datacap systems, inc.

100 New Britain Blvd
Chalfont, PA 18914
Tel: 215-997-8989
E-mail: datacap@dcap.com

dsiPDCX

Out of Scope (OOS)
Core Integration Specification

V01.42

Copyright and Trademark Notices

Notice

This document contains information proprietary to Datacap Systems Inc. The only acceptable use for the information contained herein is to interface third party systems exclusively to Datacap's ePay server and Tran products. Any other use is strictly prohibited.

Revision History

00.01	06 Dec 2010	• Preliminary Draft Release
00.02	09 Dec 2010	• Incorporated <TranType> and <PadType> tags for 'SecureDeviceInit' Admin transaction.
00.03	10 Dec 2010	• Incorporated advice on Check and EBT transaction capability
01.00	26 May 2011	• Formal Release • Added configuration and data flow information for VFI Vx810 support • Added new values for <PadType> and <SecureDevice>
01.12	30 Aug 2011	• Added configuration and data flow information for UIC PP795 support • Added new values for <PadType> and <SecureDevice>
01.15	20 Apr 2012	• Added configuration and data flow information for VivoPay 4500m support
01.16	24 May 2012	• Document reorganization; Core Integration Guide and XML transaction detail in separate sections by class • Added configuration and data flow information for MagTek MiniMSR and SureSwipe • Added new TranType <GetPrePaidStripe> • Added 'void CancelCommand' method
01.17	31 May 2012	• Updated TStream DTD to include MinLen, MaxLen, Preamble and PrePaidStripeTimeout elements
01.18	08 Jun 2012	• Updated TStream and RStream DTD to include GetPrePaidStripe, PrePiadTrack2 and PrePaidAcctNo elements
01.19	10 Jul 2012	• Added new TranCode <GetSignature> • Added new SecureDevices
01.21	29 Aug 2012	• Added new tags <LastFourCheck> and <CardholderID> • Added configuration and data flow information for J2 650 POS Internal MSR support
01.22	30 Sep 2012	• Added new section for Payair mobile transaction support • Added FDMS Buypass Host to NETePay Compatible Server List • Added new supplement (#10) for Check transactions
01.23	15 Oct 2012	• Added configuration and data flow information for Equinox L500 support • Added configuration and data flow information for Flrich FEC Gladius Smart AL7385 support
01.28	14 Jul 2013	• Added configuration and data flow information for IBM SurePOS 500 MSR support • Added configuration and data flow information for FEC AER Touch POS Terminal support • Added configuration and data flow information for Micros PCWS 2015 System support • Added configuration and data flow information for POSX EVO TP4 All-In-One POS support • Added new tags <SignMaximumX> and <SignMaximumY> in <Signature> response
01.28	04 Oct 2013	• Added updates for IPTranLT

01.30	22 Nov 2013	• Added GetDevicesInfo method
01.32	25 Jan 2014	• Added new SecureDevices
01.33	01 Mar 2014	• Added reference to Loyalty for new <Admin> XML for <AddAlias>, <NewAliasAcct>, <RemoveAlias>, <Transfer> • Added ONTRAN explanation for SecureDevice
01.34	13 Apr 2014	• Added new SecureDevices (HP Retail Jacket, IDTech SecuRED MSR)
01.38	24 Sep 2014	• Added new SecureDevices
01.42	13 Sep 2014	• Revised BatchSummary and BatchClose instructions

Table Of Contents

1.0 Introduction

dsiPDCX is a Windows ActiveX control that provides POS applications with the ability to process an extensive assortment of electronic payments in a multi-tiered client-server architecture without handling sensitive cardholder data (i.e. account number, expiration date, CVV/CID, PIN). Applications integrated with dsiPDCX act as a client to any of Datacap's NETePay payment servers to process payments.

dsiPDCX insulates a POS application from handling cardholder data by directly controlling the card account input device, such as an MSR, PIN pad, contactless reader, etc. Since dsiPDCX directly controls the input cardholder account information to process a payment transaction, it can facilitate an 'out of scope' position from a PCI-DSC (Payment Card Industry Data Security Council) PA-DSS (Payment Application – Data Security Standard) perspective for a POS application.

dsiPDCX provides support for a variety of cardholder account input devices (referred to as SecureDevices in this document) such as MSRs (magnetic stripe readers), PIN pads, contactless readers, etc. A complete list of the currently supported SecureDevices, including specifics on capabilities, models, drivers, etc. is available from Datacap as a separate document. Datacap is continually developing support for new SecureDevices in dsiPDCX; if you require specific device support that is not currently listed. Contact your Datacap Developer Support representative for a detailed listing of supported devices or to discuss your needs for a specific device.

dsiPDCX is designed for use exclusively with Datacap's NETePay server and IPTranLT products using Internet Protocol (IP). Datacap ePay servers are available for in-store or enterprise configurations and are designed to communicate with specific payment systems providers. Both ePay servers and IPTranLT are available with a range of communications to the payment systems processor, including Internet, Frame Relay, Wireless, and Dial. A single dsiPDCX integration effort permits a POS developer to offer a variety of payment processing access methods that best suit their client's performance and budget needs.

dsiPDCX incorporates automatic failover support for up to 10 Datacap NETePay payment servers assuring high availability. Servers can provide redundant connections utilizing different connection methods to the processing host transparently. Multiple Datacap payment servers or IPTranLT's can be simultaneously supported by dsiPDCX for different payment processing hosts; for example, one server using processor for credit cards and another server using a different processor for check processing.

dsiPDCX does not use any storage on the client machine; the Datacap NETePay server software and IPTranLt's provides consolidated transaction data storage, logging and data management functions. All Datacap NETePay server and IPTranLT products have been assessed and are compliant with PA-DSS 1.2 requirements.

The dsiPDCX control uses XML formatted requests and responses for transaction processing requests.

DSIClientX is another available version of an ActiveX control for NETePay access where the POS application manages cardholder data input directly. DSIClientX does not create an out of scope environment for POS applications.

1.1 System Configurations Supported Using dsiPDCX

The dsiPDCX ActiveX control is typically installed on a PC-POS workstation along with the POS application. The POS application issues payment processing requests in XML format to dsiPDCX. If the payment processing request requires cardholder information, such as a magnetic stripe entry, dsiPDCX will obtain the required input(s) through direct software control of the appropriate input device (SecureDevice), such as a PIN pad, MSR, etc.

The types of transactions that can be processed with any particular SecureDevice is determined by it capabilities. For example, debit transactions require a SecureDevice with PIN input capabilities. The SecureDevice identifier value should be supplied as the argument for the <SecureDevice> tag in every transaction request (Refer to XML definitions in separate dsiPDCX Transaction Definitions sections).

In the event that a SecureDevice is incapable or unable to read cardholder account information, dsiPDCX provides a mechanism for entry on the POS terminal screen as an input window controlled completely outside of the POS software. A POS developer is able to control timeouts for input at a SecureDevice for enabling manual entry by dsiPDCX.

The following lists the currently supported SecureDevice configurations and their capabilities by dsiPDCX. Datacap Systems is continually developing support for new types of SecureDevices; contact your Datacap representative for the most current listing of supported SecureDevices and their specifications.

dsiPDCX Supported Devices	SecureDeviceID	PadType	Comm Interface	Windows Driver Required	Notes
None	NONE	None	None	No	
MSR or PIN Pad on Tran Device	ONTRAN	None	None	No	To use ai input device connected to a serial port on a Tran device (specified in PSCS parameter file)
Datacap PDC with optional VFI1000se PIN pad	PDC	VFI1000 or None	RS-232	No	
Datacap PDC with optional VFI1000se PIN pad	PDC2	VFI1000 or None	RS-232	No	Blowfish Encryption between PDC and dsiPDCX
Datacap PDC with optional VFI1000se PIN pad	PDC	VFI1000 or None	USB - Virtual Com Port	Yes	
Datacap PDC with optional VFI1000se PIN pad	PDC2	VFI1000 or None	USB - Virtual Com Port	Yes	Blowfish Encryption between PDC and dsiPDCX
IDTech MSR	IDTMSRHID	None	USB - HID	No	ValueMag or SecureMag without Encryption Enabled
IDTech SecureMag MSR	IDTSECUREMAGHID	None	USB - HID	No	SecureMag with Datacap MSR Encryption Key and Encryption Enabled
MagTek MINI MSR	MTMINIMSRHID	None	USB - HID	No	
MagTek SureSwipe MSR	MTSURESWIPEHID	None	USB - HID	No	
MagTek IPAD PIN pad	MTIPADHID	IPAD100	USB - HID	No	For Mercury E2E only - with Mercury MSR Encryption Key
MagTek MINI MSR	MTMINIMSR	None	RS-232	No	
MagTek MINI MSR	MTMINIMSRVCOM	None	USB - Virtual Com Port	Yes	Windows Driver supported on Windows XP Only
MagTek SureSwipe MSR	MTSURESWIPEVCOM	None	USB - Virtual Com Port	Yes	Windows Driver supported

					on Windows XP Only
UIC 795 PIN pad	UIC795	UIC795	RS-232	No	
UIC 795 PIN pad	UIC795	UIC795	USB - Virtual Com Port	Yes	
Verifone Vx810 PIN pad running XPI	VX810XPI	VX810	RS-232	No	
Verifone Vx810 PIN pad running XPI	VX810XPI	VX810	USB - Virtual Com Port	Yes	
ViVoPay 4500m contactless with MSR	VIVO4500M	None	RS-232	No	Contactless capable plus swipe
ViVoPay 4500m contactless with MSR	VIVO4500M	None	USB - Virtual Com Port	Yes	Contactless capable plus swipe
J2 650 POS Internal MSR	J2650MSRVCOM	None	Internal	Yes	Requires driver from J2 Retail Systems that can change the MSR interface from keyboard wedge to serial COM port. The MSR must be in OPOS mode.
Equinox Payments L5300 PIN pad	EQUINOXL5300	L5300	RS-232	No	Contactless capable plus swipe
Firich FEC Gladius Smart AL7385 Touch POS Terminal	FECMSRHID	None	USB - HID	No	An FEC USB HID reader is required along with a driver from FEC.
IBM SurePOS 500 MSR	IBMSUREPOSMSR	None	RS-232 (19200 baud)	No	
FEC AER Touch POS Terminal	FECAERMSRHID	None	USB - HID	No	
Micros PCWS 2015 System	MICROSPCWSMSR	None	Internal Micros PCWS	Requires Micros PCWS device driver	MSR configured for Special Mode
POSX EVO TP4 All-In-One POS	POSXMSRHID	None	USB - HID	No	
AnyShop POS MSR	ANYSHOPPOSMSRHID	None	USB - HID	No	
Elo ETT10A1 Tablet MSR	ELOTABLETMSR	None	RS-232 or USB/VCOM	Yes if USB/ VCOM	
Verifone MX915 PinPad	VERIFONEMX915	MX915	RS-232	No	Contactless capable plus swipe
PioneerPOS StealthTouch MSR	PIONEERSTM5MSRHID	None	USB - HID	No	
HP Retail Jacket for HP ElitePad	HPRETAILJACKET	None	USB - HID	No	
IDTECH SecuRED Encrypted MSR	IDTSECUREDMSRHID	None	USB - HID	No	Use Datacap encryption key - part number IDT-KEYINJ-063-A.
UIC 795SE PIN pad	UIC795SE		RS-232 or USB	No	
UIC 795 SE PIN pad with TSYS VOLTAGE	UIC795SE_TSYS_VOLT		RS-232 or USB	No	
Verifone Vx805 PIN pad running XPI	VX805XPI	VX805	RS-232 or USB	No	Swipe only
Verifone Vx805 PIN pad running XPI	VX805XPI_MERCURY_E 2E	VX805	RS-232 or USB	No	Mercury E2E Encryption - contactless capable plus swipe
Verifone Vx805 PIN pad running XPI	VX805XPI_CTLS	VX805	RS-232 or USB	No	Contactless capable plus swipe
ID Innovations Classic MSR	IDINNOVATIONS_CLASS IC_MSRHID	None	USB - HID	No	
ID Innovations Value MSR	IDINNOVATIONS_VALUE _MSRHID	None	USB - HID	No	
Verifone MX850 PinPad	VERIFONEMX850	VX850	RS-232	No	Contactless capable plus swipe
Ingenico iSC250 PinPad	INGENICOISC250	ISC250	RS-232	No	
HP RP77800	HP_RP77800	None	USB - HID	No	
None	NONE	None	None	No	

1.2 Integration Strategy – Payment Type and Market Selection

dsiPDCX provides a developer an extensible software interface for many types of payments. While the flexibility and transaction types included in dsiPDCX will protect an integrator's development investment in electronic payments, very few need all the capabilities in their market specific applications.

Before integrating dsiPDCX, a software developer should first assess the payment requirements for the markets they serve. dsiPDCX is designed for applications in three major payment type categories as follows:

Category	*Types of Retail Businesses*
Retail	Non Food Retail (specialty, discount, department) Grocery Stores Convenience Stores (without gas pump interface) Quick Service Restaurants (no tips) Unattended paypoints (parking lots, kiosks vending)
Restaurant	Table Service Restaurants with Tips Personal Service with Tips (Barber, Beauty, etc)
MOTO	Mail Order Telephone Order

Payment processing providers will put a business into one of these processing categories when establishing their merchant account. The categories imply the type of processing expected and determines the rates and discount structure for the payment processing service.

In addition to the payment categories, there are six classes of payment transactions which should be evaluated for the merchant target market. dsiPDCX can support the following transaction categories:

Category
Credit Cards
Debit (ATM) Cards (US)
EBT (Electronic Benefits Transfer)
PrePaid (Gift Cards, Stored Value Cards)
Loyalty
CardLookup
Mobile Payments

Almost all merchants want to accept credit cards; however many restaurants do not accept ATM debit cards because they require the customer to input a PIN (Personal Identification Number) which is difficult in a table service environment. Food retailers of all type have an interest in accepting government sponsored benefits payments (called EBT) for foodstamp and cash purchases. Many retailers want to use Gift Cards (electronic gift certificates) as a program to foster loyalty and many businesses have a high percentage of check payments.

1.2.1 Transaction Codes Supported in dsiPDCX

TranType	TranCode	Stripe	Manual	AVS	Partial
Credit	Sale	X	X	X	X
	Return	X	X		
	VoidSale	X	X		
	VoidSaleByRecordNo				
	VoidReturn	X	X		
	VoidReturnByRecordNo				
	AuthOnly	X	X	X	
	PreAuth	X	X		X
	VoiceAuth	X	X		
	PreAuthCapture	X	X		
	Adjust	X	X		
	AdjustByRecordNo				
	Balance	X	X		
	ZeroAuth	X	X	X	
Debit (US)	Sale	X			
	Return	X			
	PreAuth	X			
	PreAuthCapture	X			
EBT	Sale (Foodstamp)	X	X		
	Return (Foodstamp)	X	X		
	Balance (Foodstamp)	X	X		
	Sale (Cash)	X	X		
	Return (Cash)				
	Balance (Cash)	X	X		
	Voucher (Foodstamp)		X		
	Voucher Return – (Foodstamp)		X		
FSA	Credit Sale	X	X		
	Credit Reverse	X	X		
	Debit Sale	X			
	Debit Reverse	X			
PrePaid	Issue	X	X		
	Sale	X	X		
	Return	X	X		
	VoidSale	X	X		
	VoidReturn	X	X		
	Balance	X	X		
	NoNSFSale	X	X		
	CashOut	X	X		
	GetPrePaidStripe	X	X		

Loyalty	Issue	X	X		
	Add	X	X		
	Subtract	X	X		
	VoidIssue	X	X		
	VoidAdd	X	X		
	VoidSubtract	X	X		
	Balance	X	X		

Admin	BatchSummary				
	ItemDetail				
	BatchClear				
	BatchClose				

CardLookup	BIN Lookup – Card Capabilities	X	X		

GetSignature	Get digitized signature				

Important:

Support for any particular transaction code depends on the server version with which dsiPDCX is communicating. Not all servers support all transaction codes or the variants; refer to Datacap's dsiPDCX Compatible Server List at the end of this document for the transaction codes supported.

1.3 Integrating dsiPDCX

dsiPDCX is a Windows ActiveX control which can be used with most Windows application development environments and languages.

1.3.1 Requirements

1. dsiPDCX is compatible with Windows 98 SE, ME, 2000, NT (SP6), XP, Vista and Windows7.
2. dsiPDCX control requires that TCP/IP networking is installed and operating correctly. Since the client-server architecture allows LAN and/or WAN system configurations, the appropriate network adapter is determined by individual requirements.
3. dsiPDCX requires an available Windows COM port for attaching a PDC or PIN pad device. RS-232 devices require a Windows COM port; USB devices require a USB port and the appropriate software driver.
4. dsiPDCX utilizes Windows cryptographic services – the 128bit cipher (high encryption) must be installed on the system running dsiPDCX. Encryption support updates may be downloaded from Microsoft's website as required.

1.3.2 Installation

The latest version of dsiPDCX ActiveX control is available for download from Datacap Systems' Software Download website at www.datacapepay.com as a Windows self-extracting installer.

dsiPDCX is also included in the DSIClient ZIP archive install for integrators who may want to include Datacap's controls within their install package. In addition to dsiPDCX, this archive contains:

.

DSIClient	2.50
DSIClientX	3.86
dsiPDCX	1.34
Ver1000X	1.10
Ver2000X	1.40
EverestMultiPayX	1.03
PDCX	1.10
PDC Config	1.05
IngenicoRBAX	1.20

Integrators who intend to rely exclusively on dsiPDCX to manage any attached payment peripherals need only install dsiPDCX.

1.3.3 dsiPDCX Initialization

Before using dsiPDCX to process transactions, an application must establish a Server IP Addressing strategy to be used to process transactions as outlined below. In addition, SecureDeviceInit method must be performed to identify and initialize the attached PDC or SecureDevice.

1.3.3.1 Automatic Server IP Addressing

dsiPDCX uses hard IP addresses to reach a server. The IP address may be local or a legitimate external address. dsiPDCX can maintain an internal list of up to 10 IP addresses to support automatic failover in the event that a *ProcessTransaction* call is unable to get a response from a server. These addresses must be resolved by calling the *ServerIPConfig* method before processing any transaction with the *ProcessTransaction* method.

The *ServerIPConfig* method uses DNS to resolve host names – or it can be supplied with hard IP addresses, which it will use directly. The IP addresses resolved by *ServerIPConfig* are stored internally to the control and used for subsequent calls to *ProcessTransaction*. The list is ordered in the same sequence as the arguments are passed. This address list is maintained by the control until it is released/destructed – *it must be performed each time the control is loaded*. See section 2.1 for details on the *ServerIPConfig* method.

The *ServerIPConfig* method resolves and creates the internal host address list but it does not verify the operational status of the servers at those addresses. To verify which addresses have operational servers, the *PingStoredServerList* method should be called next. This method does not login or process any transactions but returns the number of addresses, which responded to an IP ping request. At least one active server is required to process transactions.

1.3.3.2 Manual Server IP Addressing

As an alternative to using the dsiPDCX automatic rollover management of IP addresses, an application can optionally include a single hard IP address (not a host name) in the *ProcessTransaction* call. When an address is included in the call, dsiPDCX will use that address and ignore its internal host list and will not failover to any of the addresses in the list. In this case, it is the responsibility of the application to retry the transaction with a different server address or let the control use its internal list on a subsequent call.

By including a hard IP address in the *ProcessTransaction* call, you can direct a processing request to a particular server. This is a method for directing certain types of transactions among multiple servers. To support this technique of managing server addresses, dsiPDCX provides a method, *GetIpAddressFromHostName*, which allows an application to do DNS to get a hard server IP address, if necessary.

Generally, DNS services should be avoided on a transaction by transaction basis since the response time is highly variable and it can significantly lengthen the time to process a transaction overall. It's more efficient for an application to resolve all its required host names at startup and remember them for subsequent calls to *ProcessTransaction*.

1.3.3.3 SecureDevice Initialization

dsiPDCX requires an Admin command to initialize the attached PDC or attached SecureDevice. This command should be performed during startup of the POS system with the optional PIN pad attached. It should not be performed prior to every transaction as it takes several seconds to complete with an attached PIN pad. The command named 'SecureDeviceInit' is as follows:

XML Template: **SecureDeviceInit Request**

```
<?xml version="1.0"?>
<TStream>
  <Admin>
    <IpAddress>999.999.999.999</IpAddress>
    <IpPort>99999</IpPort>
    <TranDeviceID>TranDeviceID</TranDeviceID>
    <MerchantID>MerchantID</MerchantID>
    <TerminalID>TerminalID</TerminalID>
    <OperatorID>OperatorID</OperatorID>
    <TranCode>SecureDeviceInit</TranCode>
    <TranType>TranType</TranType>
    <PadType>PadType</PadType>
    <SecureDevice>ValidSecureDeviceID</SecureDevice>
    <ComPort>ComPort</ComPort>
    <SequenceNo>SequenceNo</SequenceNo>
      <TerminalName>TerminalName</TerminalName>
    <ShiftID>ShiftID</ShiftID>
    <Signature>Signature</Signature>
  </Admin>
</TStream>
```

Element	Req	Min	Max	Type	Description
IpAddress	O	7	15	AN	IP address of server to use for this transaction. This address will override the addresses obtained via ServerIPConfig. If an IPTranLT is the device to receive the command, this tag should be omitted.
IpPort	O	1	5	N	IP port number on server to use for this transaction. If omitted, default port is 9000. If an IPTranLT is the device to receive the command, this tag should be omitted.
TranDeviceID	O	1	24	AN	If an IPTranLT is the device to receive the command, this tag must be included. The TranDeviceID (DID) value is printed on a sticker on the bottom of every IPTranLT.
MerchantID	Y	1	24	AN	Merchant identification assigned by processor.
TerminalID	O	1	24	AN	***For general use*** Terminal ID data must be supplied in this tag only if provided by the processor or merchant service provider; otherwise this tag should not be included. ***For reversal support with selected processors*** Use a unique POS workstation number (exactly 3 digits 000-999) for multi-workstation locations. See 1.3.7 in the DSIClientX documentation.
OperatorID	O	1	10	N	Operator (clerk, server, etc.) associated with the inquiry.
TranCode	Y	1	40	A	Use value “SecureDeviceInit”
TranType	Y	1	40	A	Use value “Setup”
PadType	Y	1	24	A	PIN Pad Type: “VFI1000” - if VeriFone 1000SE PIN pad attached to PDC. ‘None’ - if no PIN pad is attached to the PDC (credit only).

					"VX810" - when using a VeriFone Vx810 PIN pad attached to a PC; PDC is not used. "UIC795" - when using a UIC PP795 PIN pad attached to a PC; PDC is not used. "IPAD100" - when using a MagTek IPAD PIN pad attached to a PC; PDC is not used. "L5300" - when using an Equinox Payments L5300 PIN pad attached to a PC; PDC is not used.
SecureDevice	Y	1	40	AN	ValidSecureDeviceID values by device type: "PDC" - when using a legacy PDC with or without an attached VFI1000SE PIN pad. Blowfish encryption support requires latest PDC firmware update. "PDC2" - when using a PDC with or without an attached VFI1000SE PIN pad. Blowfish encryption support is standard. "VX810XPI" - when using a VeriFone Vx810 PIN pad attached to a PC; PDC is not used. "UIC795" - when using a UIC PP795 PIN pad attached to a PC; PDC is not used. "UIC795SE" - when using a UIC PP795SE PIN pad attached to a PC; PDC is not used. "UIC795SE_TSYS_VOLT" - when using a UIC PP795SE PIN pad for TSYS VOLT attached to a PC; PDC is not used. "VIVO4500M" -when using a VivoPay 4500m contactless reader / MSR attached to a PC; PDC is not used. "MTSURESWIPEVCOM" - when using a MagTek SureSwipe MSR attached to a PC; PDC is not used. "MTMINIMSRVCOM" - when using a MagTek MiniMSR MSR attached to a PC via USB; PDC is not used. "MTMINIMSR" - when using a MagTek MiniMSR MSR attached to a PC via RS-232; PDC is not used. "IDTMSRHID" - when using an IDTech MSR attached to a PC via USB and using Windows native HID driver; PDC is not used. "IDTSECUREMAGHID" - when using an IDTech SecureMag MSR attached to a PC via USB and using Windows native HID driver; PDC is not used. "MTMINIMSRHID" - when using a MagTek MINI MSR attached to a PC via USB and using Windows native HID driver; PDC is not used. "MTSURESWIPEHID" - when using a MagTek SureSwipe MSR attached to a PC via USB and using Windows native HID driver; PDC is not used. "MTIPADHID" - when using a MagTek IPAD PIN Pad attached to a PC via USB and using Windows native HID driver; PDC is not used.

					"J2650MSRVCOM" – when using internal MSR on J2 650 POS. Requires driver from J2 Retail Systems that can change the MSR interface from keyboard wedge to serial COM port. The MSR must be in OPOS mode. PDC is not used. "EQUINOXL5300" - when using an Equinox Payments L5300 PIN Pad attached to a PC via RS-232; PDC is not used. "FECMSRHID" - when using a Firich FEC Gladius Smart AL7385 Touch POS Terminal. An FEC USB HID reader is required along with a driver from FEC; PDC is not used. "IBMSUREPOSMSR" - when using a IBM SurePOS 500 MSR attached to a PC via RS-232; PDC is not used. "FECAERMSRHID" - when using a FEC AER Touch POS Terminal. PDC is not used. "MICROSPCWSMSR" - when using a Micros PCWS 2015 System. Requires Micros PCWS device driver; MSR configured for Special Mode; PDC is not used. "POSXMSRHID" - when using a POSX EVO TP4 All-In-One POS system attached to a PC via USB and using Windows native HID driver; PDC is not used. "ANYSHOPPOSMSRHID" - when using an AnyShop POS MSR attached to a PC via USB and using Windows native HID driver; PDC is not used. "ELOTABLETMSR" - when using an Elo ETT10A1 Tablet MSR attached to a PC via RS-232 or USB and using Windows native HID or Elo VCOM driver; PDC is not used. "ONTRAN" - when using a MSR or PIN Pad on Tran Device attached to a PC; PDC is not used. "VERIFONEMX915" - when using a Verifone MX915 PinPad attached to a PC via RS-232; PDC is not used. "PIONEERSTM5MSRHID" - when using a PioneerPOS StealthTouch MSR attached to a PC via USB and using Windows native HID driver; PDC is not used. "HPRETAILJACKET" - when using an HP Retail Jacket for HP ElitePad MSR attached to a PC via USB and using Windows native HID driver; PDC is not used. "IDTSECUREDMSRHID" - when using an IDTech SecuRED Encrypted MSR attached to a PC via USB and using Windows native HID driver; PDC is not used. "VX805XPI" - when using a Verifone MX805 PinPad attached to a PC via RS-232; PDC is not used. "IDINNOVATIONS_CLASSIC_MSRHID" - when using an ID Innovations Classic MSR attached to a PC via USB and using Windows native HID driver; PDC is not used.

					"IDINNOVATIONS_VALUE_MSRHID" - when using an ID Innovations Value MSR attached to a PC via USB and using Windows native HID driver; PDC is not used. "VERIFONEMX850" - when using a Verifone MX850 PinPad attached to a PC via RS-232; PDC is not used. "INGENICOISC250" - when using an Ingenico iSC250 PIN pad attached to a PC via USB and using Windows native HID driver; PDC is not used.
ComPort	Y	1	3	N	The true RS-232 serial port or virtual serial port for USB devices that use a virtual serial port driver. Port number to which SecureDevice is attached (0-255). For USB-HID devices, a ComPort value of '0' (zero) should be used.
SequenceNo	O	1	20	AN	***For reversal support with selected processors*** Use value '00000000'
TerminalName	O	1	20		Terminal name
ShiftID	O	1	20		Shift identification
Signature	O	1	2048		Digitized signature

1.3.4 Server Timeout Control

The time for dsiPDCX to fail to connect to a particular server address can be controlled with the initialization method *SetConnectTimeout*. When a timeout value is set with this method, calls to *ProcessTransaction* will only wait that long to connect before trying another server address in the internal list. If an IP address was supplied in the *ProcessTransaction* call, dsiPDCX will return after the timeout expires without attempting to use any internally stored server addresses. The default value for the server connect timeout is 10 seconds if *SetConnectTimeout* is never called. See the *SetConnectTimeout* method section for details.

1.4 Processing Transaction Requests

The *ProcessTransaction* method provides processing for all transaction types and codes. dsiPDCX uses this single method with an XML format string argument to define what type of transaction processing will take place.

The XML transaction requests are ASCII text strings with tag pairs (similar to HTML) to identify the transaction code as well as the data elements, such as account numbers, charge amount etc. to be processed.

An understanding of XML is not required to use dsiPDCX – however XML services available from Windows can make the creation, exchange and validation of transaction requests seamless with many applications. Sections 4 through 9 detail the XML transactions and provide templates which you can cut and paste into your application - substitute required data in the appropriate tags and you're ready to process transactions.

1.4.1 Information Required for All Transactions Codes

The main integration task for processing transaction requests is collecting the specific data required by the type of payment desired. All transaction codes have a common set of data required for processing, which an application should always have available. The common data required for every transaction code consists of:

Data	*Description*
Merchant Identification No	Merchant identification assigned by processor.
Operator Identification No	Operator (clerk, server, etc.) associated with the transaction.
Transaction Type	Credit, Debit, EBT, PrePaid, Check
Transaction Code	The type of transaction (e.g. sale, return ,void, etc)
Invoice Number	Invoice number – the sequential receipt number or check number of the transaction.
Reference Number	Usually the same data as the Invoice number. In some cases, the reference number returned by a previous transaction is required.
SecureDevice	A value that identifies which Secure Device configuration is to be controlled by dsiPDCX.
ComPort	A value that identifies which Windows COM port will be used to control the connected SecureDevice.

In addition to these fields, each transaction code will require additional transaction specific information.

1.4.2 Example XML for Credit Card Sale Using dsiPDCX

The following is an example of the XML to process a credit card with an attached PDC with MSR. The transaction will use a server IP address from the dsiPDCX internal table defined by the ServerIPConfig method.

```
<?xml version="1.0"?>
<TStream>
  <Transaction>
    <MerchantID>123456789</MerchantID>
    <OperatorID>4444</OperatorID>
    <TranType>Credit</TranType>
    <TranCode>Sale</TranCode>
    <SecureDevice>PDC2</SecureDevice>
    <ComPort>4</ComPort>
    <InvoiceNo>456</InvoiceNo>
    <RefNo>456</RefNo>
    <Account>
      <AcctNo>SecureDevice</AcctNo>
    </Account>
    <Amount>
      <Purchase>24.95</Purchase>
    </Amount>
  </Transaction>
</TStream>
```

Separate sections detail the complete range of XML transactions supported by dsiPDCX and provide templates that you can cut and paste into your application with the appropriate data substituted in the tags.

1.4.4 Restaurant Category Considerations

Restaurant category merchants (i.e. merchants that needs to handle tips) require a two-step transaction process to complete payments with tips. When a check is first presented to a patron (cardholder) in a restaurant, it includes the cost of the services rendered without a tip (gratuity). The cardholder will give their card to a server who then will authorize the card using a *PreAuth* transaction. The *PreAuth* transaction verifies that the card is valid and has enough available credit for the check amount but it does not post the transaction to the processing system for payment.

At this point, a credit draft should be printed by the application with the authorization code (AuthCode) and reference number (RefNo) information returned in the successful *PreAuth* response. The draft should have an open tip line for the cardholder to enter the tip amount they want to include on the charge. Once the tip amount is entered, the transaction can be finalized with the exact total for the payment including the tip using the *PreAuthCapture* transaction.

The *PreAuthCapture* transaction will use the authorization code (AuthCode) and reference number (RefNo) information returned in the successful *PreAuth* response, as well as the tip (Gratuity) amount. The *PreAuthCapture* will finalize the payment for settlement with the tip amount specified by the cardholder.

Important:

1. ***PreAuth*** *transactions which are not closed with a corresponding* ***PreAuthCapture*** *transaction will not be settled (i.e. paid to the merchant account). It is important that a system enforces performing a* ***PreAuthCapture*** *transaction for each* ***PreAuth*** *within a reasonable period of time (usually 48 hours).*

2. *Only one* ***PreAuthCapture*** *should be performed per* ***PreAuth*** *since each additional* ***PreAuthCapture*** *performed results in another settlement record. This will result in multiple charges against the cardholder account processed by the original* ***PreAuth*** *transaction.*

3. *If a gratuity amount must be changed, use an* ***Adjust*** *transaction with a Gratuity tag* ***after*** *the* ***PreAuthCapture*** *transaction has been performed.*

4. *All* ***PreAuth*** *transactions should be captured using corresponding* ***PreAuthCapture*** *transactions. Transactions that are not to be settled should be selected by performing a* ***Void*** *on the associated* ***PreAuthCapture*** *transaction.*

1.4.5 Partial Authorization Considerations

Compliance requirements issued by the card associations in 2009 indicate that significant costs will eventually be assessed on merchants whose POS/ECR systems are unable to process authorization requests that are approved for an amount different (lower) than requested. In anticipation of the requirement to accept partial authorizations, integrators should implement support for the <PartialAuth> tag in the Credit-Sale and Credit-PreAuth transactions.

When implementing support for the <PartialAuth> tag, the logic should inspect the amount returned in the <Authorize> tag of the response to determine how much was actually authorized. The ECR/POS tendering logic should be prepared to ask for additional tenders in the event that an authorization is approved for a lower amount than requested.

1.4.6 Automatic Reversal Considerations

Selected processors offer an automatic reversal of transactions which fail due to an error condition. This capability reduces the possible occurrences of duplicate transactions when an operator attempts a transaction again after some failure to get a response. Not all processors currently support this capability but your integration should be implemented to take advantage of this capability as additional processors add this capability.

Automatic reversal operation depends on sequence numbering in transaction requests that is coordinated with the processing host. When a transaction is submitted to the processor, the included sequence number is compared to the last sequence number the host supplied in the previous response. If the sequence number matches the last supplied number from the host, the transaction processes normally and the host replies with a new sequence number to be used in the next request.

If the sequence number in a request does not match the last sequence number supplied in that last response from the processor, then the processor assumes that the sequence number in the request refers to a previous transaction and that transaction is replaced with the current request. If the transaction data also matches in the account number and amount, the processor assumes that it is a duplicate request and returns an approval without adding the request to the batch, thus avoiding a duplicate.

The integration rules for proper operation of automatic reversals are summarized as follows:

1. The last <SequenceNo> received in a processor response for *any* transaction must be supplied in the *next* transaction request of any type. Regardless of the last response type or intervening failed attempts, that *last received* <SequenceNo> must be used on the next request of any type. Note that there may be some responses that don't include a <SequenceNo> value but the *last* <SequenceNo> received in any response should be persisted for the next transaction request.

   The exception to the above rule is for new installations where no transactions have yet been processed and therefore no <SequenceNo> value sent by the processor. In this situation use '00000000' as the <SequenceNo> value for the first transaction request; after that use the last response value for <SequenceNo>.

2. The <TerminalID> value included in each transaction request must be unique to each POS workstation in a multiterminal location. Permitted values for <TerminalID> are 3 digits from 000 to 999 and must be unique among POS workstations that are using the same NETePay server. The <TerminalID> value should be assigned once and should remain static.

1.5 Handling Transaction Responses

The *ProcessTransaction* method returns an XML format string defining the outcome of the processing request. The format of the XML response depends on the transaction type and code requested.

Every XML response includes a *CmdResponse* element which includes the following tags:

Tag	Type	Description
ResponseOrigin	Y	Indicates the source of the response: "Client" = generated by dsiPDCX control "Server" = generated by Datacap ePay server "Bridge" = generated by Datacap Bridge server "Processor" = generated by payment processor
DSIXReturnCode	Y	Six digit return code which identifies the error type. See Section 3 on error codes for possible values
CmdStatus	Y	Indicates the outcome of the command: "Approved" = transaction approved by payment processor "Declined" = transaction declined by payment processor "Success" = command completed successfully "Error" = error processing command. Check DSIXReturnCode and TextResponse for additional info on error
TextResponse	Y	For Successful responses, this field can contain an optional message from the server or processor. For other response codes, see Section 3 on Error Codes for values.
UserTraceData	O	Echo of data supplied by the user system in the request; for use by the user system for internal tracking. May be null if no user system data was supplied in the request

If *CmdStatus* is "**Approved**" and the *TranType* submitted in the request was "Credit", "Debit", "EBT", "PrePaid", or "Check", that means the transaction was approved by the payment processor.

If *CmdStatus* is "**Declined**" and the *TranType* submitted in the request was "Credit", "Debit", "EBT", "PrePaid", or "Check", that means the transaction was *not* approved by the payment processor. *TextResponse* should be displayed.

If *CmdStatus* is "**Success**" and the *TranType* submitted in the request was "Admin", that means the request was successfully processed. Refer to Section 4 for a description of the Admin response formats.

If *CmdStatus* is "**Error**" for any *TranType* submitted in the request, that means that *ProcessTransaction* encountered an error and that the remainder of the response may be incomplete or corrupt. *ResponseOrigin*, *DSIXReturnCode* and *TextResponse* should be examined and displayed to determine the nature of the error.

The interpretation of CmdStatus, TranType and actions can be summarized in the following chart:

CmdStatus	*TranType*	*Action*
“Approved”	“Credit” “Debit” “EBT” “PrePaid”	Transaction was approved by payment processor.
“Declined”	“Credit “Debit” “EBT” “PrePaid”	Transaction was declined by payment processor. An application should display the *TextResponse* (which may contain reason/action from the payment processor) for the operator and allow another payment.
“Success”	“Admin”	An Admin type transaction completed successfully. Refer to Section 10 for a description of the Admin response formats.
“Error”	“Credit “Debit” “EBT” “PrePaid” “Admin”	The transaction encountered an error and did not get valid response data from the payment processor. An application should examine and display *ResponseOrigin*, *DSIXReturnCode* and *TextResponse* to determine the nature of the error.

Every XML response includes a *TranResponse* element which includes the transaction specific response elements. *See Section 4 for complete detail on TranResponse elements, use and meaning.*

Once an application has determined that the payment was approved, it should always save the following data about the transaction from the *TranResponse* element: *RefNo*, *AuthCode*. Optional data should also be saved, if present: *AVSResult*, *CVVResult*, *Memo*, and *AcqRefData*. These fields may be required to process subsequent transactions, such as VoidSale, VoidReturn and PreAuthCapture.

An application should also print a draft for signature by the cardholder (for card present transactions). A draft is a separate document in addition to a normal POS/ERC receipt. The cardholder does not need to have a copy of a draft if their receipt includes identification of the payment, date, approval code, reference number and account number. An application should make a provision to reprint a copy of the draft for the customer who demands it or in the event that the original draft is torn or does not print properly.

A sample credit draft for a 24-column format (wider formats may be used) would look as follows:

```
Merchant name ..................... **** MERCHANT  NAME ****
Merchant address - line 1 ......... ** MERCHANT ADDRESS 1 **
Merchant address - line 2 ......... ** MERCHANT ADDRESS 2 **
Merchant phone number ............. ** MERCHANT PHONE NO. **
Transaction date, time..……..... mm/dd/yy           hh:mm
Sale type and amount .............. CRED SALE       $9999.99
Account number and exp. date ..... XXXXXXXXXXX99999   XX/XX
Approval code (if available) ..... APP: CCCCCCCCCCCC
Reference number (if available) ... REF: RRRRRRRRRRRRRRRRRR
Cardholder acknowledgment ..... I AGREE TO PAY ABOVE
                                 TOTAL AMOUNT ACCORDING
                                TO CARD ISSUER AGREEMENT


Line for cardholder signature ..... X_____________________
```

2.0 dsiPDCX Methods and Events

2.1 Method: ServerIpConfig

Use: To create a list of server IP addresses from HostNames to be used automatically by the control for subsequent transaction requests.

Syntax: **`BSTR ServerIpConfig(BSTR HostList, SHORT ProcessControl)`**

Notes: This method should be the first one called after the dsiPDCX control is loaded. The list of IP addresses generated by this method is retained internally by the control as long as it is in memory. Until this method is called, any other methods which require a server IP address will fail. The semicolon-separated list of host names (HostList) can contain both host names and IP addresses. Supplied host names are looked up via DNS. The host list should be arranged with the desired primary server listed first, secondary sever next, etc. The control will attempt to use the server addresses in that order. It is recommended that at least 2 server addresses be successfully configured in order to enable fail over operation.

Arguments: HostList – up to 10 host names or IP addresses separated by semicolon (;). The host names entries in the list will be looked up via DNS.

ProcessControl – This determines how the control will process the request. Allowed values are:

0 Process using visible dialog boxes
1 Process without using visible dialog boxes

Returns: XML formatted string response of type RStream as follows:

```
<?xml version="1.0"?>
<RStream>
  <CmdResponse>
    <ResponseOrigin>Client</ResponseOrigin>
    <DSIXReturnCode>DSIXReturnCode</DSIXReturnCode>
    <CmdStatus>CmdStatus</CmdStatus>
    <TextResponse>9</TextResponse>
  </CmdResponse>
</RStream>
```

Element	Min	Max	Type	Description
ResponseOrigin	1	10	A	Indicates the source of the response: "Client" = generated by dsiPDCX control
DSIXReturnCode	6	6	N	Six digit return code which identifies the error type. See Section 3 on error codes for possible values
CmdStatus	1	1	A	Indicates the outcome of the command: "Success" = command completed successfully "Error" = error processing command. Check DSIXReturnCode and TextResponse for additional info on error
TextResponse	1	40	AN	For Successful responses, this field contains the number of IP addresses that were stored. For other response codes, see Section 3 on Error Codes for values

2.2 Method: PingStoredServerList

Use: To determine the state of the server(s) defined by the ServerIPConfig method. This method does not login or process any transactions. This method should be called after the ServerIPConfig method to determine if there are an acceptable number of available servers.

Syntax: **BSTR PingStoredServerList (BSTR IpPort, SHORT ProcessControl)**

Arguments: IpPort – This specifies the IP port number that will be pinged on the server. The default server port is 9000 but may be changed by the server operator. The allowed values for IpPort are:

""	Null - Use default port value (9000)
"5000"-"32000"	Use the specified port value

ProcessControl – This determines how the control will process the request. Allowed values are:

0 Process using visible dialog boxes
1 Process without using visible dialog boxes

Returns: XML formatted string response of type RStream as follows:

```
<?xml version="1.0"?>
<RStream>
  <CmdResponse>
    <ResponseOrigin>Client</ResponseOrigin>
    <DSIXReturnCode>DSIXReturnCode</DSIXReturnCode>
    <CmdStatus>CmdStatus</CmdStatus>
    <TextResponse>n Server(s) Active</TextResponse>
  </CmdResponse>
</RStream>
```

Element	Min	Max	Type	Description
ResponseOrigin	1	10	A	Indicates the source of the response: "Client" = generated by dsiPDCX control
DSIXReturnCode	6	6	N	Six digit return code which identifies the error type. See Section 3 on error codes for possible values
CmdStatus	1	1	A	Indicates the outcome of the command: "Success" = command completed successfully "Error" = error processing command. Check DSIXReturnCode and TextResponse for additional info on error
TextResponse	1	40	AN	For Successful responses, this field contains string "n Server(s) Active" where n is a number from 0-10. For other response codes, see Section 3 on error codes for values

2.3 Method: PingServer

Use: To determine if a single server exists at a specified IP address. This method does not login or process any transactions. This method can be called to determine if there is a server active at that address.

Syntax: `BSTR PingServer(BSTR IpAddress, BSTR IpPort, SHORT ProcessControl)`

Arguments: IpAddress – This specifies the IP address to be pinged. It is in 999.999.999.999 format.

IpPort – This specifies the IP port number that will be pinged on the server. The default server port is 9000 but may be changed by the server operator. The allowed values for IpPort are:

"0"	Use default port value (9000)
"5000"-"32000"	Use the specified port value

ProcessControl – This determines how the control will process the request. Allowed values are:

0 Process using visible dialog boxes
1 Process without using visible dialog boxes

Returns: XML formatted string response of type RStream as follows:

```
<?xml version="1.0"?>
<RStream>
  <CmdResponse>
    <ResponseOrigin>Client</ResponseOrigin>
    <DSIXReturnCode>DSIXReturnCode</DSIXReturnCode>
    <CmdStatus>CmdStatus</CmdStatus>
    <TextResponse>n Server(s) Active</TextResponse>
  </CmdResponse>
</RStream>
```

Element	Min	Max	Type	Description
ResponseOrigin	1	10	A	Indicates the source of the response: "Client" = generated by dsiPDCX control
DSIXReturnCode	6	6	N	Six digit return code which identifies the error type. See Section 3 on error codes for possible values
CmdStatus	1	1	A	Indicates the outcome of the command: "Success" = command completed successfully "Error" = error processing command. Check DSIXReturnCode and TextResponse for additional info on error
TextResponse	1	40	AN	For Successful responses, this field contains string "1 Server(s) Active" or "0 Server(s) Active". For other response codes, see Section 3 on error codes for values

2.4 Method: GetIpAddressFromHostName

Use: To perform a DNS lookup on a single Hostname and get an IpAddress. The IpAddress can be used by the POS application in the XML field.

Syntax: **`BSTR GetIpAddressFromHostName (BSTR HostName, SHORT ProcessControl)`**

Arguments: ProcessControl – This determines how the control will process the request. Allowed values are:

0 Process using visible dialog boxes
1 Process without using visible dialog boxes

Returns: XML formatted string response of type RStream as follows:

```
<?xml version="1.0"?>
<RStream>
  <CmdResponse>
    <ResponseOrigin>Client</ResponseOrigin>
    <DSIXReturnCode>DSIXReturnCode</DSIXReturnCode>
    <CmdStatus>CmdStatus</CmdStatus>
    <TextResponse></TextResponse>
    <IpAddress>255.255.255.255</IpAddress>
  </CmdResponse>
</RStream>
```

Element	Min	Max	Type	Description
ResponseOrigin	1	10	A	Indicates the source of the response: "Client" = generated by dsiPDCX control
DSIXReturnCode	6	6	N	Six digit return code which identifies the error type. See Section 3 on error codes for possible values
CmdStatus	1	1	A	Indicates the outcome of the command: "Success" = command completed successfully "Error" = error processing command. Check DSIXReturnCode and TextResponse for additional info on error
TextResponse	1	40	AN	Null
IpAddress	7	15	AN	The IP address of the HostName submitted. If CmdStatus is Error, the returned value may be 0.0.0.0 or null.

2.5 Method: ProcessTransaction

Use: To process payment commands defined in XML formatted command strings. See subsequent sections on available XML commands, formats and usage.

Syntax:
```
BSTR ProcessTransaction     (BSTR   XMLCommand,
                             SHORT  ProcessControl,
                             BSTR   ClientServerPassword,
                             BSTR   UserTraceData)
```

Notes: When there is already a ProcessTransaction request outstanding, submitting another ProcessTransaction request will result in an immediate error response as follows: ResponseOrigin = “Client”, DSIXReturnCode = “003002”, CmdStatus = “Success”, and TextResponse = “InProcess!” and the new ProcessTransaction requested will not be processed.

When the ProcessControl parameter is enabled (1), the ProcessTransaction method will not display any dialog boxes.

Arguments: BSTR XML Command – An XML formatted string of type <TStream> containing the details of the transaction request. See subsequent sections on available XML TStream transaction types, formats and usage.

ProcessControl – This determines how the control will process the request. Allowed values are:

0 Process using visible dialog boxes
1 Process without using visible dialog boxes

ClientServerPassword – A password consisting of 1-12 characters which is defined by the server. If the server is configured such that a ClientServerPassword is not required, then a null string should be supplied.

UserTraceData – Optional value which will be returned unaltered with response to allow application to identify a particular response.

Returns: XML formatted string response of type <RStream>. *See Section 4.1 for complete definitions, use and requirements for <RStream> response elements.*

2.7 Method: SetConnectTimeout

Use: To set the connection timeout for PingServer and timeout to failover to the next server address in the list generated by ServerIPConfig while processing transactions with ProcessTransaction.

Syntax: `BSTR SetConnectTimeout (SHORT Timeout)`

Arguments: Timeout – The timeout value in seconds (5 – 60).

Returns: SHORT value of actual timeout value set.

Notes: The default connection timeout value is 10 seconds if SetConnectTimeout is never called.

2.8 Method: SetResponseTimeout

Use: To set the response timeout while processing transactions with ProcessTransaction.

Syntax: `BSTR SetResponseTimeout (SHORT Timeout)`

Arguments: Timeout – The timeout value in seconds (60 – 3900).

Returns: SHORT value of actual timeout value set.

Notes: The default connection timeout value is 300 seconds if SetResponseTimeout is never called. The default value should be used unless unusual response performance is experienced or as advised for a particular processing connection.

2.9 Method: CancelRequest

Use: Send a 'cancel' message to the current in-process command.

Syntax: `void CancelRequest`

Arguments: None

Returns: None

Notes: Depending on the progress state of the in-process command, the CancelRequest may or may not be effective. The CancelRequest method should be called only once and should be sent from the same thread of execution as the original command. The CancelRequest method produces the same effect as the Cancel button in visible dialog box when the ProcessTransaction method is called with the ProcessControl argument equal 0.

2.A Method: GetDevicesInfo

Use: To determine the number of SecureDevices supported in the version of dsiPDCX and the device capabiltites and characteristics..

Syntax: **`BSTR GetDevicesInfo()`**

Arguments: None

Returns: XML formatted string response of type Devices as follows:

```
<Devices>
<NumSecureDevices>SSSSSS</NumSecureDevices>
<SecureDeviceNN>
    <DescriptionNN>Description</DescriptionNN>
    <InterfaceNN>Interface</InterfaceNN>
    <SecureDeviceIDNN>SecureDeviceID</SecureDeviceIDNN>
    <NumPadTypesNN>NumPadTypes</NumPadTypesNN>
    <SecureDeviceNNPadType1>SecureDevicePadType1</SecureDeviceNNPadType1>
    <SecureDeviceNNPadType2>SecureDevicePadType2</SecureDeviceNNPadType2>
</SecureDeviceNN>
</Devices>
```

Element	Min	Max	Type	Description
NumSecureDevices	1	6	N	Where SSSSSS ndicates the number of Secure Devices supported in the version of dsiPDCX
SecureDeviceNN: DescriptionNN	1	40	A	Description text of SecureDeviceNN. NN = 1 to SSSSSS.
SecureDeviceNN: InterfaceNN	1	40	AN	Interface type(s) supported for SecureDeviceNN. NN = 1 to SSSSSS.
SecureDeviceNN: SecureDeviceIDNN	1	40	AN	SecureDeviceID for SecureDeviceNN. NN = 1 to SSSSSS.
SecureDeviceNN: NumPadTypesNN	1	40	AN	Description text of SecureDeviceNN. NN = 1 to SSSSSS.
SecureDeviceNN: SecureDeviceNNPadType1	1	40	AN	Description text of SecureDeviceNN. NN = 1 to SSSSSS.
SecureDeviceNN: SecureDeviceNNPadType2	1	40	AN	Description text of SecureDeviceNN. NN = 1 to SSSSSS.

Sample Response:

```
<?xml version="1.0"?>
<Devices>
<NumSecureDevices>39</NumSecureDevices>
<SecureDevice1>
        <Description1>Datacap PDC</Description1>
        <Interface1>RS-232 or USB/VCom</Interface1>
        <SecureDeviceID1>PDC</SecureDeviceID1>
        <NumPadTypes1>2</NumPadTypes1>
```

```
        <SecureDevice1PadType1>VFI1000</SecureDevice1PadType1>
        <SecureDevice1PadType2>None</SecureDevice1PadType2>
</SecureDevice1>
<SecureDevice2>
        <Description2>Datacap PDC2</Description2>
        <Interface2>RS-232 or USB/VCom</Interface2>
        <SecureDeviceID2>PDC2</SecureDeviceID2>
        <NumPadTypes2>2</NumPadTypes2>
        <SecureDevice2PadType1>VFI1000</SecureDevice2PadType1>
        <SecureDevice2PadType2>None</SecureDevice2PadType2>
</SecureDevice2>
<SecureDevice3>
        <Description3>Verifone Vx810 PIN pad running XPI</Description3>
        <Interface3>RS-232 or USB/VCom</Interface3>
        <SecureDeviceID3>VX810XPI</SecureDeviceID3>
        <NumPadTypes3>1</NumPadTypes3>
        <SecureDevice3PadType1>VX810</SecureDevice3PadType1>
</SecureDevice3>
<SecureDevice4>
        <Description4>UIC 795 PIN pad</Description4>
        <Interface4>RS-232 or USB/VCom</Interface4>
        <SecureDeviceID4>UIC795</SecureDeviceID4>
        <NumPadTypes4>1</NumPadTypes4>
        <SecureDevice4PadType1>UIC795</SecureDevice4PadType1>
</SecureDevice4>
<SecureDevice5>
        <Description5>ViVoPay 4500m contactless with MSR</Description5>
        <Interface5>RS-232 or USB/VCom</Interface5>
        <SecureDeviceID5>VIVO4500M</SecureDeviceID5>
        <NumPadTypes5>1</NumPadTypes5>
        <SecureDevice5PadType1>None</SecureDevice5PadType1>
</SecureDevice5>
<SecureDevice6>
        <Description6>MagTek SureSwipe MSR</Description6>
        <Interface6>RS-232 or USB/VCom</Interface6>
        <SecureDeviceID6>MTSURESWIPEVCOM</SecureDeviceID6>
        <NumPadTypes6>1</NumPadTypes6>
        <SecureDevice6PadType1>None</SecureDevice6PadType1>
</SecureDevice6>
<SecureDevice7>
        <Description7>MagTek MINI MSR</Description7>
        <Interface7>RS-232 or USB/VCom</Interface7>
        <SecureDeviceID7>MTMINIMSRVCOM</SecureDeviceID7>
        <NumPadTypes7>1</NumPadTypes7>
        <SecureDevice7PadType1>None</SecureDevice7PadType1>
</SecureDevice7>
<SecureDevice8>
        <Description8>MagTek MINI MSR</Description8>
        <Interface8>RS-232</Interface8>
        <SecureDeviceID8>MTMINIMSR</SecureDeviceID8>
        <NumPadTypes8>1</NumPadTypes8>
        <SecureDevice8PadType1>None</SecureDevice8PadType1>
</SecureDevice8>
<SecureDevice9>
        <Description9>MagTek SureSwipe MSR</Description9>
        <Interface9>USB/HID</Interface9>
```

```
        <SecureDeviceID9>MTSURESWIPEHID</SecureDeviceID9>
        <NumPadTypes9>1</NumPadTypes9>
        <SecureDevice9PadType1>None</SecureDevice9PadType1>
</SecureDevice9>
<SecureDevice10>
        <Description10>MagTek MINI MSR</Description10>
        <Interface10>USB/HID</Interface10>
        <SecureDeviceID10>MTMINIMSRHID</SecureDeviceID10>
        <NumPadTypes10>1</NumPadTypes10>
        <SecureDevice10PadType1>None</SecureDevice10PadType1>
</SecureDevice10>
<SecureDevice11>
        <Description11>IDTech MSR</Description11>
        <Interface11>USB/HID</Interface11>
        <SecureDeviceID11>IDTMSRHID</SecureDeviceID11>
        <NumPadTypes11>1</NumPadTypes11>
        <SecureDevice11PadType1>None</SecureDevice11PadType1>
</SecureDevice11>
<SecureDevice12>
        <Description12>IDTech MSR no Encrypt</Description12>
        <Interface12>USB/HID</Interface12>
        <SecureDeviceID12>IDTMSRHID</SecureDeviceID12>
        <NumPadTypes12>1</NumPadTypes12>
        <SecureDevice12PadType1>None</SecureDevice12PadType1>
</SecureDevice12>
<SecureDevice13>
        <Description13>IDTech MSR Encrypted</Description13>
        <Interface13>USB/HID</Interface13>
        <SecureDeviceID13>IDTSECUREMAGHID</SecureDeviceID13>
        <NumPadTypes13>1</NumPadTypes13>
        <SecureDevice13PadType1>None</SecureDevice13PadType1>
</SecureDevice13>
<SecureDevice14>
        <Description14>MagTek IPAD PIN pad</Description14>
        <Interface14>USB/HID</Interface14>
        <SecureDeviceID14>MTIPADHID</SecureDeviceID14>
        <NumPadTypes14>2</NumPadTypes14>
        <SecureDevice14PadType1>IPAD100</SecureDevice14PadType1>
        <SecureDevice14PadType2>None</SecureDevice14PadType2>
</SecureDevice14>
<SecureDevice15>
        <Description15>Equinox Payments L5300 PIN pad</Description15>
        <Interface15>RS-232</Interface15>
        <SecureDeviceID15>EQUINOXL5300</SecureDeviceID15>
        <NumPadTypes15>2</NumPadTypes15>
        <SecureDevice15PadType1>L5300</SecureDevice15PadType1>
        <SecureDevice15PadType2>None</SecureDevice15PadType2>
</SecureDevice15>
<SecureDevice16>
        <Description16>J2 650 POS Internal MSR</Description16>
        <Interface16>RS-232</Interface16>
        <SecureDeviceID16>J2650MSRVCOM</SecureDeviceID16>
        <NumPadTypes16>1</NumPadTypes16>
        <SecureDevice16PadType1>None</SecureDevice16PadType1>
</SecureDevice16>
<SecureDevice17>
```

```
        <Description17>Firich FEC Gladius Smart AL7385 MSR</Description17>
        <Interface17>USB/HID</Interface17>
        <SecureDeviceID17>FECMSRHID</SecureDeviceID17>
        <NumPadTypes17>1</NumPadTypes17>
        <SecureDevice17PadType1>None</SecureDevice17PadType1>
</SecureDevice17>
<SecureDevice18>
        <Description18>FEC AER Touch POS Terminal MSR</Description18>
        <Interface18>USB/HID</Interface18>
        <SecureDeviceID18>FECAERMSRHID</SecureDeviceID18>
        <NumPadTypes18>1</NumPadTypes18>
        <SecureDevice18PadType1>None</SecureDevice18PadType1>
</SecureDevice18>
<SecureDevice19>
        <Description19>IBM SurePOS 500 MSR</Description19>
        <Interface19>RS-232 (19200 baud)</Interface19>
        <SecureDeviceID19>IBMSUREPOSMSR</SecureDeviceID19>
        <NumPadTypes19>1</NumPadTypes19>
        <SecureDevice19PadType1>None</SecureDevice19PadType1>
</SecureDevice19>
<SecureDevice20>
        <Description20>Micros PCWS 2015 System MSR</Description20>
        <Interface20>Internal Micros PCWS</Interface20>
        <SecureDeviceID20>MICROSPCWSMSR</SecureDeviceID20>
        <NumPadTypes20>1</NumPadTypes20>
        <SecureDevice20PadType1>None</SecureDevice20PadType1>
</SecureDevice20>
<SecureDevice21>
        <Description21>POSX EVO TP4 All-In-One POS MSR</Description21>
        <Interface21>USB/HID</Interface21>
        <SecureDeviceID21>POSXMSRHID</SecureDeviceID21>
        <NumPadTypes21>1</NumPadTypes21>
        <SecureDevice21PadType1>None</SecureDevice21PadType1>
</SecureDevice21>
<SecureDevice22>
        <Description22>AnyShop POS MSR</Description22>
        <Interface22>USB/HID</Interface22>
        <SecureDeviceID22>ANYSHOPPOSMSRHID</SecureDeviceID22>
        <NumPadTypes22>1</NumPadTypes22>
        <SecureDevice22PadType1>None</SecureDevice22PadType1>
</SecureDevice22>
<SecureDevice23>
        <Description23>Elo ETT10A1 Tablet MSR</Description23>
        <Interface23>RS-232 or USB/VCom</Interface23>
        <SecureDeviceID23>ELOTABLETMSR</SecureDeviceID23>
        <NumPadTypes23>1</NumPadTypes23>
        <SecureDevice23PadType1>None</SecureDevice23PadType1>
</SecureDevice23>
<SecureDevice24>
        <Description24>MSR or Pad on Tran</Description24>
        <Interface24>IP</Interface24>
        <SecureDeviceID24>ONTRAN</SecureDeviceID24>
        <NumPadTypes24>1</NumPadTypes24>
        <SecureDevice24PadType1>None</SecureDevice24PadType1>
</SecureDevice24>
<SecureDevice25>
```

```
        <Description25>Verifone MX915 PinPad</Description25>
        <Interface25>RS-232</Interface25>
        <SecureDeviceID25>VERIFONEMX915</SecureDeviceID25>
        <NumPadTypes25>2</NumPadTypes25>
        <SecureDevice25PadType1>MX915</SecureDevice25PadType1>
        <SecureDevice25PadType2>None</SecureDevice25PadType2>
</SecureDevice25>
<SecureDevice26>
        <Description26>PioneerPOS StealthTouch MSR</Description26>
        <Interface26>USB/HID</Interface26>
        <SecureDeviceID26>PIONEERSTM5MSRHID</SecureDeviceID26>
        <NumPadTypes26>1</NumPadTypes26>
        <SecureDevice26PadType1>None</SecureDevice26PadType1>
</SecureDevice26>
<SecureDevice27>
        <Description27>HP Retail Jacket</Description27>
        <Interface27>USB/HID</Interface27>
        <SecureDeviceID27>HPRETAILJACKET</SecureDeviceID27>
        <NumPadTypes27>1</NumPadTypes27>
        <SecureDevice27PadType1>None</SecureDevice27PadType1>
</SecureDevice27>
<SecureDevice28>
        <Description28>IDTech SecuRED Encrypted</Description28>
        <Interface28>USB/HID</Interface28>
        <SecureDeviceID28>IDTSECUREDMSRHID</SecureDeviceID28>
        <NumPadTypes28>1</NumPadTypes28>
        <SecureDevice28PadType1>None</SecureDevice28PadType1>
</SecureDevice28>
<SecureDevice29>
        <Description29>UIC 795SE PIN pad</Description29>
        <Interface29>RS-232 or USB/VCom</Interface29>
        <SecureDeviceID29>UIC795SE</SecureDeviceID29>
        <NumPadTypes29>1</NumPadTypes29>
        <SecureDevice29PadType1>UIC795SE</SecureDevice29PadType1>
</SecureDevice29>
<SecureDevice30>
        <Description30>UIC 795 SE PIN pad with TSYS VOLTAGE</Description30>
        <Interface30>RS-232 or USB/VCom</Interface30>
        <SecureDeviceID30>UIC795SE_TSYS_VOLT</SecureDeviceID30>
        <NumPadTypes30>1</NumPadTypes30>
        <SecureDevice30PadType1>UIC795SE</SecureDevice30PadType1>
</SecureDevice30>
<SecureDevice31>
        <Description31>Verifone Vx805 PIN pad running XPI</Description31>
        <Interface31>RS-232 or USB/VCom</Interface31>
        <SecureDeviceID31>VX805XPI</SecureDeviceID31>
        <NumPadTypes31>1</NumPadTypes31>
        <SecureDevice31PadType1>VX805</SecureDevice31PadType1>
</SecureDevice31>
<SecureDevice32>
        <Description32>ID Innovations Classic MSR</Description32>
        <Interface32>USB/HID</Interface32>
        <SecureDeviceID32>IDINNOVATIONS_CLASSIC_MSRHID</SecureDeviceID32>
        <NumPadTypes32>1</NumPadTypes32>
        <SecureDevice32PadType1>None</SecureDevice32PadType1>
</SecureDevice32>
```

```
<SecureDevice33>
        <Description33>ID Innovations Value MSR</Description33>
        <Interface33>USB/HID</Interface33>
        <SecureDeviceID33>IDINNOVATIONS_VALUE_MSRHID</SecureDeviceID33>
        <NumPadTypes33>1</NumPadTypes33>
        <SecureDevice33PadType1>None</SecureDevice33PadType1>
</SecureDevice33>
<SecureDevice34>
        <Description34>Verifone MX850 PinPad</Description34>
        <Interface34>RS-232</Interface34>
        <SecureDeviceID34>VERIFONEMX850</SecureDeviceID34>
        <NumPadTypes34>2</NumPadTypes34>
        <SecureDevice34PadType1>MX850</SecureDevice34PadType1>
        <SecureDevice34PadType2>None</SecureDevice34PadType2>
</SecureDevice34>
<SecureDevice35>
        <Description35>Ingenico iSC250 PinPad</Description35>
        <Interface35>RS-232</Interface35>
        <SecureDeviceID35>INGENICOISC250</SecureDeviceID35>
        <NumPadTypes35>2</NumPadTypes35>
        <SecureDevice35PadType1>ISC250</SecureDevice35PadType1>
        <SecureDevice35PadType2>None</SecureDevice35PadType2>
</SecureDevice35>
<SecureDevice36>
        <Description36>HP RP77800</Description36>
        <Interface36>USB/HID</Interface36>
        <SecureDeviceID36>HP_RP77800</SecureDeviceID36>
        <NumPadTypes36>1</NumPadTypes36>
        <SecureDevice36PadType1>None</SecureDevice36PadType1>
</SecureDevice36>
<SecureDevice37>
        <Description37>Verifone Vx805 Pinpad running XPI with Mercury E2E
Encryption</Description37>
        <Interface37>RS-232 or USB/VCom</Interface37>
        <SecureDeviceID37>VX805XPI_MERCURY_E2E</SecureDeviceID37>
        <NumPadTypes37>1</NumPadTypes37>
        <SecureDevice37PadType1>VX805</SecureDevice37PadType1>
</SecureDevice37>
<SecureDevice38>
        <Description38>Verifone Vx805 PIN pad running XPI with Contactless</Description38>
        <Interface38>RS-232 or USB/VCom</Interface38>
        <SecureDeviceID38>VX805XPI_CTLS</SecureDeviceID38>
        <NumPadTypes38>1</NumPadTypes38>
        <SecureDevice38PadType1>VX805</SecureDevice38PadType1>
</SecureDevice38>
<SecureDevice39>
        <Description39>None</Description39>
        <Interface39>None</Interface39>
        <SecureDeviceID39>NONE</SecureDeviceID39>
        <NumPadTypes39>1</NumPadTypes39>
        <SecureDevice39PadType1>None</SecureDevice39PadType1>
</SecureDevice39>
</Devices>
```

3.0 XML Admin Requests

3.1 BatchSummary Request

Use: To get batch summary information (totals and counts by card type).

Note: A BatchSummary request must be performed just prior to a BatchClose request and all relevant response fields in the BatchSummary should be supplied in the Batch Close if available.

Note: The BatchSummary is an <Admin> rather than a <Transaction> type of <TStream>. Additional information is necessary to perform <Admin> transactions – contact your Datacap technical representative for further information or use Datacap's ePay Administrator program to perform batch management functions.

Note: *Retain tags and values returned in the BatchSummary response eclosed between the <BatchSummary> tags for use in the BatchClose command.*

Response: See Section 5.3.

XML Template: **Batch Summary Request**

```
<?xml version="1.0"?>
<TStream>
  <Admin>
    <IpAddress>999.999.999.999</IpAddress>
    <IpPort>99999</IpPort>
    <TranDeviceID>TranDeviceID</TranDeviceID>
    <MerchantID>MerchantID</MerchantID>
    <TerminalID>TerminalID</TerminalID>
    <OperatorID>OperatorID</OperatorID>
    <TranCode>BatchSummary</TranCode>
    <SecureDevice>ValidSecureDeviceID</SecureDevice>
    <ComPort>ComPort</ComPort>
    <SequenceNo>SequenceNo</SequenceNo>
      <TerminalName>TerminalName</TerminalName>
    <ShiftID>ShiftID</ShiftID>
  </Admin>
</TStream>
```

Element	Req	Min	Max	Type	Description
IpAddress	O	7	15	AN	IP address of server to use for this transaction. This address will override the addresses obtained via ServerIPConfig. If an IPTranLT is the device to receive the command, this tag should be omitted.
IpPort	O	1	5	N	IP port number on server to use for this transaction. If omitted, default port is 9000. If an IPTranLT is the device to receive the command, this tag should be omitted.

TranDeviceID	O	1	24	AN	If an IPTranLT is the device to receive the command, this tag must be included. The TranDeviceID (DID) value is printed on a sticker on the bottom of every IPTranLT.
MerchantID	Y	1	24	AN	Merchant identification assigned by processor.
TerminalID	O	1	24	AN	***For general use*** Terminal ID data must be supplied in this tag only if provided by the processing provider; otherwise this tag should not be included. ***For reversal support with selected processors*** Use a unique POS workstation number (exactly 3 digits 000-999) for multi-workstation locations.
OperatorID	Y	1	10	N	Operator (clerk, server, etc.) associated with the inquiry.
TranCode	Y	1	40	A	"BatchSummary"
SecureDevice	Y	1	40	AN	Secure Device type. See Sections 1.1 and 1.3.3.3 of dsiPDCX Core Integration Specification for ValidSecureDeviceID values.
ComPort	Y	1	3	N	The true RS-232 serial port or virtual serial port for USB devices that use a virtual serial port driver. Port number to which SecureDevice is attached (0-255).
SequenceNo	O	1	20	AN	***For reversal support with selected processors*** Use '00000000' for first transaction in new installations. For all subsequent transactions, use the SequenceNo value returned in the previous transaction response.
TerminalName	O	1	20	AN	TerminalName
ShiftID	O	1	20	AN	Shift identification

3.2 BatchClear Request

Use: To clear the currently open batch. *CAUTION!* Clearing the batch erases all transaction information. Make sure that the batch is settled successfully and/or that you have printed copies for all drafts in the batch should they need to be manually re-entered.

Note: The BatchClear is an <Admin> rather than a <Transaction> type of <TStream>. Additional information is necessary to perform <Admin> transactions – contact your Datacap technical representative for further information or use Datacap's ePay Administrator program to perform batch management functions.

Response: See Section 5.4.

XML Template: **Batch Clear Request**

```
<?xml version="1.0"?>
<TStream>
  <Admin>
    <IpAddress>999.999.999.999</IpAddress>
    <IpPort>99999</IpPort>
    <TranDeviceID>TranDeviceID</TranDeviceID>
    <MerchantID>MerchantID</MerchantID>
    <TerminalID>TerminalID</TerminalID>
    <OperatorID>OperatorID</OperatorID>
    <SequenceNo>SequenceNo</SequenceNo>
    <TranCode>BatchClear</TranCode>
    <SecureDevice>ValidSecureDeviceID</SecureDevice>
    <ComPort>ComPort</ComPort>
    <SequenceNo>SequenceNo</SequenceNo>
      <TerminalName>TerminalName</TerminalName>
    <ShiftID>ShiftID</ShiftID>
  </Admin>
</TStream>
```

Element	Req	Min	Max	Type	Description
IpAddress	O	7	15	AN	IP address of server to use for this transaction. This address will override the addresses obtained via ServerIPConfig. If an IPTranLT is the device to receive the command, this tag should be omitted.
IpPort	O	1	5	N	IP port number on server to use for this transaction. If omitted, default port is 9000. If an IPTranLT is the device to receive the command, this tag should be omitted.
TranDeviceID	O	1	24	AN	If an IPTranLT is the device to receive the command, this tag must be included. The TranDeviceID (DID) value is printed on a sticker on the bottom of every IPTranLT.
MerchantID	Y	1	24	AN	Merchant identification assigned by processor.
TerminalID	O	1	24	AN	***For general use***

					Terminal ID data must be supplied in this tag only if provided by the processor or merchant service provider; otherwise this tag should not be included. ***For reversal support with selected processors*** Use a unique POS workstation number (exactly 3 digits 000-999) for multi-workstation locations.
OperatorID	Y	1	10	N	Operator (clerk, server, etc.) associated with the inquiry.
TranCode	Y	1	40	A	"BatchClear"
SecureDevice	Y	1	40	AN	Secure Device type. See Sections 1.1 and 1.3.3.3 of dsiPDCX Core Integration Specification for ValidSecureDeviceID values.
ComPort	Y	1	3	N	The true RS-232 serial port or virtual serial port for USB devices that use a virtual serial port driver. Port number to which SecureDevice is attached (0-255).
SequenceNo	O	1	20	AN	***For reversal support with selected processors*** Use '00000000' for first transaction in new installations. For all subsequent transactions, use the SequenceNo value returned in the previous transaction response.
TerminalName	O	1	20		Terminal name
ShiftID	O	1	20		Shift identification

3.3 BatchClose Request

Use: To close the currently open batch.

Note: A BatchSummary request must be performed just prior to a BatchClose request and all relevant (matching) response fields in the BatchSummary should be supplied in the Batch Close if available.

Note: The BatchClose is an <Admin> rather than a <Transaction> type of <TStream>. Additional information is necessary to perform <Admin> transactions – contact your Datacap technical representative for further information or use Datacap's ePay Administrator program to perform batch management functions.

Note: A BatchSummary request must be performed immediately prior to a BatchClose request and all *matching* response fields (including those with 0.00 values) in the BatchSummary response must be supplied in the BatchClose request.

Response: See Section 5.5.

XML Template: **Batch Close Request**

```
<?xml version="1.0"?>
<TStream>
  <Admin>
    <IpAddress>999.999.999.999</IpAddress>
    <IpPort>99999</IpPort>
    <TranDeviceID>TranDeviceID</TranDeviceID>
    <MerchantID>MerchantID</MerchantID>
    <TerminalID>TerminalID</TerminalID>
    <OperatorID>OperatorID</OperatorID>
    <TranCode>BatchClose</TranCode>
    <SecureDevice>ValidSecureDeviceID</SecureDevice>
    <ComPort>ComPort</ComPort>
    <BatchNo>BatchNo</BatchNo>
    <BatchItemCount>BatchItemCount</BatchItemCount>
    <NetBatchTotal>NetBatchTotal</NetBatchTotal>
    <CreditPurchaseCount>CreditPurchaseCount</CreditPurchaseCount>
    <CreditPurchaseAmount>CreditPurchaseAmount</CreditPurchaseAmount>
    <CreditReturnCount>CreditReturnCount</CreditReturnCount>
    <CreditReturnAmount>CreditReturnAmount</CreditReturnAmount>
    <DebitPurchaseCount>DebitPurchaseCount</DebitPurchaseCount>
    <DebitPurchaseAmount>DebitPurchaseAmount</DebitPurchaseAmount>
    <DebitReturnCount>DebitReturnCount</DebitReturnCount>
    <DebitReturnAmount>DebitReturnAmount</DebitReturnAmount>
    <SequenceNo>SequenceNo</SequenceNo>
    <TerminalName>TerminalName</TerminalName>
    <ShiftID>ShiftID</ShiftID>
  </Admin>
</TStream>
```

Element	Req	Min	Max	Type	Description
IpAddress	O	7	15	AN	IP address of server to use for this transaction. This address will override the addresses obtained via ServerIPConfig. If an IPTranLT is the device to receive the command, this tag should be omitted.
IpPort	O	1	5	N	IP port number on server to use for this transaction. If omitted, default port is 9000. If an IPTranLT is the device to receive the command, this tag should be omitted.
TranDeviceID	O	1	24	AN	If an IPTranLT is the device to receive the command, this tag must be included. The TranDeviceID (DID) value is printed on a sticker on the bottom of every IPTranLT.
MerchantID	Y	1	24	AN	Merchant identification assigned by processor.
TerminalID	O	1	24	AN	***For general use*** Terminal ID data must be supplied in this tag only if provided by the processor or merchant service provider; otherwise this tag should not be included. ***For reversal support with selected processors*** Use a unique POS workstation number (exactly 3 digits 000-999) for multi-workstation locations.
OperatorID	Y	1	10	N	Operator (clerk, server, etc.) associated with the inquiry.
TranCode	Y	1	40	A	"BatchClose"
SecureDevice	Y	1	40	AN	Secure Device type. See Sections 1.1 and 1.3.3.3 of dsiPDCX Core Integration Specification for ValidSecureDeviceID values.
ComPort	Y	1	3	N	The true RS-232 serial port or virtual serial port for USB devices that use a virtual serial port driver. Port number to which SecureDevice is attached (0-255).
BatchNo	Y	1	6	AN	Batch number returned by BatchSummary request
BatchItemCount	Y	1	8	N	Number of total items in batch returned by BatchSummary request
NetBatchTotal	Y	1	10	N	Net of all transactions in batch returned by BatchSummary request
CreditPurchaseCount	O	1	8	N	Number of credit purchase transactions in batch returned by BatchSummary request. Note: This value must be included in the BatchClose request if previously obtained in BatchSummary response.
CreditPurchaseAmount	O	1	10	N	Net of all credit purchase transactions in batch returned by BatchSummary request. Note: This value must be included in the BatchClose request if previously returned in BatchSummary response.
CreditReturnCount	O	1	8	N	Number of credit return transactions in batch returned by BatchSummary request. Note: This value must be included in the BatchClose request if previously returned in BatchSummary response.
CreditReturnAmount	O	1	10	N	Net of all credit return transactions in batch returned by BatchSummary request. Note: This value must be included in the BatchClose request if previously returned in BatchSummary response.

DebitPurchaseCount	O	1	8	N	Number of debit purchase transactions in batch returned by BatchSummary request. Note: This value must be included in the BatchClose request if previously returned in BatchSummary response.
DebitPurchaseAmount	O	1	10	N	Net of all debit purchase transactions in batch returned by BatchSummary request. Note: This value must be included in the BatchClose request if previously returned in BatchSummary response.
DebitReturnCount	O	1	8	N	Number of debit return transactions in batch returned by BatchSummary request. Note: This value must be included in the BatchClose request if previously returned in BatchSummary response.
DebitReturnAmount	O	1	10	N	Net of all debit return transactions in batch returned by BatchSummary request. Note: This value must be included in the BatchClose request if previously returned in BatchSummary response.
SequenceNo	O	1	20	AN	***For reversal support with selected processors*** Use '00000000' for first transaction in new installations. For all subsequent transactions, use the SequenceNo value returned in the previous transaction response.
TerminalName	O	1	20	AN	Terminal name
ShiftID	O	1	20	AN	Shift identification

3.4 Batch Number Change Request

Use: To change the number of the currently open batch.

Note: A BatchSummary request should only be performed on the recommendation of processing provider support services.

Note: The BatchSummary is an <Admin> rather than a <Transaction> type of <TStream>. Additional information is necessary to perform <Admin> transactions – contact your Datacap technical representative for further information or use Datacap's ePay Administrator program to perform batch management functions.

Response: See Section 5.3.

XML Template: **BatchNumber Request**

```
<?xml version="1.0"?>
<TStream>
  <Admin>
    <IpAddress>999.999.999.999</IpAddress>
    <IpPort>99999</IpPort>
    <TranDeviceID>TranDeviceID</TranDeviceID>
    <MerchantID>MerchantID</MerchantID>
    <TerminalID>TerminalID</TerminalID>
    <OperatorID>OperatorID</OperatorID>
    <TranCode>BatchNumber</TranCode>
    <SecureDevice>ValidSecureDeviceID</SecureDevice>
    <ComPort>ComPort</ComPort>
    <SequenceNo>SequenceNo</SequenceNo>
      <TerminalName>TerminalName</TerminalName>
    <ShiftID>ShiftID</ShiftID>
    <BatchNumber>BatchNumber</BatchNumber>
  </Admin>
</TStream>
```

Element	Req	Min	Max	Type	Description
IpAddress	O	7	15	AN	IP address of server to use for this transaction. This address will override the addresses obtained via ServerIPConfig. If an IPTranLT is the device to receive the command, this tag should be omitted.
IpPort	O	1	5	N	IP port number on server to use for this transaction. If omitted, default port is 9000. If an IPTranLT is the device to receive the command, this tag should be omitted.
TranDeviceID	O	1	24	AN	If an IPTranLT is the device to receive the command, this tag must be included. The TranDeviceID (DID) value is printed on a sticker on the bottom of every IPTranLT.
MerchantID	Y	1	24	AN	Merchant identification assigned by processor.

TerminalID	O	1	24	AN	***For general use*** Terminal ID data must be supplied in this tag only if provided by the processing provider; otherwise this tag should not be included. ***For reversal support with selected processors*** Use a unique POS workstation number (exactly 3 digits 000-999) for multi-workstation locations.
OperatorID	Y	1	10	N	Operator (clerk, server, etc.) associated with the inquiry.
TranCode	Y	1	40	A	"BatchSummary"
SecureDevice	Y	1	40	AN	Secure Device type. See Sections 1.1 and 1.3.3.3 of dsiPDCX Core Integration Specification for ValidSecureDeviceID values.
ComPort	Y	1	3	N	The true RS-232 serial port or virtual serial port for USB devices that use a virtual serial port driver. Port number to which SecureDevice is attached (0-255).
SequenceNo	O	1	20	AN	***For reversal support with selected processors*** Use '00000000' for first transaction in new installations. For all subsequent transactions, use the SequenceNo value returned in the previous transaction response.
TerminalName	O	1	20	AN	TerminalName
ShiftID	O	1	20	AN	Shift identification
BatchNumber	Y	1	10	N	Number to which the currently open batch will be changed.

3.5 ServerVersion Request

Use: To obtain the information on the NETePay, DIALePay or GIFTePay server. Returned information includes server type (NETePay, DIALePay or GIFTePay), type of server (host or terminal), processing service (VITAL, Concord, CardNet, etc) and server version.

Note: The ServerVersion is an <Admin> rather than a <Transaction> type of <TStream>. Additional information is necessary to perform <Admin> transactions – contact your Datacap technical representative for further information.

Response: See Section 5.6.

XML Template: **ServerVersion Request**

```
<?xml version="1.0"?>
<TStream>
  <Admin>
    <IpAddress>999.999.999.999</IpAddress>
    <IpPort>99999</IpPort>
    <TranDeviceID>TranDeviceID</TranDeviceID>
    <MerchantID>MerchantID</MerchantID>
    <TerminalID>TerminalID</TerminalID>
    <OperatorID>OperatorID</OperatorID>
    <TranCode>ServerVersion</TranCode>
    <SecureDevice>ValidSecureDeviceID</SecureDevice>
    <ComPort>ComPort</ComPort>
    <SequenceNo>SequenceNo</SequenceNo>
      <TerminalName>TerminalName</TerminalName>
    <ShiftID>ShiftID</ShiftID>
  </Admin>
</TStream>
```

Element	Req	Min	Max	Type	Description
IpAddress	O	7	15	AN	IP address of server to use for this transaction. This address will override the addresses obtained via ServerIPConfig.
IpPort	O	1	5	N	IP port number on server to use for this transaction. If omitted, default port is 9000.
TranDeviceID	O	1	24	AN	If an IPTranLT is the device to receive the command, this tag must be included. The TranDeviceID (DID) value is printed on a sticker on the bottom of every IPTranLT.
MerchantID	Y	1	24	AN	Merchant identification assigned by processor.
TerminalID	O	1	24	AN	***For general use*** Terminal ID data must be supplied in this tag only if provided by the processor or merchant service provider; otherwise this tag should not be included. ***For reversal support with selected processors*** Use a unique POS workstation number (exactly 3 digits

					000-999) for multi-workstation locations.
OperatorID	Y	1	10	N	Operator (clerk, server, etc.) associated with the inquiry.
TranCode	Y	1	40	A	“ServerVersion”
SecureDevice	Y	1	40	AN	Secure Device type. See Sections 1.1 and 1.3.3.3 of dsiPDCX Core Integration Specification for ValidSecureDeviceID values.
ComPort	Y	1	3	N	The true RS-232 serial port or virtual serial port for USB devices that use a virtual serial port driver. Port number to which SecureDevice is attached (0-255).
SequenceNo	O	1	20	AN	***For reversal support with selected processors*** Use ‘00000000’ for first transaction in new installations. For all subsequent transactions, use the SequenceNo value returned in the previous transaction response.
TerminalName	O	1	20		Terminal name
ShiftID	O	1	20		Shift identification

3.6 Loyalty Admin Transactions

Use: New <Admin> class transactions have been added to support selected Loyalty/PrePaid processing services.

Note: See the dsiPDCX Loyalty Transactions XML Specification - Supplement 7 for specific implementation details.

4.0 GetSignature Request

Use: To obtain a digitized signature from a SecureDevice equipped with a digitizing pad.

Note: At the time the latest version of this documentation was released, the SecureDevice 'UIC795' supported the GetSignature command when equipped with a signature pad.

Response: See Section 5.7.

XML Template: **GetSignature Request**

```
<?xml version="1.0"?>
<TStream>
  <Transaction>
    <IpAddress>999.999.999.999</IpAddress>
    <IpPort>99999</IpPort>
    <MerchantID>MerchantID</MerchantID>
    <TerminalID>TerminalID</TerminalID>
    <OperatorID>OperatorID</OperatorID>
    <TranCode>GetSignature</TranCode>
    <SecureDevice>ValidSecureDeviceID</SecureDevice>
    <Account>
      <AcctNo>SecureDevice</AcctNo>
    </Account>
  </Transaction>
</TStream>
```

Element	Req	Min	Max	Type	Description
IpAddress	O	7	15	AN	IP address of server to use for this transaction. This address will override the addresses obtained via ServerIPConfig.
IpPort	O	1	5	N	IP port number on server to use for this transaction. If omitted, default port is 9000.
MerchantID	Y	1	24	AN	Merchant identification assigned by processor.
TerminalID	O	1	24	AN	***For general use*** Terminal ID data must be supplied in this tag only if provided by the processor or merchant service provider; otherwise this tag should not be included. ***For reversal support with selected processors*** Use a unique POS workstation number (exactly 3 digits 000-999) for multi-workstation locations.
OperatorID	Y	1	10	N	Operator (clerk, server, etc.) associated with the inquiry.
TranCode	Y	1	40	A	"GetSignature"
SecureDevice	Y	1	40	AN	Secure Device type. See Sections 1.1 and 1.3.3.3 of dsiPDCX Core Integration Specification for ValidSecureDeviceID values.
Account:AcctNo	Y	1	40	AN	"SecureDevice"

5.0 XML Responses

5.1 Transaction Response

```
<?xml version="1.0"?>
<RStream>
  <CmdResponse>
    <ResponseOrigin>ResponseOrigin</ResponseOrigin>
    <DSIXReturnCode>DSIXReturnCode</DSIXReturnCode>
    <CmdStatus>CmdStatus</CmdStatus>
    <TextResponse>TextResponse</TextResponse>
    <UserTraceData>UserTraceData</UserTraceData>
    <SequenceNo>SequenceNo</SequenceNo>
  </CmdResponse>
  <TranResponse>
     <MerchantID>MerchantID</MerchantID>
    <TerminalID>TerminalID</TerminalID>
    <AcctNo>AcctNo</AcctNo>
    <ExpDate>ExpDate</ExpDate>
    <CardType>CardType</CardType>
    <TranCode>TranCode</TranCode>
    <AuthCode>AuthCode</AuthCode>
    <AVSResult>AVSResult</AVSResult>
    <CVVResult>CVVResult</CVVResult>
    <VoucherNo>VoucherNo</VoucherNo>
    <CaptureStatus>CaptureStatus</CaptureStatus>
    <RefNo>RefNo</RefNo>
     <InvoiceNo>InvoiceNo</InvoiceNo>
    <OperatorID>OperatorID</OperatorID>
    <Memo>Memo</Memo>
    <Amount>
      <Purchase>Purchase</Purchase>
      <Authorize>Authorize</Authorize>
      <Gratuity>Gratuity</Gratuity>
      <CashBack>CashBack</CashBack>
      <Balance>Balance</Balance>
    </Amount>
    <AcqRefData>AcqRefData</AcqRefData>
    <RecordNo>RecordNo</RecordNo>
  <IssueCurrency>IssueCurrency</IssueCurrency>
    <IssueCurrencyPreviousBalance>IssueCurrencyPreviousBalance</IssueCurrencyPreviousBalance>
    <IssueCurrencyEndingBalance>IssueCurrencyEndingBalance</IssueCurrencyEndingBalance>
    <ExchangeRate>ExchangeRate</ExchangeRate>
    <PrePaidExp>PrePaidExp</PrePaidExp>
  </TranResponse>
</RStream>
```

Element	Return	Min	Max	Type	Description
ResponseOrigin	Y	1	10	A	Indicates the source of the response: "Client" = generated by DSIClientX control "Server" = generated by Datacap server "Processor" = generated by payment processor
DSIXReturnCode	Y	6	6	N	Six digit return code which identifies the error type. See Section 3 on error codes for possible values
CmdStatus	Y	1	10	A	Indicates the outcome of the command: "Approved" = transaction approved by payment processor "Declined" = transaction declined by payment processor "Success" = command completed successfully "Error" = error processing command. Check DSIXReturnCode and TextResponse for additional info on error
TextResponse	Y	1	40	AN	For Successful responses, this field can contain an optional message from the server or processor. For other response codes, see Section 3 on Error Codes for values.
UserTraceData	Y	0	40	AN	Echo of data supplied by the user system in the request; for use by the user system for internal tracking. May be null if no user system data was supplied in the request
SequenceNo	O	1	20	N	Transaction sequence number (included in Canadian Debit transaction responses)
MerchantID	Y	1	24	AN	Merchant identification assigned by processor.
TerminalID	O	0	24	AN	Terminal ID data must be supplied in this tag only if provided by the processor or merchant service provider; otherwise this tag should not be included.
AcctNo	Y	1	19	N	Cardholder Account number.
ExpDate	Y	4	4	N	Expiration date as MMYY
CardType	O	3	20	AN	"VISA" "M/C" "AMEX" "DCLB" "DCVR" "JCB" "DEBIT" "OTHER" "Foodstamp" "Cash" "PrePaid"
TranCode	Y	1	40	A	"Sale" "Return" "VoidSale" "VoidReturn" "AuthOnly" "PreAuth" "VoiceAuth" "PreAuthCapture" "Voucher" "Issue" "Balance" "Adjust"
AuthCode	O	1	16	AN	The authorization code returned by the processor for the approved

					transaction. ***The AuthCode value should be saved since it is required for VoidSale, Adjust, VoiceAuth and PreAuthCapture transactions.***
AVSResult	O	0	40	AN	The result of an AVS transaction returned by the processor.
CVVResult	O	0	40	AN	The result of an AVS transaction returned by the processor.
VoucherNo	O	0	40	AN	The of an EBT transaction returned by the processor.
CaptureStatus	O	1	16	AN	"Captured" "NotCaptured"
RefNo	O	1	16	AN	A reference returned by the processor. ***The RefNo value returned by the processor should be saved since it is required for subsequent VoidSale, Adjust, VoiceAuth and PreAuthCapture transactions.***
InvoiceNo	Y	1	16	AN	Use the InvoiceNo value supplied in the original request.
OperatorID	Y	1	10	N	Operator (clerk, server, etc.) associated with the Transaction.
Memo	O	0	40	AN	Echo of data supplied by the user system in the request; for use by the user system for internal tracking.
Amount:Purchase	Y	1	8	N	Purchase price (with 2 place decimal)
Amount:Authorize	O	1	8	N	Authorized amount (with 2 place decimal)
Amount:Gratuity	O	1	8	N	Gratuity amount (with 2 place decimal)
Amount:CashBack	O	1	8	N	CashBack amount (with 2 place decimal)
Amount:Balance	O	1	8	N	Balance amount (with 2 place decimal) for EBT or PrePaid transactions.
AcqRefData	O	1	200	AN	Acquirer Reference returned by processor on PreAuth Transaction requests – will be null for other TranCodes. ***The AcqRefData value returned by the processor should be saved since it is required for subsequent PreAuthCapture transactions.***
RecordNo	O	1	99	AN	Indicates record identifier in current batch. The RecordNo value can be saved and used as a token to process subsequent VoidSaleByRecordNo, VoidReturnByRecordNo and AdjustByRecordNo transactions without using account number data.
IssueCurrency	O	3	3	AN	Indicates the currency type in which the prepaid card account was originally issued. Supported values are: "CAN" - Canadian dollars "USD" - U.S. dollars **Note:** If this tag is returned on a prepaid transaction, it must be printed on the receipt with the legend "Currency" and the returned value.
IssueCurrencyPreviousBalance	O	1	8	N	Balance amount in issue currency (with 2 place decimal) at the start of the transaction. **Note:** If this tag is returned on a prepaid transaction, it must be printed on the receipt under the Currency line with the legend "Previous Balance" and the returned value.
IssueCurrencyEndingBalance	O	1	8	N	Balance amount in issue currency (with 2 place decimal) at the completion of the transaction. **Note:** If this tag is returned on a prepaid transaction, it must be printed on the receipt under the Previous Balance line with the legend "Ending Balance" and the returned value.
ExchangeRate	O	1	8	N	Exchange rate used to calculate from local to issued currency.

					Note: If this tag is returned on a prepaid transaction, it may be optionally printed under the Ending Balance line on the receipt with the legend “Exchange Rate” and the returned value.
PrePaidExp	O	1	8	AN	PrePaid card expiration date OR Trace Number. Date format: 12/31/04 or 2004-12-31 Trace Number Format: 99999999

5.2 Loyalty Transaction Response

Loyalty Response - Format 1

```
<?xml version="1.0"?>
<RStream>
  <CmdResponse>
    <ResponseOrigin>ResponseOrigin</ResponseOrigin>
    <DSIXReturnCode>DSIXReturnCode</DSIXReturnCode>
    <CmdStatus>CmdStatus</CmdStatus>
    <TextResponse>TextResponse</TextResponse>
    <UserTraceData>UserTraceData</UserTraceData>
    <SequenceNo>SequenceNo</SequenceNo>
  </CmdResponse>
  <TranResponse>
    <MerchantID>MerchantID</MerchantID>
    <TerminalID>TerminalID</TerminalID>
    <AcctNo>AcctNo</AcctNo>
    <ExpDate>ExpDate</ExpDate>
    <CardType>CardType</CardType>
    <TranCode>TranCode</TranCode>
    <AuthCode>AuthCode</AuthCode>
    <CaptureStatus>CaptureStatus</CaptureStatus>
    <RefNo>RefNo</RefNo>
    <InvoiceNo>InvoiceNo</InvoiceNo>
    <OperatorID>OperatorID</OperatorID>
    <Format>1</Format>
    <Memo>Memo</Memo>
    <Amount>
      <Purchase>Purchase</Purchase>
      <Authorize>Authorize</Authorize>
    </Amount>
    <LoyaltyExp>LoyaltyExp</LoyaltyExp>
    <Items>
      <PointsBalance>PointsBalance</PointsBalance>
      <UnitsBalance>UnitsBalance</UnitsBalance>
      <PriceBalance>PriceBalance</PriceBalance>
    </Items>
  </TranResponse>
</RStream>
```

Element	Type	Min	Max	Type	Description
ResponseOrigin	Y	1	10	A	Indicates the source of the response: "Client" = generated by DSIClientX control "Server" = generated by Datacap ePay server "Bridge" = generated by Datacap Bridge server "Processor" = generated by payment processor
DSIXReturnCode	Y	6	6	N	Six digit return code which identifies the error type. See Section 3 on error codes for possible values
CmdStatus	Y	1	10	A	Indicates the outcome of the command: "Approved" = transaction approved by payment processor

					"Declined" = transaction declined by payment processor "Success" = command completed successfully "Error" = error processing command. Check DSIXReturnCode and TextResponse for additional info on error
TextResponse	Y	1	40	AN	For Successful responses, this field can contain an optional message from the server or processor. For other response codes, see Section 3 on Error Codes for values.
UserTraceData	Y	0	40	AN	Echo of data supplied by the user system in the request; for use by the user system for internal tracking. May be null if no user system data was supplied in the request
SequenceNo	O	1	20	N	Transaction sequence number (included in Canadian Debit transaction responses)
MerchantID	Y	1	24	AN	Merchant identification assigned by processor.
TerminalID	O	0	24	AN	Terminal ID data must be supplied in this tag only if provided by the processor or merchant service provider; otherwise this tag should not be included.
AcctNo	Y	1	19	N	Cardholder Account number. Keep original AcctNo information used in Loyalty request.
ExpDate	Y	4	4	N	Expiration date as MMYY
CardType	O	3	20	AN	"Loyalty"
TranCode	Y	1	40	A	"Issue" "Add" "Subtract" "VoidIssue" "VoidAdd" "VoidSubtract" "Balance"
AuthCode	O	1	16	AN	The authorization code returned by the processor for the approved Transaction.
RefNo	O	1	16	AN	A reference returned by the processor. If no InvoiceNo returned in response, retain original InvoiceNo. The RefNo should be retained as it is necessary for all Void transaction requests.
InvoiceNo	Y	1	16	AN	Use the InvoiceNo value supplied in the original request
OperatorID	O	1	10	N	Operator (clerk, server, etc.) associated with the Transaction.
Format	Y	1	3	N	"1"
Memo	O	0	40	AN	Echo of data supplied by the user system in the request; for use by the user system for internal tracking.
Amount:Purchase	Y	Y	1	8	Purchase price (with 2 place decimal)
Amount:Authorize	O	Y	1	8	Authorized amount (with 2 place decimal)
LoyaltyExp	O	1	8	AN	Loyalty card expiration date OR Trace Number.
Items:PointsBalance	Y	1	8	N	Item 1 Points balance
Items:UnitsBalance	Y	1	8	N	Item 1 Units balance
Items:PriceBalance	Y	1	8	N	Item 1 Price/amount
Items:PriceBalance8	O	1	8	N	Item 8 Price/amount balance

Loyalty Response - Format 2

```
<?xml version="1.0"?>
<RStream>
  <CmdResponse>
    <ResponseOrigin>ResponseOrigin</ResponseOrigin>
    <DSIXReturnCode>DSIXReturnCode</DSIXReturnCode>
    <CmdStatus>CmdStatus</CmdStatus>
    <TextResponse>TextResponse</TextResponse>
    <UserTraceData>UserTraceData</UserTraceData>
    <SequenceNo>SequenceNo</SequenceNo>
  </CmdResponse>
  <TranResponse>
    <MerchantID>MerchantID</MerchantID>
    <TerminalID>TerminalID</TerminalID>
    <AcctNo>AcctNo</AcctNo>
    <ExpDate>ExpDate</ExpDate>
    <CardType>CardType</CardType>
    <TranCode>TranCode</TranCode>
    <AuthCode>AuthCode</AuthCode>
    <CaptureStatus>CaptureStatus</CaptureStatus>
    <RefNo>RefNo</RefNo>
    <InvoiceNo>InvoiceNo</InvoiceNo>
    <OperatorID>OperatorID</OperatorID>
    <Format>2</Format>
    <Memo>Memo</Memo>
    <Amount>
      <Purchase>Purchase</Purchase>
      <Authorize>Authorize</Authorize>
    </Amount>
    <LoyaltyExp>LoyaltyExp</LoyaltyExp>
    <Items>
      <PointsBalance>PointsBalance</PointsBalance>
      <UnitsBalance>UnitsBalance</UnitsBalance>
      <PriceBalance>PriceBalance</PriceBalance>
      <PointsBalance2>PointsBalance2</PointsBalance2>
      <UnitsBalance2>UnitsBalance2</UnitsBalance2>
      <PriceBalance2>PriceBalance2</PriceBalance2>
      <PointsBalance3>PointsBalance3</PointsBalance3>
      <UnitsBalance3>UnitsBalance3</UnitsBalance3>
      <PriceBalance3>PriceBalance3</PriceBalance3>
      <PointsBalance4>PointsBalance4</PointsBalance4>
      <UnitsBalance4>UnitsBalance4</UnitsBalance4>
      <PriceBalance4>PriceBalance4</PriceBalance4>
      <PointsBalance5>PointsBalance5</PointsBalance5>
      <UnitsBalance5>UnitsBalance5</UnitsBalance5>
      <PriceBalance5>PriceBalance5</PriceBalance5>
      <PointsBalance6>PointsBalance6</PointsBalance6>
      <UnitsBalance6>UnitsBalance6</UnitsBalance6>
      <PriceBalance6>PriceBalance6</PriceBalance6>
      <PointsBalance7>PointsBalance7</PointsBalance7>
      <UnitsBalance7>UnitsBalance7</UnitsBalance7>
      <PriceBalance7>PriceBalance7</PriceBalance7>
      <PointsBalance8>PointsBalance8</PointsBalance8>
      <UnitsBalance8>UnitsBalance8</UnitsBalance8>
      <PriceBalance8>PriceBalance8</PriceBalance8>
    </Items>
```

```
    <TotalPointsBalance>TotalPointsBalance</TotalPointsBalance>
    <TotalUnitsBalance>TotalUnitsBalance</TotalUnitsBalance>
    <TotalPriceBalance>TotalPriceBalance</TotalPriceBalance>
    <WriteReceipt>Receipt Print Data</WriteReceipt>
  </TranResponse>
</RStream>
```

Element	Type	Min	Max	Type	Description
ResponseOrigin	Y	1	10	A	Indicates the source of the response: "Client" = generated by DSIClientX control "Server" = generated by Datacap ePay server "Bridge" = generated by Datacap Bridge server "Processor" = generated by payment processor
DSIXReturnCode	Y	6	6	N	Six digit return code which identifies the error type. See Section 3 on error codes for possible values
CmdStatus	Y	1	10	A	Indicates the outcome of the command: "Approved" = transaction approved by payment processor "Declined" = transaction declined by payment processor "Success" = command completed successfully "Error" = error processing command. Check DSIXReturnCode and TextResponse for additional info on error
TextResponse	Y	1	40	AN	For Successful responses, this field can contain an optional message from the server or processor. For other response codes, see Section 3 on Error Codes for values.
UserTraceData	Y	0	40	AN	Echo of data supplied by the user system in the request; for use by the user system for internal tracking. May be null if no user system data was supplied in the request
SequenceNo	O	1	20	N	Transaction sequence number (included in Canadian Debit transaction responses)
MerchantID	Y	1	24	AN	Merchant identification assigned by processor.
TerminalID	O	0	24	AN	Terminal ID data must be supplied in this tag only if provided by the processor or merchant service provider; otherwise this tag should not be included.
AcctNo	Y	1	19	N	Cardholder Account number. Keep original AcctNo information used in Loyalty request.
ExpDate	Y	4	4	N	Expiration date as MMYY
CardType	O	3	20	AN	"Loyalty"
TranCode	Y	1	40	A	"Issue" "Add" "Subtract" "VoidIssue" "VoidAdd" "VoidSubtract" "Balance"
AuthCode	O	1	16	AN	The authorization code returned by the processor for the approved Transaction.

RefNo	O	1	16	AN	A reference returned by the processor. If no InvoiceNo returned in response, retain original InvoiceNo. The RefNo should be retained as it is necessary for all Void transaction requests.
InvoiceNo	Y	1	16	AN	Use the InvoiceNo value supplied in the original request
OperatorID	O	1	10	N	Operator (clerk, server, etc.) associated with the Transaction.
Format	Y	1	3	N	“2”
Memo	O	0	40	AN	Echo of data supplied by the user system in the request; for use by the user system for internal tracking.
Amount:Purchase	Y	1	8	N	Purchase price (with 2 place decimal)
Amount:Authorize	O	1	8	N	Authorized amount (with 2 place decimal)
LoyaltyExp	O	1	8	AN	Loyalty card expiration date OR Trace Number.
Items:PointsBalance	Y	1	8	N	Item 1 Points balance
Items:UnitsBalance	Y	1	8	N	Item 1 Units balance
Items:PriceBalance	Y	1	8	N	Item 1 Price/amount
Items:PointsBalance2	O	1	8	N	Item 2 Points balance
Items:UnitsBalance2	O	1	8	N	Item 2 Units balance
Items:PriceBalance2	O	1	8	N	Item 2 Price/amount balance
Items:PointsBalance3	O	1	8	N	Item 3 Points balance
Items:UnitsBalance3	O	1	8	N	Item 3 Units balance
Items:PriceBalance3	O	1	8	N	Item 3 Price/amount balance
Items:PointsBalance4	O	1	8	N	Item 4 Points balance
Items:UnitsBalance4	O	1	8	N	Item 4 Units balance
Items:PriceBalance4	O	1	8	N	Item 4 Price/amount balance
Items:PointsBalance5	O	1	8	N	Item 5 Points balance
Items:UnitsBalance5	O	1	8	N	Item 5 Units balance
Items:PriceBalance5	O	1	8	N	Item 5 Price/amount balance
Items:DescBalance5	O	1	40	AN	Item 5 Description
Items:PointsBalance6	O	1	8	N	Item 6 Points balance
Items:UnitsBalance6	O	1	8	N	Item 6 Units balance
Items:PriceBalance6	O	1	8	N	Item 6 Price/amount balance
Items:PointsBalance7	O	1	8	N	Item 7 Points balance
Items:UnitsBalance7	O	1	8	N	Item 7 Units balance
Items:PriceBalance7	O	1	8	N	Item 7 Price/amount balance
Items:PointsBalance8	O	1	8	N	Item 8 Points balance
Items:UnitsBalance8	O	1	8	N	Item 8 Units balance
Items:PriceBalance8	O	1	8	N	Item 8 Price/amount balance

5.3 BatchSummary Response

```
<?xml version="1.0"?>
<RStream>
  <CmdResponse>
    <ResponseOrigin>ResponseOrigin</ResponseOrigin>
    <DSIXReturnCode>DSIXReturnCode</DSIXReturnCode>
    <CmdStatus>CmdStatus</CmdStatus>
    <TextResponse>TextResponse</TextResponse>
    <UserTraceData>UserTraceData</UserTraceData>
  </CmdResponse>
  <BatchSummary>
    <MerchantID>MerchantID</MerchantID>
    <OperatorID>OperatorID</OperatorID>
    <TerminalID>TerminalID</TerminalID>
    <BatchNo>BatchNo</BatchNo>
    <BatchItemCount>BatchItemCount</BatchItemCount >
    <NetBatchTotal>NetBatchTotal</NetBatchTotal>
    <CreditPurchaseCount>CreditPurchaseCount</CreditPurchaseCount>
    <CreditPurchaseAmount>CreditPurchaseAmount</CreditPurchaseAmount>
    <CreditReturnCount>CreditReturnCount</CreditReturnCount>
    <CreditReturnAmount>CreditReturnAmount</CreditReturnAmount>
    <DebitPurchaseCount>DebitPurchaseCount</DebitPurchaseCount>
    <DebitPurchaseAmount>DebitPurchaseAmount</DebitPurchaseAmount>
    <DebitReturnCount>DebitReturnCount</DebitReturnCount>
    <DebitReturnAmount>DebitReturnAmount</DebitReturnAmount>
  </BatchSummary>
</RStream>
```

Note: *Retain tags and values returned in the BatchSummary response eclosed between the <BatchSummary> tags for use in the BatchClose command.*

Element	Min	Max	Type	Description
ResponseOrigin	1	10	A	Indicates the source of the response: "Client" = generated by DSIClientX control "Server" = generated by Datacap ePay server "Bridge" = generated by Datacap Bridge server "Processor" = generated by payment processor
DSIXReturnCode	6	6	N	Six digit return code which identifies an error type. See Section 3 on error codes for possible values
CmdStatus	1	10	A	Indicates the outcome of the command: "Success" = command completed successfully "Error" = error processing command. Check DSIXReturnCode and TextResponse for additional info on error
TextResponse	1	40	AN	For Successful responses, this field can contain an optional message from the server or processor. For other response codes, see Section 3 on Error Codes for values
UserTraceData	0	40	AN	Echo of data supplied by the user system in the request; for use by the user system for internal tracking.
MerchantID	1	24	AN	Merchant identification assigned by processor.

TerminalID	0	24	AN	Terminal ID data must be supplied in this tag only if provided by the processor or merchant service provider; otherwise this tag should not be included.
BatchNo	1	6	AN	Batch number returned by processor
BatchItemCount	1	8	N	Number of total items in batch
NetBatchTotal	1	10	N	Net of all transactions in batch
CreditPurchaseCount	1	8	N	Number of credit purchase transactions in batch returned by BatchSummary request. Note: This value should be saved for use by the BatchClose command.
CreditPurchaseAmount	1	10	N	Net of all credit purchase transactions in batch returned by BatchSummary request. Note: This value should be saved for use by the BatchClose command.
CreditReturnCount	1	8	N	Number of credit return transactions in batch returned by BatchSummary request. Note: This value should be saved for use by the BatchClose command.
CreditReturnAmount	1	10	N	Net of all credit return transactions in batch returned by BatchSummary request. Note: This value should be saved for use by the BatchClose command.
DebitPurchaseCount	1	8	N	Number of debit purchase transactions in batch returned by BatchSummary request. Note: This value should be saved for use by the BatchClose command.
DebitPurchaseAmount	1	10	N	Net of all debit purchase transactions in batch returned by BatchSummary request. Note: This value should be saved for use by the BatchClose command.
DebitReturnCount	1	8	N	Number of debit return transactions in batch returned by BatchSummary request. Note: This value should be saved for use by the BatchClose command.
DebitReturnAmount	1	10	N	Net of all debit return transactions in batch returned by BatchSummary request. Note: This value should be saved for use by the BatchClose command.

5.4 BatchClear Response

```
<?xml version="1.0"?>
<RStream>
  <CmdResponse>
    <ResponseOrigin>ResponseOrigin</ResponseOrigin>
    <DSIXReturnCode>DSIXReturnCode</DSIXReturnCode>
    <CmdStatus>CmdStatus</CmdStatus>
    <TextResponse>TextResponse</TextResponse>
    <UserTraceData>UserTraceData</UserTraceData>
  </CmdResponse>
  <BatchClear>
    <MerchantID>MerchantID</MerchantID>
    <OperatorID>OperatorID</OperatorID>
    <TerminalID>TerminalID</TerminalID>
    <BatchNo>BatchNo</BatchNo>
    <BatchItemCount>BatchItemCount</BatchItemCount >
    <NetBatchTotal>NetBatchTotal</NetBatchTotal>
    <ControlNo>ControlNo</ControlNo>
  </BatchClear>
</RStream>
```

Element	Min	Max	Type	Description
ResponseOrigin	1	10	A	Indicates the source of the response: "Client" = generated by DSIClientX control "Server" = generated by Datacap ePay server "Bridge" = generated by Datacap Bridge server "Processor" = generated by payment processor
DSIXReturnCode	6	6	N	Six digit return code which identifies an error type. See Section 3 on error codes for possible values
CmdStatus	1	10	A	Indicates the outcome of the command: "Success" = command completed successfully "Error" = error processing command. Check DSIXReturnCode and TextResponse for additional info on error
TextResponse	1	40	AN	For Successful responses, this field can contain an optional message from the server or processor. For other response codes, see Section 3 on Error Codes for values
UserTraceData	0	40	AN	Echo of data supplied by the user system in the request; for use by the user system for internal tracking.
MerchantID	1	24	AN	Merchant identification assigned by processor.
TerminalID	0	24	AN	Terminal ID data must be supplied in this tag only if provided by the processor or merchant service provider; otherwise this tag should not be included.
BatchNo	1	6	AN	Batch number returned by processor
BatchItemCount	1	8	N	Number of total items in batch
NetBatchTotal	1	10	N	Net of all transactions in batch
ControlNo	1	6	N	Processor supplied number

5.5 BatchClose Response

```
<?xml version="1.0"?>
<RStream>
  <CmdResponse>
    <ResponseOrigin>ResponseOrigin</ResponseOrigin>
    <DSIXReturnCode>DSIXReturnCode</DSIXReturnCode>
    <CmdStatus>CmdStatus</CmdStatus>
    <TextResponse>TextResponse</TextResponse>
    <UserTraceData>UserTraceData</UserTraceData>
  </CmdResponse>
  <BatchClose>
    <MerchantID>MerchantID</MerchantID>
    <OperatorID>OperatorID</OperatorID>
    <TerminalID>TerminalID</TerminalID>
    <BatchNo>BatchNo</BatchNo>
    <BatchItemCount>BatchItemCount</BatchItemCount >
    <NetBatchTotal>NetBatchTotal</NetBatchTotal>
    <CreditPurchaseCount>CreditPurchaseCount</CreditPurchaseCount>
    <CreditPurchaseAmount>CreditPurchaseAmount</CreditPurchaseAmount>
    <CreditReturnCount>CreditReturnCount</CreditReturnCount>
    <CreditReturnAmount>CreditReturnAmount</CreditReturnAmount>
    <DebitPurchaseCount>DebitPurchaseCount</DebitPurchaseCount>
    <DebitPurchaseAmount>DebitPurchaseAmount</DebitPurchaseAmount>
    <DebitReturnCount>DebitReturnCount</DebitReturnCount>
    <DebitReturnAmount>DebitReturnAmount</DebitReturnAmount>
    <ControlNo>ControlNo</ControlNo>
  </BatchClose>
</RStream>
```

Element	Min	Max	Type	Description
ResponseOrigin	1	10	A	Indicates the source of the response: "Server" = generated by DSI server "Processor" = generated by processor "Client" = generated by DSIClientX control
DSIXReturnCode	6	6	N	Six digit return code which identifies an error type. See Section 3 on error codes for possible values
CmdStatus	1	10	A	Indicates the outcome of the command: "Success" = command completed successfully "Error" = error processing command. Check DSIXReturnCode and TextResponse for additional info on error
TextResponse	1	40	AN	For Successful responses, this field can contain an optional message from the server or processor. For other response codes, see Section 3 on Error Codes for values
UserTraceData	0	40	AN	Echo of data supplied by the user system in the request; for use by the user system for internal tracking.
MerchantID	1	24	AN	Merchant identification assigned by processor.
TerminalID	0	24	AN	Terminal ID data must be supplied in this tag only if provided by the processor or merchant service provider; otherwise this tag

				should not be included.
BatchNo	1	6	AN	Batch number returned by processor
BatchItemCount	1	8	N	Number of total items in batch
NetBatchTotal	1	10	N	Net of all transactions in batch
CreditPurchaseCount	1	8	N	Number of credit purchase transactions in batch returned by BatchSummary request
CreditPurchaseAmount	1	10	N	Net of all credit purchase transactions in batch returned by BatchSummary request
CreditReturnCount	1	8	N	Number of credit return transactions in batch returned by BatchSummary request
CreditReturnAmount	1	10	N	Net of all credit return transactions in batch returned by BatchSummary request
DebitPurchaseCount	1	8	N	Number of debit purchase transactions in batch returned by BatchSummary request
DebitPurchaseAmount	1	10	N	Net of all debit purchase transactions in batch returned by BatchSummary request
DebitReturnCount	1	8	N	Number of debit return transactions in batch returned by BatchSummary request
DebitReturnAmount	1	10	N	Net of all debit return transactions in batch returned by BatchSummary request
ControlNo	1	6	N	Processor supplied number

5.6 ServerVersion Response

```
<?xml version="1.0"?>
<RStream>
  <CmdResponse>
    <ResponseOrigin>ResponseOrigin</ResponseOrigin>
    <DSIXReturnCode>DSIXReturnCode</DSIXReturnCode>
    <CmdStatus>CmdStatus</CmdStatus>
    <TextResponse>TextResponse</TextResponse>
    <UserTraceData>UserTraceData</UserTraceData>
  </CmdResponse>
  <ServerVersion>
    <ProductName>ProductName</ProductName>
    <ProductClass>ProductClass</ProductClass>
    <Provider>Provider</Provider>
    <ProductVersion>ProductVersion</ProductVersion>
  </ServerVersion>
</RStream>
```

Element	Min	Max	Type	Description
ResponseOrigin	1	10	A	Indicates the source of the response: "Client" = generated by DSIClientX control "Server" = generated by Datacap ePay server "Bridge" = generated by Datacap Bridge server "Processor" = generated by payment processor
DSIXReturnCode	6	6	N	Six digit return code which identifies an error type. See Section 3 on error codes for possible values
CmdStatus	1	10	A	Indicates the outcome of the command: "Success" = command completed successfully "Error" = error processing command. Check DSIXReturnCode and TextResponse for additional info on error
TextResponse	1	40	AN	For Successful responses, this field can contain an optional message from the server or processor. For other response codes, see Section 3 on Error Codes for values
UserTraceData	0	40	AN	Echo of data supplied by the user system in the request; for use by the user system for internal tracking.
ProductName	1	24	AN	Server product name. (NETePay | DIALePay | GIFTePay)
ProductClass	1	24	AN	Type of processing server. (Terminal Based | Host Based)
Provider	1	24	AN	Processing provider identifier. (VITAL | Concord | CardNet | NOVA | Mercury)
ProductVersion	1	40	AN	Version information for the Datacap server.

5.7 GetSignature Response

```
<?xml version="1.0"?>
<RStream>
  <CmdResponse>
    <ResponseOrigin>ResponseOrigin</ResponseOrigin>
    <DSIXReturnCode>DSIXReturnCode</DSIXReturnCode>
    <CmdStatus>CmdStatus</CmdStatus>
    <TextResponse>TextResponse</TextResponse>
    <UserTraceData>UserTraceData</UserTraceData>
  </CmdResponse>
  <Signature>Signature</Signature>
  <SignMaximumX>SignMaximumX</SignMaximumX>
  <SignMaximumY>SignMaximumY</SignMaximumY>
</RStream>
```

Element	Min	Max	Type	Description
ResponseOrigin	1	10	A	Indicates the source of the response: "Client" = generated by DSIClientX control "Server" = generated by Datacap ePay server "Bridge" = generated by Datacap Bridge server "Processor" = generated by payment processor
DSIXReturnCode	6	6	N	Six digit return code which identifies an error type. See Section 3 on error codes for possible values
CmdStatus	1	10	A	Indicates the outcome of the command: "Success" = command completed successfully "Error" = error processing command. Check DSIXReturnCode and TextResponse for additional info on error
TextResponse	1	40	AN	For Successful responses, this field can contain an optional message from the server or processor. For other response codes, see Section 3 on Error Codes for values
UserTraceData	0	40	AN	Echo of data supplied by the user system in the request; for use by the user system for internal tracking.
Signature	1	2048	AN	Digitized signature data (See note)
SignMaximumX	1	3	N	Maximum X axis value for signature data
SignMaximumY	1	3	N	Maximum Y axis value for signature data

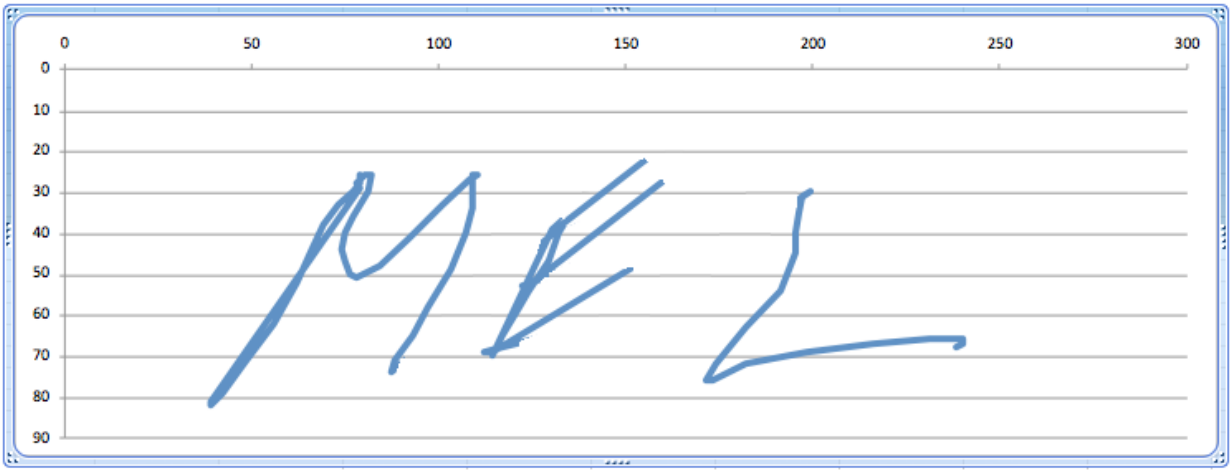
Note:

The Signature response data is a string of up to 2048 characters and is returned as a series of "strokes" and within each stroke is a series of X,Y coordinate points. The first point of a stroke is always a "move to" position. All subsequent points within a stroke are "draw to" positions. The "end stroke point" is simply "#,#". That would be followed by the first point of the next stroke. X and Y coordinates are separated by a comma and end with a colon.

Example <Signature> tag data:

<Signature>79,27:79,27:79,26:79,26:79,26:79,27:79,28:77,30:73,33:69,38:66,44:62,52:56,62:48,72:42,79:39,82:39,81:#,#:79,29:79,29:78,28:80,26:82,26:81,30:77,36:75,40:74,44:75,47:76,50:78,51:84,48:92,41:101,33:108,27:110,26:109,26:109,28:109,30:109,31:109,34:107,40:103,49:97,58:93,65:88,71:87,74:#,#:133,38:133,38:133,38:132,40:129,47:124,54:121,59:117,65:114,70:#,#:130,39:130,39:130,39:135,35:145,29:157,24:166,23:#,#:125,52:125,52:122,53:125,52:136,48:147,47:152,48:#,#:114,69:114,69:112,69:117,68:129,65:139,63:147,62:#,#:199,30:199,30:197,31:196,35:195,40:195,45:191,54:182,63:174,72:171,76:173,76:182,72:198,69:216,67:231,66:239,66:240,67:238,68:#,#:</Signature>

The graphical representation of this sample signature data is approximately as follows:



6.0 XML Document Type Definitions (DTDs)

6.1 TStream

```
<?xml version="1.0"?>

<!DOCTYPE TStream [

<!-- TStream is the document request message template
-->

<!ELEMENT TStream (Transaction | Admin)*>

<!ELEMENT   Transaction  (

            IpAddress?,
            IpPort?,
            MerchantID,
            TerminalID?,
            OperatorID?,
            DateTime?,
            TranType,
            PadType?,
            Format?,
            CardType?,
            TranCode,
            MinLen?,
            MaxLen?,
            Preamble?,
            PrePaidStripeTimeout?,
            SecureDevice?,
            ComPort?,
            Duplicate?,
            InvoiceNo,
            Memo?,
            RefNo,
            PartialAuth?,
            AccountSource?,
            CardHolderID?,
            LastFourCheck?,
            Account,
            Amount,
            Points?,
            AVS?,
            CVVData?,
            PIN?,
            SequenceNo?,
            TerminalName?,
            ShiftID?,
            Signature?,
            TranInfo?,
            DemographicInfo?
```

```
)>


<!ELEMENT   Admin  (

            IpAddress?,
            IpPort?,
            MerchantID,
            TerminalID?,
            OperatorID?,
            TranType?,
            TranCode,
            SecureDevice?,
            ComPort?,
            InvoiceNo?,
            RefNo?,
            Account?,
            Amount?,
            TerminalName?,
            ShiftID?,
            Signature?,
            TranInfo?,
            BatchInfo?
)>

<!ELEMENT   IpAddress               (#PCDATA)>

<!ELEMENT   IpPort                  (#PCDATA)>

<!ELEMENT   MerchantID              (#PCDATA)>

<!ELEMENT   TerminalID              (#PCDATA)>

<!ELEMENT   DateTime                (Date, Time)>
<!ELEMENT   Date                    (#PCDATA)>
<!ELEMENT   Time                    (#PCDATA)>

<!ELEMENT   TranType                (#PCDATA)>

<!ELEMENT   PadType                 (#PCDATA)>

<!ELEMENT   Format                  (#PCDATA)>

<!ELEMENT   CardType                (#PCDATA)>

<!ELEMENT   TranCode                (#PCDATA)>

<!ELEMENT   MinLen                  (#PCDATA)>

<!ELEMENT   MaxLen                  (#PCDATA)>

<!ELEMENT   Preamble                (#PCDATA)>

<!ELEMENT   PrePaidStripeTimeout    (#PCDATA)>

<!ELEMENT   SecureDevice            (#PCDATA)>
```

```
<!ELEMENT   ComPort                 (#PCDATA)>

<!ELEMENT   Duplicate               (#PCDATA)>

<!ELEMENT   InvoiceNo               (#PCDATA)>

<!ELEMENT   OperatorID              (#PCDATA)>

<!ELEMENT   Memo                    (#PCDATA)>

<!ELEMENT   RefNo                   (#PCDATA)>

<!ELEMENT   AccountSource           (#PCDATA)>

<!ELEMENT   CardHolderID            (#PCDATA)>

<!ELEMENT   LastFourCheck           (#PCDATA)>

<!ELEMENT   Account           ((Name?, Track2) | (Name?,AcctNo))>
<!ELEMENT   Name                    (#PCDATA)>
<!ELEMENT   Track2                  (#PCDATA)>
<!ELEMENT   AcctNo                  (#PCDATA)>


<!ELEMENT   Amount                  (
            Purchase,
            CashBack?,
            Authorize?,
            Gratuiity?,
            Tax?,
            FSAPrescription?,
            FSAVision?,
            FSAClinical?,
            FSADental?
)>

<!ELEMENT   Purchase                (#PCDATA)>
<!ELEMENT   CashBack                (#PCDATA)>
<!ELEMENT   Authorize               (#PCDATA)>
<!ELEMENT   Gratuity                (#PCDATA)>
<!ELEMENT   Tax                     (#PCDATA)>

<!ELEMENT   Points                  (#PCDATA)>

<!ELEMENT   Items                   (

            Points,
            Units,
            Price,
            PromoID,
            Desc?,

            Points2?,
            Units2?,
            Price2?,
            PromoID2?,
            Desc2?,
```

```
            Points3?,
            Units3?,
            Price3?,
            PromoID3?,
            Desc3?,

            Points4?,
            Units4?,
            Price4?,
            PromoID4?,
            Desc4?,

            Points5?,
            Units5?,
            Price5?,
            PromoID5?,
            Desc5?,

            Points6?,
            Units6?,
            Price6?,
            PromoID6?,
            Desc6?,

            Points7?,
            Units7?,
            Price7?,
            PromoID7?,
            Desc7?,

            Points8?,
            Units8?,
            Price8?,
            PromoID8?,
            Desc8?
)>


<!ELEMENT   Points          #PCDATA)
<!ELEMENT   Units           #PCDATA)
<!ELEMENT   Price           #PCDATA)
<!ELEMENT   PromoID         #PCDATA)
<!ELEMENT   Desc            #PCDATA)

<!ELEMENT   Points2         #PCDATA)
<!ELEMENT   Units2          #PCDATA)
<!ELEMENT   Price2          #PCDATA)
<!ELEMENT   PromoID2        #PCDATA)
<!ELEMENT   Desc2           #PCDATA)

<!ELEMENT   Points3         #PCDATA)
<!ELEMENT   Units3          #PCDATA)
<!ELEMENT   Price3          #PCDATA)
<!ELEMENT   PromoID3        #PCDATA)
<!ELEMENT   Desc3           #PCDATA)
```

```
<!ELEMENT   Points4             #PCDATA)
<!ELEMENT   Units4              #PCDATA)
<!ELEMENT   Price4              #PCDATA)
<!ELEMENT   PromoID4            #PCDATA)
<!ELEMENT   Desc4               #PCDATA)

<!ELEMENT   Points5             #PCDATA)
<!ELEMENT   Units5              #PCDATA)
<!ELEMENT   Price5              #PCDATA)
<!ELEMENT   PromoID5            #PCDATA)
<!ELEMENT   Desc5               #PCDATA)

<!ELEMENT   Points6             #PCDATA)
<!ELEMENT   Units6              #PCDATA)
<!ELEMENT   Price6              #PCDATA)
<!ELEMENT   PromoID6            #PCDATA)
<!ELEMENT   Desc6               #PCDATA)

<!ELEMENT   Points7             #PCDATA)
<!ELEMENT   Units7              #PCDATA)
<!ELEMENT   Price7              #PCDATA)
<!ELEMENT   PromoID7            #PCDATA)
<!ELEMENT   Desc7               #PCDATA)

<!ELEMENT   Points8             #PCDATA)
<!ELEMENT   Units8              #PCDATA)
<!ELEMENT   Price8              #PCDATA)
<!ELEMENT   PromoID8            #PCDATA)
<!ELEMENT   Desc8               #PCDATA)


<!ELEMENT   AVS                     (
            Address,
            Zip)
>
<!ELEMENT   Address                 (#PCDATA)>
<!ELEMENT   Zip                     (#PCDATA)>

<!ELEMENT   CVVData                 (#PCDATA)>

<!ELEMENT   SequenceNo              (#PCDATA)>

<!ELEMENT   PIN                     (
            PINBlock,
            DervdKey
)>

<!ELEMENT   PINBlock                (#PCDATA)>
<!ELEMENT   DervdKey                (#PCDATA)>

<!ELEMENT   TranInfo                (
            AuthCode?,
            AcqRefData?,
            CVVResult?,
            AVSResult?,
            CustomerCode?
)>
```

```
<!ELEMENT   AuthCode                (#PCDATA)>
<!ELEMENT   Memo                    (#PCDATA)>
<!ELEMENT   AcqRefData              (#PCDATA)>
<!ELEMENT   RecordNo                (#PCDATA)>
<!ELEMENT   CVVResult               (#PCDATA)>
<!ELEMENT   AVSResult               (#PCDATA)>
<!ELEMENT   CustomerCode            (#PCDATA)>

<!ELEMENT   DemographicInfo         (
            CustEmail?,
            CustName?,
            CustPhone?,
            CustAddr1?,
            CustAddr2?,
            CustCity?,
            CustState?,
            CustZip?,
            CustLanguage?
)>

<!ELEMENT   CustEmail               (#PCDATA)>
<!ELEMENT   CustName                (#PCDATA)>
<!ELEMENT   CustPhone               (#PCDATA)>
<!ELEMENT   CustAddr1               (#PCDATA)>
<!ELEMENT   CustAddr2               (#PCDATA)>
<!ELEMENT   CustState               (#PCDATA)>
<!ELEMENT   CustZip                 (#PCDATA)>
<!ELEMENT   CustLanguage            (#PCDATA)>

<!ELEMENT   Purchase                (#PCDATA)>

<!ELEMENT   TerminalName            (#PCDATA)>
<!ELEMENT   ShiftID                 (#PCDATA)>
<!ELEMENT   Signature               (#PCDATA)>

<!ELEMENT   BatchNo                 (#PCDATA)>
<!ELEMENT   BatchItemCount          (#PCDATA)>
<!ELEMENT   NetBatchTotal           (#PCDATA)>
<!ELEMENT   CreditPurchaseCount     (#PCDATA)>
<!ELEMENT   CreditPurchaseAmount    (#PCDATA)>
<!ELEMENT   CreditReturnCount       (#PCDATA)>
<!ELEMENT   CreditReturnAmount      (#PCDATA)>
<!ELEMENT   DebitPurchaseCount      (#PCDATA)>
<!ELEMENT   DebitPurchaseAmount     (#PCDATA)>
<!ELEMENT   DebitReturnCount        (#PCDATA)>
<!ELEMENT   DebitReturnAmount       (#PCDATA)>

]>

<TStream>
</TStream>
```

6.2 RStream

```
<?xml version="1.0"?>

<!-- RStream is the document response message template
-->

<!DOCTYPE   RStream [

<!ELEMENT   RStream (

CmdResponse |
(CmdResponse, TranResponse) |
(CmdResponse, TranResponse, PrintData) |
(CmdResponse, TranResponse, ItemBalances, PrintData) |
(CmdResponse, BatchClose) |
(CmdResponse, BatchSummary) |
(CmdResponse, ItemDetail) |
(CmdResponse, BatchClear) |
(CmdResponse, ServerVersion |
(CmdResponse, GetPrePaidStripe)

)*>



<!-- CmdResponse Element -->

<!ELEMENT   CmdResponse  (

            ResponseOrigin,
            DSIXReturnCode,
            CmdStatus,
            TextResponse,
            IpAddress?,
            UserTraceData?,
            SequenceNo?
)>


<!-- TranResponse Element -->

<!ELEMENT   TranResponse  (

            MerchantID,
            TerminalID?,
            AcctNo,
            ExpDate?,
            CardType,
            Selection?,
            TranCode,
            AuthCode?,
            CaptureStatus?,
            RefNo,
            InvoiceNo?,
```

```
            OperatorID?,
            Memo?,
            Amount,
            AVSResult?,
            CVVResult?,
            AcqRefData?,
            PrePaidExp?,
            BankRespCode?,
            ISORespCode?,
            TranDate?,
            TranTime?,
)>


<!-- ItemBalances Element -->

<!ELEMENT   ItemBalances (

            TotalPoints,
            TotalUnits,
            TotalPrice,
            TotalPoints2?,
            TotalUnits2?,
            TotaiPrice2?,
            TotalPoints3?,
            TotalUnits3?,
            TotaiPrice3?,
            TotalPoints4?,
            TotalUnits4?,
            TotaiPrice4?,
            TotalPoints5?,
            TotalUnits5?,
            TotaiPrice5?,
            TotalPoints6?,
            TotalUnits6?,
            TotaiPrice6?,
            TotalPoints7?,
            TotalUnits7?,
            TotalPrice7?,
            TotalPoints8?,
            TotalUnits8?,
            TotaiPrice8?

)>


<!-- PrintData Element -->

<!ELEMENT   PrintData  (

            Line1,
            Line2,
            Line3,
            Line4,
            Line5,
            Line6,
            Line7,
```

```
            Line8,
            Line9,
            Line10,
            Line11,
            Line12,
            Line13
)>


<!-- BatchClose Element -->

<!ELEMENT   BatchClose  (

            MerchantID,
            TerminalID?,
            BatchNo,
            BatchNumber?,
            BatchItemCount,
            NetBatchTotal,
            CreditPurchaseCount?,
            CreditPurchaseAmount?,
            CreditReturnCount?,
            CreditReturnAmount?,
            DebitPurchaseCount?,
            DebitPurchaseAmount?,
            DebitReturnCount?,
            DebitReturnAmount?,
            ControlNo?
)>

<!-- BatchSummary Element -->

<!ELEMENT   BatchSummary  (

            MerchantID,
            TerminalID?,
            BatchNo,
            BatchNumber?,
            BatchItemCount,
            NetBatchTotal,
            CreditPurchaseCount?,
            CreditPurchaseAmount?,
            CreditReturnCount?,
            CreditReturnAmount?,
            DebitPurchaseCount?,
            DebitPurchaseAmount?,
            DebitReturnCount?,
            DebitReturnAmount?
)>


<!-- BatchClear Element -->

<!ELEMENT   BatchClear  (

            MerchantID,
            TerminalID?,
```

```
            BatchNo,
            BatchNumber?,
            BatchItemCount,
            ControlNo?
)>


<!-- ServerVersion Element -->

<!ELEMENT   ServerVersion     (

            ProductName,
            ProductClass,
            Provider,
            ProductVersion
)>


<!-- ItemDetail Element -->

<!ELEMENT   ItemDetail  (

            MerchantID,
            TerminalID?,
            BatchNo,
            TranDate?,
            TranTime?,
            AcctNo,
            ExpDate?,
            ItemAmount1,
            ItemAmount2?,
            AuthCode?,
            RefNo?
)>


<!-- GetPrePaidStripe Element -->

<!ELEMENT   GetPrePaidStripe  (

            PrePaidTrack2?,
            PrePaidAcctNo?
)>


<!ELEMENT   Amount            (
            Purchase,
            Purchase1?,
            Purchase2?,
            Purchase3?,
            Purchase4?,
            Purchase5?,
            Purchase6?,
            Purchase7?,
            Purchase8?,
            CashBack?,
            Authorize?,
```

```
            Gratuity?,
            Tax?,
            FSAPrescription?,
            FSAVision?,
            FSAClinical?,
            FSADental?
)>

<!ELEMENT   Value             (
            Points1,
            Points2?,
            Points3?,
            Points4?,
            Points5?,
            Points6?,
            Points7?,
            Points8?,
)>

<!ELEMENT      ResponseOrigin          (#PCDATA)>
<!ELEMENT      DSIXReturnCode          (#PCDATA)>
<!ELEMENT      TextResponse            (#PCDATA)>
<!ELEMENT      IpAddress               (#PCDATA)>
<!ELEMENT      MerchantID              (#PCDATA)>
<!ELEMENT      TerminalID              (#PCDATA)>
<!ELEMENT      SequenceNo              (#PCDATA)>
<!ELEMENT      AcctNo                  (#PCDATA)>
<!ELEMENT      ExpDate                 (#PCDATA)>
<!ELEMENT      CardType                (#PCDATA)>
<!ELEMENT      Selection               (#PCDATA)>
<!ELEMENT      TranCode                (#PCDATA)>
<!ELEMENT      AuthCode                (#PCDATA)>
<!ELEMENT      CaptureStatus           (#PCDATA)>
<!ELEMENT      RefNo                   (#PCDATA)>
<!ELEMENT      InvoiceNo               (#PCDATA)>
<!ELEMENT      OperatorID              (#PCDATA)>
<!ELEMENT      Memo                    (#PCDATA)>
<!ELEMENT      Purchase                (#PCDATA)>
<!ELEMENT      CashBack                (#PCDATA)>
<!ELEMENT      Authorize               (#PCDATA)>
<!ELEMENT      Gratuity                (#PCDATA)>
<!ELEMENT      Balance                 (#PCDATA)>
<!ELEMENT      UserTraceData           (#PCDATA)>
<!ELEMENT      AVSResult               (#PCDATA)>
<!ELEMENT      CVVResult               (#PCDATA)>
<!ELEMENT      AcqRefData              (#PCDATA)>
<!ELEMENT      RecordNo                (#PCDATA)>
<!ELEMENT      PrePaidExp              (#PCDATA)>
<!ELEMENT      BankRespCode            (#PCDATA)>
<!ELEMENT      ISORespCode             (#PCDATA)>
<!ELEMENT      ControlNo               (#PCDATA)>
<!ELEMENT      TranDate                (#PCDATA)>
<!ELEMENT      TranTime                (#PCDATA)>
<!ELEMENT      ItemAmount1             (#PCDATA)>
<!ELEMENT      ItemAmount2             (#PCDATA)>
<!ELEMENT      BatchItemCount          (#PCDATA)>
<!ELEMENT      NetBatchTotal           (#PCDATA)>
```

```
<!ELEMENT      CreditPurchaseCount     (#PCDATA)>
<!ELEMENT      CreditPurchaseAmount    (#PCDATA)>
<!ELEMENT      CreditReturnCount       (#PCDATA)>
<!ELEMENT      CreditReturnAmount      (#PCDATA)>
<!ELEMENT      DebitPurchaseCount      (#PCDATA)>
<!ELEMENT      DebitPurchaseAmount     (#PCDATA)>
<!ELEMENT      DebitReturnCount        (#PCDATA)>
<!ELEMENT      DebitReturnAmount       (#PCDATA)>
<!ELEMENT      BatchNo                 (#PCDATA)>
<!ELEMENT      ProductName             (#PCDATA)>
<!ELEMENT      ProductClass            (#PCDATA)>
<!ELEMENT      Provider                (#PCDATA)>
<!ELEMENT      ProductVersion          (#PCDATA)>

]>

<RStream>
</RStream>
```

7.0 dsiPDCX Compatible Servers

7.1 NETePay Compatible Servers

The following chart indicates which transactions are supported by the available NETePay servers.

(Continued for additional processors on next page.)

CREDIT Transactions	ADS Host	MPS Host	Nova Host Host	Nova Term Term	Concord EFS Host	Fifth Third Host	VITAL TSYS Term	VITAL TSYS Host	CardNet FDMS Term	Sterling Host	RBS Lynk Term	Heart land Term
Sale	X	X	X	X	X	X	X	X	X	X	X	X
Return	X	X	X	X	X	X	X	X	X	X	X	X
Void Sale	X	X	X	X	X	X	X	X	X	X	X	X
Void Sale by Record				X		X	X		X		X	X
Void Return	X	X	X	X	X	X	X	X	X	X	X	X
Void Return by Record				X		X	X		X		X	X
AuthOnly	X	X	X	X	X	X	X	X	X	X	X	X
PreAuth		X		X	X	X	X		X	X	X	X
VoiceAuth	X	X	X	X	X	X	X	X	X	X	X	X
PreAuth Capture		X		X	X	X	X		X	X	X	X
Adjust		X		X		X	X	X	X		X	X
Adjust by Record				X		X	X		X		X	X
Balance		X	X						X			
Zero Auth (Verify Card)									X		X	

DEBIT Transactions												
Sale	X	X	X		X	X	X	X	X	X	X	X
Return	X	X	X		X	X	X	X		X	X	X
Void Sale												
Void Return												

EBT Transactions												
Foodstamp Sale		X	X		X	X	X	X		X	X	
Foodstamp Return		X	X		X	X	X	X		X	X	
Cash Sale		X	X		X	X	X	X		X	X	
Cash Return												
Cash Balance		X	X		X	X		X		X		
Foodstamp - Voucher – Manual Entry		X	X		X	X	X	X		X	X	
Foodstamp – Voucher Return												
Foodstamp – Balance		X	X		X	X		X		X		

FSA Transactions												
FSA Credit Sale		X	X				X		X	X		X
FSA Credit Sale Reversal		X	X				X		X	X		X

ADMIN Requests												
Batch Summary Request	X*	X*	X*	X*	X*	X*	X*	X*	X*	X*	X*	X*
Batch Clear Request		X*					X*				X*	X*
Batch Close Request	X*	X*	X*	X*	X*	X*	X*	X*	X*	X*	X*	X*
Batch Change Number							X*				X*	X*
Server Version	X*	X*	X*	X*	X*	X*	X*	X*	X*	X*	X*	X*

NETePay Compatible Servers (continued)

CREDIT Transactions	GPS Term US	GPS Host US	Paymentech Host	Paymentech Term	FDMS Nashville	FDMS Buypass Host
Sale	X	X	X	X	X	X
Return	X	X	X	X	X	X
Void Sale	X	X	X	X	X	X
Void Sale by Record	X			X	X	
Void Return	X	X	X	X	X	X
Void Return br Record	X			X	X	
AuthOnly	X	X	X	X	X	X
PreAuth	X	X	X	X	X	X
VoiceAuth	X	X	X	X	X	X
PreAuth Capture	X	X	X	X	X	X
Adjust	X	X		X	X	
Adjust by Record	X			X	X	
Balance			X		X	X
Zero Auth (Verify Card)			X		X	X

DEBIT Transactions						
Sale	X	X	X		X	X
Return	X	X	X			X
Void Sale						
Void Return						

EBT Transactions						
Foodstamp Sale	X	X	X			X
Foodstamp Return	X	X	X			X
Cash Sale	X	X	X			X
Cash Return						
Cash Balance	X	X	X			X
Foodstamp - Voucher – Manual Entry	X	X	X			X
Foodstamp – Voucher Return						
Foodstamp – Balance	X	X	X			X

FSA Transactions						
FSA Credit Sale		X			X	X
FSA Credit Sale Reversal		X			X	X

ADMIN Requests						
Batch Summary Request	X*	X*	X*	X*	X*	X*
Item Detail Request						
Batch Clear Request				X*		
Batch Close Request	X*	X*	X*	X*	X*	X*
Batch Change Number				X*		
Server Version	X*	X*	X*	X*	X*	X*

* - These commands are supported only from the ePay Administrator Application supplied by Datacap. These commands are not available through the XML interface of DSIClientX.

7.2 GIFTePay Compatible Servers

The following chart indicates which transactions are supported by the available GIFTePay servers.

PREPAID Transactions	Chockstone GIFT	Concord EFS GIFT	EPI GIFT	Fifth Third GIFT	Givex GIFT	Nova GIFT	Paymentech GIFT	MPS GIFT	Secure Pay GIFT	Sterling GIFT	SVS GCS GIFT	Value Link GIFT	Valutec GIFT
Issue	x	x	x	x	x	x	x	x	x	x	x	x	x
Void Issue	x	x	x	x	x	x	x	x	x	x	x	x	x
Sale	x	x	x	x	x	x	x	x	x	x	x	x	x
Return	x	x	x	x	x	x	x	x	x	x	x	x	x
Void Sale	x	x	x	x	x	x	x	x	x	x	x	x	x
Void Return	x	x	x	x	x	x	x	x	x	x	x	x	x
NoNSFSale	x	x	x	x	x	x	x	x	x	x	x	x	x
Reload	x	x	x	x	x	x	x	x	x	x	x	x	x
Void Reload	x	x	x	x	x	x	x	x	x	x	x	x	x
CashOut	x	x		x					x		x	x	x
Balance	x	x	x	x	x	x	x	x	x	x	x	x	x

Loyalty Transactions													
Issue					x			x					
Add					x			x					
Subtract					x			x					
VoidIssue					x			x					
VoidAdd					x			x					
VoidSubtract					x			x					
Balance					x			x					

GIFTePay Compatible Servers (continued)

PREPAID Transactions	NPC GIFT	Profit Point GIFT	Smart Trans Sys GIFT	EVO GIFT	Uni-Perk GIFT	Card Dog GIFT	Spark Base GIFT	RBS World Pay GIFT	World Gift Card GIFT	Archer Heart - land GIFT
Issue	x	x	x	x	x	x	x	x	x	x
Void Issue	x	x	x	x	x	x	x	x	x	x
Sale	x	x	x	x	x	x	x	x	x	x
Return	x	x	x	x	x	x	x	x	x	x
Void Sale	x	x	x	x	x	x	x	x	x	x
Void Return	x	x	x	x	x	x	x	x	x	x
NoNSFSale	x	x	x	x	x	x	x	x	x	x
Reload	x	x	x	x	x	x	x	x	x	x
Void Reload	x	x	x	x	x	x	x	x	x	x
CashOut	x	x	x	x	x	x	x	x	x	
Balance	x	x	x	x	x	x	x	x	x	x
Add Alias *										x
Remove Alias *										x
New Alias Acct *										x
Transfer *										x

(*) – These transactions implemented as <Admin> commands.

7.3 DIALePay Server by DataTran Application

The following chart indicates which transactions are supported by the DIALePay server with various DataTran applications.

	NDC	NDH	MAP	NOV	ENV	CIT	FDR	NAB	VS2	PNS	PNH	MPT	MPH	SGH	LNK	TVA	IPS
CREDIT Transactions																	
Sale	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Return	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Void Sale	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Void Return	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
AuthOnly	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
PreAuth	X			X	X	X		X	X	X		X			X	X	
VoiceAuth	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
PreAuth Capture	X			X	X	X		X	X	X		X			X	X	
Adjust	X			X	X	X		X	X	X		X			X	X	
DEBIT Transactions																	
Sale	X	X		X		X			X		X		X	X	X		
Return	X	X		X					X		X		X	X	X		
Void Sale																	
Void Return																	
EBT Transactions																	
Foodstamp Sale		X							X				X				
Foodstamp Return		X							X				X				
Cash Sale		X							X				X				
Cash Return																	
Cash Balance																	
Foodstamp - Voucher – Manual Entry		X							X				X				
Foodstamp – Voucher Return																	
Foodstamp – Balance																	
ADMIN Requests																	
Batch Summary Request	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Item Detail Request																	
Batch Clear Request	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Batch Close Request	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Batch Change Number									1	1					1	1	1
Server Version	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

1 - These commands are supported only from the ePay Administrator Application supplied by Datacap. These commands are not available through the XML interface of DSIClientX.

	Givex GIFT	Paymentech GIFT	Valutec GIFT	SVS GIFT	Merccury GIFT
PREPAID (Gift) Transactions					
Issue	X	X	X	X	X
Void Issue	X	X	X	X	X
Sale	X	X	X	X	X
Return	X	X	X	X	X
Void Sale	X	X	X	X	X
Void Return	X	X	X	X	X
NoNSFSale	X	X		X	X
Reload	X	X	X	X	X
Void Reload	X	X	X	X	X
Balance	X	X	X	X	X