

XML Authorization Interface



**User Guide
Version 1.0**

Introduction to XML Authorization Interface

ProPay offers individuals and small business owners the opportunity to authorize, settle, and void credit card transactions, by easily communicating with ProPay using XML.

With ProPay's XML Authorization Interface, you have the freedom to authorize credit card transactions and either settle or void them at your convenience.

ProPay makes buying on the Internet easier

ProPay's XML Authorization Interface makes it easier for you to utilize the Internet to get paid because it eliminates a step in the Internet transaction process. The XML Authorization Interface can be integrated into your website, thus becoming "invisible" to your customers. Invisibility allows customers to pay with their credit cards on the same website eliminating the need to be moved to another website to make their purchases.

XML eases transmission of information

ProPay uses XML (Extensible Markup Language) because it provides an easily recognized standard that allows companies great flexibility in sending detailed financial transactions to ProPay. XML uses *elements* and *attributes* that define, or set the limits of, data, and allows it to be read and interpreted by a computer application. Because W3C is still defining the XML standard, some elements may change in the future.

Single-user version

This version of ProPay is for a single-user and contains only credit card processing. A multi-user version of the XML Interface is also available that offers additional customizable services:

- Create new ProPay accounts
- Pay to e-mail address
- Pay to bank account

Contact ProPay at websitetools@propay.com for information on these services.

This document will instruct you on how to use ProPay's XML Authorization Interface.

Contents

1.0 Fees for the XML Authorization Interface	3
2.0 Security Precautions	3
3.0 Set Up of the XML Authorization Interface	4
4.0 XML Authorization Interface Services	5
4.1 Credit Card Authorization Only	5
4.2 Credit Card Settlement	7
4.3 Void Credit Card Authorization	9
5.0 Definition of Elements	11
6.0 XML Schemas and DTDs	13

1.0 Fees for the XML Authorization Interface

ProPay charges an initial integration fee of \$395.00 for the XML Authorization Interface and (2) two hours of technical support. Additional technical support is \$150 per hour, also billed in 15-minute increments.

A renewal fee of \$195 will be charged after every year of use.

Other small fees will be charged depending on the services used. See the service descriptions given below for exact fees on the specific services.

For further questions please contact Dan England at (801) 852-4100 or dengland@propay.com

2.0 Security Precautions

ProPay recognizes the importance of handling financial transactions in a secure manner and has taken steps to ensure that ProPay offers the best transmission security available.

1. ProPay ensures that data is always sent via a secure connection by using Secure Socket Layer (SSL), a protocol developed for the transmission of private documents and transactions over the Internet. SSL creates a secure connection between a client and a server, over which encrypted data is sent. SSL is built into all major browsers and web servers. ProPay recommends that you use SSL Version 3.0 or higher.
2. The XML Authorization Interface also requires 128-bit encryption of data instead of the more common 40- or 56- bit encryption, which is too weak for transmitting financial data. 128-bit encryption can be downloaded free from Netscape (for Netscape Navigator) or Microsoft (for Internet Explorer).
 - Follow this link to upgrade to 128-bit encryption on Microsoft Internet Explorer:
<http://www.microsoft.com/windows/ie/download/128bit/intro.htm>
 - Follow this link to upgrade to 128-bit encryption on Netscape Navigator:
<http://home.netscape.com/security>
3. In order to authenticate that the incoming data is from the correct source, ProPay generates a unique Certification String (certStr) and gives it to the XML user to be included with the XML data. This Certification String is the equivalent of a password. It is extremely important that it be kept confidential.

3.0 Set Up of the XML Authorization Interface

To take advantage of ProPay's XML Authorization Interface, follow these steps:

1. Complete and return your licensing agreement to ProPay.
2. Ensure that your browser supports 128-bit encryption (see above).
3. Send ProPay payment for the XML Authorization Interface.
4. ProPay will send you a unique certification string to use on ProPay's test server.
5. Set up your software to send and receive XML data.
6. Test the XML Authorization Interface by sending test transmissions to ProPay's test server.
7. ProPay will send you a unique certification string to use when sending real transactions to ProPay for processing. This certification information should be held confidential to prevent unauthorized access to financial transactions.

You are now ready to use ProPay's XML Interface.

4.0 XML Authorization Services

4.1 Credit Card Authorization Only

Use this service to authorize a credit card transaction, but not the transfer of the funds. A credit card authorization reserves the money on the credit card, but does not transfer it. Settling a credit card transaction actually transfers the money from the credit card to the receiving account.

According to Visa and MasterCard regulations, payment should not be settled more than 48 hours before the purchased product is shipped or service supplied.

If the transaction is not settled within 30 days, it is automatically voided. You can settle the transaction from the Authorization Only Reports page on the ProPay website, or send further XML data requesting settlement.

The following criteria apply to this service:

Requirements	Not applicable
Transaction Limits	\$1.00 Minimum Maximum is based on the reserve amount. See ProPay's website for information.
Rules	Transaction will be automatically voided if not settled within 30 days
Fees	Fees charged at settlement
Funds Available	After settlement

The following elements are sent and received for this service:

Request Data Sent	Transaction Data Sent	Data Returned
username certStr class	transType = 05 amount addr (optional) aptNum (optional) zip (optional) sourceEmail ccNum expDate CVV2 (optional) invNum (optional)	transType sourceEmail invNum status transNum authCode

Example of XML data sent to ProPay:

```
<?xml version="1.0"?>
<!DOCTYPE Request.dtd>
<XMLRequest>
  <username>xmlTesting@propay.com</username>
  <certStr>111XMLCertStr111</certStr>
  <class>partner</class>
  <XMLTrans>
    <transType>05</transType>
    <amount>9997</amount>
    <sourceEmail>jeff@propay.com</sourceEmail>
    <ccNum>1544444444443333</ccNum>
    <expDate>0102</expDate>
    <CVV2>312</CVV2>
    <invNum>G67613E</invNum>
  </XMLTrans>
</XMLRequest>
```

Example of XML data returned from ProPay:

```
<?xml version="1.0"?>
<!DOCTYPE Response.dtd>
<XMLResponse>
  <XMLTrans>
    <transType>05</transType>
    <sourceEmail>jeff@propay.com</sourceEmail>
    <invNum>G67613E</invNum>
    <status>67</status>
    <transNum>74</transNum>
    <authCode>712312022933</authCode>
  </XMLTrans>
</XMLResponse>
```

4.2 Credit Card Settlement

Use this service to settle a credit card transaction that was initiated by the Authorization Only service. This service can also be done online from the Authorization Only Reports page of the ProPay website.

The following criteria apply to this service:

Requirements	Previous authorization on this credit card
Transaction Limits	Not applicable
Rules	Not applicable
Fees	Usually 3.5% of the transaction plus \$.35
Funds Available	Two business days

The following elements are sent and received for this service:

Request Data Sent	Transaction Data Sent	Data Returned
username certStr class	transType = 06 sourceEmail transNum*	transType sourceEmail transNum status

*transNum is the value returned to you in the credit card authorization-only response.

Example of XML data sent to ProPay:

```
<?xml version= "1.0"?>
<!DOCTYPE Request.dtd>
<XML Request>
  <username>xmlTesting@propay.com</username>
  <certStr>111XMLCertStr111</certStr>
  <class>partner</class>
  <XMLTrans>
    <transType>06</transType>
    <sourceEmail>jeff@propay.com</sourceEmail>
    <transNum>477263</transNum>
  </XMLTrans>
</XMLRequest>
```

Example of XML data returned from ProPay:

```
<?xml version="1.0"?>
<!DOCTYPE Response.dtd>
<XMLResponse>
  <XMLTrans>
    <transType>06</transType>
    <sourceEmail>jeff@propay.com</sourceEmail>
    <transNum>477263</transNum>
    <status>57</status>
  </XMLTrans>
</XMLResponse>
```


4.3 Void Credit Card Authorization

Use this service to void a transaction that was initiated with the Authorization Only service but hasn't settled yet. This can also be done online from the Authorization Only Reports page of the ProPay website.

The following criteria apply to this service:

Requirements	Previous authorization on this credit card
Transaction Limits	Not applicable
Rules	Not applicable
Fees	No charge
Funds Available	Not applicable

The following elements are sent and received for this service:

Request Data Sent	Transaction Data Sent	Data Returned
username certStr class	transType = 07 sourceEmail transNum* amount	transType sourceEmail transNum status

*transNum is the value returned to you in the credit card authorization-only response.

Example of XML data sent to ProPay:

```
<?xml version= "1.0"?>
<!DOCTYPE Request.dtd>
<XML Request>
  <username>xmlTesting@propay.com</username>
  <certStr>111XMLCertStr111</certStr>
  <class>partner</class>
  <XMLTrans>
    <transType>07</transType>
    <sourceEmail>jeff@propay.com</sourceEmail>
    <transNum>453123</transNum>
    <amount>4000</amount>
  </XMLTrans>
</XMLRequest>
```

Example of XML data returned from ProPay:

```
<?xml version="1.0"?>
<!DOCTYPE Response.dtd>
<XMLResponse>
  <XMLTrans>
    <transType>07</transType>
    <sourceEmail>jeff@propay.com</sourceEmail>
    <transNum>453123</transNum>
    <status>00</status>
  </XMLTrans>
</XMLResponse>
```

5.0 Definition of Elements

5.1 Elements Sent to ProPay

Element	Description	Length	Valid Values
addr	street address	1-50	valid 'string' values
amount	amount of transaction in cents (Minimum \$1.00 for credit card transactions.)	>1	0-9 Must be >= 100 for credit card transactions
aptNum	apartment number	1-10	valid 'string' values
ccNum	credit card number	13-16 for VISA, 16 for MasterCard	0-9 For VISA, numbers beginning with 400000- 49999. For MasterCard, numbers beginning with 510000 - 559999.
certStr	certification string created by ProPay	6-19	valid 'string' values
class	type of user. Must be the string 'partner'. Do not substitute your company name.	7	partner
CVV2	credit card CVV2 number. This is the 3 digits that follow the credit card number on the back of the card.	3	0-9
expDate	credit card expiration date as MMY	4	0-9
invNum	invoice number. To identify this transaction for your own use. This value is passed back from ProPay without being used.	0-20	a-z,A-Z,0-9
sourceEmail	e-mail address	6-50	a-z,A-Z,0-9,-,.,@
transNum	transaction number Generated by Propay (passed back with the authorization response).	>1	valid 'integer' values
transType	transaction type	2	05 - Credit card authorization only 06 - Credit card settlement 07 - Void credit card

			authorization only
username	sign-in name of ProPay account holder	6-55	valid 'string' values
zip	zip code	5-10	-,0-9

5.2 Elements Returned from ProPay

Element	Description	Value
authCode	credit card authorization code	valid 'integer' values
invNum	invoice number. To identify this transaction for your own use. This is the same value passed to ProPay.	same as sent to ProPay
sourceEmail	e-mail address	same as sent to ProPay
status	success or failure status	00 - Success 20 - Invalid username 21 - Invalid transType 24 - Invalid sourceEmail 28 - Invalid addr 29 - Invalid aptNum 32 - Invalid zip 44 - Invalid amount 45 - Invalid invNum 48 - Invalid ccNum 49 - Invalid expDate 50 - Invalid cvv2 57 - Cannot settle transaction because it already expired 58 - Credit card declined 59 - Credit card not authenticated 60 - Credit card authorization timed out; retry at a later time 61 - Amount exceeds single transaction limit 62 - Amount exceeds monthly volume limit 63 - Insufficient funds in account 64 - Over credit card use limit 65 - Miscellaneous error 67 - Unauthorized service requested
transNum	transaction number generated by ProPay	valid 'integer' values
transType	transaction type	same as sent to ProPay

6.0 XML Schemas and DTDs

6.1 Data Type Definitions

```
<?xml version="1.0"?>
<!-- edited with XML Spy v3.5 NT (http://www.xmlspy.com) by Wes Smith
(ProPay) -->
<PPD:schema xmlns:PPD="http://www.w3.org/2000/10/XMLSchema"
elementFormDefault="unqualified" attributeFormDefault="unqualified"
version="1.0">
  <PPD:simpleType name="requestTypes">
    <PPD:annotation>
      <PPD:documentation>one of the following transaction
types must be used for each transaction.
      </PPD:documentation>
    </PPD:annotation>
    <PPD:restriction base="PPD:string">
      <PPD:enumeration value="01"/>
      <PPD:enumeration value="02"/>
      <PPD:enumeration value="03"/>
      <PPD:enumeration value="04"/>
      <PPD:enumeration value="05"/>
      <PPD:enumeration value="06"/>
      <PPD:enumeration value="07"/>
      <PPD:enumeration value="08"/>
      <PPD:enumeration value="09"/>
      <PPD:enumeration value="10"/>
    </PPD:restriction>
  </PPD:simpleType>
  <PPD:simpleType name="responseTypes">
    <PPD:annotation>
      <PPD:documentation>one of the following response
codes will be sent for each transaction. will change as needed.
      </PPD:documentation>
    </PPD:annotation>
    <PPD:restriction base="PPD:string">
      <PPD:enumeration value="00">
        <PPD:annotation>
          <PPD:documentation>success</PPD:documentation>
        </PPD:annotation>
      </PPD:enumeration>
      <PPD:enumeration value="00">
        <PPD:annotation>
          <PPD:documentation>success</PPD:documentation>
        </PPD:annotation>
      </PPD:enumeration>
      <PPD:enumeration value="20">
        <PPD:annotation>
          <PPD:documentation>invalid
username</PPD:documentation>
        </PPD:annotation>
      </PPD:enumeration>
      <PPD:enumeration value="21">
        <PPD:annotation>
```

```

                                <PPD:documentation>invalid
transType</PPD:documentation>
                                </PPD:annotation>
                                </PPD:enumeration>
                                <PPD:enumeration value="23">
                                <PPD:annotation>
                                <PPD:documentation>invalid
accountType</PPD:documentation>
                                </PPD:annotation>
                                </PPD:enumeration>
                                <PPD:enumeration value="24">
                                <PPD:annotation>
                                <PPD:documentation>invalid e-
mail</PPD:documentation>
                                </PPD:annotation>
                                </PPD:enumeration>
                                <PPD:enumeration value="25">
                                <PPD:annotation>
                                <PPD:documentation>invalid
firstName</PPD:documentation>
                                </PPD:annotation>
                                </PPD:enumeration>
                                <PPD:enumeration value="26">
                                <PPD:annotation>
                                <PPD:documentation>invalid
mInitial</PPD:documentation>
                                </PPD:annotation>
                                </PPD:enumeration>
                                <PPD:enumeration value="27">
                                <PPD:annotation>
                                <PPD:documentation>invalid
lastName</PPD:documentation>
                                </PPD:annotation>
                                </PPD:enumeration>
                                <PPD:enumeration value="28">
                                <PPD:annotation>
                                <PPD:documentation>invalid
addr</PPD:documentation>
                                </PPD:annotation>
                                </PPD:enumeration>
                                <PPD:enumeration value="29">
                                <PPD:annotation>
                                <PPD:documentation>invalid
aptNum</PPD:documentation>
                                </PPD:annotation>
                                </PPD:enumeration>
                                <PPD:enumeration value="30">
                                <PPD:annotation>
                                <PPD:documentation>invalid
city</PPD:documentation>
                                </PPD:annotation>
                                </PPD:enumeration>
                                <PPD:enumeration value="31">
                                <PPD:annotation>
                                <PPD:documentation>invalid
state</PPD:documentation>
                                </PPD:annotation>

```

```

        </PPD:enumeration>
        <PPD:enumeration value="32">
            <PPD:annotation>
                <PPD:documentation>invalid
zip</PPD:documentation>
            </PPD:annotation>
        </PPD:enumeration>
        <PPD:enumeration value="33">
            <PPD:annotation>
                <PPD:documentation>invalid
mailAddr</PPD:documentation>
            </PPD:annotation>
        </PPD:enumeration>
        <PPD:enumeration value="34">
            <PPD:annotation>
                <PPD:documentation>invalid
mailApt</PPD:documentation>
            </PPD:annotation>
        </PPD:enumeration>
        <PPD:enumeration value="35">
            <PPD:annotation>
                <PPD:documentation>invalid
mailCity</PPD:documentation>
            </PPD:annotation>
        </PPD:enumeration>
        <PPD:enumeration value="36">
            <PPD:annotation>
                <PPD:documentation>invalid
mailState</PPD:documentation>
            </PPD:annotation>
        </PPD:enumeration>
        <PPD:enumeration value="37">
            <PPD:annotation>
                <PPD:documentation>invalid
mailZip</PPD:documentation>
            </PPD:annotation>
        </PPD:enumeration>
        <PPD:enumeration value="38">
            <PPD:annotation>
                <PPD:documentation>invalid
dayPhone</PPD:documentation>
            </PPD:annotation>
        </PPD:enumeration>
        <PPD:enumeration value="39">
            <PPD:annotation>
                <PPD:documentation>invalid
evenPhone</PPD:documentation>
            </PPD:annotation>
        </PPD:enumeration>
        <PPD:enumeration value="40">
            <PPD:annotation>
                <PPD:documentation>invalid
ssn</PPD:documentation>
            </PPD:annotation>
        </PPD:enumeration>
        <PPD:enumeration value="41">
            <PPD:annotation>

```

```

                                <PPD:documentation>invalid
dob</PPD:documentation>
                                </PPD:annotation>
                                </PPD:enumeration>
                                <PPD:enumeration value="42">
                                <PPD:annotation>
                                <PPD:documentation>invalid
recEmail</PPD:documentation>
                                </PPD:annotation>
                                </PPD:enumeration>
                                <PPD:enumeration value="43">
                                <PPD:annotation>
                                <PPD:documentation>invalid
knownAccount</PPD:documentation>
                                </PPD:annotation>
                                </PPD:enumeration>
                                <PPD:enumeration value="44">
                                <PPD:annotation>
                                <PPD:documentation>invalid
amount</PPD:documentation>
                                </PPD:annotation>
                                </PPD:enumeration>
                                <PPD:enumeration value="45">
                                <PPD:annotation>
                                <PPD:documentation>invalid
state</PPD:documentation>
                                </PPD:annotation>
                                </PPD:enumeration>
                                <PPD:enumeration value="46">
                                <PPD:annotation>
                                <PPD:documentation>invalid
invNum</PPD:documentation>
                                </PPD:annotation>
                                </PPD:enumeration>
                                <PPD:enumeration value="47">
                                <PPD:annotation>
                                <PPD:documentation>invalid
rtNum</PPD:documentation>
                                </PPD:annotation>
                                </PPD:enumeration>
                                <PPD:enumeration value="48">
                                <PPD:annotation>
                                <PPD:documentation>invalid
acctNum</PPD:documentation>
                                </PPD:annotation>
                                </PPD:enumeration>
                                <PPD:enumeration value="49">
                                <PPD:annotation>
                                <PPD:documentation>invalid
expData</PPD:documentation>
                                </PPD:annotation>
                                </PPD:enumeration>
                                <PPD:enumeration value="50">
                                <PPD:annotation>
                                <PPD:documentation>invalid
CVV2</PPD:documentation>
                                </PPD:annotation>

```



```

        </PPD:enumeration>
        <PPD:enumeration value="51">
            <PPD:annotation>
                <PPD:documentation>invalid
transNum</PPD:documentation>
            </PPD:annotation>
        </PPD:enumeration>
        <PPD:enumeration value="52">
            <PPD:annotation>
                <PPD:documentation>invalid
splitNum</PPD:documentation>
            </PPD:annotation>
        </PPD:enumeration>
        <PPD:enumeration value="53">
            <PPD:annotation>
                <PPD:documentation>a ProPay account for
this e-mail address already exists
            </PPD:documentation>
        </PPD:annotation>
        </PPD:enumeration>
        <PPD:enumeration value="54">
            <PPD:annotation>
                <PPD:documentation>a ProPay account for
this ssn already exists
            </PPD:documentation>
        </PPD:annotation>
        </PPD:enumeration>
        <PPD:enumeration value="55">
            <PPD:annotation>
                <PPD:documentation>Recipient e-mail
address doesn't exist</PPD:documentation>
            </PPD:annotation>
        </PPD:enumeration>
        <PPD:enumeration value="56">
            <PPD:annotation>
                <PPD:documentation>Recipient e-mail
address already exists</PPD:documentation>
            </PPD:annotation>
        </PPD:enumeration>
        <PPD:enumeration value="57">
            <PPD:annotation>
                <PPD:documentation>Cannot settle
transaction because it already expired
            </PPD:documentation>
        </PPD:annotation>
        </PPD:enumeration>
        <PPD:enumeration value="58">
            <PPD:annotation>
                <PPD:documentation>Credit card
declined</PPD:documentation>
            </PPD:annotation>
        </PPD:enumeration>
        <PPD:enumeration value="59">
            <PPD:annotation>
                <PPD:documentation>Not
authenticated</PPD:documentation>
            </PPD:annotation>

```

```

        </PPD:enumeration>
        <PPD:enumeration value="60">
            <PPD:annotation>
                <PPD:documentation>Auth timed
out</PPD:documentation>
            </PPD:annotation>
        </PPD:enumeration>
        <PPD:enumeration value="61">
            <PPD:annotation>
                <PPD:documentation>over transaction
limit</PPD:documentation>
            </PPD:annotation>
        </PPD:enumeration>
        <PPD:enumeration value="62">
            <PPD:annotation>
                <PPD:documentation>over monthly
volume</PPD:documentation>
            </PPD:annotation>
        </PPD:enumeration>
        <PPD:enumeration value="63">
            <PPD:annotation>
                <PPD:documentation>Over
available</PPD:documentation>
            </PPD:annotation>
        </PPD:enumeration>
        <PPD:enumeration value="64">
            <PPD:annotation>
                <PPD:documentation>over credit card
use</PPD:documentation>
            </PPD:annotation>
        </PPD:enumeration>
        <PPD:enumeration value="65">
            <PPD:annotation>
                <PPD:documentation>misc
error</PPD:documentation>
            </PPD:annotation>
        </PPD:enumeration>
        <PPD:enumeration value="66">
            <PPD:annotation>
                <PPD:documentation>denied a propay
account</PPD:documentation>
            </PPD:annotation>
        </PPD:enumeration>
        <PPD:enumeration value="67">
            <PPD:annotation>
                <PPD:documentation>unauthorized service
requested</PPD:documentation>
            </PPD:annotation>
        </PPD:enumeration>
    </PPD:restriction>
</PPD:simpleType>
<PPD:simpleType name="classTypes">
    <PPD:restriction base="PPD:string">
        <PPD:enumeration value="palm"/>
        <PPD:enumeration value="partner"/>
    </PPD:restriction>
</PPD:simpleType>

```

```

<PPD:simpleType name="propay">
  <PPD:restriction base="PPD:string">
    <PPD:enumeration value="yes"/>
    <PPD:enumeration value="no"/>
    <PPD:enumeration value="unknown"/>
  </PPD:restriction>
</PPD:simpleType>
<PPD:simpleType name="email">
  <PPD:restriction base="account">
    <PPD:minLength value="6"/>
    <PPD:maxLength value="50"/>
    <PPD:pattern value="/^.+@([\?][a-zA-Z0-9\-\.\.]+\.[a-zA-Z]{2,3}|[0-9]{1,3})[\?)]$/"/>
  </PPD:restriction>
</PPD:simpleType>
<PPD:simpleType name="fName">
  <PPD:restriction base="PPD:string">
    <PPD:maxLength value="20"/>
    <PPD:pattern value="[-a-zA-Z\s\x27]"/>
  </PPD:restriction>
</PPD:simpleType>
<PPD:simpleType name="initial">
  <PPD:restriction base="PPD:string">
    <PPD:maxLength value="1"/>
    <PPD:pattern value="[a-zA-Z]"/>
  </PPD:restriction>
</PPD:simpleType>
<PPD:simpleType name="lName">
  <PPD:restriction base="PPD:string">
    <PPD:maxLength value="25"/>
    <PPD:pattern value="[-a-zA-Z\s\x27]"/>
  </PPD:restriction>
</PPD:simpleType>
<PPD:simpleType name="address">
  <PPD:restriction base="PPD:string">
    <PPD:maxLength value="50"/>
  </PPD:restriction>
</PPD:simpleType>
<PPD:simpleType name="apartment">
  <PPD:restriction base="PPD:string">
    <PPD:maxLength value="10"/>
  </PPD:restriction>
</PPD:simpleType>
<PPD:simpleType name="city">
  <PPD:restriction base="PPD:string">
    <PPD:minLength value="2"/>
    <PPD:maxLength value="20"/>
    <PPD:pattern value="[a-zA-Z\s]"/>
  </PPD:restriction>
</PPD:simpleType>
<PPD:simpleType name="states">
  <PPD:restriction base="PPD:string">
    <PPD:enumeration value="AL"/>
    <PPD:enumeration value="AK"/>
    <PPD:enumeration value="AR"/>
    <PPD:enumeration value="AZ"/>
    <PPD:enumeration value="CA"/>
  </PPD:restriction>

```

```

    <PPD:enumeration value="CO"/>
    <PPD:enumeration value="CT"/>
    <PPD:enumeration value="DC"/>
    <PPD:enumeration value="DE"/>
    <PPD:enumeration value="FL"/>
    <PPD:enumeration value="GA"/>
    <PPD:enumeration value="HA"/>
    <PPD:enumeration value="IA"/>
    <PPD:enumeration value="ID"/>
    <PPD:enumeration value="IL"/>
    <PPD:enumeration value="IN"/>
    <PPD:enumeration value="KS"/>
    <PPD:enumeration value="KY"/>
    <PPD:enumeration value="LA"/>
    <PPD:enumeration value="MA"/>
    <PPD:enumeration value="MD"/>
    <PPD:enumeration value="ME"/>
    <PPD:enumeration value="MI"/>
    <PPD:enumeration value="MN"/>
    <PPD:enumeration value="MO"/>
    <PPD:enumeration value="MS"/>
    <PPD:enumeration value="MT"/>
    <PPD:enumeration value="NC"/>
    <PPD:enumeration value="ND"/>
    <PPD:enumeration value="NE"/>
    <PPD:enumeration value="NH"/>
    <PPD:enumeration value="NJ"/>
    <PPD:enumeration value="NM"/>
    <PPD:enumeration value="NV"/>
    <PPD:enumeration value="NY"/>
    <PPD:enumeration value="OH"/>
    <PPD:enumeration value="OK"/>
    <PPD:enumeration value="OR"/>
    <PPD:enumeration value="PA"/>
    <PPD:enumeration value="PR"/>
    <PPD:enumeration value="RI"/>
    <PPD:enumeration value="SC"/>
    <PPD:enumeration value="SD"/>
    <PPD:enumeration value="TN"/>
    <PPD:enumeration value="TX"/>
    <PPD:enumeration value="UT"/>
    <PPD:enumeration value="VA"/>
    <PPD:enumeration value="VT"/>
    <PPD:enumeration value="WA"/>
    <PPD:enumeration value="WI"/>
    <PPD:enumeration value="WV"/>
    <PPD:enumeration value="WY"/>
  </PPD:restriction>
</PPD:simpleType>
<PPD:simpleType name="zip">
  <PPD:restriction base="PPD:string">
    <PPD:minLength value="5"/>
    <PPD:maxLength value="10"/>
    <PPD:pattern value="[-\d]"/>
  </PPD:restriction>
</PPD:simpleType>
<PPD:simpleType name="phone">

```

```

        <PPD:restriction base="PPD:string">
            <PPD:length value="10" fixed="true"/>
            <PPD:pattern value="\d"/>
        </PPD:restriction>
    </PPD:simpleType>
    <PPD:simpleType name="ssn">
        <PPD:restriction base="PPD:string">
            <PPD:length value="9" fixed="true"/>
            <PPD:pattern value="\d"/>
        </PPD:restriction>
    </PPD:simpleType>
    <PPD:simpleType name="dob">
        <PPD:annotation>
            <PPD:documentation>
                must be of the form MM-DD-YYYY
            </PPD:documentation>
        </PPD:annotation>
        <PPD:restriction base="PPD:string">
            <PPD:pattern value="[-\d]"/>
        </PPD:restriction>
    </PPD:simpleType>
    <PPD:simpleType name="amount">
        <PPD:restriction base="PPD:integer">
            <PPD:minInclusive value="100"/>
        </PPD:restriction>
    </PPD:simpleType>
    <PPD:simpleType name="invoice">
        <PPD:restriction base="PPD:string">
            <PPD:minLength value="0"/>
            <PPD:maxLength value="20"/>
            <PPD:pattern value="[a-zA-Z\d]"/>
        </PPD:restriction>
    </PPD:simpleType>
    <PPD:simpleType name="routing">
        <PPD:restriction base="PPD:string">
            <PPD:length value="9" fixed="true"/>
            <PPD:pattern value="\d"/>
        </PPD:restriction>
    </PPD:simpleType>
    <PPD:simpleType name="account">
        <PPD:restriction base="PPD:string">
            <PPD:maxLength value="17"/>
            <PPD:pattern value="\d"/>
        </PPD:restriction>
    </PPD:simpleType>
    <PPD:simpleType name="creditCard">
        <PPD:restriction base="PPD:string">
            <PPD:minLength value="13"/>
            <PPD:maxLength value="16"/>
            <PPD:pattern value="\d"/>
        </PPD:restriction>
    </PPD:simpleType>
    <PPD:simpleType name="expiration">
        <PPD:annotation>
            <PPD:documentation>
                must be in the of the form MMY
            </PPD:documentation>
        </PPD:annotation>
    </PPD:simpleType>

```

```

        </PPD:annotation>
        <PPD:restriction base="PPD:string">
            <PPD:pattern value="[\d]"/>
        </PPD:restriction>
    </PPD:simpleType>
    <PPD:simpleType name="cvv2">
        <PPD:restriction base="PPD:string">
            <PPD:length value="3" fixed="true"/>
            <PPD:pattern value="[\d]"/>
        </PPD:restriction>
    </PPD:simpleType>
    <PPD:simpleType name="user">
        <PPD:restriction base="PPD:string">
            <PPD:minLength value="6"/>
            <PPD:maxLength value="55"/>
        </PPD:restriction>
    </PPD:simpleType>
    <PPD:simpleType name="cert">
        <PPD:restriction base="PPD:string">
            <PPD:minLength value="6"/>
            <PPD:maxLength value="19"/>
        </PPD:restriction>
    </PPD:simpleType>
</PPD:schema>

```

6.2 Transaction Requests

```

<?xml version="1.0"?>
<!-- edited with XML Spy v3.5 NT (http://www.xmlspy.com) by Wes Smith
(ProPay) -->
<req:schema xmlns:req="http://www.w3.org/2000/10/XMLSchema">
    <req:include
schemaLocation="http://www.propay.com/xml/PPDataTypes.xsd"/>
    <req:element name="username" type="user">
        <req:annotation>
            <req:documentation>username
            </req:documentation>
        </req:annotation>
    </req:element>
    <req:element name="certStr" type="cert">
        <req:annotation>
            <req:documentation>certification string, provided by
ProPay
            </req:documentation>
        </req:annotation>
    </req:element>
    <req:element name="class" type="classTypes">
        <req:annotation>
            <req:documentation>user class (palm/partner)
            </req:documentation>
        </req:annotation>
    </req:element>
    <req:element name="transType" type="requestTypes"/>
    <req:element name="sourceEmail" type="email">

```

```

        <req:annotation>
            <req:documentation>email address of account for which
transaction is being processed e-mail address of initiating account, or
e-mail for new signup
            </req:documentation>
        </req:annotation>
    </req:element>
    <req:element name="firstName" type="fName"/>
    <req:element name="mInitial" type="initial"/>
    <req:element name="lastName" type="lName"/>
    <req:element name="addr" type="address">
        <req:annotation>
            <req:documentation>billing address for credit card
owner, or street address for new signup
        </req:documentation>
        </req:annotation>
    </req:element>
    <req:element name="aptNum" type="apartment"/>
    <req:element name="city" type="city"/>
    <req:element name="state" type="states"/>
    <req:element name="zip" type="zip"/>
    <req:element name="mailAddr" type="address">
        <req:annotation>
            <req:documentation>mailing address for new signup;
optional if the same as street address for new signup
        </req:documentation>
        </req:annotation>
    </req:element>
    <req:element name="mailApt" type="apartment"/>
    <req:element name="mailCity" type="city"/>
    <req:element name="mailState" type="states"/>
    <req:element name="mailZip" type="zip"/>
    <req:element name="dayPhone" type="phone"/>
    <req:element name="evenPhone" type="phone"/>
    <req:element name="ssn" type="ssn"/>
    <req:element name="dob" type="dob"/>
    <req:element name="amount" type="amount">
        <req:annotation>
            <req:documentation>amount of money to be transferred
        </req:documentation>
        </req:annotation>
    </req:element>
    <req:element name="recEmail" type="email">
        <req:annotation>
            <req:documentation> e-mail address for destination of
propay to propay transaction
        </req:documentation>
        </req:annotation>
    </req:element>
    <req:element name="knownAccount" type="propay">
        <req:annotation>
            <req:documentation> designation of a propay account
for recEmail
        </req:documentation>
        </req:annotation>
    </req:element>
    <req:element name="invNum" type="invoice">
        <req:annotation>

```

```

        <req:documentation> invoice number (identifier for
customer use created by customer for his own use)
        </req:documentation>
    </req:annotation>
</req:element>
<req:element name="rtNum" type="routing">
    <req:annotation>
        <req:documentation> routing number for checking
account
        </req:documentation>
    </req:annotation>
</req:element>
<req:element name="acctNum" type="account">
    <req:annotation>
        <req:documentation> checking account number
        </req:documentation>
    </req:annotation>
</req:element>
<req:element name="ccNum" type="creditCard">
    <req:annotation>
        <req:documentation> credit card number, only VISA
cards, with numbers beginning from 400000- 49999 and either 13-16
numbers long, or MasterCard with numbers beginning from 510000 - 559999
and 16 numbers will be accepted
        </req:documentation>
    </req:annotation>
</req:element>
<req:element name="expDate" type="expiration">
    <req:annotation>
        <req:documentation> credit card expiration date
        </req:documentation>
    </req:annotation>
</req:element>
<req:element name="CVV2" type="cvv2">
    <req:annotation>
        <req:documentation>credit card validation number
usually the last 3 digits on the back of the credit card following the
credit card number
        </req:documentation>
    </req:annotation>
</req:element>
<req:element name="splitNum" type="req:integer">
    <req:annotation>
        <req:documentation>identifier for alternate
fee/distribution(coordinate with ProPay)
        </req:documentation>
    </req:annotation>
</req:element>
<req:element name="transNum" type="req:integer">
    <req:annotation>
        <req:documentation> transaction number assigned by
ProPay for processing account
        </req:documentation>
    </req:annotation>
</req:element>
<req:group name="signup">
    <req:annotation>

```



```

        <req:documentation>create new ProPay
account(transType 01)
        </req:documentation>
    </req:annotation>
    <req:sequence>
        <req:element ref="transType"/>
        <req:element ref="sourceEmail"/>
        <req:element ref="firstName"/>
        <req:element ref="mInitial" minOccurs="0"/>
        <req:element ref="lastName"/>
        <req:element ref="addr"/>
        <req:element ref="aptNum" minOccurs="0"/>
        <req:element ref="city"/>
        <req:element ref="state"/>
        <req:element ref="zip"/>
        <req:element ref="mailAddr" minOccurs="0"/>
        <req:element ref="mailApt" minOccurs="0"/>
        <req:element ref="mailCity" minOccurs="0"/>
        <req:element ref="mailState" minOccurs="0"/>
        <req:element ref="mailZip" minOccurs="0"/>
        <req:element ref="dayPhone"/>
        <req:element ref="evenPhone"/>
        <req:element ref="ssn"/>
        <req:element ref="dob"/>
    </req:sequence>
</req:group>
<req:group name="PPToPP">
    <req:annotation>
        <req:documentation>propay to propay transactionPay to
ProPay Account(transType 02)
        </req:documentation>
    </req:annotation>
    <req:sequence>
        <req:element ref="transType"/>
        <req:element ref="amount"/>
        <req:element ref="sourceEmail"/>
        <req:element ref="recEmail"/>
        <req:element ref="knownAccount"/>
        <req:element ref="invNum" minOccurs="0"/>
    </req:sequence>
</req:group>
<req:group name="ACH">
    <req:annotation>
        <req:documentation>Propay account to Checking
accountPay to a Bank Account(transType 03)
        </req:documentation>
    </req:annotation>
    <req:sequence>
        <req:element ref="transType"/>
        <req:element ref="amount"/>
        <req:element ref="firstName"/>
        <req:element ref="lastName"/>
        <req:element ref="rtNum"/>
        <req:element ref="acctNum"/>
        <req:element ref="invNum" minOccurs="0"/>
    </req:sequence>
</req:group>

```

```

    <req:group name="CC">
      <req:annotation>
        <req:documentation> credit card processing, auth-
only, credit card splitPerform a credit card transaction, authorization
only, or credit card split(transType 04, 05, 08, 09)
        </req:documentation>
      </req:annotation>
      <req:sequence>
        <req:element ref="transType"/>
        <req:element ref="amount"/>
        <req:element ref="addr" minOccurs="0"/>
        <req:element ref="aptNum" minOccurs="0"/>
        <req:element ref="zip" minOccurs="0"/>
        <req:element ref="sourceEmail"/>
        <req:element ref="ccNum"/>
        <req:element ref="expDate"/>
        <req:element ref="CVV2" minOccurs="0"/>
        <req:element ref="invNum" minOccurs="0"/>
        <req:element ref="splitNum" minOccurs="0"/>
      </req:sequence>
    </req:group>
    <req:group name="CCAdmin">
      <req:annotation>
        <req:documentation> settlement/voidSettle or void a
credit card transaction(transType 06, 07)
        </req:documentation>
      </req:annotation>
      <req:sequence>
        <req:element ref="transType"/>
        <req:element ref="amount" minOccurs="0"/>
        <req:element ref="sourceEmail"/>
        <req:element ref="transNum"/>
      </req:sequence>
    </req:group>
    <req:element name="XMLRequest">
      <req:complexType>
        <req:sequence>
          <req:element ref="username"/>
          <req:element ref="certStr"/>
          <req:element ref="class"/>
          <req:element name="XMLTrans">
            <req:complexType>
              <req:choice>
                <req:group ref="signup"/>
                <req:group ref="PPToPP"/>
                <req:group ref="ACH"/>
                <req:group ref="CC"/>
                <req:group ref="CCAdmin"/>
                <req:element
ref="transType"/>
              </req:choice>
            </req:complexType>
          </req:element>
        </req:sequence>
      </req:complexType>
    </req:element>
  </req:schema>

```

6.3 Responses

```
<?xml version="1.0"?>
<!-- edited with XML Spy v3.5 NT (http://www.xmlspy.com) by Wes Smith
(ProPay) -->
<res:schema xmlns:res="http://www.w3.org/2000/10/XMLSchema">
  <res:include
schemaLocation="http://www.propay.com/xml/PPDataTypes.xsd"/>
  <res:element name="transType" type="requestTypes"/>
  <res:element name="status" type="responseTypes"/>
  <res:element name="sourceEmail" type="email">
    <res:annotation>
      <res:documentation>email address for new signup
      </res:documentation>
    </res:annotation>
  </res:element>
  <res:element name="invNum" type="invoice">
    <res:annotation>
      <res:documentation> invoice number (identifier for
customer use)
      </res:documentation>
    </res:annotation>
  </res:element>
  <res:element name="transNum" type="res:integer">
    <res:annotation>
      <res:documentation> transaction number for processing
account
      </res:documentation>
    </res:annotation>
  </res:element>
  <res:element name="authCode" type="res:integer">
    <res:annotation>
      <res:documentation>authorization code for cc trans
      </res:documentation>
    </res:annotation>
  </res:element>
  <res:group name="signup">
    <res:annotation>
      <res:documentation>response for signup request
      </res:documentation>
    </res:annotation>
    <res:sequence>
      <res:element ref="transType"/>
      <res:element ref="sourceEmail"/>
      <res:element ref="status"/>
    </res:sequence>
  </res:group>
  <res:group name="admin">
    <res:annotation>
      <res:documentation>response from palm initiation
      </res:documentation>
    </res:annotation>
  </res:group>
</res:schema>
</res:sequence>
```

```

        <res:element ref="status"/>
    </res:sequence>
</res:group>
<res:group name="CC">
    <res:annotation>
        <res:documentation>response for credit card
transaction
    </res:documentation>
    </res:annotation>
    <res:sequence>
        <res:element ref="transType"/>
        <res:element ref="sourceEmail"/>
        <res:element ref="invNum"/>
        <res:element ref="status"/>
        <res:element ref="transNum"/>
        <res:element ref="authCode"/>
    </res:sequence>
</res:group>
<res:group name="trans">
    <res:annotation>
        <res:documentation>response for non-cc
transaction(PPToPP, ACH)
    </res:documentation>
    </res:annotation>
    <res:sequence>
        <res:element ref="transType"/>
        <res:element ref="invNum"/>
        <res:element ref="status"/>
        <res:element ref="transNum"/>
    </res:sequence>
</res:group>
<res:group name="CCAdmin">
    <res:annotation>
        <res:documentation>response settle/void
    </res:documentation>
    </res:annotation>
    <res:sequence>
        <res:element ref="transType"/>
        <res:element ref="sourceEmail"/>
        <res:element ref="status"/>
        <res:element ref="transNum"/>
    </res:sequence>
</res:group>
<res:element name="XMLRequest">
    <res:complexType>
        <res:choice>
            <res:group ref="signup"/>
            <res:group ref="trans"/>
            <res:group ref="admin"/>
            <res:group ref="CC"/>
            <res:group ref="CCAdmin"/>
        </res:choice>
    </res:complexType>
</res:element>
</res:schema>

```

