



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

# ***dsiPDCX***

## ***Programming Interface Specification***

***V00.01***

***Preliminary***

### **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 Tran™ and ePay™ server products.  
Any other use is strictly prohibited.

Copyright © 2010 – Datacap Systems Inc.

## Copyright Notice

Copyright © 2010 – Datacap Systems Inc.  
100 New Britain Blvd.  
Chalfont, Pennsylvania 18914 USA  
All rights reserved.

## 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 products. Any other use is strictly prohibited.**

The DSIClientX™, DSIEMVClientX™, dsiPDCX™, DataTran™, NETePay™, DIALePay™, GIFTePay™, ePay™, ePay Administrator™, DataTran™, DialTran™, IPTran™, TwinTran™ and Datacap names and logos and all related trademarks, tradenames, and other intellectual property are the property of Datacap Systems Inc. and cannot be used without its express prior written permission.

## ***Revision History***

00.01      05 Dec 2010      • Preliminary Draft Release.

## **Table Of Contents**

<b>1.0 Introduction</b>	<b>5</b>
1.1 <i>System Diagram and Card Data Flow Using dsiPDCX</i>	6
<b>2.0 Integration of dsiPDCX</b>	<b>7</b>
2.1 Initialization	7
2.2 Processing Transaction Requests	8
2.3 Canadian Debit Requests	8
<b>3.0 XML Transaction Requests</b>	<b>9</b>
4.0 <i>XML Document Type Definitions (DTDs)</i>	12
4.01 <i>TStream</i>	12
4.02 <i>RStream</i>	18

## 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 utilizes a Datacap PDC (Peripheral Device Controller) attached to a PC serial port. The PDC incorporates a magnetic stripe reader (MSR) and expansion port(s) for other payment devices such as a PIN pad. dsiPDCX manages all interaction with the PDC to get card stripe or PIN input so the POS application is relieved from handling any sensitive cardholder data. The mode of operation can facilitate an 'out of scope' position from a PCI-DSS (Payment Card Industry Data Security Council) PA-DSS (Payment Application – Data Security Standard) perspective for a POS application. To be completely out of scope, a POS application must not handle cardholder data in any manner [that \(change from than\)](#) might subject the application to PA-DSS compliance.

The dsiPDCX software is designed to communicate exclusively with Datacap's NETePay server products using Internet Protocol (IP). Messages exchanged between the dsiPDCX and a NETePay server are encrypted for secure transmission over open networks (such as the Internet). This secure communications architecture provides the flexibility to configure systems using LAN and/or WAN networks.

Datacap servers are available for in-store or enterprise configurations and are designed to communicate with specific payment systems providers. Servers are available with a range of communications to the payment systems processor, including Internet, Frame Relay, Wireless, and Dial. A single dsiPDCX integration allows a systems vendor to offer a variety of payment processing access methods that most 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 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 provides consolidated transaction data storage, logging and data management functions. All Datacap NETePay 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.

*This dsiPDCX interface document must be used in conjunction with Datacap's DSIClientX Programming Interface Specification.*

DSIClientX is another version of the ActiveX control for NETePay access that allows the POS application to manage cardholder data input directly and supplying that data to the ActiveX control to be sent securely to NETePay for processing. dsiPDCX manages all cardholder data input so that it is never exposed to the POS application. The basic transaction definitions and capabilities used by dsiPDCX are presented in the document DSIClientX Programming Interface Specification V4.24 (or later) and a POS developer should employ that document for basic development planning. dsiPDCX performs all the transaction functions that are available in the standard DSIClientX but has extensions and process flow differences to manage cardholder input independently of the POS application.

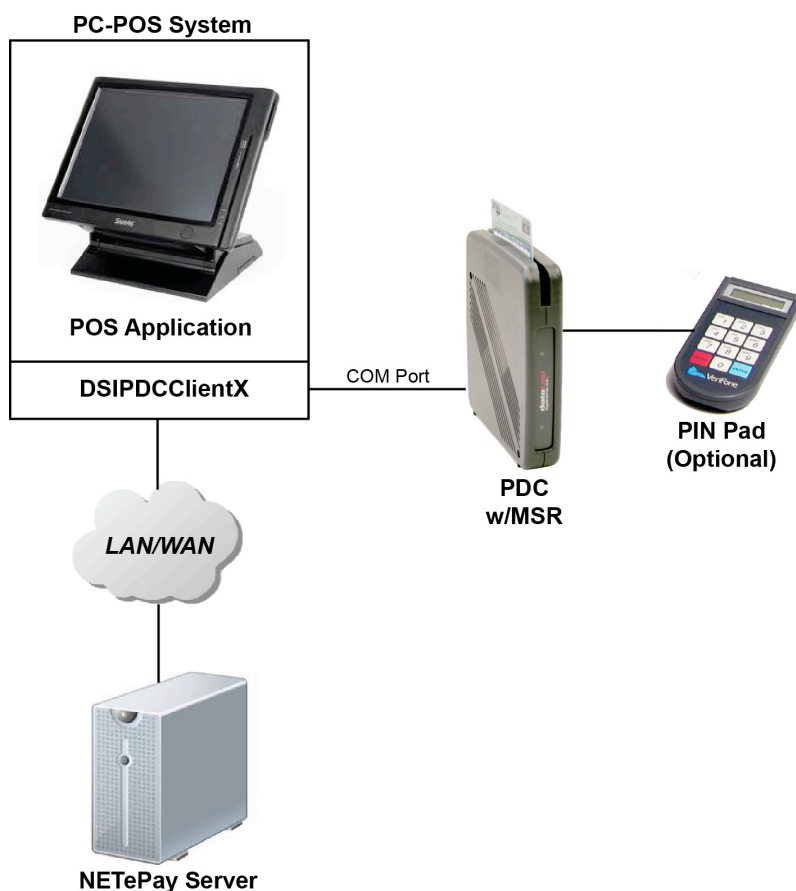
## 1.1 System Diagram and Card Data Flow Using dsiPDCX

The dsiPDCX component is typically installed on the 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 activate the MSR on the PDC and wait for stripe input with no further intervention from the POS application. If a stripe is not successfully read within 20 (remove 15) seconds, dsiPDCX returns a timeout error code to the POS application which should issue the payment request again.

If the stripe on the card is damaged or otherwise unreadable by the MSR on the PDC, the POS application can request that dsiPDCX get the account input manually through the PC keyboard. When so instructed, dsiPDCX will present a dialog box to the POS operator with fields to enter the card account number, expiration date, CVV/CVD, address, and zip code.

After card information is obtained, dsiPDCX encrypts and transmits the entire request to NETePay using TCP/IP. NETePay will then communicate with the processing host to (remove to) for authorization. NETePay will return the results of the authorization attempt to dsiPDCX which will then pass the results to the POS application.

For transactions that require the entry of a PIN (Personal Identification Number), such as debit transactions, dsiPDCX will activate an optionally attached PIN pad on the PDC after magnetic stripe entry has completed. dsiPDCX maintains timeout control for each input operation and will return a timeout error to the POS application if input has not been successfully accomplished within the timeout period (replace with the timeout); the POS application should decide whether to try again or use manual input instead. Change control name to dsiPDCX in diagram below



## 2.0 Integration of dsiPDCX

dsiPDCX is integrated with a POS application in the same manner as the standard DSIClientX. The following explains (add the s) any different integration considerations for dsiPDCX. Refer to the “DSIClientX Programming Interface Specification” document for the integration guidelines.

### 2.1 Initialization

In addition to the initialization commands required for DSIClientX, dsiPDCX requires an additional Admin command to initialize the attached PDC. 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>
    <MerchantID>MerchantID</MerchantID>
    <TerminalID>TerminalID</TerminalID>
    <OperatorID>OperatorID</OperatorID>
    <TranCode>SecureDeviceInit</TranCode>
    <SecureDevice>SecureDevice</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.
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	<p><b>For general use</b> 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.</p> <p><b>For reversal support with selected processors</b> Use a unique POS workstation number (exactly 3 digits 000-999) for multi-workstation locations. See 1.3.7.</p>
OperatorID	O	1	10	N	Operator (clerk, server, etc.) associated with the inquiry.
TranCode	Y	1	40	A	“SecureDeviceInit”

SecureDevice	Y	1	40	AN	Secure device type; use value "PDC"
ComPort	Y	1	3	N	COM (Serial) port number to which a PDC is attached (1-255).
SequenceNo	O	1	20	AN	<b>For reversal support with selected processors</b> Use value '00000000'
TerminalName	O	1	20		Terminal name
ShiftID	O	1	20		Shift identification
Signature	O	1	2048		Digitized signature

## **2.2 Processing Transaction Requests**

dsiPDCX supports all the methods and events made available by the standard DSIClientX. dsiPDCX however only supports the following values for the ProcessControl argument:

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

The value of 2 (to process asynchronously without visible dialog boxes and fire an event) is not supported in dsiPDCX.

## **2.3 Canadian Debit Requests**

dsiPDCX does not support the Canadian Debit transactions available in the standard DSIClientX.

### 3.0 XML Transaction Requests

---

The XML transaction definitions supported in the standard DSIClientX [are \(remove and\)](#) also available in dsiPDCX. In order to support the input of cardholder information via the PDC, dsiPDCX requires two additional tags and a different use of an existing tag.

The new tags to be added to the standard DSIClientX XML definitions are:

**<SecureDevice>** Indicates which type of secure input device will process cardholder data  
**<ComPort>** Indicates to which PC COM port the Secure Device (i.e. PDC) is attached

In order to support both automatic (via PDC) or manual (via dialog box and keyboard) of cardholder data, the **<Account:AcctNo>** tag will only accept two arguments:

**SecureDevice** Indicates that the secure input device (PDC) should be used to process cardholder data.

**Prompt (remove Manual)** Indicates that a dialog box should be presented to the POS operator for keyboard entry of cardholder data as follows:

Account Number	Required
Expiration Date	Required
CVV/CVD	Optional
<b>Address</b>	<b>Optional</b>
Zip Code (AVS)	Optional

For SecureDevice entry, the POS application can supply optional values for **<AVS:Address>**, **<AVS:Zip>** and **<CVVData>**; for Manual entry, values supplied in those tags by the POS application will be overridden by the manually input values, if any.

Both SecureDevice and Manual entry routines will expire if no useable input is processed with a timeout period. When SecureDevice input times out, the POS application should consider manual input; since manual input results in a higher processing charge to the merchant, it should be used only when automatic input is not possible.

The following is the XML template for a Credit Sale with the dsiPDCX tag extensions and changes. The tag extensions and changes should be incorporated into every XML request when using dsiPDCX.

#### XML Template: **Credit Sale**

```
<?xml version="1.0"?>
<TStream>
  <Transaction>
    <IpAddress>999.999.999.999</IpAddress>
    <IpPort>99999</IpPort>
    <MerchantID>MerchantID</MerchantID>
    <TerminalID>TerminalID</TerminalID>
    <OperatorID>OperatorID</OperatorID>
    <TranType>Credit</TranType>
    <Duplicate>Override</Duplicate>
    <CardType>CardType</CardType>
    <TranCode>Sale</TranCode>
    <SecureDevice>SecureDevice</SecureDevice>
    <ComPort>ComPort</ComPort>
    <InvoiceNo>InvoiceNo</InvoiceNo>
    <RefNo>RefNo</RefNo>
    <PartialAuth>PartialAuth</PartialAuth>
```

```

<Account>
  <AcctNo>SecureDevice</AcctNo> REMOVE ...<Track2>Track2</Track2>
</Account>
<Amount>
  <Purchase>Purchase</Purchase>
  <Tax>Tax</Tax>
</Amount>
<SequenceNo>SequenceNo</SequenceNo>
<AVS>
  <Address>Address</Address>
  <Zip>Zip</Zip>
</AVS>
<CVVData>CVVData</CVVData>
<SequenceNo>SequenceNo</SequenceNo>
<TerminalName>TerminalName</TerminalName>
<ShiftID>ShiftID</ShiftID>
<Signature>Signature</Signature>
<TranInfo>
  <CustomerCode>CustomerCode</CustomerCode>
</TranInfo>
</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	<p><b>For general use</b> 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.</p> <p><b>For reversal support with selected processors</b> Use a unique POS workstation number (exactly 3 digits 000-999) for multi-workstation locations. See 1.3.7.</p>
OperatorID	Y	1	10	N	Operator (clerk, server, etc.) associated with the Transaction.
TranType	Y	1	20	A	"Credit"
Duplicate	O	1	20	AN	"Override" - To force approval of a transaction that was previously rejected as a duplicate by the processor.
CardType	O	3	20	AN	"VISA", "M/C", "AMEX", "DCLB", "DCVR", "JCB", "OTHER"
TranCode	Y	1	40	A	"Sale"
SecureDevice	Y	1	40	AN	Secure device type; use value "PDC"
ComPort	Y	1	3	N	COM (Serial) port number to which a PDC is attached (1-255).
InvoiceNo	Y	1	16	AN	Invoice number – sequential receipt number, check number, or other unique transaction identifier.
RefNo	Y	1	16	AN	Use the same data as InvoiceNo

PartialAuth	O	1	16	AN	If value equals "Allow", then transaction may be authorized for an amount different than supplied in <b>&lt;Amount&gt;:&lt;Purchase&gt;</b> tag. See the <b>&lt;Authorize&gt;</b> tag in the response to determine how much was actually authorized. Note: The ability to support Partial Authorization will be mandated by the processors in the near future.
Account:AcctNo	Y	1	19	AN	To use PDC for cardholder data, use value 'SecureDevice' To use the PC keyboard for manual input of cardholder data, use 'Prompt' ( <a href="#">remove Manual</a> )
Amount:Purchase	Y	1	8	N	Purchase price (with 2 place decimal – eg. 29.95)
Amount:Tax	O	1	8	N	<b>For Purchase Card Level II transactions</b> – Sales Tax amount (with 2 place decimal – eg. 29.95)
SequenceNo	O	1	20	AN	<b>For reversal support with selected processors</b> Use '00000000' for first transaction in new installations. For all subsequent transactions, use the SequenceNo value returned in the previous transaction response. See 1.3.7.
AVS:Address	O	1	8	N	<b>For manually entered (hand-keyed) account numbers</b> Address (building street) number. Supply this data whenever using a manually entered account number to ensure best transaction qualification.
AVS:Zip	O	1	9	N	<b>For manually entered (hand-keyed) account numbers</b> Zip code. Supply this data whenever using a manually entered account number to ensure best transaction qualification.
CVVData	O	3	4	N	<b>For manually entered (hand-keyed) account numbers</b> Three or four CVV card digits or "None" or "Illegible". Supply this data whenever using a manually entered account number to ensure best transaction qualification.
TerminalName	O	1	20	AN	Terminal name
ShiftID	O	1	20	AN	Shift identification
Signature	O	1	2048	AN	Digitized signature
TranInfo:CustomerCode	O	1	17	AN	<b>For Purchase Card Level II transactions</b> – customer's identifying information; such as PO number.

## 4.0 XML Document Type Definitions (DTDs)

### 4.01 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,
    SecureDevice?,
    ComPort?,
    Duplicate?,
    InvoiceNo,
    Memo?,
    RefNo,
    PartialAuth?,
    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 SecureDevice (#PCDATA)>
<!ELEMENT ComPort (#PCDATA)>
<!ELEMENT Duplicate (#PCDATA)>
<!ELEMENT InvoiceNo (#PCDATA)>
<!ELEMENT OperatorID (#PCDATA)>
<!ELEMENT Memo (#PCDATA)>
<!ELEMENT RefNo (#PCDATA)>
<!ELEMENT Account ((Name?, Track2) | (Name?, AcctNo, ExpDate))>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Track2 (#PCDATA)>
<!ELEMENT AcctNo (#PCDATA)>
<!ELEMENT ExpDate (#PCDATA)>
<!ELEMENT Amount (
Purchase,

```

```

CashBack?,
Authorize?,
Gratuuity?,
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>
```

## 4.02 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 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?,
    TotalPrice2?,
    TotalPoints3?,
    TotalUnits3?,
    TotalPrice3?,
    TotalPoints4?,
    TotalUnits4?,
    TotalPrice4?,
    TotalPoints5?,
    TotalUnits5?,
    TotalPrice5?,
    TotalPoints6?,
    TotalUnits6?,
    TotalPrice6?,
    TotalPoints7?,
    TotalUnits7?,
    TotalPrice7?,
    TotalPoints8?,
    TotalUnits8?,
    TotalPrice8?
)>
```

```
<!-- 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?
)>
```

```
<!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>

```