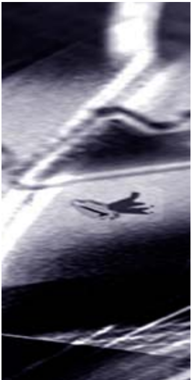
VeriSign Payment Services

XMLPay 1.1 Core Specification

USER GUIDE



Customer Support: 1-888-883-9770

vps-support@verisign.com

VeriSign, Inc. 00000016/Rev. 2

VeriSign Payment Services XMLPay 1.1 Core Specification

VeriSign, Inc. 00000016/Rev. 2

Printed in the United States of America.

Publication date: January 2003

This document supports **VeriSign Payment Services XMLPay Version 1.1** and all subsequent releases unless otherwise indicated in a new edition or release notes.

Trademark Notices

Note This document may describe features and/or functionality that are not present in your software or your service agreement. Contact your account representative to learn more about what is available with this VeriSign product.

Contents

Summary of Revisions

VeriSign Payment Services XMLPay 1.1 Core Specification

VeriSign, Inc. 00000016/Rev. 2

The following changes were made to this document since revision 1:

Appendix A, “XMLPay Schemas”	The **RepeatSale** element is defined, but not supported. The following corrections were made to **RepeatSale** in the XMLPay schema: The **RepeatSale** element declaration has been added to the **RepeatSale** element definition (immediately after the **ForceCapture** element definition). **RepeatSale** has been added as a possible sub-element in the **Transaction** element.
Appendix C, “VeriSign Transaction Results”	This appendix now includes error codes for communication errors.

VeriSign, Inc. 00000016

The following changes were made to this document since revision 0:

Reference	Description
XMLPay Examples	Transaction examples have been updated to reflect the new specification. See Chapter 5, “XMLPay Examples.”
Schema updated	The XMLPay schema has been updated. See Appendix A, “XMLPay Schemas.”
Items element	The **ExtAmt** and **ShippingAmt** attributes have been removed and the **TotalAmt** attribute has been changed. See “Items” on page 26.
RepeatSale element	The **RepeatSale** element is not supported.

CHAPTER 1

Introduction

This document defines an XML syntax for payment transaction requests, responses, and receipts in a payment processing network.

Intended Audience

The typical user of XMLPay is an Internet merchant or merchant aggregator who wants to dispatch credit card, corporate purchase card, Automated Clearing House (ACH), or other payment requests to a financial processing network.

About this Document

This document is organized as follows:

- Chapter 2, “XMLPay Overview” describes XML and XMLPay, presenting processing models, networking, messaging and related specifications.
- Chapter 3, “XMLPay Syntax” presents the syntax for transaction requests, responses, and receipts using a simplified notation.
- Chapter 4, “XMLPay Transaction Profiles” lists the transactions supported for each tender type—ACH, Card, Check—along with the data elements used for each of those transactions.
- Chapter 5, “XMLPay Examples” gives several XMLPay document samples.
- Appendix A, “XMLPay Schemas” provides standard W3C schemas for XMLPay and XMLPay Types.
- Appendix B, “XMLPay DTD” presents the Document Type Definition XMLPay schema.

- Appendix C, “VeriSign Transaction Results” lists VeriSign transaction result codes and response messages as well as Address Verification Service (AVS) result codes.

Related Document

VeriSign Payflow Pro Developer’s Guide VeriSign Payflow Pro is a high performance TCP/IP-based Internet payment gateway solution. Payflow Pro is pre-integrated with leading e-commerce solutions and is also available as a downloadable software development kit (SDK).

CHAPTER 2

XMLPay Overview

About XML

XML (e**X**tensible **M**arkup **L**anguage) is derived from Standardized General Markup Language (SGML) and HyperText Markup Language (HTML). In a sense, XML is SGML "lite", but XML manages to maintain SGML's strength as well as HTML's simplicity. What's more, XML can be converted to HTML.

The main advantage of XML is that text can be meaningfully annotated. In XML, markers identify and tag the text. But the markers themselves have no defined meaning; it is the applications that define the markers.

XML allows complex transactions to be structured. Client integration is simplified through the exchange of XML documents. Since XML provides support for digital signatures, documents from unknown sources can be trusted. In addition, XML can easily produce large documents such as transaction logs and reports.

Benefits of XML

The main benefits of XML are that it:

- Allows text annotation
- Presents text, data, and content to applications as a structured document
- Facilitates integration of diverse applications

In addition to these benefits, XML is easy to:

- Read (all text)
- Parse and validate
- Search for content
- Produce

Well-formed XML Document

A well-formed XML document conforms to XML syntax and must have:

- An XML processing instruction at the beginning (prolog)
- A single root element
- Matching (case sensitive) start and end tags for all elements
- All XML elements properly nested
- Attribute values in quotes

Example 1

```
<?xml version="1.0”?>
<Card>
   <CardType> MasterCard </CardType>
   <CardNumber>495576040004</CardNumber>
   <ExpDate>200011</ExpDate>
   CVNum>828</CVNum>
</Card>
```

Example 2

```
<?xmlversion="1.0”?>
<Items ID="IDTI123”>
   <Item Number="1”>
      <Description>Electric Toothbrush</Description>
      <Quantity>1</Quantity>
      <UnitPrice>100</UnitPrice>
      <TotalAmt>100</TotalAmt>
   </Item>
   <Item Number="2”>
      <Description>ToothPaste</Description>
      <Quantity>20</Quantity>
      <UnitPrice>2</UnitPrice>
      <TotalAmt>40</TotalAmt>
   </Item>
   …
</Items>
```

Using XMLPay

XMLPay defines an XML syntax for payment transaction requests, responses, and receipts in a payment processing network.

The typical user of XMLPay is an Internet merchant or merchant aggregator who wants to dispatch credit card, corporate purchase card, Automated Clearing House (ACH), Internet check, or other payment requests to a financial processing network.

Using the data type definitions specified by XMLPay, a user creates a client payment request and dispatches it—using a mechanism left unspecified by XMLPay—to an associated XMLPay-compliant server component. Responses (also formatted in XML) convey the results of the payment requests to the client.

Note For specific examples of how to submit XML documents using the Payflow Pro client API, see the Payflow Pro SDK download package.

XMLPay Instruments

XMLPay supports payment processing using the following payment instruments:

- Retail credit and debit cards
- Corporate purchase cards: Levels 1, 2, and 3
- Internet checks
- ACH

XMLPay Operations

Typical XMLPay operations include:

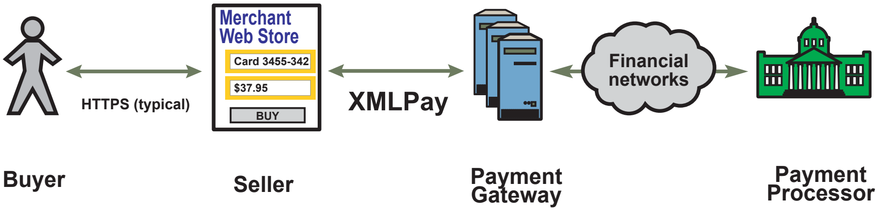
- Funds authorization and capture
- Sales and repeat sales
- Voiding of transactions

XMLPay Processing Models

XMLPay is intended for use in both Business-to-Consumer (B2C) and Business-to-Business (B2B) payment processing applications.

Business-to-Consumer

In a B2C Sale transaction, the Buyer presents a payment instrument (for example, a credit card number) to a Seller to transfer money from the Buyer to the Seller.

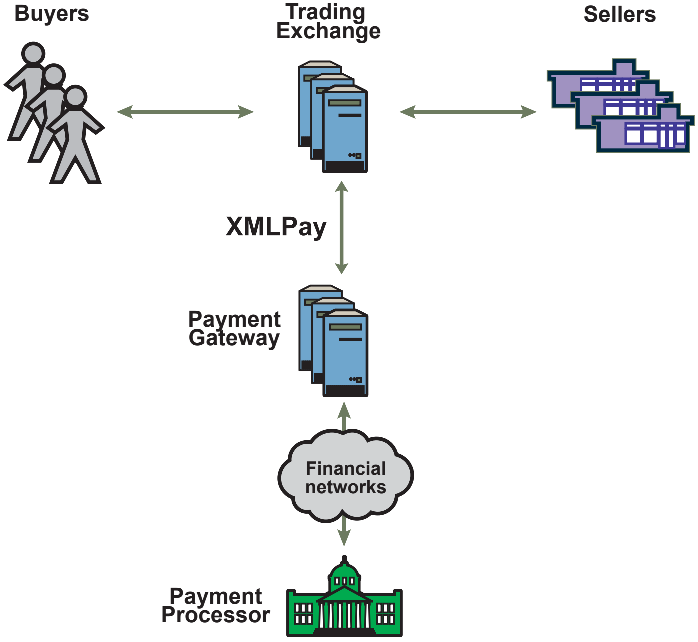


The Seller uses XMLPay to forward the Buyer's payment information to a Payment Processor. The Seller formats an **`XMLPayRequest`** and submits it either directly to an XMLPay-compliant payment processor or, as pictured, indirectly via an XMLPay-compliant Payment Gateway. Responses have the type **`XMLPayResponse`**.

The Buyer-to-Seller and Payment Gateway-to-Processor channels are typically left unaffected by use of XMLPay. For example, XMLPay is typically not used in direct communications between the buyer and the seller. Instead, conventional HTML form submission or other Internet communication methods are typically used. Similarly, because Payment Processors often differ considerably in the formats they specify for payment requests, XMLPay server logic is usually localized at the Payment Gateway, leaving the legacy connections between gateways and processors unchanged.

Business-to-Business

When used in support of B2B transactions, the Seller does not typically initiate XMLPay requests. Instead, an aggregator or trading exchange uses XMLPay to communicate business-focused purchasing information (such as level 3 corporate purchase card data) to a payment gateway.



In this way, the Trading Exchange links payment execution to other XML-based communications between Buyers and Sellers such as Advance Shipping Notice delivery, Purchase Order communication, or other B2B communication functions.

XMLPay Networking

XMLPay uses a flexible approach to the networking aspects of payment communication. Network security and efficient communications are the primary goals.

Data Transport

The mechanics of passing XMLPay requests to server processing components is outside the scope of the core XMLPay specification.

The typical application scenario involves passing payment requests synchronously to a server over a persistent TCP/IP connection that is not torn down until a response is received or a timeout occurs. Alternatively, asynchronous processing paradigms (such as store and forward) are also possible.

Security and Authentication

Payment transactions typically include highly sensitive data such as credit card numbers, their associated expiration dates, payment amounts, and line-item invoice details. Such information must be protected when transmitted over insecure networks. It is recommended that XMLPay messages be encrypted using the strongest encryption available when passing over insecure networks, for example, using either SSL or S/MIME as appropriate for the chosen processing model.

Because payment transactions also typically modify or reveal sensitive data such as account balances, it is essential that the identity of the submitting party be authenticated. XMLPay supports the User ID & password authentication mechanism.

Use of neither authentication method is mandated. Authentication is not required at the XMLPay level because it is often more conveniently implemented at the transport level. For example, HTTP user ID and passwords, SSL client authentication, or S/MIME signed messages can provide suitable authentication mechanisms for XMLPay data.

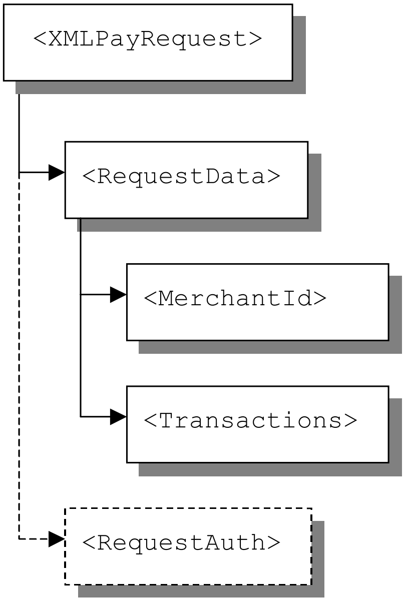
Finally, some environments require authentication of buyers in e-commerce transactions. XMLPay provides a mechanism for including buyer authentication information, in a variety of formats, in payment transactions. For more information on how this mechanism can be employed, see VeriSign's *Authenticated Payment*™ white paper.

XMLPay Messaging

The highest-level XMLPay structures represent payment transaction requests, responses, and receipts.

XMLPayRequest

Payment transactions are submitted, one or more at a time, as **`XMLPayRequest`** documents. The high-level structure of a request looks like this:



Merchant ID identifies the *merchant of record* for the transaction within the target payment processing network. The merchant of record may be different from the submitting party in a delegated processing model.

Transactions is the list of payment transactions to be processed. XMLPay supports up to 32 transactions per XMLPay document submission.

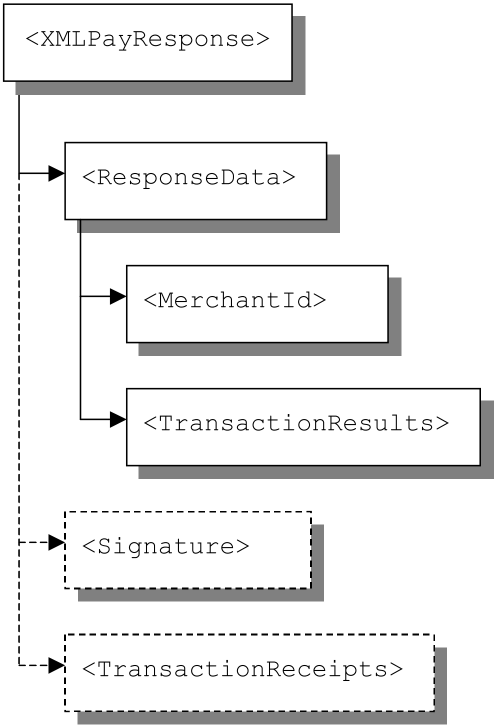
RequestAuth is an optional structure used to authenticate the submitting party, in the absence of transport level authentication.

See Chapter 3, “XMLPay Syntax” for a detailed description of request documents.

XMLPayResponse

Each XMLPayRequest submission produces a corresponding XMLPayResponse document containing results for each submitted transaction request. The high-level structure of a response looks like this:

Note **`Signature`** and **`TransactionReceipts`** are not supported on the Payment server.



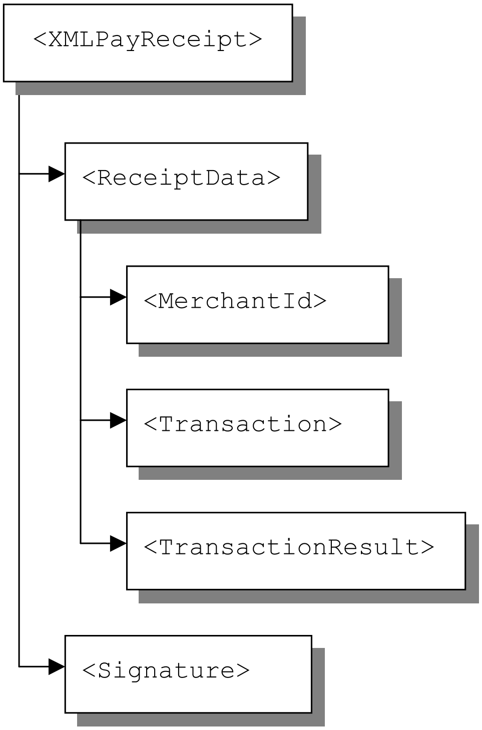
See Chapter 3, “XMLPay Syntax” for a detailed description of response documents.

XMLPayReceipt

Receipts provide verifiable records of transactions. Using XMLPayReceipt documents, a payment processing network may optionally provide receipts for each XMLPay transaction processed.

Note XMLPayReceipt is not supported on the Payment server.

The high-level structure of a receipt looks like this:



See Chapter 3, “XMLPay Syntax” for a detailed description of receipt documents.

CHAPTER 3

XMLPay Syntax

This chapter presents the syntax for transaction requests, responses, and receipts using a simplified notation.

- Appendix A, “XMLPay Schemas,” provides the complete syntax, expressed in W3C XML-schema notation.
- Appendix B, “XMLPay DTD,” provides a document type definition (DTD) representation of the schema.

Syntax Notation

The following example presents the notation used to describe XMLPay document:

```
<Example>
   (element)
   (optionalElement)?
   (alternateElement1|alternateElement2)
   (element)+
   (element)*
</Example>
```

element	Indicates the occurrence of a (possibly complex) XML element (for example, <element>...</element>) defined elsewhere.
?	Indicates an optional element.
|	Separates alternative elements, any one of which is allowed.
+	Indicates that one or more occurrences of an element are allowed.
*****	Indicates that zero or more occurrences of an element are allowed.

Note The Payflow Pro SDK download package provides specific examples of XML documents using the Payflow Pro client API.

The XMLPayRequest Document

```
<XMLPayRequest Timeout=?>
   <RequestData>
      (Vendor)
      (Partner)
      <Transactions>
            (Transaction)+
      </Transactions>
   </RequestData>
   (RequestAuth)?
</XMLPayRequest>
```

Attribute	Description
Vendor	Identifies the merchant of record for the transaction within the target payment processing network. In a delegated processing model, the merchant of record may be different from the submitting party.
Partner	Identifies the submitting party.
Transaction	Defined on page 15. XMLPay supports up to 32 transactions per XMLPay document submission.
RequestAuth	Defined on page 20.
Timeout	Optional attribute that puts an upper bound on the time that the server will spend processing the request. Multiple transactions submitted within a single request execute independently of one another. Results are returned for any transactions in a request that complete before a timeout occurs.

Transaction

XMLPay supports up to 32 transactions per XMLPay document submission.

```
<Transaction Id=? CustRef=?>
   (Authorization|Capture|Sale|Credit|Void|ForceCapture|
   RepeatSale|GetStatus)
</Transaction>
```

Attribute	Description
Id	Optional attribute that tracks the transaction through the payment-processing network. The submitting merchant generates this transaction identifier, which should be unique among all transactions submitted by that merchant. **Id** need not be globally unique across merchants, since the payment-processing network interprets it within the context of the merchant associated with the transaction. If an **Id** attribute is provided in a transaction, it will be included in the matching **TransactionResult** in the resultant **XMLPayResult**. Similarly, **CustRef** is a merchant-generated ID identifying a specific customer of this merchant and associating it with this transaction.

Authorization Transaction

An authorization transaction verifies the availability of funds and reserves them for later capture.

```
<Authorization>
   <PayData>
      (Invoice)
      (Tender)
   </PayData>
   (PayDataAuth)?
   (ExtData)*
</Authorization>
```

Attribute	Description
PayData	Specifies the details of the purchase, within Invoice, as well as the payment **Tender** to use.
PayDataAuth	Optional element that coveys the payer's authorization, for use in Authenticated Payment.
ExtData	Optional element that may carry extended data (outside the syntax of the XMLPay schema).

Capture Transaction

A capture transaction transfers the funds secured by a previous authorization transaction, identified by `PNRef`, into the merchant’s account.

```
<Capture>
   (PNRef)
   (Invoice)?
   (ExtData)*
</Capture>
```

Attribute	Description
Invoice	An updated Invoice may optionally be provided, specifying any changes in the purchase details from the original invoice in the reference authorization.
ExtData	Optional element that may carry extended data (outside the syntax of the XMLPay schema).

Sale Transaction

A sale transaction verifies the availability of funds and captures funds in one step.

```
<Sale>
   <PayData>
      (Invoice)
      (Tender)
   </PayData>
   (PayDataAuth)?
   (ExtData)*
</Sale>
```

Attribute	Description
PayData	Specifies the details of the purchase, within **Invoice**, as well as the payment **Tender** to use.
PayDataAuth	Optional element that coveys the payer's authorization, for use in Authenticated Payment.
ExtData	Optional element that may carry extended data (outside the syntax of the XMLPay schema).

Credit Transaction

A credit transaction reverses a previous sale or capture transaction.

```
<Credit>
   (PNRef|Tender)
   (Invoice)?
   (ExtData)*
</Credit>
```

Attribute	Description
PNRef	The transaction to be credited is identified by **PNRef**. A credit may be run without a **PNRef** by providing the **Tender** for the account to be credited and Invoice for the amount.
Invoice	In the case of a partial credit, you must provide **Invoice** and provide details on the items being returned.
ExtData	Optional element that may carry extended data (outside the syntax of the XMLPay schema).

Void Transaction

A void transaction cancels a pending sale, capture, or credit.

```
<Void>
   (PNRef)
   (ExtData)*
</Void>
```

Attribute	Description
PNRef	The transaction to be cancelled is identified by **PNRef**. If the referenced transaction has already been processed, the void fails.

ForceCapture Transaction

A ForceCapture transaction captures funds reserved through an out-of-band authorization (for example, a voice authorization received over the phone).

```
<ForceCapture>
   <PayData>
      (Invoice)
      (Tender)
   </PayData>
   (PayDataAuth)?
   (AuthCode)
   (ExtData)*
</ForceCapture>
```

Attribute	Description
AuthCode	Authorization code received out-of-band.
PayData	Specifies the details of the purchase, within Invoice, as well as the payment **Tender** to use.
PayDataAuth	Optional element that coveys the payer's authorization, for use in Authenticated Payment.
ExtData	Optional element that may carry extended data (outside the syntax of the XMLPay schema).

RepeatSale Transaction

Note The RepeatSale transaction is not supported on the Payments server.

```
<RepeatSale>
   (PNRef)
   (Invoice)?
   (ExtData)*
</RepeatSale>
```

The RepeatSale transaction repeats a previous sale transaction.

The previous sale transaction is identified by `PNRef`. Any information provided in `Invoice` overrides the corresponding information in the referenced transaction. In particular, this transaction is allowed to increase or decrease the amount of the original sale.

`ExtData` is an optional element that may carry extended data (outside the syntax of the XMLPay schema).

GetStatus Transaction

A GetStatus transaction queries the status of a previous transaction.

```
<GetStatus>
   (PNRef)
   (ExtData)*
</GetStatus>
```

Attribute	Description
PNRef	The transaction to query is identified by **PNRef**.
ExtData	Optional element that may carry extended data (outside the syntax of the XMLPay schema).

RequestAuth

The RequestAuth element provides authentication of the requestor through either a username and password, using UserPass, or a digital signature, using Signature.

```
<RequestAuth>
   (UserPass|Signature)
</RequestAuth>
```

In the case of a digital signature, the W3C XML Signature syntax is used and the signature is executed over the `RequestData`.

UserPass

```
<UserPass>
   (User)
   (UserDomain)?
   (Password)
</UserPass>
```

Attribute	Description
User	String identifier assigned to a user.
UserDomain	Names a partner or a vendor under whose auspice a transaction is being submitted.
Password	User's password (string).

The XMLPayResponse Document

```
<XMLPayResponse>
   <ResponseData>
      (Vendor)
      (Partner)
      <TransactionResults>
            (TransactionResult)+
      </TransactionResults>
   </ResponseData>
   (Signature)?
   (TransactionReceipts)?
</XMLPayResponse>
```

Attribute	Description
Vendor	Identifies the merchant of record for the transaction within the payment processing network. **Partner** identifies the partner who submitted the transaction on behalf of the vendor.
TransactionResult	Defined on page 22.
Signature	Optional signature over **ResponseData**, executed by the payment processing gateway using the W3C XML Signature syntax. This signature may be used to provide integrity protection of the response data and/or authentication of the responder, if needed (transport-level security may also be used to provide these protections).
TransactionReceipts	Optional list of receipts from the payment processing network.

TransactionResult

```
<TransactionResult Id=?>
   (Result)
   (AVSResult)?
   (CVResult)?
   (Message)?
   (PNRef)?
   (AuthCode)?
   (HostCode)?
   (HostURL)?
   (OrigResult)?
   (Status)?
   (ReceiptURL)?
   (ExtData)*
</TransactionResult>
```

Attribute	Description
Result	Number that indicates outcome of the transaction (see Appendix C, “VeriSign Transaction Results”).
AVSResult	Results of the AVS check, if appropriate.
CVResult	Results of the CV check, if appropriate. Possible values: match, no match, service not available, or service not requested.
Message	Descriptive message describing **Result**.
PNRef	Identifier assigned to the transaction by the payment processing network.
AuthCode	Authorization code for the transaction provided by the bank, if any.
HostCode	Result code returned by the payment processor, if any. Whereas **Result** provides a normalized view the transaction status, **HostCode** passes through the back-end processor status unmodified.
HostURL	URL returned by the payment processor, if any, to use in referring to the transaction.
ReceiptURL	URL returned by the payment processor, if any, to use in referring to a receipt for the transaction.
OrigResult	Original result for a transaction queried with **GetStatus**.
Status	Current status for a transaction queried with **GetStatus**.
ExtData	Optional element that may carry extended data (outside the syntax of the XMLPay schema).
Id	Identifier assigned to the transaction by the merchant (if one was provided in the transaction request).

AVSResult

```
<AVSResult>
    (StreetMatch)
    (ZipMatch)
</AVSResult>
```

Attribute	Description
StreetMatch	Indicates whether or not the billing street address matched the bank's records. Possible values: match, no match, service not available, or service not requested.
ZipMatch	Indicates whether or not the billing zip matched the bank's records. Possible values: match, no match, service not available, or service not requested.

Core Structures

PayData

```
<PayData>
    (Invoice)
    (Tender)
</PayData>
```

Attribute	Description
Invoice	Describes the details of a purchase. Defined on page 24.
Tender	Describes the payment instrument. Defined on page 27.

PayDataAuth

The `PayDataAuth` element provides authentication of the payer for an associated PayData, using either a PKCS-7 format or a W3C XML Signature format digital signature.

```
<PayDataAuth>
    (PKCS7Signature|Signature)
</PayDataAuth>
```

Invoice

```
<Invoice>
   (InvNum)?
   (Date)?
   <BillFrom>
      (Name)?
      (Address)?
      (EMail)?
      (Phone)?
      (Fax)?
      (URL)?
   </BillFrom>
   <BillTo>
      (Name)?
      (Address)?
      (EMail)?
      (Phone)?
      (Fax)?
      (CustCode)?
      (PONum)?
      (TaxExempt)?
   </BillTo>
   <ShipFrom>
      (Name)?
      (Address)?
      (EMail)?
      (Phone)?
      (Fax)?
   </ShipFrom>
   <ShipTo>
      (Name)?
      (Address)?
      (EMail)?
      (Phone)?
      (Fax)?
   </ShipTo>
   (Description)*
   (Items)?
   (DiscountAmt)?
   (ShippingAmt)?
   (DutyAmt)?
   (TaxAmt)?
   (NationalTaxIncl)?
   (TotalAmt)?
   (Comment)?
   (ExtData)*
</Invoice>
```

Attribute	Description
InvNum	Invoice number.
Date	Invoice date: YYYYMMDD, ISO 8601.
BillFrom Name, Address, EMail, Phone, Fax, and URL	information about the biller.
BillTo Name, Address, EMail, Phone, and Fax	information about the buyer.
BillTo CustCode	Code, chosen by the merchant, that may be included in the invoice to identify the customer (buyer).
BillTo PONum	Buyer's purchase order number.
BillTo TaxExempt	Indicates that the buyer is a tax exempt entity.
ShipFrom and ShipTo	Information about the shipping addresses, if different from **BillFrom** and **BillTo** respectively.
Description	summary description of the purchase. This field, in the case of an Amex purchase card, can contain up to four separate descriptions of 40 characters each.
Items	Full line-item breakdown of the purchase. Defined on page 26.
DiscountAmt	Discount to be applied to the item subtotal.
ShippingAmt	Total of all shipping and handling charges.
DutyAmt	Total duty amount of this invoice, if applicable.
TaxAmt	Total of all taxes.
NationalTaxIncl	Boolean which when true, indicates that the national tax in included in the **TaxAmt**.
TotalAmt	Grand total (**item subtotal - DiscountAmt + ShippingAmt + TaxAmt**).
Comment	Free-form comment about the purchase.
ExtData	Optional element that may carry extended data (outside the syntax of the XMLPay schema).

Items

Items is a list of line item detail records. `Item` is defined below.

```
<Items>
    (Item)+
</Item>
```

Item

```
<Item Number=>
    (SKU)?
    (UPC)?
    (Description)?
    (Quantity)?
    (UnitOfMeasurement)?
    (UnitPrice)?
    (DiscountAmt)?
    (TaxAmt)?
    (TotalAmt)?
    (ExtData)*
</Item>
```

Attribute	Description
Number	Line number for the item in the invoice.
SKU	Merchant's product code for the item (stock keeping unit).
UPC	Item's universal product code.
Description	Item's description.
Quantity	Number of units of this item. **UnitOfMeasurement** provides the units for Quantity (ISO 31).
UnitPrice	Cost of each unit.
DiscountAmt	Discount to be applied to this line item.
TaxAmt	Total of all taxes for this line item.
TotalAmt	Total amount including tax and discount for this line item: **TaxAmt - DiscountAmt**.
ExtData	Optional element that may carry extended data (outside the syntax of the XMLPay schema).

Address

```
<Address>
    (Street)?
    (City)?
    (State)?
    (Zip)?
    (Country)?
</Address>
```

Attribute	Description
Street	Street address, including number.
City	City name.
State	State or province. For US addresses, two character state codes should be used.
Zip	Postal code.
Country	Country code (ISO 3166). Default is **US**.

Tender

```
<Tender>
    (ACH|Card|Check)
</Tender>
```

ACH

ACH (Automated Clearing House) tender detail.

```
<ACH>
    (AcctType)
    (AcctNum)
    (ABA)
    (Prenote)?
    (ExtData)*
</ACH>
```

Attribute	Description
AcctType	Type of the bank account: checking or savings.
AcctNum	Account number.

Attribute	Description
ABA	Bank routing number.
Prenote	Boolean. If **Y**, then the purpose of this transaction is not to move money, but to establish authorization for future transactions to be submitted on a recurring basis.
ExtData	Optional element that may carry extended data (outside the syntax of the XMLPay schema).

Card

Retail Credit/Debit and Corporate Purchase Card tender detail.

```
<Card>
   (CardType)
   (CardNum)
   (ExpDate)
   (CVNum)?
   (MagData)?
   (NameOnCard)?
   (ExtData)*
</Card>
```

Attribute	Description
CardType	Indicates the type of the card: Amex, DinersClub, Discover, JCB, MasterCard, or Visa.
CardNum	Account number.
ExpDate	Card expiration date: YYYYMM or ISO 8601.
CVNum	Card verification number (typically printed on the back of the card, but not embossed on the front).
MagData	Data located on the magnetic strip of a credit card.
NameOnCard	Card holder's name as printed on the card.
ExtData	Optional element that may carry extended data (outside the syntax of the XMLPay schema).

Check

Check tender detail.

```
<Check>
    (CheckType)
    (CheckNum)
    (MICR)
    (DL)?
    (SS)?
    (DOB)?
    (ExtData)*
</Check>
```

Attribute	Description
CheckType	Indicates the type of the check: corporate, personal, or government.
CheckNum	Account holder's next unused check number.
MICR	Magnetic Ink Check Reader (MICR) is the entire line of numbers at the bottom of the check. It includes the transit number, account number, and check number.
DL	Account holder's driver's license number. **XxNnnnnnnn** format, where **Xx** is the state code; **Nnnnnnnn** is the number
SS	Account holder's social security number.
DOB	Account holder's date of birth YYYYMMDD, ISO 8601
ExtData	Optional element that may carry extended data (outside the syntax of the XMLPay schema).

ExtData

```
<ExtData Name= Value= />
```

Attribute	Description
Name	Name of the extended data element.
Value	Value of the extended data element.

CHAPTER 4

XMLPay Transaction Profiles

The following tables document the transactions supported for each tender, along with the data elements used for each of those transactions.

ACH Transactions

ACH supports the following transactions: Sale, Credit, and Void.

The following data are used to process ACH transactions (required elements for sale are in **bold text**):

- Invoice.TotalAmt
- ACH.AcctType
- ACH.AcctNum
- ACH.ABA
- ACH.Prenote
- BillTo.Name

Card Transactions

Retail credit and debit cards and corporate purchase cards support the following transactions: Sale, Authorization, Delay Capture, Credit, and Void.

The following data are used to process card transactions (required elements for sale/authorization are in **bold text**):

- Invoice.TotalAmt
- Card.CardNum

- Card.ExpDate
- Card.NameOnCard (Note: Defaults to BillTo.Name)
- BillTo.Street, ZIP (Note: Required for AVS)
- CardInfo.CVNum

Purchase Card Level I

Level I purchase cards use the following additional data:

- BillTo.PONum
- Invoice.TaxAmt

Purchase Card Level II

Level II purchase cards use the following additional data:

- BillTo.TaxExempt
- ShipTo.ZIP

Purchase Card Level III

Level III purchase cards use the following additional data:

- Invoice.Date
- BillTo.CustCode
- ShipTo.Country
- ShipFrom.ZIP
- Invoice.ShippingAmt
- Invoice.DiscountAmt
- Item.SKU, UPC, Description, Quantity, UnitOfMeasurement, UnitPrice, ExtAmt, DiscountAmt, TaxAmt, TotalAmt, DutyAmt, NationalTaxIncl

Note Minimum requirements for describing an item are Description, Quantity, and UnitPrice.
ExtAmt = Quantity * UnitPrice.
TotalAmt = TaxAmt – DiscountAmt

Check Transactions

Checks support the following transaction: Sale. The following data are used to process check transactions (required elements are in **bold text**):

- Invoice.TotalAmt
- Check.CheckType
- Check.CheckNum
- Check.MICR
- BillTo.Name
- BillTo.Address
- BillTo.EMail
- BillTo.Phone
- **BillTo.DL** (DL is optional if SS is provided)
- **BillTo.SS** (SS is optional if DL is provided)
- BillTo.DOB
- Invoice.InvNum

CHAPTER 5

XMLPay Examples

This chapter provides several examples of XMLPay documents.

For specific examples of how to submit XML documents using the Payflow Pro client API, see the Payflow Pro SDK download package itself.

Note To receive namespaces in the XML response document, you must set the XMLPayRequest `version` attribute to "1.1". Leaving this attribute out or setting it to any value less than 1.1 leaves the namespace out of the XML response document.

In this Chapter

An example response follows each example request.

ACH Sale Request

```
<?xml version="1.0" encoding="UTF-8"?>

<XMLPayRequest Timeout="40" version="1.1"
xmlns="http://www.verisign.com/XMLPay"
>  <RequestData>
    <Vendor>vendor</Vendor>
    <Partner>VeriSign</Partner>
    <Transactions>
      <Transaction>
        <Sale>
          <PayData>
            <Invoice>
              <InvNum>1</InvNum>
              <Date>20020208</Date>
              <BillTo>
                <Name>billtoname</Name>
                <TaxExempt>Y</TaxExempt>
              </BillTo>
              <NationalTaxIncl>Y</NationalTaxIncl>
              <TotalAmt>24.97</TotalAmt>
            </Invoice>
            <Tender>
              <ACH>
                <AcctType>C</AcctType>
                <AcctNum>6355059797</AcctNum>
                <ABA>091000019</ABA>
                <Prenote>N</Prenote>
              </ACH>
            </Tender>
          </PayData>
        </Sale>
      </Transaction>
    </Transactions>
  </RequestData>
  <RequestAuth>
    <UserPass>
      <User>user</User>
      <Password>password</Password>
    </UserPass>
  </RequestAuth>
</XMLPayRequest>
```

ACH Sale Response

```
<?xml version="1.0" encoding="UTF-8"?>

<XMLPayResponse xmlns="http://www.verisign.com/XMLPay">
  <ResponseData>
    <Vendor>vendor</Vendor>
    <Partner>verisign</Partner>
    <TransactionResults>
      <TransactionResult>
        <Result>0</Result>
        <AVSResult>
          <StreetMatch>Service Not Requested</StreetMatch>
          <ZipMatch>Service Not Requested</ZipMatch>
        </AVSResult>
        <CVResult>Service Not Requested</CVResult>
        <Message>Approved</Message>
        <PNRef>VTHD55395864</PNRef>
        <OrigResult>0</OrigResult>
      </TransactionResult>
    </TransactionResults>
  </ResponseData>
</XMLPayResponse>
```

Card Authorization Request

```
<?xml version="1.0" encoding="UTF-8"?>

<XMLPayRequest Timeout="40"
xmlns="http://www.verisign.com/XMLPay">
  <RequestData>
    <Vendor>vendor</Vendor>
    <Partner>VeriSign</Partner>
    <Transactions>
      <Transaction>
        <Authorization>
          <PayData>
            <Invoice>
              <NationalTaxIncl>N</NationalTaxIncl>
              <TotalAmt>24.97</TotalAmt>
            </Invoice>
            <Tender>
              <Card>
                <CardType>visa</CardType>
                <CardNum>5105105105105100</CardNum>
                <ExpDate>200911</ExpDate>
                <NameOnCard/>
              </Card>
            </Tender>
          </PayData>
        </Authorization>
      </Transaction>
    </Transactions>
  </RequestData>
  <RequestAuth>
    <UserPass>
      <User>user</User>
      <Password>password</Password>
    </UserPass>
  </RequestAuth>
</XMLPayRequest>
```

Card Authorization and Response

```
<?xml version="1.0" encoding="UTF-8"?>

<XMLPayResponse>
  <ResponseData>
    <Vendor>vendor</Vendor>
    <Partner>verisign</Partner>
    <TransactionResults>
      <TransactionResult>
        <Result>0</Result>
        <AVSResult>
          <StreetMatch>Service Not Available</StreetMatch>
          <ZipMatch>Service Not Available</ZipMatch>
        </AVSResult>
        <CVResult>Service Not Requested</CVResult>
        <Message>Approved</Message>
        <PNRef>V63A09910356</PNRef>
        <AuthCode>747PNI</AuthCode>
        <HostCode>00</HostCode>
        <OrigResult>0</OrigResult>
      </TransactionResult>
    </TransactionResults>
  </ResponseData>
</XMLPayResponse>
```

Card Capture Request

```
<XMLPayRequest Timeout="40"
xmlns="http://www.verisign.com/XMLPay">
  <RequestData>
    <Vendor>vendor</Vendor>
    <Partner>VeriSign</Partner>
    <Transactions>
      <Transaction>
        <Capture>
          <PNRef>V63A09910356</PNRef>
        </Capture>
      </Transaction>
    </Transactions>
  </RequestData>
  <RequestAuth>
    <UserPass>
      <User>user</User>
      <Password>password</Password>
    </UserPass>
  </RequestAuth>
</XMLPayRequest>
```

Card Capture Response

```
<?xml version="1.0" encoding="UTF-8"?>

<XMLPayResponse>
  <ResponseData>
    <Vendor>vendor</Vendor>
    <Partner>verisign</Partner>
    <TransactionResults>
      <TransactionResult>
        <Result>0</Result>
        <AVSResult>
          <StreetMatch>Service Not Available</StreetMatch>
          <ZipMatch>Service Not Available</ZipMatch>
        </AVSResult>
        <CVResult>Service Not Requested</CVResult>
        <Message>Approved</Message>
        <PNRef>V53A09206640</PNRef>
        <AuthCode>747PNI</AuthCode>
        <HostCode>00</HostCode>
        <OrigResult>0</OrigResult>
      </TransactionResult>
    </TransactionResults>
  </ResponseData>
</XMLPayResponse>
```

Card Credit Request

```
<?xml version="1.0" encoding="UTF-8"?>

<XMLPayRequest Timeout="40"
xmlns="http://www.verisign.com/XMLPay">
  <RequestData>
    <Vendor>vendor</Vendor>
    <Partner>VeriSign</Partner>
    <Transactions>
      <Transaction>
        <Credit>
          <PNRef>V53A09206652</PNRef>
        </Credit>
      </Transaction>
    </Transactions>
  </RequestData>
  <RequestAuth>
    <UserPass>
      <User>user</User>
      <Password>password</Password>
    </UserPass>
  </RequestAuth>
</XMLPayRequest>
```

Card Credit Response

```
<?xml version="1.0" encoding="UTF-8"?>

<XMLPayResponse>
  <ResponseData>
    <Vendor>vendor</Vendor>
    <Partner>verisign</Partner>
    <TransactionResults>
      <TransactionResult>
        <Result>105</Result>
        <AVSResult>
          <StreetMatch>Service Not Requested</StreetMatch>
          <ZipMatch>Service Not Requested</ZipMatch>
        </AVSResult>
        <CVResult>Service Not Requested</CVResult>
        <Message>Credit error</Message>
        <PNRef>V54A09206685</PNRef>
        <OrigResult>0</OrigResult>
      </TransactionResult>
    </TransactionResults>
  </ResponseData>
</XMLPayResponse>
```

Card Force Capture Request

```
<?xml version="1.0" encoding="UTF-8"?>

<XMLPayRequest Timeout="40"
xmlns="http://www.verisign.com/XMLPay">
  <RequestData>
    <Vendor>vendor</Vendor>
    <Partner>VeriSign</Partner>
    <Transactions>
      <Transaction>
        <ForceCapture>
          <PayData>
            <Invoice>
              <NationalTaxIncl>N</NationalTaxIncl>
              <TotalAmt>24.97</TotalAmt>
            </Invoice>
            <Tender>
              <Card>
                <CardType>visa</CardType>
                <CardNum>5105105105105100</CardNum>
                <ExpDate>200911</ExpDate>
                <NameOnCard/>
              </Card>
            </Tender>
          </PayData>
          <AuthCode>AUTHCODE</AuthCode>
        </ForceCapture>
      </Transaction>
    </Transactions>
  </RequestData>
  <RequestAuth>
    <UserPass>
      <User>user</User>
      <Password>password</Password>
    </UserPass>
  </RequestAuth>
</XMLPayRequest>
```

Card Force Capture Response

```
<?xml version="1.0" encoding="UTF-8"?>

<XMLPayResponse>
  <ResponseData>
    <Vendor>vendor</Vendor>
    <Partner>verisign</Partner>
    <TransactionResults>
      <TransactionResult>
        <Result>0</Result>
        <AVSResult>
          <StreetMatch>Service Not Requested</StreetMatch>
          <ZipMatch>Service Not Requested</ZipMatch>
        </AVSResult>
        <CVResult>Service Not Requested</CVResult>
        <Message>Approved</Message>
        <PNRef>V54A09206761</PNRef>
        <AuthCode>AUTHCO</AuthCode>
        <HostCode>00</HostCode>
        <OrigResult>0</OrigResult>
      </TransactionResult>
    </TransactionResults>
  </ResponseData>
</XMLPayResponse>
```

Card Sale Request

```xml
<?xml version="1.0" encoding="UTF-8"?>

<XMLPayRequest Timeout="40"
xmlns="http://www.verisign.com/XMLPay">
  <RequestData>
    <Vendor>vendor</Vendor>
    <Partner>VeriSign</Partner>
    <Transactions>
      <Transaction>
        <Sale>
          <PayData>
            <Invoice>
              <NationalTaxIncl>N</NationalTaxIncl>
              <TotalAmt>24.97</TotalAmt>
            </Invoice>
            <Tender>
              <Card>
                <CardType>visa</CardType>
                <CardNum>5105105105105100</CardNum>
                <ExpDate>200911</ExpDate>
                <NameOnCard/>
              </Card>
            </Tender>
          </PayData>
        </Sale>
      </Transaction>
    </Transactions>
  </RequestData>
  <RequestAuth>
    <UserPass>
      <User>user</User>
      <Password>password</Password>
    </UserPass>
  </RequestAuth>
</XMLPayRequest>
```

Card Sale Response

```xml
<?xml version="1.0" encoding="UTF-8"?>

<XMLPayResponse>
  <ResponseData>
    <Vendor>vendor</Vendor>
    <Partner>verisign</Partner>
    <TransactionResults>
      <TransactionResult>
        <Result>0</Result>
        <AVSResult>
          <StreetMatch>Service Not Available</StreetMatch>
          <ZipMatch>Service Not Available</ZipMatch>
        </AVSResult>
        <CVResult>Service Not Requested</CVResult>
        <Message>Approved</Message>
        <PNRef>V64A09909896</PNRef>
        <AuthCode>968PNI</AuthCode>
        <HostCode>00</HostCode>
        <OrigResult>0</OrigResult>
      </TransactionResult>
    </TransactionResults>
  </ResponseData>
</XMLPayResponse>
```

Card Status Request

```
<?xml version="1.0" encoding="UTF-8"?>

<XMLPayRequest Timeout="40"
xmlns="http://www.verisign.com/XMLPay">
  <RequestData>
    <Vendor>vendor</Vendor>
    <Partner>VeriSign</Partner>
    <Transactions>
      <Transaction>
        <GetStatus>
          <PNRef>V54A09206691</PNRef>
        </GetStatus>
      </Transaction>
    </Transactions>
  </RequestData>
  <RequestAuth>
    <UserPass>
      <User>user</User>
      <Password>password</Password>
    </UserPass>
  </RequestAuth>
</XMLPayRequest>
```

Card Status Response

```
<?xml version="1.0" encoding="UTF-8"?>

<XMLPayResponse>
  <ResponseData>
    <Vendor>vendor</Vendor>
    <Partner>verisign</Partner>
    <TransactionResults>
      <TransactionResult>
        <Result>0</Result>
        <AVSResult>
          <StreetMatch>Service Not Requested</StreetMatch>
          <ZipMatch>Service Not Requested</ZipMatch>
        </AVSResult>
        <CVResult>Service Not Requested</CVResult>
        <Message>Approved</Message>
        <PNRef>V54A09206691</PNRef>
        <OrigResult>0</OrigResult>
      </TransactionResult>
    </TransactionResults>
  </ResponseData>
</XMLPayResponse>
```

Card Void Request

```
<?xml version="1.0" encoding="UTF-8"?>

<XMLPayRequest Timeout="40"
xmlns="http://www.verisign.com/XMLPay">
  <RequestData>
    <Vendor>vendor</Vendor>
    <Partner>VeriSign</Partner>
    <Transactions>
      <Transaction>
        <Void>
          <PNRef>V54A09206691</PNRef>
        </Void>
      </Transaction>
    </Transactions>
  </RequestData>
  <RequestAuth>
    <UserPass>
      <User>user</User>
      <Password>password</Password>
    </UserPass>
  </RequestAuth>
</XMLPayRequest>
```

Card Void Response

```
<?xml version="1.0" encoding="UTF-8"?>

<XMLPayResponse>
  <ResponseData>
    <Vendor>vendor</Vendor>
    <Partner>verisign</Partner>
    <TransactionResults>
      <TransactionResult>
        <Result>0</Result>
        <AVSResult>
          <StreetMatch>Service Not Requested</StreetMatch>
          <ZipMatch>Service Not Requested</ZipMatch>
        </AVSResult>
        <CVResult>Service Not Requested</CVResult>
        <Message>Approved</Message>
        <PNRef>V54A09206748</PNRef>
        <HostCode>00</HostCode>
        <OrigResult>0</OrigResult>
      </TransactionResult>
    </TransactionResults>
  </ResponseData>
</XMLPayResponse>
```

Check Sale Request

```
<?xml version="1.0" encoding="UTF-8"?>

<XMLPayRequest Timeout="40" version="1.1"
xmlns="http://www.verisign.com/XMLPay"
>  <RequestData>
    <Vendor>vendor</Vendor>
    <Partner>VeriSign</Partner>
    <Transactions>
      <Transaction>
        <Sale>
          <PayData>
            <Invoice>
              <InvNum>1</InvNum>
              <Date>20020208</Date>
              <BillTo>
                <Name>billtoname</Name>
                <Address>
                  <Street>street</Street>
                  <City>city</City>
                  <State>NY</State>
                  <Zip>67890</Zip>
                  <Country>US</Country>
                </Address>
                <EMail>email</EMail>
                <TaxExempt>N</TaxExempt>
              </BillTo>
              <NationalTaxIncl>Y</NationalTaxIncl>
              <TotalAmt>24.97</TotalAmt>
            </Invoice>
            <Tender>
              <Check>
                <CheckType>P</CheckType>
                <CheckNum>1001</CheckNum>
                <MICR>1234567890439085000l001</MICR>
                <DL>CA123456</DL>
                <SS>123456789</SS>
                <DOB>19000601</DOB>
              </Check>
            </Tender>
          </PayData>
        </Sale>
      </Transaction>
    </Transactions>
  </RequestData>
  <RequestAuth>
    <UserPass>
      <User>user</User>
      <Password>password</Password>
    </UserPass>
  </RequestAuth>
</XMLPayRequest>
```

Check Sale Response

```
<?xml version="1.0" encoding="UTF-8"?>

<XMLPayResponse xmlns="http://www.verisign.com/XMLPay">
  <ResponseData>
    <Vendor>vendor</Vendor>
    <Partner>verisign</Partner>
    <TransactionResults>
      <TransactionResult>
        <Result>0</Result>
        <AVSResult>
          <StreetMatch>Service Not Requested</StreetMatch>
          <ZipMatch>Service Not Requested</ZipMatch>
        </AVSResult>
        <CVResult>Service Not Requested</CVResult>
        <Message>Approved</Message>
        <PNRef>VCNB01775729</PNRef>
        <OrigResult>0</OrigResult>
      </TransactionResult>
    </TransactionResults>
  </ResponseData>
</XMLPayResponse>
```

Fraud Screen Sale Request

Note To obtain Fraud results in the XMLPay response, you must specify XMLPayRequest version = “1.1”. The results appear in the ExtData fields of the response (as shown in the Fraud Screen Sale Response example on page 52).

```
<?xml version="1.0" encoding="UTF-8"?>

<XMLPayRequest Timeout="40" version="1.1"
xmlns="http://www.verisign.com/XMLPay"
>  <RequestData>
    <Vendor>vendor</Vendor>
    <Partner>VeriSign</Partner>
    <Transactions>
      <Transaction>
        <Sale>
          <PayData>
            <Invoice>
              <InvNum>1</InvNum>
              <Date>20020208</Date>
              <BillFrom>
                <Name>billfromname</Name>
                <Address>
                  <Street>street</Street>
                  <City>city</City>
                  <State>CA</State>
                  <Zip>12345</Zip>
                  <Country>US</Country>
                </Address>
                <EMail>email</EMail>
                <Phone>1234567890</Phone>
                <Fax>9876543210</Fax>
                <URL>http://www.anydomain.com</URL>
              </BillFrom>
              <BillTo>
                <Name>billtoname</Name>
                <Address>
                  <Street>street</Street>
                  <City>city</City>
                  <State>NY</State>
                  <Zip>67890</Zip>
                  <Country>US</Country>
                </Address>
                <EMail>email</EMail>
```

```
  <Phone>1234567890</Phone>
  <Fax>9876543210</Fax>
  <CustCode>custcode</CustCode>
  <PONum>123456</PONum>
  <TaxExempt>Y</TaxExempt>
</BillTo>
<ShipFrom>
  <Name>shipfromname</Name>
  <Address>
    <Street>street</Street>
    <City>city</City>
    <State>AS</State>
    <Zip>12345</Zip>
    <Country>US</Country>
  </Address>
  <EMail>email</EMail>
  <Phone>1234567890</Phone>
  <Fax>9876543210</Fax>
</ShipFrom>
<ShipTo>
  <Name>shiptoname</Name>
  <Address>
    <Street>street</Street>
    <City>city</City>
    <State>AS</State>
    <Zip>67890</Zip>
    <Country>WA</Country>
  </Address>
  <EMail>email</EMail>
  <Phone>1234567890</Phone>
  <Fax>9876543210</Fax>
</ShipTo>
<Description>Description 1</Description>
<Description>Description 2</Description>
<Description>Description 3</Description>
<Description>Description 4</Description>
<Items>
  <Item Number="1">
    <SKU>1111</SKU>
    <UPC>9999</UPC>
    <Description>Description</Description>
    <Quantity>1</Quantity>
    <UnitOfMeasurement>INQ</UnitOfMeasurement>
    <UnitPrice>9.0</UnitPrice>
    <DiscountAmt>0.90</DiscountAmt>
    <TaxAmt>4.46</TaxAmt>
```

```
                  <TotalAmt>18.37</TotalAmt>
                </Item>
              </Items>
              <DiscountAmt>12.93</DiscountAmt>
              <ShippingAmt>3.67</ShippingAmt>
              <DutyAmt>5.23</DutyAmt>
              <TaxAmt>8.13</TaxAmt>
              <NationalTaxIncl>Y</NationalTaxIncl>
              <TotalAmt>24.97</TotalAmt>
              <Comment>Comment</Comment>
            </Invoice>
            <Tender>
              <Card>
                <CardType>mastercard</CardType>
                <CardNum>5105105105105100</CardNum>
                <ExpDate>200912</ExpDate>
              </Card>
            </Tender>
          </PayData>
        </Sale>
      </Transaction>
    </Transactions>
  </RequestData>
  <RequestAuth>
    <UserPass>
      <User>user</User>
      <Password>password</Password>
    </UserPass>
  </RequestAuth>
</XMLPayRequest>
```

Fraud Screen Sale Response

```
<?xml version="1.0" encoding="UTF-8"?>

<XMLPayResponse xmlns="http://www.verisign.com/XMLPay">
  <ResponseData>
    <Vendor>vendor</Vendor>
    <Partner>verisign</Partner>
    <TransactionResults>
    <TransactionResult>
      <Result>0</Result>
      <AVSResult>
       <StreetMatch>Service Not Requested</StreetMatch>
       <ZipMatch>Service Not Requested</ZipMatch>
      </AVSResult>
      <CVResult>Service Not Requested</CVResult>
      <Message>Approved</Message>
      <PNRef>VAAB16811448</PNRef>
      <OrigResult>0</OrigResult>
      <ExtData Name="SCORE" Value="236"></ExtData>
      <ExtData Name="REASON1" Value="8"></ExtData>
      <ExtData Name="REASON2" Value="6"></ExtData>
      <ExtData Name="REASON3" Value="1"></ExtData>
      <ExtData Name="EXCEPTION1" Value="102"></ExtData>
      <ExtData Name="EXCEPTION2" Value="105"></ExtData>
      <ExtData Name="EXCEPTION3" Value="106"></ExtData>
      <ExtData Name="EXCEPTION4" Value="107"></ExtData>
      <ExtData Name="EXCEPTION5" Value="101"></ExtData>
      <ExtData Name="EXCEPTION6" Value="102"></ExtData>
      <ExtData Name="EXCEPTION7" Value="103"></ExtData>
      <ExtData Name="FRAUDCODE" Value="0"></ExtData>
      <ExtData Name="FRAUDMSG" Value="No error
detected."></ExtData>
   </TransactionResult>
    </TransactionResults>
  </ResponseData>
</XMLPayResponse>
```

Line Item Sale Request

```
<?xml version="1.0" encoding="UTF-8"?>

<XMLPayRequest Timeout="40" version="1.1"
xmlns="http://www.verisign.com/XMLPay"
>  <RequestData>
    <Vendor>vendor</Vendor>
    <Partner>VeriSign</Partner>
    <Transactions>
      <Transaction>
        <Sale>
          <PayData>
            <Invoice>
              <BillTo>
                <Name>billtoname</Name>
                <Address>
                  <Street>street</Street>
                  <City>city</City>
                  <State>NY</State>
                  <Zip>67890</Zip>
                  <Country>US</Country>
                </Address>
                <EMail>email</EMail>
                <Phone>1234567890</Phone>
                <Fax>9876543210</Fax>
                <CustCode>custcode</CustCode>
                <PONum>123456</PONum>
                <TaxExempt>Y</TaxExempt>
              </BillTo>
              <Items>
                <Item Number="1">
                  <SKU>1111</SKU>
                  <UPC>9999</UPC>
                  <Description>Description</Description>
                  <Quantity>1</Quantity>
                  <UnitOfMeasurement>INQ</UnitOfMeasurement>
                  <UnitPrice>9.0</UnitPrice>
                  <DiscountAmt>0.90</DiscountAmt>
                  <TaxAmt>4.46</TaxAmt>
                  <TotalAmt>18.37</TotalAmt>
                </Item>
              </Items>
              <NationalTaxIncl>N</NationalTaxIncl>
              <TotalAmt>24.97</TotalAmt>
            </Invoice>
            <Tender>
              <Card>
                <CardType>mastercard</CardType>
                <CardNum>5105105105105100</CardNum>
                <ExpDate>200912</ExpDate>
              </Card>
            </Tender>
          </PayData>
        </Sale>
      </Transaction>
    </Transactions>
  </RequestData>
  <RequestAuth>
    <UserPass>
      <User>user</User>
      <Password>password</Password>
    </UserPass>
  </RequestAuth>
</XMLPayRequest>
```

Line Item Sale Response

```
<?xml version="1.0" encoding="UTF-8"?>

<XMLPayResponse xmlns="http://www.verisign.com/XMLPay">
  <ResponseData>
    <Vendor>vendor</Vendor>
    <Partner>verisign</Partner>
    <TransactionResults>
      <TransactionResult>
        <Result>0</Result>
        <AVSResult>
          <StreetMatch>Match</StreetMatch>
          <ZipMatch>Match</ZipMatch>
        </AVSResult>
        <CVResult>Service Not Requested</CVResult>
        <Message>Approved</Message>
        <PNRef>VTHA55395881</PNRef>
        <AuthCode>624PNI</AuthCode>
        <HostCode>00</HostCode>
        <OrigResult>0</OrigResult>
      </TransactionResult>
    </TransactionResults>
  </ResponseData>
</XMLPayResponse>
```

APPENDIX A

XMLPay Schemas

Note The **RepeatSale** element is defined, but not supported.

XMLPay Schema

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!--Generated by XML Authority. Conforms to w3c
http://www.w3.org/2000/10/XMLSchema-->
<xsd:schema xmlns = "http://www.verisign.com/XMLPay"
     targetNamespace = "http://www.verisign.com/XMLPay"
     xmlns:xsd = "http://www.w3.org/2001/XMLSchema"
      xmlns:ds = "http://www.w3.org/2000/09/xmldsig#"
     elementFormDefault = "qualified"
     attributeFormDefault = "unqualified">

    <!-- <xsd:include schemaLocation = "XMLPayTypes.xsd"/> -->

    <xsd:element name = "XMLPayRequest">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref = "RequestData"/>
                <xsd:element ref = "RequestAuth"/>
            </xsd:sequence>
            <xsd:attribute name = "version" type = "xsd:string"/>
            <xsd:attribute name = "Timeout" type = "xsd:int"/>
            <xsd:attribute name = "UniqueTransactionIdentifier" type = "xsd:string"/>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name = "RequestData">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name = "Vendor" type = "UserIdType"/>
                <xsd:element name = "Partner" type = "UserIdType"/>
                <xsd:element name = "Transactions">
                    <xsd:complexType>
                        <xsd:sequence>
```

```
                            <xsd:element ref = "Transaction" maxOccurs = "unbounded"/>
                        </xsd:sequence>
                    </xsd:complexType>
                </xsd:element>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name = "RequestAuth">
        <xsd:complexType>
            <xsd:choice>
                <xsd:element ref = "UserPass"/>
                <xsd:element ref = "ds:Signature"/>
            </xsd:choice>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name = "XMLPayResponse">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref = "ResponseData"/>
                <xsd:element ref = "ds:Signature" minOccurs = "0"/>
                <xsd:element name = "TransactionReceipts" minOccurs = "0">
                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element ref = "XMLPayReceipt" maxOccurs = "unbounded"/>
                        </xsd:sequence>
                    </xsd:complexType>
                </xsd:element>
            </xsd:sequence>
            <xsd:attribute name = "version" type = "xsd:string"/>
            <xsd:attribute name = "UniqueTransactionIdentifier" type = "xsd:string"/>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name = "ResponseData">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name = "Vendor" type = "UserIdType"/>
                <xsd:element name = "Partner" type = "UserIdType"/>
                <xsd:element name = "TransactionResults">
                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element ref = "TransactionResult" maxOccurs =
"unbounded"/>
                        </xsd:sequence>
                    </xsd:complexType>
                </xsd:element>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name = "XMLPayReceipt">
        <xsd:complexType>
```

```
            <xsd:sequence>
                <xsd:element ref = "ReceiptData"/>
                <xsd:element ref = "ds:Signature" minOccurs = "0"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name = "ReceiptData">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name = "Vendor" type = "UserIdType"/>
                <xsd:element name = "Partner" type = "UserIdType"/>
                <xsd:element ref = "Transaction"/>
                <xsd:element ref = "TransactionResult"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name = "Transaction">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:choice>
                    <xsd:element ref = "Authorization"/>
                    <xsd:element ref = "Capture"/>
                    <xsd:element ref = "Sale"/>
                    <xsd:element ref = "Credit"/>
                    <xsd:element ref = "Void"/>
                    <xsd:element ref = "ForceCapture"/>
                    <xsd:element ref = "RepeatSale"/>
                    <xsd:element ref = "GetStatus"/>
                </xsd:choice>
            </xsd:sequence>
            <xsd:attribute name = "Id" type = "xsd:ID"/>
            <xsd:attribute name = "CustRef" type = "ReferenceIdType"/>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name = "Authorization">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref = "PayData"/>
                <xsd:element ref = "PayDataAuth" minOccurs = "0"/>
                <xsd:element ref = "ExtData" minOccurs = "0" maxOccurs = "unbounded"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name = "Capture">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name = "PNRef" type = "PNRefType"/>
                <xsd:element ref = "Invoice" minOccurs = "0"/>
                <xsd:element ref = "ExtData" minOccurs = "0" maxOccurs = "unbounded"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name = "Sale">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref = "PayData"/>
                <xsd:element ref = "PayDataAuth" minOccurs = "0"/>
                <xsd:element ref = "ExtData" minOccurs = "0" maxOccurs = "unbounded"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name = "Credit">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:choice>
                    <xsd:element name = "PNRef" type = "PNRefType"/>
                    <xsd:element ref = "Tender"/>
                </xsd:choice>
                <xsd:element ref = "Invoice" minOccurs = "0"/>
                <xsd:element ref = "ExtData" minOccurs = "0" maxOccurs = "unbounded"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name = "Void">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name = "PNRef" type = "PNRefType"/>
                <xsd:element ref = "ExtData" minOccurs = "0" maxOccurs = "unbounded"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name = "ForceCapture">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref = "PayData"/>
                <xsd:element ref = "PayDataAuth" minOccurs = "0"/>
                <xsd:element name = "AuthCode" type = "AuthCodeType"/>
                <xsd:element ref = "ExtData" minOccurs = "0" maxOccurs = "unbounded"/>
            </xsd:sequence>
    </xsd:complexType>
    </xsd:element>
    <xsd:element name = "RepeatSale">
            <xsd:complexType>
            <xsd:sequence>
                <xsd:element name = "PNRef" type = "PNRefType"/>
                <xsd:element ref = "Invoice" minOccurs = "0"/>
                <xsd:element ref = "ExtData" minOccurs = "0" maxOccurs = "unbounded"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name = "GetStatus">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name = "PNRef" type = "PNRefType"/>
                <xsd:element ref = "ExtData" minOccurs = "0" maxOccurs = "unbounded"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name = "TransactionResult">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name = "Result" type = "xsd:int"/>
                <xsd:element ref = "AVSResult" minOccurs = "0"/>
                <xsd:element name = "CVResult" type = "MatchResultEnum" minOccurs =
"0"/>
                <xsd:element name = "Message" type = "xsd:string" minOccurs = "0"/>
                <xsd:element name = "PNRef" type = "PNRefType" minOccurs = "0"/>
                <xsd:element name = "AuthCode" type = "AuthCodeType" minOccurs = "0"/>
                <xsd:element name = "HostCode" type = "HostCodeType" minOccurs = "0"/>
                <xsd:element name = "HostURL" type = "xsd:anyURI" minOccurs = "0"/>
                <xsd:element name = "OrigResult" type = "xsd:byte" minOccurs = "0"/>
                <xsd:element name = "TrStatus" type = "StatusType" minOccurs = "0"/>
                <xsd:element name = "ReceiptURL" type = "xsd:anyURI" minOccurs = "0"/>
                <xsd:element ref = "ExtData" minOccurs = "0" maxOccurs = "unbounded"/>
            </xsd:sequence>
            <xsd:attribute name = "Id" type = "xsd:ID"/>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name = "AVSResult">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name = "StreetMatch" type = "MatchResultEnum"/>
                <xsd:element name = "ZipMatch" type = "MatchResultEnum"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
    <xsd:simpleType name = "AuthCodeType">
        <xsd:restriction base = "xsd:string"/>
    </xsd:simpleType>
    <xsd:simpleType name = "HostCodeType">
        <xsd:restriction base = "xsd:string"/>
    </xsd:simpleType>
    <xsd:simpleType name = "MatchResultEnum">
        <xsd:restriction base = "xsd:string"/>
    </xsd:simpleType>
    <xsd:element name = "UserPass">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name = "User" type = "UserIdType"/>
                <xsd:element name = "UserDomain" type = "UserIdType" minOccurs = "0"/>
```

```
              <xsd:element name = "Password" type = "PasswordType"/>
           </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

XMLPay Types Schema

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!--Generated by XML Authority. Conforms to w3c
http://www.w3.org/2000/10/XMLSchema-->
<xsd:schema xmlns = "http://www.verisign.com/XMLPay"
   targetNamespace = "http://www.verisign.com/XMLPay"
   xmlns:xsd = "http://www.w3.org/2001/XMLSchema"
      xmlns:ds = "http://www.w3.org/2000/09/xmldsig#"
   elementFormDefault = "qualified"
   attributeFormDefault = "unqualified">

  <import namespace="http://www.w3.org/2000/09/xmldsig#"

schemaLocation="http://www.w3.org/TR/2000/CR-xmldsig-core-20001031/xmldsig-core-sc
hema.xsd"/>

  <xsd:element name = "PayData">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref = "Invoice"/>
        <xsd:element ref = "Tender"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name = "PayDataAuth">
    <xsd:complexType>
      <xsd:choice>
        <xsd:element name = "PKCS7Signature" type = "xsd:base64Binary"/>
        <xsd:element ref = "ds:Signature"/>
      </xsd:choice>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name = "Invoice">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name = "InvNum" type = "InvNumType" minOccurs = "0"/>
        <xsd:element name = "Date" type = "DateType" minOccurs = "0"/>
        <xsd:element name = "BillFrom" minOccurs = "0">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name = "Name" type = "NameType" minOccurs = "0"/>
              <xsd:element ref = "Address" minOccurs = "0"/>
              <xsd:element name = "EMail" type = "EMailType" minOccurs = "0"/>
              <xsd:element name = "Phone" type = "PhoneNumberType" minOccurs = "0"/>
              <xsd:element name = "Fax" type = "PhoneNumberType" minOccurs = "0"/>
              <xsd:element name = "URL" type = "xsd:anyURI" minOccurs = "0"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
        <xsd:element name = "BillTo" minOccurs = "0">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name = "CustomerId" type = "UserIdType" minOccurs = "0"/>
              <xsd:element name = "Name" type = "NameType" minOccurs = "0"/>
              <xsd:element ref = "Address" minOccurs = "0"/>
              <xsd:element name = "EMail" type = "EMailType" minOccurs = "0"/>
              <xsd:element name = "Phone" type = "PhoneNumberType" minOccurs = "0"/>
              <xsd:element name = "Fax" type = "PhoneNumberType" minOccurs = "0"/>
              <xsd:element name = "CustCode" type = "CustCodeType" minOccurs = "0"/>
              <xsd:element name = "PONum" type = "PONumType" minOccurs = "0"/>
              <xsd:element name = "TaxExempt" type = "YesNoType" minOccurs = "0"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
        <xsd:element name = "ShipFrom" minOccurs = "0">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name = "Name" type = "NameType" minOccurs = "0"/>
              <xsd:element ref = "Address"/>
              <xsd:element name = "EMail" type = "EMailType" minOccurs = "0"/>
              <xsd:element name = "Phone" type = "PhoneNumberType" minOccurs = "0"/>
              <xsd:element name = "Fax" type = "PhoneNumberType" minOccurs = "0"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
        <xsd:element name = "ShipTo" minOccurs = "0">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name = "Name" type = "NameType" minOccurs = "0"/>
              <xsd:element ref = "Address"/>
              <xsd:element name = "EMail" type = "EMailType" minOccurs = "0"/>
              <xsd:element name = "Phone" type = "PhoneNumberType" minOccurs = "0"/>
              <xsd:element name = "Fax" type = "PhoneNumberType" minOccurs = "0"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
        <xsd:element name = "Description" type = "DescType" minOccurs = "0" maxOccurs
= "unbounded"/>
        <xsd:element name = "Items" minOccurs = "0">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element ref = "Item" minOccurs = "0" maxOccurs = "unbounded"/>
            </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name = "DiscountAmt" type = "CurrencyAmount" minOccurs = "0"/>
      <xsd:element name = "ShippingAmt" type = "CurrencyAmount" minOccurs = "0"/>
      <xsd:element name = "DutyAmt" type = "CurrencyAmount" minOccurs = "0"/>
      <xsd:element name = "TaxAmt" type = "CurrencyAmount" minOccurs = "0"/>
      <xsd:element name = "NationalTaxIncl" type = "YesNoType" minOccurs = "0"/>
      <xsd:element name = "TotalAmt" type = "CurrencyAmount"/>
      <xsd:element name = "Comment" type = "CommentType" minOccurs = "0"/>
      <xsd:element ref = "ExtData" minOccurs = "0" maxOccurs = "unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:simpleType name = "InvNumType">
  <xsd:restriction base = "xsd:string"/>
</xsd:simpleType>
<xsd:simpleType name = "PONumType">
  <xsd:restriction base = "xsd:string"/>
</xsd:simpleType>
<xsd:simpleType name = "DescType">
  <xsd:restriction base = "xsd:string"/>
</xsd:simpleType>
<xsd:simpleType name = "CommentType">
  <xsd:restriction base = "xsd:string"/>
</xsd:simpleType>
<xsd:simpleType name = "UnitOfMeasurementType">
  <xsd:restriction base = "xsd:string"/>
</xsd:simpleType>
<xsd:simpleType name = "SKUType">
  <xsd:restriction base = "xsd:string"/>
</xsd:simpleType>
<xsd:simpleType name = "UPCType">
  <xsd:restriction base = "xsd:string"/>
</xsd:simpleType>
<xsd:simpleType name = "NameType">
  <xsd:restriction base = "xsd:string"/>
</xsd:simpleType>
<xsd:simpleType name = "StateType">
  <xsd:restriction base = "xsd:string"/>
</xsd:simpleType>
<xsd:simpleType name = "ZipType">
  <xsd:restriction base = "xsd:string"/>
</xsd:simpleType>
<xsd:simpleType name = "CountryCode">
  <xsd:restriction base = "xsd:string"/>
</xsd:simpleType>
<xsd:simpleType name = "EMailType">
  <xsd:restriction base = "xsd:string"/>
</xsd:simpleType>
<xsd:simpleType name = "PhoneNumberType">
```

```
    <xsd:restriction base = "xsd:string"/>
  </xsd:simpleType>
  <xsd:simpleType name = "Decimal9_2">
    <xsd:restriction base = "xsd:string"/>
  </xsd:simpleType>
  <xsd:simpleType name = "CurrencyCode">
    <xsd:restriction base = "xsd:string"/>
  </xsd:simpleType>
  <xsd:simpleType name = "TenderTypeEnum">
    <xsd:restriction base = "xsd:string"/>
  </xsd:simpleType>
  <xsd:simpleType name = "TransactionTypeEnum">
    <xsd:restriction base = "xsd:string"/>
  </xsd:simpleType>
  <xsd:simpleType name = "AcctTypeEnum">
    <xsd:restriction base = "xsd:string"/>
  </xsd:simpleType>
  <xsd:simpleType name = "AcctNumType">
    <xsd:restriction base = "xsd:string"/>
  </xsd:simpleType>
  <xsd:simpleType name = "MagDataType">
    <xsd:restriction base = "xsd:string"/>
  </xsd:simpleType>
  <xsd:simpleType name = "ABAType">
    <xsd:restriction base = "xsd:string"/>
  </xsd:simpleType>
  <xsd:simpleType name = "CardTypeEnum">
    <xsd:restriction base = "xsd:string"/>
  </xsd:simpleType>
  <xsd:simpleType name = "CardNumType">
    <xsd:restriction base = "xsd:string"/>
  </xsd:simpleType>
  <xsd:simpleType name = "ExpDateType">
    <xsd:restriction base = "xsd:string"/>
  </xsd:simpleType>
  <xsd:simpleType name = "CVType">
    <xsd:restriction base = "xsd:string"/>
  </xsd:simpleType>
  <xsd:simpleType name = "CheckTypeEnum">
    <xsd:restriction base = "xsd:string"/>
  </xsd:simpleType>
  <xsd:simpleType name = "CheckNumType">
    <xsd:restriction base = "xsd:string"/>
  </xsd:simpleType>
  <xsd:simpleType name = "MICRType">
    <xsd:restriction base = "xsd:string"/>
  </xsd:simpleType>
  <xsd:simpleType name = "DLType">
    <xsd:restriction base = "xsd:string"/>
  </xsd:simpleType>
  <xsd:simpleType name = "SSType">
    <xsd:restriction base = "xsd:string"/>
  </xsd:simpleType>
  <xsd:simpleType name = "UserIdType">
    <xsd:restriction base = "xsd:string"/>
  </xsd:simpleType>
  <xsd:simpleType name = "CustCodeType">
    <xsd:restriction base = "xsd:string"/>
  </xsd:simpleType>
  <xsd:simpleType name = "PasswordType">
    <xsd:restriction base = "xsd:string"/>
  </xsd:simpleType>
  <xsd:simpleType name = "PNRefType">
    <xsd:restriction base = "xsd:string"/>
  </xsd:simpleType>
  <xsd:simpleType name = "URI">
    <xsd:restriction base = "xsd:string"/>
  </xsd:simpleType>
  <xsd:simpleType name = "SicType">
    <xsd:restriction base = "xsd:string"/>
  </xsd:simpleType>
  <xsd:simpleType name = "StatusType">
    <xsd:restriction base = "xsd:string"/>
  </xsd:simpleType>
  <xsd:simpleType name = "ReferenceIdType">
    <xsd:restriction base = "xsd:string"/>
  </xsd:simpleType>
  <xsd:simpleType name = "IpCodeType">
    <xsd:restriction base = "xsd:string"/>
  </xsd:simpleType>
  <xsd:simpleType name = "YesNoType"> <!-- Y or N -->
    <xsd:restriction base = "xsd:string"/>
  </xsd:simpleType>
  <xsd:simpleType name = "DateType"> <!-- yyyymmdd -->
    <xsd:restriction base = "xsd:string"/>
  </xsd:simpleType>
  <xsd:element name = "Item">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name = "SKU" type = "SKUType" minOccurs = "0"/>
        <xsd:element name = "UPC" type = "UPCType" minOccurs = "0"/>
        <xsd:element name = "Description" type = "CommentType" minOccurs = "0"/>
        <xsd:element name = "Quantity" type = "xsd:int" minOccurs = "0"/>
        <xsd:element name = "UnitOfMeasurement" type = "UnitOfMeasurementType" 
minOccurs = "0"/>
        <xsd:element name = "UnitPrice" type = "CurrencyAmount" minOccurs = "0"/>
        <xsd:element name = "DiscountAmt" type = "CurrencyAmount" minOccurs = "0"/>
        <xsd:element name = "TaxAmt" type = "CurrencyAmount" minOccurs = "0"/>
        <xsd:element name = "TotalAmt" type = "CurrencyAmount" minOccurs = "0"/>
        <xsd:element ref = "ExtData" minOccurs = "0" maxOccurs = "unbounded"/>
    </xsd:sequence>
    <xsd:attribute name = "Number" use = "required" type = "xsd:int"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name = "Address">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name = "Street" type = "NameType" minOccurs = "0"/>
      <xsd:element name = "City" type = "NameType" minOccurs = "0"/>
      <xsd:element name = "State" type = "StateType" minOccurs = "0"/>
      <xsd:element name = "Zip" type = "ZipType" minOccurs = "0"/>
      <xsd:element name = "Country" type = "CountryCode" minOccurs = "0"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:complexType name = "CurrencyAmount">
  <xsd:simpleContent>
    <xsd:extension base = "Decimal9_2">
      <xsd:attribute name = "Currency" type = "CurrencyCode"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
<xsd:element name = "Tender" type = "TenderInfo"/>
<xsd:complexType name = "TenderInfo">
  <xsd:choice>
    <xsd:element ref = "ACH"/>
    <xsd:element ref = "Card"/>
    <xsd:element ref = "Check"/>
  </xsd:choice>
</xsd:complexType>
<xsd:element name = "ACH">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name = "AcctType" type = "AcctTypeEnum"/>
      <xsd:element name = "AcctNum" type = "AcctNumType"/>
      <xsd:element name = "ABA" type = "ABAType"/>
      <xsd:element name = "Prenote" type = "YesNoType" minOccurs = "0"/>
      <xsd:element ref = "ExtData" minOccurs = "0" maxOccurs = "unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name = "Card">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name = "CardType" type = "CardTypeEnum" minOccurs = "0"/>
      <xsd:element name = "CardNum" type = "CardNumType"/>
      <xsd:element name = "ExpDate" type = "ExpDateType"/>
      <xsd:element name = "CVNum" type = "CVType" minOccurs = "0"/>
      <xsd:element name = "MagData" type = "MagDataType" minOccurs = "0"/>
      <xsd:element name = "NameOnCard" type = "NameType" minOccurs = "0"/>
```

```
        <xsd:element ref = "ExtData" minOccurs = "0" maxOccurs = "unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name = "Check">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name = "CheckType" type = "CheckTypeEnum"/>
        <xsd:element name = "AllianceNum" type = "xsd:string" minOccurs = "0"/>
        <xsd:element name = "CheckNum" type = "CheckNumType"/>
        <xsd:element name = "MICR" type = "MICRType"/>
        <xsd:element name = "DL" type = "DLType" minOccurs = "0"/>
        <xsd:element name = "SS" type = "SSType" minOccurs = "0"/>
        <xsd:element name = "DOB" type = "DateType" minOccurs = "0"/>
        <xsd:element ref = "ExtData" minOccurs = "0" maxOccurs = "unbounded"/>
        <xsd:element ref = "Address" minOccurs = "0"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:complexType name = "AuthInfoType">
    <xsd:sequence>
      <xsd:element name = "UserId" type = "UserIdType"/>
      <xsd:element name = "Password" type = "PasswordType" minOccurs = "0"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:element name = "ExtData">
    <xsd:complexType>
      <xsd:simpleContent>
        <xsd:extension base = "xsd:string">
          <xsd:attribute name = "Name" type = "xsd:string"/>
          <xsd:attribute name = "Value" type = "xsd:string"/>
        </xsd:extension>
      </xsd:simpleContent>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

APPENDIX B

XMLPay DTD

A document type definition (DTD) defines the structure of an XML document. With a DTD you can define the set and order of tags, as well as the attributes for each. A well-formed XML document is considered valid when it conforms to its corresponding DTD.

The following is a Document Type Definition (DTD) representation of the XMLPay schema.

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

<!ELEMENT InvNum (#PCDATA)>

<!ELEMENT Date (#PCDATA)>

<!ELEMENT Name (#PCDATA)>

<!ELEMENT Street (#PCDATA)>

<!ELEMENT City (#PCDATA)>

<!ELEMENT State (#PCDATA)>

<!ELEMENT Zip (#PCDATA)>

<!ELEMENT Country (#PCDATA)>

<!ELEMENT Address ((Street?, City?, State?, Zip?, Country?))>

<!ELEMENT EMail (#PCDATA)>

<!ELEMENT Phone (#PCDATA)>

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

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

<!ELEMENT BillFrom (Name?, Address?, EMail?, Phone?, Fax?, URL?)>

<!ELEMENT CustomerId (#PCDATA)>

<!ELEMENT CustCode (#PCDATA)>

<!ELEMENT PONum (#PCDATA)>

<!ELEMENT TaxExempt (#PCDATA)>

<!ELEMENT BillTo (CustomerId?, Name?, Address?, EMail?, Phone?,
Fax?, CustCode?, PONum?, TaxExempt?)>

<!ELEMENT ShipFrom (Name?, Address, EMail?, Phone?, Fax?)>

<!ELEMENT ShipTo (Name?, Address, EMail?, Phone?, Fax?)>

<!ELEMENT Description (#PCDATA)>

<!ELEMENT SKU (#PCDATA)>

<!ELEMENT UPC (#PCDATA)>

<!ELEMENT Quantity (#PCDATA)>

<!ELEMENT UnitOfMeasurement (#PCDATA)>

<!ELEMENT UnitPrice (#PCDATA)>
<!ATTLIST UnitPrice Currency CDATA #IMPLIED>

<!ELEMENT ExtAmt (#PCDATA)>
<!ATTLIST ExtAmt Currency CDATA #IMPLIED>

<!ELEMENT DiscountAmt (#PCDATA)>
<!ATTLIST DiscountAmt Currency CDATA #IMPLIED>

<!ELEMENT TaxAmt (#PCDATA)>
<!ATTLIST TaxAmt Currency CDATA #IMPLIED>

<!ELEMENT TotalAmt (#PCDATA)>
<!ATTLIST TotalAmt Currency CDATA #IMPLIED>

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

```
<!ATTLIST ExtData Name CDATA #IMPLIED>
<!ATTLIST ExtData Value CDATA #IMPLIED>

<!ELEMENT Item ((SKU?, UPC?, Description?, Quantity?,
UnitOfMeasurement?, UnitPrice?, ExtAmt?, DiscountAmt?, TaxAmt?,
TotalAmt?, ExtData*))>
<!ATTLIST Item Number CDATA #REQUIRED>

<!ELEMENT Items (Item*)>

<!ELEMENT ShippingAmt (#PCDATA)>
<!ATTLIST ShippingAmt Currency CDATA #IMPLIED>

<!ELEMENT DutyAmt (#PCDATA)>
<!ATTLIST DutyAmt Currency CDATA #IMPLIED>

<!ELEMENT NationalTaxIncl (#PCDATA)>

<!ELEMENT Comment (#PCDATA)>

<!ELEMENT Invoice ((InvNum?, Date?, BillFrom?, BillTo?, ShipFrom?,
ShipTo?, Description*, Items?, DiscountAmt?, ShippingAmt?,
DutyAmt?, TaxAmt?, NationalTaxIncl?, TotalAmt, Comment?,
ExtData*))>

<!ELEMENT AcctType (#PCDATA)>

<!ELEMENT AcctNum (#PCDATA)>

<!ELEMENT ABA (#PCDATA)>

<!ELEMENT Prenote (#PCDATA)>

<!ELEMENT ACH ((AcctType, AcctNum, ABA, Prenote?, ExtData*))>

<!ELEMENT CardType (#PCDATA)>

<!ELEMENT CardNum (#PCDATA)>

<!ELEMENT ExpDate (#PCDATA)>

<!ELEMENT CVNum (#PCDATA)>

<!ELEMENT MagData (#PCDATA)>

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

```
<!ELEMENT Card ((CardType?, CardNum, ExpDate, CVNum?, MagData?,
NameOnCard?, ExtData*))>

<!ELEMENT CheckType (#PCDATA)>

<!ELEMENT AllianceNum (#PCDATA)>

<!ELEMENT CheckNum (#PCDATA)>

<!ELEMENT MICR (#PCDATA)>

<!ELEMENT DL (#PCDATA)>

<!ELEMENT SS (#PCDATA)>

<!ELEMENT DOB (#PCDATA)>

<!ELEMENT Check ((CheckType, AllianceNum?, CheckNum, MICR, DL?,
SS?, DOB?, ExtData*, Address?))>

<!ELEMENT Tender ((ACH| Card| Check))>

<!ELEMENT PayData ((Invoice, Tender))>

<!ELEMENT PKCS7Signature (#PCDATA)>

<!ELEMENT PayDataAuth ((PKCS7Signature| Signature))>

<!ELEMENT Vendor (#PCDATA)>

<!ELEMENT Partner (#PCDATA)>

<!ELEMENT Authorization ((PayData, PayDataAuth?, ExtData*))>

<!ELEMENT PNRef (#PCDATA)>

<!ELEMENT Capture ((PNRef, Invoice?, ExtData*))>

<!ELEMENT Sale ((PayData, PayDataAuth?, ExtData*))>

<!ELEMENT Credit (((PNRef| Tender), Invoice?, ExtData*))>

<!ELEMENT Void ((PNRef, ExtData*))>

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

```
<!ELEMENT ForceCapture ((PayData, PayDataAuth?, AuthCode,
ExtData*))>

<!ELEMENT GetStatus ((PNRef, ExtData*))>

<!ELEMENT Transaction (((Authorization| Capture| Sale| Credit|
Void| ForceCapture|GetStatus)))>
<!ATTLIST Transaction Id CDATA #IMPLIED>
<!ATTLIST Transaction CustRef CDATA #IMPLIED>

<!ELEMENT Transactions (Transaction+)>

<!ELEMENT RequestData ((Vendor, Partner, Transactions))>

<!ELEMENT User (#PCDATA)>

<!ELEMENT UserDomain (#PCDATA)>

<!ELEMENT Password (#PCDATA)>

<!ELEMENT UserPass ((User, UserDomain?, Password))>

<!ELEMENT RequestAuth ((UserPass| Signature))>

<!ELEMENT XMLPayRequest ((RequestData, RequestAuth))>
<!ATTLIST XMLPayRequest version CDATA #IMPLIED>
<!ATTLIST XMLPayRequest Timeout CDATA #IMPLIED>
<!ATTLIST XMLPayRequest UniqueTransactionIdentifier CDATA
#IMPLIED>

<!ELEMENT Result (#PCDATA)>

<!ELEMENT StreetMatch (#PCDATA)>

<!ELEMENT ZipMatch (#PCDATA)>

<!ELEMENT AVSResult ((StreetMatch, ZipMatch))>

<!ELEMENT CVResult (#PCDATA)>

<!ELEMENT Message (#PCDATA)>

<!ELEMENT HostCode (#PCDATA)>

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

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

<!ELEMENT TrStatus (#PCDATA)>

<!ELEMENT ReceiptURL (#PCDATA)>

<!ELEMENT TransactionResult ((Result, AVSResult?, CVResult?,
Message?, PNRef?, AuthCode?, HostCode?, HostURL?, OrigResult?,
TrStatus?, ReceiptURL?, ExtData*))>
<!ATTLIST TransactionResult Id CDATA #IMPLIED>

<!ELEMENT TransactionResults (TransactionResult+)>

<!ELEMENT ResponseData ((Vendor, Partner, TransactionResults))>

<!ELEMENT ReceiptData ((Vendor, Partner, Transaction,
TransactionResult))>

<!ELEMENT XMLPayReceipt ((ReceiptData, Signature?))>

<!ELEMENT TransactionReceipts (XMLPayReceipt+)>

<!ELEMENT XMLPayResponse ((ResponseData, Signature?,
TransactionReceipts?))>
<!ATTLIST XMLPayResponse version CDATA #IMPLIED>
<!ATTLIST XMLPayResponse UniqueTransactionIdentifier CDATA
#IMPLIED>
```

APPENDIX C

VeriSign Transaction Results

The transaction result status response (RESULT) is the first value returned from the VeriSign server. This value indicates the overall status of the transaction attempt. The response message (RESPMSG) provides a brief description for decline or error results.

- A value of `0` (zero) indicates that no errors occurred and the transaction was approved.
- A value less than zero indicates that a communication error occurred. In this case, no transaction is attempted. See “RESULT Values for Communications Errors” on page 76.
- A value greater than zero indicates a decline or error. See “RESULT Values for Transaction Declines or Errors” on page 78.

AVS result codes are described on page 81.

RESULT Values for Communications Errors

A value for **RESULT** less than zero indicates that a communication error occurred. In this case, no transaction is attempted.

A value of -1 or -2 usually indicates a configuration error. Either the VeriSign server is unavailable, or incorrect server/socket pairs have been specified. A value of -1 can also result when there are Internet connectivity errors. Refer other errors to VeriSign at **vps-support@verisign.com.**

RESULT	Description
-1	Failed to connect to host
-2	Failed to resolve hostname
-5	Failed to initialize SSL context
-6	Parameter list format error: & in name
-7	Parameter list format error: invalid [ ] name length clause
-8	SSL failed to connect to host
-9	SSL read failed
-10	SSL write failed
-11	Proxy authorization failed
-12	Timeout waiting for response
-13	Select failure
-14	Too many connections
-15	Failed to set socket options
-20	Proxy read failed
-21	Proxy write failed
-22	Failed to initialize SSL certificate
-23	Host address not specified
-24	Invalid transaction type
-25	Failed to create a socket
-26	Failed to initialize socket layer
-27	Parameter list format error: invalid [ ] name length clause
-28	Parameter list format error: name
-29	Failed to initialize SSL connection
-30	Invalid timeout value
-31	The certificate chain did not validate, no local certificate found
-32	The certificate chain did not validate, common name did not match URL
-99	Out of memory

RESULT Values for Transaction Declines or Errors

A RESULT greater than zero indicates a decline or error.

For this type of error, a RESPMSG name/value pair is included. The exact wording of the RESPMSG (shown in **bold**) may vary. Sometimes a colon appears after the initial RESPMSG followed by more detailed information.

Table C-1 VeriSign Transaction RESULTs/RESPMSGs

RESULT	RESPMSG/Explanation
0	**Approved**
1	User authentication failed
2	**Invalid tender type.** Your merchant bank account does not support the following credit card type that was submitted.
3	**Invalid transaction type.** Transaction type is not appropriate for this transaction. For example, you cannot credit an authorization-only transaction.
4	**Invalid amount format**
5	**Invalid merchant information.** Processor does not recognize your merchant account information. Contact your bank account acquirer to resolve this problem.
7	**Field format error**. Invalid information entered. See RESPMSG.
8	**Not a transaction server**
9	**Too many parameters or invalid stream**
10	**Too many line items**
11	Client time-out waiting for response
12	**Declined.** Check the credit card number and transaction information to make sure they were entered correctly. If this does not resolve the problem, have the customer call the credit card issuer to resolve.
13	**Referral.** Transaction was declined but could be approved with a verbal authorization from the bank that issued the card. Submit a manual Voice Authorization transaction and enter the verbal auth code.
19	**Original transaction ID not found.** The transaction ID you entered for this transaction is not valid. See RESPMSG.
20	**Cannot find the customer reference number**

Table C-1 VeriSign Transaction RESULTs/RESPMSGs (Continued)

RESULT	RESPMSG/Explanation
22	**Invalid ABA number**
23	**Invalid account number.** Check credit card number and re-submit.
24	**Invalid expiration date.** Check and re-submit.
25	**Invalid Host Mapping.** Transaction type not mapped to this host
26	**Invalid vendor account**
27	**Insufficient partner permissions**
28	**Insufficient user permissions**
29	**Invalid XML document**. This could be caused by an unrecognized XML tag or a bad XML format that cannot be parsed by the system.
30	**Duplicate transaction**
31	**Error in adding the recurring profile**
32	**Error in modifying the recurring profile**
33	**Error in canceling the recurring profile**
34	**Error in forcing the recurring profile**
35	**Error in reactivating the recurring profile**
36	**OLTP Transaction failed**
50	**Insufficient funds available in account**
99	**General error.** See RESPMSG.
100	**Transaction type not supported by host**
101	**Time-out value too small**
102	**Processor not available**
103	**Error reading response from host**
104	**Timeout waiting for processor response.** Try your transaction again.
105	**Credit error.** Make sure you have not already credited this transaction, or that this transaction ID is for a creditable transaction. (For example, you cannot credit an authorization.)
106	**Host not available**

Table C-1 VeriSign Transaction RESULTs/RESPMSGs (Continued)

RESULT	RESPMSG/Explanation
107	**Duplicate suppression time-out**
108	**Void error.** See RESPMSG. Make sure the transaction ID entered has not already been voided. If not, then look at the Transaction Detail screen for this transaction to see if it has settled. (The Batch field is set to a number greater than zero if the transaction has been settled). If the transaction has already settled, your only recourse is a reversal (credit a payment or submit a payment for a credit).
109	**Time-out waiting for host response**
111	**Capture error.** Only authorization transactions can be captured.
112	**Failed AVS check.** Address and ZIP code do not match. An authorization may still exist on the cardholder’s account.
113	**Merchant sale total will exceed the cap with current transaction**
114	**Card Security Code (CSC) Mismatch**. An authorization may still exist on the cardholder’s account.
115	**System busy, try again later**
116	**VPS Internal error - Failed to lock terminal number**
117	**Failed merchant rule check**. An attempt was made to submit a transaction that failed to meet the security settings specified on the VeriSign Manager *Security Settings* page. See *VeriSign Manager User’s Guide*.
118	**Invalid keywords found in string fields**
1000	**Generic host error.** See RESPMSG. This is a generic message returned by your credit card processor. The message itself will contain more information describing the error.

AVS Result Codes

For US customers the Address Verification Service (AVS) compares the street address and zip code submitted with that on file at the cardholder's bank. Any one of the following results can appear in the AVS Street Match and AVS Zip Match fields on the transaction detail screen:

Result	Meaning
Y	Information submitted matches information on file with cardholder's bank.
N	Information submitted does not match information on file with the cardholder's bank.
X	Cardholder's bank does not support AVS checking for this information.

Note Results can vary on the same transaction detail screen. In other words, AVS Street Match = Y and AVS Zip Match = N (and vice versa) could appear on the same transaction detail screen. When service is unavailable, no code is returned.

Index