICABLE METHODS: DoTransaction

5.2.1.2 ActiveMerchants [BSTR (string)]

Call **DoGetMerchants** to load this property with a list of all the merchants set up in PC EFTPOS. This property has the following structure:

Parameter	Size	Description
Number of Merchants active	3 chars	Indicates the number of merchant records following (Merchant Number and Merchant Name are repeated for each one)
Merchant Number	3 chars	Number of this merchant - this is what needs to be set in the Merchant property
Merchant Name	13 chars	Description as entered in the PC EFTPOS Client. This can be used to aid the user in selecting a merchant from the POS.

For example four merchants active, merchants 1, 3, 4, 8, might look like this:

```
004001MERCHANT ONE 003MERCHANT TWO 004MERCHANT
FOUR008MERCHANT EIGHT
```

APPLICABLE METHODS: **DoGetMerchants**

5.2.1.3 AllowCredit [VARIANT_BOOL (boolean)]

Format: 1=TRUE or 0=FALSE.

Before calling **DoTransaction**, set **AllowCredit** to **TRUE** to inform the Client that the POS wants to allow Credit account selection during a transaction. If the POS wants to restrict the cardholder from using the credit account, set this flag to **FALSE**. If this property is not set, the Client will default the transaction to allow credit account selection.

APPLICABLE METHODS: DoTransaction

5.2.1.4 AmtCash [CY (currency)]

Format: Up to 12 digits.

Amount of cash out transaction.

APPLICABLE METHODS: DoTransaction

5.2.1.5 AmtPurchase [CY (currency)]

Format: Up to 12 digits.

Purchase amount.

APPLICABLE METHODS: DoTransaction

5.2.1.6 AuthCode [BSTR (string)]

Format: 6 characters.

Authorisation code. Returned by **TransactionEvent** if the EFTPOS Acquirer returned the Auth Code field.

APPLICABLE METHODS: DoTransaction

5.2.1.7 Caid [BSTR (string)]

Format: Up to 15 characters.

Card Acceptor ID.

APPLICABLE METHODS: DoTransaction, DoLogon, **DoGetLastTransaction**

5.2.1.8 CardType [BSTR (string)]

Format: Up to 15 characters.

Returned by **TransactionEvent**. Indicates the card type used for the transaction. **CardType** will be one of the following string literals:

Card Description	ANZ OCX Value
AA Rewards	AAREWARDS
AA Smartfuel	AA SMARTFUEL
American Express	AMEX
ATS Card (Canterbury co-operative)	ATS
ASB active	activa
Bankcard	BANKCARD
BP Fuel Card	BP Fuel Card
BP GIFTCARD	BP GIFTCARD
Bartercard	BARTERCARD
Card link	CARD LINK
Card net	CARD NET
Combined Rural Traders (CRT)	CRT
Diners	DINERS
Ecard	ECARD
Ecardz	ECARDZ
Elders	EVIA
Ezipay	EZIPAY
Farmers (Amcal, Placemakers)	FF CARD
Farmlands	FARML
Finzsoft Solutions Ltd	FINZSOFT
Flexipay	FLEXIPAY
Fisher & Paykel Finance	FPPGIFT
Gift Card (Finzsoft Solutions Ltd)	GIFT CARD
Hells Pizza	HELLS PIZZA
Indue Ltd	INDUE
Japan Credit Bureau	JCB
Leftfield	WIN*WIN
Marketsmart	SLOYAL
Mastercard	MASTERCARD

MTA Giftcard	MTA GIFTCARD
RD1.COM	RD1.COM
Shell	SHELL CARD
Superpoints	SUPERPTS
Transactor Technologies	TRANSACTOR
True Rewards	True Rewards
Fisher & Paykel Qcard	QCARD
Indue Ltd	VII
UnionPay International	UnionPay
Visa	VISA
Vodafone Top Up	VODAFONE
Westfield Gift Card	WGIFT
W&I PAYMENT (WINZ)	W&I PAYMENT
All other credit cards	CREDIT
*All other debit cards	DEBIT

APPLICABLE METHODS: DoTransaction

5.2.1.9 ChqAccount [BSTR (string)]

Format: Up to 10 characters.

Cheque Account Number.

APPLICABLE METHODS: DoChequeVerify

5.2.1.10 ChqBranch [BSTR (string)]

Format: Up to 8 characters.

Cheque Branch Number.

APPLICABLE METHODS: DoChequeVerify

5.2.1.11 ChqSerialNumber [BSTR (string)]

Format: Up to 8 characters for TEL cheque verification, or 15 characters for ANZ cheque verification

Cheque Serial Number.

APPLICABLE METHODS: DoChequeVerify

5.2.1.12 CreditOnly [BSTR (string)]

Format: 1 numeric character

Set to 1 to inform PC EFTPOS that the POS wants to force Credit account ONLY during account selection. Debit account options will not be presented. Default value is 0.

Cheque Serial Number.

APPLICABLE METHODS: **DoChequeVerify**

5.2.1.13 Date [BSTR (string)]

Format: Up to 8 characters (DDMMYYYY)

Date of transaction (as returned by host). Returned by **DoTransactionEvent**.

APPLICABLE METHODS: DoTransaction, DoLogon, **DoGetLastTransaction**

5.2.1.14 DateSettlement [BSTR (string)]

Format: Up to 8 characters (DDMMYYYY)

Used as input to **DoSettlementEnquiry** to specify date for which settlement details are to be retrieved. **DateSettlement** should be left blank if the current day's totals are to be retrieved.

APPLICABLE METHODS: **DoSettlementEnquiry**

5.2.1.15 DialogPosition [BSTR (string)]

Format: Up to 12 characters.

Positions the EFTPOS operator panel. For **DialogPosition** to be recognized ensure that **DialogX** and **DialogY** properties are both set to zero. The operator panel is positioned with a 20 pixel gap to between dialog frame and the edge of the screen. To position the dialog panel at a specific X-Y screen position, use **DialogX** and **DialogY** parameters instead. The following relative screen positions are valid for **DialogPosition**:

Value	Description
Centre	Centre of the screen
BottomLeft	Lower left hand corner
BottomCentre	Centred 20 pixels from bottom edge
BottomRight	Centred 20 pixels from bottom right edge
MidLeft	Middle left of screen
MidCentre	Same as Centre
MidRight	Middle right of screen
TopLeft	Top left of screen
TopCentre	Centred at the top of the screen
TopRight	Top right of screen

APPLICABLE METHODS: DoChequeVerify, DoLogon, DoTransaction, DoSettlementCutover, DoSettlementEnquiry, DoQueryCard

5.2.1.16 DialogTitle [BSTR (string)]

Format: Up to 100 characters.

Determines the Title for the EFTPOS Dialog. The title can be a maximum of 100 characters, but this might be truncated depending on the size of the window and the window title font.

APPLICABLE METHODS: DoChequeVerify, DoLogon, DoQueryCard, DoTransaction, DoSettlementCutover, DoSettlementEnquiry

5.2.1.17 DialogX [long (32-bit integer)]

Format: Up to 4 digits.

Positions the EFTPOS operator panel at a specific screen location. Used in conjunction with **DialogY** property to specify the absolute screen coordinates for the top left corner of the selected operator dialog panel. Set **DialogX** to a non-zero value less than the screen width minus the width of the selected dialog. PC EFTPOS will not permit the operator panel to be positioned with any part of the panel off-screen. If the value supplied in **DialogX** is too large, it is automatically reduced to place the rightmost edge of the operator dialog at the edge of the screen. **DialogX** value of 1 represents leftmost edge of screen.



Note: It is recommended that the **DialogPosition** property is used for all new POS integrations. POS applications should no longer use **DialogX** and **DialogY** properties where possible.

APPLICABLE METHODS: DoChequeVerify, DoLogon, DoQueryCard, DoTransaction, DoSettlementCutover, DoSettlementEnquiry

5.2.1.18 DialogY [long (32-bit integer)]

Format: Up to 4 digits.

Positions the EFTPOS operator panel at a specific screen location. Used in conjunction with **DialogX** property to specify the absolute screen coordinates for the top left corner of the selected operator dialog panel. Set **DialogY** to a non-zero value less than the screen height minus the height of the selected dialog. PC EFTPOS will not permit the operator panel to be positioned with any part of the panel off-screen. If the value supplied in **DialogY** is too large, it is automatically reduced to place the rightmost edge of the operator dialog at the edge of the screen. **DialogY** value of 1 represents topmost edge of screen.



Note: It is recommended that the **DialogPosition** property is used for all new POS integrations. POS applications should no longer use **DialogX** and **DialogY** properties where possible.

APPLICABLE METHODS: DoChequeVerify, DoLogon, DoQueryCard, DoTransaction, DoSettlementCutover, DoSettlementEnquiry

5.2.1.19 EnableCashoutDuplicate [VARIANT_BOOL (boolean)]

Format: 1=TRUE or 0=FALSE.

If set to TRUE, an additional duplicate receipt is printed for cashout transactions to capture a customer signature.

APPLICABLE METHODS: DoTransaction

5.2.1.20 EnableInfoEvent [BSTR (string)]

Format: 1 numeric character

Set to 1 to inform PC EFTPOS that the POS wants to receive InfoEvents. Default value is 0.

5.2.1.21 Field48Data [BSTR (string)]

Format: Up to 999 characters.

Proprietary field – For use by prior arrangement of ENZ only

APPLICABLE METHODS: DoTransaction

5.2.1.22 LastTxnSuccess [VARIANT_BOOL (boolean)]

Format: 1=TRUE or 0=FALSE.

Set to TRUE when the request was successful and Accepted by the host as per the following table:

Events	"Accepted" ResponseCode
TransactionEvent, GetLastTransactionEvent	"00" or "08"
LogonEvent	"00"
SettlementCutoverEvent, SettlementEnquiryEvent	"00", "90", or "97"

APPLICABLE METHODS: DoTransaction, DoGetLastTransaction, DoLogon, DoSettlementCutover, DoSettlementEnquiry

5.2.1.23 LoggedOn [VARIANT_BOOL (boolean)]

Format: 1=TRUE or 0=FALSE.

Set to TRUE if the PIN pad is in a logged on state. FALSE if a logon is required. It is never necessary for an application to use **DoLogon** to explicitly logon a PIN pad as auto-logon is automatically performed.

APPLICABLE METHODS: DoStatus

5.2.1.24 Merchant [long (32-bit integer)]

Format: Up to 4 digits.

Indicates the active merchant (1-8). Call **DoGetMerchants** to determine which merchants are available.

Before calling one of the following methods set the merchant required. If no merchant is set then PC EFTPOS Client will default to merchant 1 to process the transaction.

APPLICABLE METHODS: DoLogon, DoTransaction, DoSettlementCutover, DoSettlementEnquiry, DoChequeVerify, DoGetLastReceipt, DoGetLastTransaction, DoQueryCard

5.2.1.25 MessageType [long (32-bit integer)]

Format: Up to 4 digits.

EFTPOS Message Type, this property is set by the PC EFTPOS Client.

Value	Description
210	EFTPOS Financial transaction
510	Settlement
810	Logon

APPLICABLE METHODS: DoTransaction

5.2.1.26 Nii [BSTR (string)]

Format: Up to 12 characters.

EFTPOS Network International Identifier. Not used by all EFTPOS Acquirers.

This property is set by PC EFTPOS Client.

APPLICABLE METHODS: DoStatus

5.2.1.27 Pan [BSTR (string)]

Format: Up to 32 characters.

Primary Account Number. Loaded with customer card number for manually entered transactions. Only credit card details may be manually entered. Due to Scheme restrictions first 6 and last 4 digits will only be populated for this property if present.

APPLICABLE METHODS: DoTransaction

5.2.1.28 PinpadSerialNumber [BSTR (string)]

Format: 8 characters with leading zeros.

The hardware ID or serial number of the attached PIN pad.

APPLICABLE METHODS: DoStatus

5.2.1.29 PinpadVersion [BSTR (string)]

Format: Up to 16 characters.

The PIN pad software name and version.

APPLICABLE METHODS: DoStatus

5.2.1.30 POSProductId [BSTR (string)]

Format: Up to 10 characters.

The name of the POS application that is using ENZEFT.OCX. This property must be initialized by the application before using any ENZEFT.OCX methods. It is only necessary to set this property once throughout the life of the application

APPLICABLE METHODS: DoStatus, DoLogon, DoTransaction, DoSettlementCutover, DoSettlementEnquiry, DoChequeVerify, DoGetLastReceipt, DoGetLastTransaction, DoQueryCard

5.2.1.31 POSVendor [BSTR (string)]

Format: Up to 32 characters.

The name of the vendor of the POS application that is using ENZEFT.OCX. This property must be initialized by the application before using any ENZEFT.OCX methods. It is only necessary to set this property once throughout the life of the application.

APPLICABLE METHODS: DoStatus, DoLogon, DoTransaction, DoSettlementCutover, DoSettlementEnquiry, DoChequeVerify, DoGetLastReceipt, DoGetLastTransaction, DoQueryCard

5.2.1.32 POSVersion [BSTR (string)]

Format: Up to 4 characters. Suggested format is xxxy where xx is major and yy minor version numbers e.g.: Version 1.23 would be set as 0123.

The version of the POS application that is using ENZEFT.OCX e.g.: "V1.23". This property must be initialized by the application before using any ENZEFT.OCX methods. It is only necessary to set this property once throughout the life of the application.

APPLICABLE METHODS: DoStatus, DoLogon, DoTransaction, DoSettlementCutover, DoSettlementEnquiry, DoChequeVerify, DoGetLastReceipt, DoGetLastTransaction, DoQueryCard

5.2.1.33 Ready [VARIANT_BOOL (boolean)]

Format: 1=TRUE or 0=FALSE.

Indicates if the EFTPOS Subsystem has been successfully initialised and is ready to accept commands. Note that a TRUE status for this property does not guarantee that any subsequent method call will be successfully executed, but can be used to avoid unnecessary delays if the EFTPOS subsystem has determined that EFTPOS is not currently available (PIN pad offline or processing a reversal, PC EFTPOS Client process not running etc.) See also Installed property.

APPLICABLE METHODS: None - property is continuously updated by ENZEFT.OCX

5.2.1.34 Receipt [BSTR (string)]

Formatted EFTPOS receipt information. The **Receipt** property is loaded with a pre-formatted EFTPOS receipt. This is available during the **GetLastReceipt Event** and **PrintReceiptEvent** events. In the latter case, the receipt must be printed by the POS.

APPLICABLE METHODS: DoGetLastReceipt, DoTransaction, DoChequeVerify, DoLogon, DoSettlementCutover, DoSettlementEnquiry

5.2.1.35 ResponseCode [BSTR (string)]

Format: Up to 2 characters.

EFTPOS **ResponseCode**.

Financial transactions may return a 00, 05, 08, Y1 or Y3 **ResponseCode** when they are approved. All other **ResponseCodes** listed in the table below reflect financial transaction failure.

Code	Description
A1	Recursive Call
A2	General Failure
A4	Invalid Merchant
A5	Invalid Dialog
A7	OCX Message too small
A8	Internal Error 101
A9	Internal Error 109
AA	PIN pad Already In Use
AX	Unknown Error
B1	Invalid Timeout
B2	Unsupported Operation
B3	Client Offline
B4	OCX Message too big
B5	Invalid Amount
B6	Invalid Dialog Params
B7	Invalid TxnType
B8	Invalid TxnRef
B9	Invalid Branch
BY	PIN pad busy
C0	Invalid Serial Number
C1	Invalid Account
C2	Invalid Pan Source
C3	Client Version Mismatch
C4	Duplicate ENZEFT
C6	Client Not Installed
C7	Win Create Failed

C8	Unable to connect to EFT Server
PF	PIN pad offline
T5	Declined the signature from an "Accept with Signature" transaction
T7	Customer cancelled transaction via PIN pad cancel key
T8	POS Operator cancelled transaction via operator cancel key
U9	Timeout awaiting host response
VO	OCX Version error
VS	Server Version
VP	PIN pad Version
Y8	Declined Cheque Auth transaction. TEL always return an "00" for either an accepted or declined. The operator can control the fate of the final outcome of this transaction
Y9	Duplicate Transaction
X0	Timeout waiting for operator input - no rec
X1	Communications error (Server not present)
X2	Error during dial/connect
X3	Error during HDLC phase

APPLICABLE METHODS: DoTransaction, DoChequeVerify, DoGetLastTransaction, DoLogon, DoSettlementCutover, Do Settlement Inquiry

5.2.1.36 ResponseText [BSTR (string)]

Format: Up to 20 characters.

ResponseText associated with the EFTPOS **ResponseCode**.

APPLICABLE METHODS: DoLogon, DoTransaction, DoSettlementCutover, DoSettlementEnquiry, DoGetLastTransaction, DoQueryCard

5.2.1.37 ResponseType [long (32-bit integer)]

Format: Up to 4 digits.

Last Transaction message type.

Value	Description
210	EFTPOS Financial transaction
510	Settlement
810	Logon

APPLICABLE METHODS: DoGetLastTransaction

5.2.1.38 Stan [long (32-bit integer)]

Format: Up to 6 digits.

EFTPOS Systems Trace Audit Number (STAN). Ranges from 1 to 999999.

APPLICABLE METHODS: DoGetLastTransaction, DoGetLastReceipt, DoTransaction, DoLogon, DoSettlementEnquiry, DoSettlementCutover

5.2.1.39 Success [VARIANT_BOOL (boolean)]

Format: 1=TRUE or 0=FALSE.

Indicates success or failure of the relevant method. Each method fires an event to indicate that it is complete. If there were errors during the processing of the method, the Success property will be set to FALSE during the associated event. To determine if a financial transaction or settlement was accepted or declined, use the **LastTxnSuccess** property.

APPLICABLE METHODS: All

5.2.1.40 Time [long (32-bit integer)]

Format: Up to 6 digits, HHMMSS format. An empty string will be returned if a valid host response was not received. Otherwise the transaction time at which the transaction was completed by the Bank host will be returned.

APPLICABLE METHODS: DoTransaction

5.2.1.41 Timeout [long (32-bit integer)]

Format: Up to 2 digits, Value is from 10-99 seconds. Default is 30 seconds and is not usually altered.

EFTPOS transaction time-out. Determines the maximum amount of time that the PIN pad will wait for a reply from the EFTPOS network after a message has been transmitted.

APPLICABLE METHODS: DoLogon, Do Settlement, DoTransaction

5.2.1.42 Track2 [BSTR (string)]

Format: Up to 37 characters.

Contents of magnetic card track 2 after calling **DoQueryCard**.



NOTE: This property will be returned only if card is present in the PCE_CPTFILE.DAT file. This file is encrypted and is maintained solely by ENZ. If additional loyalty/gift/third party cards are to be supported using **Query Card**, please request that ENZ add the required card details to the PCE_CPTFILE.DAT..DAT file.

APPLICABLE METHODS: DoQueryCard

5.2.1.43 Track1 [BSTR (string)]

Format: Up to 128 characters.

Contents of magnetic card track 1 after calling **DoQueryCard**.



NOTE: This property will be returned only if card is present in the PCE_CPTFILE.DAT file. This file is encrypted and is maintained solely by ENZ. If additional loyalty/gift/third party cards are to be supported using **Query Card**, please request that ENZ add the required card details to the PCE_CPTFILE.DAT.

APPLICABLE METHODS: DoQueryCard

5.2.1.44 TxnRef [BSTR (string)]

Format: Up to 16 characters.

Application supplied reference. **TxnRef** can contain any ASCII character between the values 32_{decimal} to 127_{decimal}. *Important! TxnRef must be unique for consecutive calls to DoTransaction.*

TxnRef can contain any ASCII character between the values 32_{decimal} (' ', space character) to 127_{decimal} ('~ ' tilde character).

APPLICABLE METHODS: DoLogon, DoTransaction, DoSettlementCutover, DoSettlementEnquiry, DoChequeVerify, DoGetLastReceipt, DoGetLastTransaction, DoQueryCard

5.2.1.45 TxnType [BSTR (string)]

Format: Up to 1 character.

Indicates the type of transaction to be performed when calling **DoTransaction**.

Value	Description
"P"	Purchase/Purchase + Cash/Cash out
"R"	Refund

APPLICABLE METHODS: DoTransaction

5.2.1.46 Version Major, Version Minor, Version Revision [long (32-bit integer)]

Format: Up to 2 digits.

Major version number, minor version number, and revision level for ENZEFT control respectively. Refer to section **Version Control** for further information.

APPLICABLE METHODS: none

6 ENZEFT METHODS

ENZEFT.OCX control offers a number of methods to initiate transactions, alter PIN pad settings and retrieve information. A call to a method always returns immediately. The methods do not return a status, but notify the caller of completion or error conditions via an event. All methods have corresponding events. An event will be generated when the method processing is complete. For example, a successful call to **DoCardSwipe** method will result in a **CardSwipeEvent** being fired after a short delay. The event driven nature of the control means that the calling thread is free to process other activity while the event is awaited. An event is always fired, even if the EFTPOS system is unable to perform the requested event. The **Success**, **ResponseCode** and **ResponseText** properties are set before the event fires and provide indication to caller of the result of the method.

6.1 DOCANCEL

Instructs the PIN pad to cancel the current transaction. This method is only valid up to and including account selection. It should not be assumed that this has been successful - completion of the original request may continue. However, if successful the Transaction Event will be sent with a ResponseCode of "T8". No matching cancel event should be expected.

Parameter	Description
Merchant	Merchant number (1-8)

6.2 DOCHEQUEVERIFY

Instructs the PIN pad to perform a cheque verification transaction. The following properties must be loaded with valid values before calling **DoChequeVerify**:

Parameter	Description
AmtPurchase	The Cheque Amount
ChqSerialNumber	The serial number of the cheque 1-6 digits
ChqBranch	The bank and branch of the cheque maximum 8 digit
ChqAccount	The account number of the cheque (maximum 10 digit)
Merchant	Merchant number (1-8)
PanSource	If the cheque has been read by a MICR reader must be set to "S", other set to blank or 'K' to indicate manual entry of cheque digits.

6.3 DOCONTROLPANEL

Causes a Control Panel to be displayed, allowing the user to initiate EFTPOS operations and perform EFTPOS configuration (passwords are required for EFTPOS configuration). Event **ControlPanelEvent** is fired when the command completes and the control panel has been successfully displayed. While the control panel is active, any further ENZEFT.OCX method calls

will fail. The operator can dismiss the EFTPOS control panel. When the control panel is dismissed, ENZEFT.OCX methods are once again enabled.

6.4 DOGETLASTRECEIPT

Retrieves last receipt produced by the PIN pad. Set Merchant property before calling this method. **GetLastReceiptEvent** is fired when this operation completes or times-out. Use **DoGetMerchants** to determine which merchants are available.

Parameter	Description
Merchant	Merchant Number (1-8).

6.5 DOGETLASTTRANSACTION

Retrieves details of the last transaction processed by the PIN pad. No input properties are required by **DoGetLastTransaction**. This method could be used by a Point Of Sale application to determine whether a transaction was successful or not following a power failure while an EFTPOS transaction was in progress. Refer to Exception Handling for further information.

The POS application should in this case check the contents of **LastTxnSuccess** and **TxnRef** properties after a successful call to **DoGetLastTransaction**. A value of TRUE in **LastTxnSuccess** indicates the EFTPOS transaction retrieved by **DoGetLastTransaction** was successful. Note that the Success property refers to the outcome of the call to **DoGetLastTransaction**, not to the result of the EFTPOS transaction retrieved by **DoGetLastTransaction** - the result of the original transaction is held in **LastTxnSuccess**. Event **GetLastTransactionEvent** is fired when this operation completes. If a transaction was recovered, the Success property will be set to TRUE. If Success is set to true, all other properties have been reset to the values they had when the last transaction completed. Check the Stan and **TxnRef** properties to determine if this is the same transaction as the POS has recorded.

The properties returned by this event are the same as for **TransactionEvent** - refer to **DoTransaction** method documentation for details. Note that the Receipt property is not loaded by **DoGetLastTransaction**. Use **DoGetLastReceipt** method to retrieve receipt data for a specified merchant.

Parameter	Description
Merchant	Merchant Number (1-8).

6.6 DOGETMERCHANTS

Queries the PC EFTPOS Client for a list of merchants that are currently active. **GetMerchantsEvent** is fired when this operation completes or times-out.

Parameter	Description
POSName	POSName. E.g. "OCX-POS"

POSVersion	POSVersion. E.g. "0020"
------------	-------------------------

6.7 DOJOURNALVIEWER

Invokes electronic journal viewer utility. **JournalViewerEvent** fires when this method has completed. In the current release, this method has the same effect as **DoControlPanel**.

Parameter	Description
Merchant	Merchant required or 001
POSName	POS Name. E.g. "OCX-POS"
POSVersion	POS Version. E.g. "0020"

6.8 DOLOGON

Initiates an EFTPOS Logon, regardless of whether the PIN pad is already logged on or not. Set the Merchant property before calling **DoLogon**. Use **DoGetMerchants** to determine which merchants are available. Event **LogonEvent** is fired when this operation completes.

Parameter	Description																						
Merchant	Merchant required or 001																						
POSName	POSName. E.g. "OCX-POS"																						
POSVersion	POSVersion. E.g. "0020"																						
DialogX	Window "X" coordinate of top left hand corner of the Transaction Dialog Box in Pels.																						
DialogY	Window "Y" coordinate of top left hand corner of the Transaction Dialog Box in Pels.																						
DialogPOS	Positions the EFTPOS operator panel. DialogPosition: <table border="0"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Centre</td> <td>Centre of the screen</td> </tr> <tr> <td>BottomLeft</td> <td>Lower left hand corner</td> </tr> <tr> <td>BottomCentre</td> <td>Centred 20 pixels from bottom edge.</td> </tr> <tr> <td>BottomRight</td> <td>Centred 20 pixels from bottom right edge.</td> </tr> <tr> <td>MidLeft</td> <td>Middle left of screen</td> </tr> <tr> <td>MidCentre</td> <td>Same as Centre</td> </tr> <tr> <td>MidRight</td> <td>Middle right of screen</td> </tr> <tr> <td>TopLeft</td> <td>Top left of screen</td> </tr> <tr> <td>TopCentre</td> <td>Centred at the top of the screen</td> </tr> <tr> <td>TopRight</td> <td>Top right of screen</td> </tr> </tbody> </table>	Value	Description	Centre	Centre of the screen	BottomLeft	Lower left hand corner	BottomCentre	Centred 20 pixels from bottom edge.	BottomRight	Centred 20 pixels from bottom right edge.	MidLeft	Middle left of screen	MidCentre	Same as Centre	MidRight	Middle right of screen	TopLeft	Top left of screen	TopCentre	Centred at the top of the screen	TopRight	Top right of screen
Value	Description																						
Centre	Centre of the screen																						
BottomLeft	Lower left hand corner																						
BottomCentre	Centred 20 pixels from bottom edge.																						
BottomRight	Centred 20 pixels from bottom right edge.																						
MidLeft	Middle left of screen																						
MidCentre	Same as Centre																						
MidRight	Middle right of screen																						
TopLeft	Top left of screen																						
TopCentre	Centred at the top of the screen																						
TopRight	Top right of screen																						

6.9 DOQUERYCARD

The PIN pad will prompt "ENTER CARD" and read the inserted magnetic card. **Track2** contents are returned in **Track2** property. Load the **DialogX** and **DialogY** parameters before calling

DoQueryCard to ensure the correct operator dialog box is displayed at the desired screen location. Event **QueryCardEvent** is fired when this operation completes or times-out.



NOTE: Only cards present in the PCE_CPTFILE.DAT file will be returned in this response. This is an encrypted file maintained by ENZ. Please contact ENZ to include additional cards that are not supported.

Parameter	Description																						
Merchant	Merchant required or 001																						
POSName	POS Name. E.g. "OCX-POS"																						
POSVersion	POS Version. E.g. "0020"																						
DialogX	Window "X" coordinate of top left hand corner of the Transaction Dialog Box in Pels.																						
DialogY	Window "Y" coordinate of top left hand corner of the Transaction Dialog Box in Pels.																						
DialogPOS	Positions the EFTPOS operator panel. DialogPosition: <table border="0"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Centre</td> <td>Centre of the screen</td> </tr> <tr> <td>BottomLeft</td> <td>Lower left hand corner</td> </tr> <tr> <td>BottomCentre</td> <td>Centred 20 pixels from bottom edge.</td> </tr> <tr> <td>BottomRight</td> <td>Centred 20 pixels from bottom right edge.</td> </tr> <tr> <td>MidLeft</td> <td>Middle left of screen</td> </tr> <tr> <td>MidCentre</td> <td>Same as Centre</td> </tr> <tr> <td>MidRight</td> <td>Middle right of screen</td> </tr> <tr> <td>TopLeft</td> <td>Top left of screen</td> </tr> <tr> <td>TopCentre</td> <td>Centred at the top of the screen</td> </tr> <tr> <td>TopRight</td> <td>Top right of screen</td> </tr> </tbody> </table>	Value	Description	Centre	Centre of the screen	BottomLeft	Lower left hand corner	BottomCentre	Centred 20 pixels from bottom edge.	BottomRight	Centred 20 pixels from bottom right edge.	MidLeft	Middle left of screen	MidCentre	Same as Centre	MidRight	Middle right of screen	TopLeft	Top left of screen	TopCentre	Centred at the top of the screen	TopRight	Top right of screen
Value	Description																						
Centre	Centre of the screen																						
BottomLeft	Lower left hand corner																						
BottomCentre	Centred 20 pixels from bottom edge.																						
BottomRight	Centred 20 pixels from bottom right edge.																						
MidLeft	Middle left of screen																						
MidCentre	Same as Centre																						
MidRight	Middle right of screen																						
TopLeft	Top left of screen																						
TopCentre	Centred at the top of the screen																						
TopRight	Top right of screen																						

6.10 DOREPRINTRECEIPT

Instructs the PIN pad to print a copy of the last financial receipt. Set the Merchant property before calling **DoReprintReceipt**. Event **RePrintReceiptEvent** is fired when this operation completes.

Parameter	Description
Merchant	Merchant required or 001
POSName	POS Name. E.g. "OCX-POS"
POSVersion	POS Version. E.g. "0020"

6.11 DORESET

Resets all ENZEFT control properties to default settings. No properties are required to be set before calling **DoReset**.

6.12 DOSETTLEMENTCUTOVER

Performs an EFTPOS settlement cutover. Set the **Merchant** property before calling **DoSettlementCutover**. The **Success** property will be set to TRUE when the **SettlementCutoverEvent** fires if the operation was successful. Returned settlement receipt data for printing would then be present in property **Receipt**.

Parameter	Description																						
Merchant	Merchant for which settlement cutover is to be performed																						
POSName	POS Name. E.g. "OCX-POS"																						
POSVersion	POS Version. E.g. "0020"																						
DialogX	Window "X" coordinate of top left hand corner of the Transaction Dialog Box in Pels.																						
DialogY	Window "Y" coordinate of top left hand corner of the Transaction Dialog Box in Pels.																						
DialogPOS	Positions the EFTPOS operator panel. DialogPosition: <table border="0"> <tr> <td>Value</td> <td>Description</td> </tr> <tr> <td>Centre</td> <td>Centre of the screen</td> </tr> <tr> <td>BottomLeft</td> <td>Lower left hand corner</td> </tr> <tr> <td>BottomCentre</td> <td>Centred 20 pixels from bottom edge.</td> </tr> <tr> <td>BottomRight</td> <td>Centred 20 pixels from bottom right edge.</td> </tr> <tr> <td>MidLeft</td> <td>Middle left of screen</td> </tr> <tr> <td>MidCentre</td> <td>Same as Centre</td> </tr> <tr> <td>MidRight</td> <td>Middle right of screen</td> </tr> <tr> <td>TopLeft</td> <td>Top left of screen</td> </tr> <tr> <td>TopCentre</td> <td>Centred at the top of the screen</td> </tr> <tr> <td>TopRight</td> <td>Top right of screen</td> </tr> </table>	Value	Description	Centre	Centre of the screen	BottomLeft	Lower left hand corner	BottomCentre	Centred 20 pixels from bottom edge.	BottomRight	Centred 20 pixels from bottom right edge.	MidLeft	Middle left of screen	MidCentre	Same as Centre	MidRight	Middle right of screen	TopLeft	Top left of screen	TopCentre	Centred at the top of the screen	TopRight	Top right of screen
Value	Description																						
Centre	Centre of the screen																						
BottomLeft	Lower left hand corner																						
BottomCentre	Centred 20 pixels from bottom edge.																						
BottomRight	Centred 20 pixels from bottom right edge.																						
MidLeft	Middle left of screen																						
MidCentre	Same as Centre																						
MidRight	Middle right of screen																						
TopLeft	Top left of screen																						
TopCentre	Centred at the top of the screen																						
TopRight	Top right of screen																						

6.13 DOSETTLEMENTENQUIRY

Performs an EFTPOS settlement Enquiry. Set the **Merchant** property before calling **DoSettlementEnquiry**. The **Success** property will be set to TRUE when the **SettlementEnquiryEvent** fires if the operation was successful. Returned settlement receipt data for printing would then be present in property **Receipt**.

Parameter	Description
Merchant	Merchant for which settlement enquiry is to be performed
POSName	POS Name. E.g. "OCX-POS"

POSVersion	POS Version. E.g. "0020"																						
DateSettlement	Settlement date requested, DDMMYYYY																						
DialogX	Window "X" coordinate of top left hand corner of the Transaction Dialog Box in Pels.																						
DialogY	Window "Y" coordinate of top left hand corner of the Transaction Dialog Box in Pels.																						
DialogPOS	Positions the EFTPOS operator panel. DialogPosition: <table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Centre</td> <td>Centre of the screen</td> </tr> <tr> <td>BottomLeft</td> <td>Lower left hand corner</td> </tr> <tr> <td>BottomCentre</td> <td>Centred 20 pixels from bottom edge.</td> </tr> <tr> <td>BottomRight</td> <td>Centred 20 pixels from bottom right edge.</td> </tr> <tr> <td>MidLeft</td> <td>Middle left of screen</td> </tr> <tr> <td>MidCentre</td> <td>Same as Centre</td> </tr> <tr> <td>MidRight</td> <td>Middle right of screen</td> </tr> <tr> <td>TopLeft</td> <td>Top left of screen</td> </tr> <tr> <td>TopCentre</td> <td>Centred at the top of the screen</td> </tr> <tr> <td>TopRight</td> <td>Top right of screen</td> </tr> </tbody> </table>	Value	Description	Centre	Centre of the screen	BottomLeft	Lower left hand corner	BottomCentre	Centred 20 pixels from bottom edge.	BottomRight	Centred 20 pixels from bottom right edge.	MidLeft	Middle left of screen	MidCentre	Same as Centre	MidRight	Middle right of screen	TopLeft	Top left of screen	TopCentre	Centred at the top of the screen	TopRight	Top right of screen
Value	Description																						
Centre	Centre of the screen																						
BottomLeft	Lower left hand corner																						
BottomCentre	Centred 20 pixels from bottom edge.																						
BottomRight	Centred 20 pixels from bottom right edge.																						
MidLeft	Middle left of screen																						
MidCentre	Same as Centre																						
MidRight	Middle right of screen																						
TopLeft	Top left of screen																						
TopCentre	Centred at the top of the screen																						
TopRight	Top right of screen																						

6.14 DOSTATUS

Queries EFTPOS PIN pad status including software version and terminal local totals. Set the Merchant property before calling **DoStatus**. Event **StatusEvent** is fired when this operation completes.

6.15 DOTRANSACTION

Instructs the PIN pad to perform a financial transaction. Event **TransactionEvent** is fired when the transaction completes. The following properties must be loaded with valid values before calling **DoTransaction**:

Parameter	Description
AmtPurchase	Purchase amount
AmtCash	Cash out Amount
CutReceipt	Determines whether a "cut" command is sent to the printer following the receipt print data.
DialogX	Window "X" coordinate of top left hand corner of the Transaction Dialog Box in Pels. If this value is such that EFTPOS-32 Client cannot fit the dialog box on screen it is automatically adjusted to a valid value. A value of "0" for both DialogX and DialogY will instruct EFTPOS-32 Client to centre the EFTPOS dialog in the screen.

DialogY	Window "Y" coordinate of top left hand corner of the Transaction Dialog Box in Pels. If this value is such that EFTPOS-32 Client cannot fit the dialog box on screen it is automatically adjusted to a valid value. A value of "0" for both DialogX and DialogY will instruct EFTPOS-32 Client to centre the EFTPOS dialog in the screen.																						
DialogPOS	Positions the EFTPOS operator panel. For DialogPosition to be recognized ensure that DialogX and DialogY properties are both set to zero. The operator panel is positioned with a 20 pixel gap to between dialog frame and the edge of the screen. To position the dialog panel at an arbitrary X-Y screen position, use DialogX and DialogY parameters instead. The following relative screen positions are valid for DialogPosition: <table border="0"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Centre</td> <td>Centre of the screen</td> </tr> <tr> <td>BottomLeft</td> <td>Lower left hand corner</td> </tr> <tr> <td>BottomCentre</td> <td>Centred 20 pixels from bottom edge.</td> </tr> <tr> <td>BottomRight</td> <td>Centred 20 pixels from bottom right edge.</td> </tr> <tr> <td>MidLeft</td> <td>Middle left of screen</td> </tr> <tr> <td>MidCentre</td> <td>Same as Centre</td> </tr> <tr> <td>MidRight</td> <td>Middle right of screen</td> </tr> <tr> <td>TopLeft</td> <td>Top left of screen</td> </tr> <tr> <td>TopCentre</td> <td>Centred at the top of the screen</td> </tr> <tr> <td>TopRight</td> <td>Top right of screen</td> </tr> </tbody> </table>	Value	Description	Centre	Centre of the screen	BottomLeft	Lower left hand corner	BottomCentre	Centred 20 pixels from bottom edge.	BottomRight	Centred 20 pixels from bottom right edge.	MidLeft	Middle left of screen	MidCentre	Same as Centre	MidRight	Middle right of screen	TopLeft	Top left of screen	TopCentre	Centred at the top of the screen	TopRight	Top right of screen
Value	Description																						
Centre	Centre of the screen																						
BottomLeft	Lower left hand corner																						
BottomCentre	Centred 20 pixels from bottom edge.																						
BottomRight	Centred 20 pixels from bottom right edge.																						
MidLeft	Middle left of screen																						
MidCentre	Same as Centre																						
MidRight	Middle right of screen																						
TopLeft	Top left of screen																						
TopCentre	Centred at the top of the screen																						
TopRight	Top right of screen																						
Merchant	Merchant number (1-8). Call DoGetMerchants to determine which merchants are available.																						
TxnRef Num	Store reference number. Not checked or validated by PIN pad																						
TxnType	'P'=Purchase, Purchase + Cashout or Cashout 'R'=Refund																						
EnableManualPan	1=EFTPOS will allow manual entry of card details at ENTER CARD prompt.																						
AllowCredit	Set to 1 to inform the Client that the POS wants to allow Credit account selection during transaction. If the POS wants to restrict the cardholder from using the credit account, set this flag to FALSE.																						
CreditOnly	Set to 1 to inform the Client that the POS wants to perform Credit account only . If set to 1, debit account selections will not be offered on the pinpad. If set to 0, both debit and credit accounts will be offered (unless AllowCredit is set to 0). Default value is 0.																						
Field48Data	Proprietary field – For use by prior arrangement of ENZ only																						

7 ENZEFT EVENTS

Events are fired to indicate completion of method calls. A method call always has a corresponding event and the event is always fired regardless of whether the method was completed successfully or not.

7.1 CHEQUEVERIFYEVENT

Fired upon completion of a **DoChequeVerify** request.

Parameter	Values	Description
ResponseCode	2 character	ResponseText code
ResponseText	20 character	Associated status text

7.2 CONTROLPANELEVENT

Fired when the PC EFTPOS Client Control Panel is closed. The control panel may be shown by calling the **DoControlPanel** method, or manually by the operator. The application may start a transaction or perform other tasks while the Control Panel is displayed, however the Control Panel windows may obscure activity in the background.

Parameter	Value	Description
ResponseCode	2 character	ResponseText code
ResponseText	20 character	Associated status text

7.3 GETLASTRECEIPEVENT

Fired upon completion of a **DoGetLastReceipt** request.

Parameter	Description
Merchant	Merchant for which receipt has been retrieved.
Stan	EFTPOS STAN associated with this receipt.
Receipt	Formatted EFTPOS receipt. Each line is terminated with a CR LF pair. No other formatting is performed. If the receipt is printed or stored for later retrieval, it must be printed or stored in its entirety to conform to EFTPOS code of practice.
TxnRef	POS supplied TxnRef originally associated with this receipt.
Success	This property must not be checked for this method.

7.4 GETLASTTRANSACTIONEVENT

Fired upon completion of a **DoGetLastTransaction** request.

Parameter	Description
LastTxnSuccess	True if retrieved transaction was originally accepted. Otherwise false.

Merchant	Merchant for which details have been retrieved.
MessageType	210: EFTPOS Financial Transaction 510: EFTPOS Settlement 810: EFTPOS Logon
Stan	EFTPOS STAN associated with this receipt.
TxnRef	POS supplied TxnRef originally associated with this receipt.

7.5 GETMERCHANTSEVENT

Fired upon completion of a **DoGetMerchants** request.

Parameter	Values	Description
ActiveMerchants	0<2048 byte	Merchants active (3 characters), merchants number (3 characters) & merchant name (13 characters)
Success	Boolean	True indicates successful completion

7.6 INFOEVENT (BSTR INFOTYPE)

Fired at various stages of a **DoTransaction** request.

Info Type	Description
1	PAN read, Pan property populated with the Pan
2	Account selected, AccountType property contains the selected account
3	Application selected, App property contains the selected application AID

7.7 JOURNALVIEWEREVENT

Fired upon completion of a **DoJournalViewer** request. In the current version, this is the same as the **ControlPanelEvent**.

7.8 INPROGRESS (BOOL INPROGRESS)

Fired upon start and completion of EFT processing. Parameter indicates busy or not busy" status. Useful for user interface to provide visual indication of EFT status.

7.9 LOGON EVENT

Fired upon completion of a **DoLogon** request.

Parameter	Values	Description
Caid		
Date		DD/MM/YYYY
PinpadVersion		
ResponseCode	2 character	Response Text code

ResponseText	20 character max	Associated status text
Stan	6 character numeric	STAN of EFTPOS Logon transaction
Time		HH:MM:SS

7.10 PRINTRECEIPEVENT

Fired when PC EFTPOS wants a receipt to be printed by the POS. The pre-formatted EFTPOS receipt is contained in the Receipt property. See the section on Receipt Printing for more information.

The **ReceiptType** parameter attached to this event indicates the type of receipt to be printed. This can be used to determine what optional information to print on the receipt. The receipt should be printed even when the POS doesn't recognise the **ReceiptType** that has been sent.

A customer receipt is a customer EFTPOS financial receipt. A merchant receipt is the merchant copy of an **ACCEPT WITH SIGNATURE** transaction and is used to record the customer signature. A logon receipt may be produced if an explicit EFTPOS logon is attempted (via **DoLogon** or control panel "Logon" button). A Settlement receipt is produced by EFTPOS settlement procedures.

Parameter	Values	Description
ReceiptType	String	"C" Customer Receipt
		"M" Merchant Receipt
		"S" Settlement Receipt
		"L" Logon Receipt
		"R" Default Receipt Type
Receipt	String	Formatted EFTPOS receipt information

7.11 QUERYCARDEVENT

Fired upon completion of a **DoQueryCard** request.

Parameter	Values	Description
ResponseCode	2 character	Response Text code
ResponseText	20 character max	Associated status text
Track2	64 character max	Card Track2 data.
Track1	128 characters max.	Card Track1 data if present



NOTE: Only cards present in the PCE_CPTFILE.DAT file will be returned in this response. This is an encrypted file maintained by ENZ. Please contact ENZ to include additional cards that are not supported.

7.12 REPRINTRECEIPEVENT

Fired upon completion of a **DoReprintReceipt** request. This event will be followed by a **PrintReceiptEvent** which should be used to do the printing.

Parameter	Description
LastTxnSuccess	True if retrieved transaction was originally accepted. Otherwise false.
Merchant	Merchant for which details have been retrieved.
MessageType	210: EFTPOS Financial Transaction 510: EFTPOS Settlement 810: EFTPOS Logon
Stan	EFTPOS STAN associated with this receipt.
TxnRef	POS supplied TxnRef originally associated with this receipt.

7.13 SETTLEMENT CUTOVER EVENT

Fired upon completion of a **DoSettlementCutover** request.

Parameter	Values	Description
ResponseCode	2 characters	ResponseText code
ResponseText	20 characters	Associated status text
Success	Boolean	True indicates successful completion of method only
LastTxnSuccess	Boolean	True indicates successful settlement completion, (EFTPOS ResponseCode "00", "90" or "97").

7.14 SETTLEMENT ENQUIRY EVENT

Fired upon completion of a **DoSettlementEnquiry** request.

Parameter	Values	Description
ResponseCode	2 characters	ResponseText code
ResponseText	20 characters	Associated status text
Success	Boolean	True indicates successful completion of method only
LastTxnSuccess	Boolean	True indicates successful settlement completion, (EFTPOS ResponseCode "00", "90" or "97").

7.15 STATUS EVENT

Fired upon completion of a **DoStatus** request.

When **StatusEvent** fires, if **Success** = TRUE

Parameter	Description
Caid	The current CAID setting for the merchant
ConfigLocked	Set to TRUE if the CATID, CAID and NII properties may not be changed.
Logged On	TRUE if Merchant is logged on. Otherwise FALSE.
Merchant	Merchant for which parameters apply.
Nii	The current NII setting for the Merchant.
Timeout	The PIN pad Transaction Time-out setting
ResponseCode	Two character EFPOS ResponseCode.
ResponseText	Text associated with ResponseCode returned.
PinpadSerialNumber	8 digit PinpadSerialNumber
PinpadVersion	PIN pad software version string
POSVendor	
POSProductId	
POSProductVersion	
Version Major	
Version Minor	
Version Revision	

7.16 TRANSACTIONEVENT

Fired upon completion of a **DoTransaction** request.

Parameter	Values	Description
AccountType	"CHQ" "SAV" "CRD"	Cheque account (Debit) Savings account (Debit) Credit account
AmtPurchase		Purchase Amount
AmtCash		Cashout Amount
AuthCode		Authorisation Code
Caid		CAID
CardName		Bank supplied name of the card used.
CardType	Refer to CardType Property for values returned.	Check the AccountType property to determine if a credit account was selected. NOTE: If the CardType value is "Debit" there is no need to check the AccountType property. However if the CardType value is not "Debit" the AccountType property must be checked.
Date		DD/MM/YYYY
LastTxnSuccess	Boolean	Indicates if the transaction was successful (EFTPOS response code "00" or "08").
Merchant	Numeric	Currently selected merchant number (1-8)

Pan		Primary Account Number. Loaded with customer card number for manually entered transactions. Only credit card details may be manually entered. Manual PAN entry, Due to Scheme restrictions first 6 and last for digits will only be populated for this property if present.
ResponseCode	2 character	Response Code
ResponseText	20 character max	Response Text
Stan	6 character numeric	STAN of EFTPOS Logon transaction, EFTPOS Systems Trace Audit Number
Time		HHMMSS time of transaction
TxnRef		POS supplied TxnRef originally associated with this receipt
Txn Type	"P" "B" "R"	P = Purchase / Cash / Purchase + Cash B = Balance R = Refund

- END -