



---

ELEMENT PAYMENT SERVICES, INC.

---

PAYMENT ACCOUNT SECURE STORAGE

THE PAYMENT CARD INDUSTRY (PCI) DATA

SECURITY STANDARD (DSS) SOLUTION FOR

PROTECTING CARDHOLDER DATA

*October 2006*

---

WHITE PAPER

---

**Confidential & Proprietary**

If you, as the reader of this document, are not the intended recipient, please be advised that any use, dissemination or copying of the concepts or contents of this document is respectfully prohibited. If you received this document in error please notify the sender immediately; thank you.

# Table of Contents

<b>1</b>	<b>INTRODUCTION.....</b>	<b>3</b>
<b>2</b>	<b>PURPOSE.....</b>	<b>5</b>
<b>3</b>	<b>COMPANY BACKGROUND .....</b>	<b>6</b>
<b>4</b>	<b>PROBLEM DEFINITION .....</b>	<b>7</b>
4.1	HOW COMPLIANCE WORKS.....	8
4.2	COMPLIANCE VALIDATION .....	9
4.3	WHY COMPLY? .....	10
<b>5</b>	<b>PASS TECHNOLOGY.....</b>	<b>11</b>
5.1	“HOW PASS WORKS” .....	12
5.1.1	<i>Step 1 – Collect Cardholder data .....</i>	<i>13</i>
5.1.2	<i>Step 2 - Securely Transmit Cardholder Data .....</i>	<i>13</i>
5.1.3	<i>Step 3 - Receive Reference Pointer to Cardholder Data .....</i>	<i>13</i>
5.1.4	<i>Step 4 - Store Reference Pointer In Place of Cardholder Data.....</i>	<i>13</i>
5.1.5	<i>Step 5 - Use Reference Pointer for Future Payment Transactions .....</i>	<i>13</i>
<b>6</b>	<b>SUMMARY .....</b>	<b>14</b>
<b>7</b>	<b>APPENDIX.....</b>	<b>15</b>
7.1	DETAILED DESCRIPTION OF PASS .....	15
7.2	METHODS.....	16
7.2.1	<i>PaymentAccountCreate .....</i>	<i>16</i>
7.2.2	<i>PaymentAccountUpdate .....</i>	<i>19</i>
7.2.3	<i>PaymentAccountDelete.....</i>	<i>22</i>
7.2.4	<i>PaymentAccountQuery .....</i>	<i>25</i>
7.3	DESIGN DIAGRAM .....	28
7.4	CARDHOLDER AND SENSITIVE AUTHENTICATION DATA .....	29

---

## 1 Introduction

---

Many businesses store sensitive customer information including, name, address, social security number, and credit card data. With the advent of the Internet, this information has become increasingly more vulnerable to hackers and fraudsters. In recent years, there has been a dramatic increase in the number of security compromises. In many cases, these incidents have resulted in large scale identity and credit card theft.

In an attempt to reduce the rise in security breaches, governments, special interest groups, and credit card companies have introduced laws and regulations to help govern the storage of sensitive data. Unfortunately, the compliance burden of these laws and regulations typically falls squarely on to the business and merchant community. Further more, to comply with the new regulations and protect themselves against embarrassing data compromises, businesses and merchants are often required to invest an inordinate amount of time and money in everything from high-tech tools to specialized staff.

The following is a small sample of recent regulations aimed at securing data storage:

- California issued state legislation through Senate Bill 1386 requiring all compromised business to notify their customers if they believe a compromise of any kind has happened; 31 states have followed suit.
- Gramm-Leach Bliley Act (GLBA) of 1999 requires all financial institutions (this includes all companies providing any kind of financial product or service for customers) to ensure confidentiality of customer records and information.
- Health Insurance Portability and Accountability Act (HIPAA) of 1996 was designed to require secure storage of patient information including all financial (billing) information.
- The Federal Information Security Management Act is designed to defend against mass attacks on a grand scale of information. For example, a compromise of financial information the size of Card Systems, Inc. (40 million credit card numbers) would be addressed at the federal level due to the nature of the compromise.

Ubiquitous breach of credit card data has been the primary catalyst for the creation of the Payment Card Industry (PCI) Data Security Standard (DSS). The PCI DSS; managed by Visa, MasterCard, American Express, Discover and JCB (Japan Credit Bureau), is designed to increase the level of security related to storing, transmitting, and/or processing <sup>1</sup>cardholder data.

The PCI DSS encompasses the requirements of the aforementioned Acts and Bills as well as address very specific cardholder data information security issues. Compliance with PCI DSS is now mandatory for all merchants and service providers that store, process,

and/or transmit cardholder data. However, the time and financial expense related to compliance with PCI DSS is significant.

Element Payment Services, Inc. has developed a unique new technology called Payment Account Secure Storage (PASS). PASS is designed to allow merchants and payment service providers alike the ability to easily comply with PCI DSS with very little time, effort, or financial impact. In addition, PASS reduces an enormous financial risk inherent with storing, processing, and/or transmitting cardholder data.

<sup>1</sup> For a detailed description of Cardholder data refer to the Appendix section [7.2 “Cardholder and Sensitive Authentication Data”](#)

---

## 2 Purpose

---

The purpose of this document is to describe Element Payment Services', Inc. new cost effective PCI DSS solution for protecting sensitive cardholder data. Likewise, how your organization can reduce a substantial financial risk. This document intends to answer the question, "How does my organization become and stay PCI DSS compliant without going bankrupt in the process?" The expected audience is IT/Business decision makers whose organization store, process, and/or transmit cardholder data.

---

### **3 Company Background**

---

Element Payment Services, Inc. is a registered Merchant Service Provider that provides credit, debit, check conversion and guarantee, EBT, and gift / loyalty card solutions to restaurant, hotel, and retail merchants throughout the United States. Incorporated in June of 2003, Element has emerged as a true innovator in the payment processing industry. Determined to set an industry precedent by offering solutions custom-tailored to fit the needs of each merchant without compromising customer care, Element was founded on a platform of these core values: Integrity, Communication, Innovation, Partnership, and Service. Today, Element services more than 30,000 merchants and 40 million transactions each year, and employs more than 60 employees in marketing, information technology, customer support, operations, administration, and management at their Phoenix-based headquarters.

---

## 4 Problem Definition

---

PCI DSS is a comprehensive set of security requirements aimed at limiting the risk exposure related to the storing, processing, and/or transmitting cardholder data.

PCI DSS, which is constantly changing to keep up with new vulnerabilities and risks, is the credit card industry's way of reducing the number of credit card data compromises.

The following are some examples of recent highly publicized data security compromises:

- CardSystems Solutions, Inc. (2005) – over 40 million card members compromised
- DSW Shoes – over 1 million card numbers compromised.
- Polo Ralph Lauren Corp – Compromise of internal servers led to fraud losses in excess of \$1.6 million.
- BJ's Wholesale Club – an unknown number of card numbers were stolen and then sold over the Internet.
- Chipotle Mexican Grill – Internal servers were hacked releasing credit card numbers to fraudsters resulting in excess of \$700,000.00 in loss.

A common misconception is that Internet/e-commerce merchants are largely responsible for the rise in data theft. Actually, according to Visa; four out of five compromises of credit cards occur in the retail (face-to-face) environment. Auditors from Ambiron-Trustwave, a leading information security and compliance company, have found that card data stored on merchants' servers have some of the highest risk. The threat of a data security breach is real and can inflict serious damage to any size organization.

---

## 4.1 How Compliance Works

---

All merchants and service providers that store, process, and/or transmit cardholder data must comply with the PCI DSS. This applies to all payment channels, including retail (brick-and-mortar), mail/telephone order, and e-commerce. Due to constantly changing requirements, PCI compliance requires annual audits. Audits require compliance with the following criteria published by Visa:

PCI Data Security Standard	
Build and Maintain a Secure Network	<ol style="list-style-type: none"><li>1. Install and maintain a firewall configuration to protect data</li><li>2. Do not use vendor-supplied defaults for system passwords and other security parameters</li></ol>
Protect Cardholder Data	<ol style="list-style-type: none"><li>3. Protect stored data</li><li>4. Encrypt transmission of cardholder data and sensitive information across public networks</li></ol>
Maintain a Vulnerability Management Program	<ol style="list-style-type: none"><li>5. Use and regularly update anti-virus software</li><li>6. Develop and maintain secure systems and applications</li></ol>
Implement Strong Access Control Measures	<ol style="list-style-type: none"><li>7. Restrict access to data by business need-to-know</li><li>8. Assign a unique ID to each person with computer access</li><li>9. Restrict physical access to cardholder data</li></ol>
Regularly Monitor and Test Networks	<ol style="list-style-type: none"><li>10. Track and monitor all access to network resources and cardholder data</li><li>11. Regularly test security systems and processes</li></ol>
Maintain an Information Security Policy	<ol style="list-style-type: none"><li>12. Maintain a policy that addresses information security</li></ol>

---

## 4.2 Compliance Validation

---

Separate and distinct from the mandate to comply with PCI is the actual validation of compliance. Validation must be performed by a Qualified Data Security Company (QDSC). The time and expense inherent with PCI validation is significant. Costs include, but are not limited to, the hiring of an independent auditor and resulting necessary post-audit changes. Post-audit changes can be very expensive if the audited business wishes to continue storing specific secured information. Validation of compliance also has many levels.

---

### 4.3 Why Comply?

---

By complying with PCI DSS, merchants and service providers not only meet their obligation to the payment card industry, but they also limit their financial risk exposure inherent to storing sensitive cardholder data. The financial risk exposure associated with storing cardholder data is large enough to bankrupt most merchants and service providers. <sup>2</sup>Penalties and fines from VISA and MasterCard range from \$100,000 to \$500,000 per incident. American Express has a \$50,000 per day penalty for any non-compliant members. These are just the penalties assessed if a business is found out of compliance with the possibility of a violation. If a violation actually exists and credit card issuers are affected; there is an additional risk of a class-action suit (by issuers due to losses and replacement of cards) and federal/state violations which could actually result in jail time.

Visa also provides the following overall benefit for compliance:

Benefits of compliance	
Everyone	Limited risk More confidence in the payment industry
Member	Protected reputation
Merchant and Service Provider	Competitive edge gained Increased revenue and improved bottom line Positive image maintained Customers are protected
Industry	"Good security neighbors" encouraged
Consumer	Information is safeguarded Identity theft prevention

---

<sup>2</sup> "Operations and Risk Management," Visa, USA Cardholder Information Security Program <http://www.visa.com/cisp>

---

## 5 PASS Technology

---

To aid compliance with PCI DSS, Element Payment Services, Inc. has developed a new Payment Account Secure Storage (PASS) technology. PASS allows merchants and service providers the ability to easily comply and remain compliant with PCI DSS.

**PASS technology has two major objectives:**

*1. Aid Merchants & Service Providers with PCI Compliance*

This objective is met by removing the responsibility of storing sensitive cardholder data. By not storing sensitive cardholder data, merchants and service providers substantially reduce their financial risk exposure as well as limit the time and expense associated with PCI compliance.

*2. Minimize the Impact on Existing Payment Processing Systems*

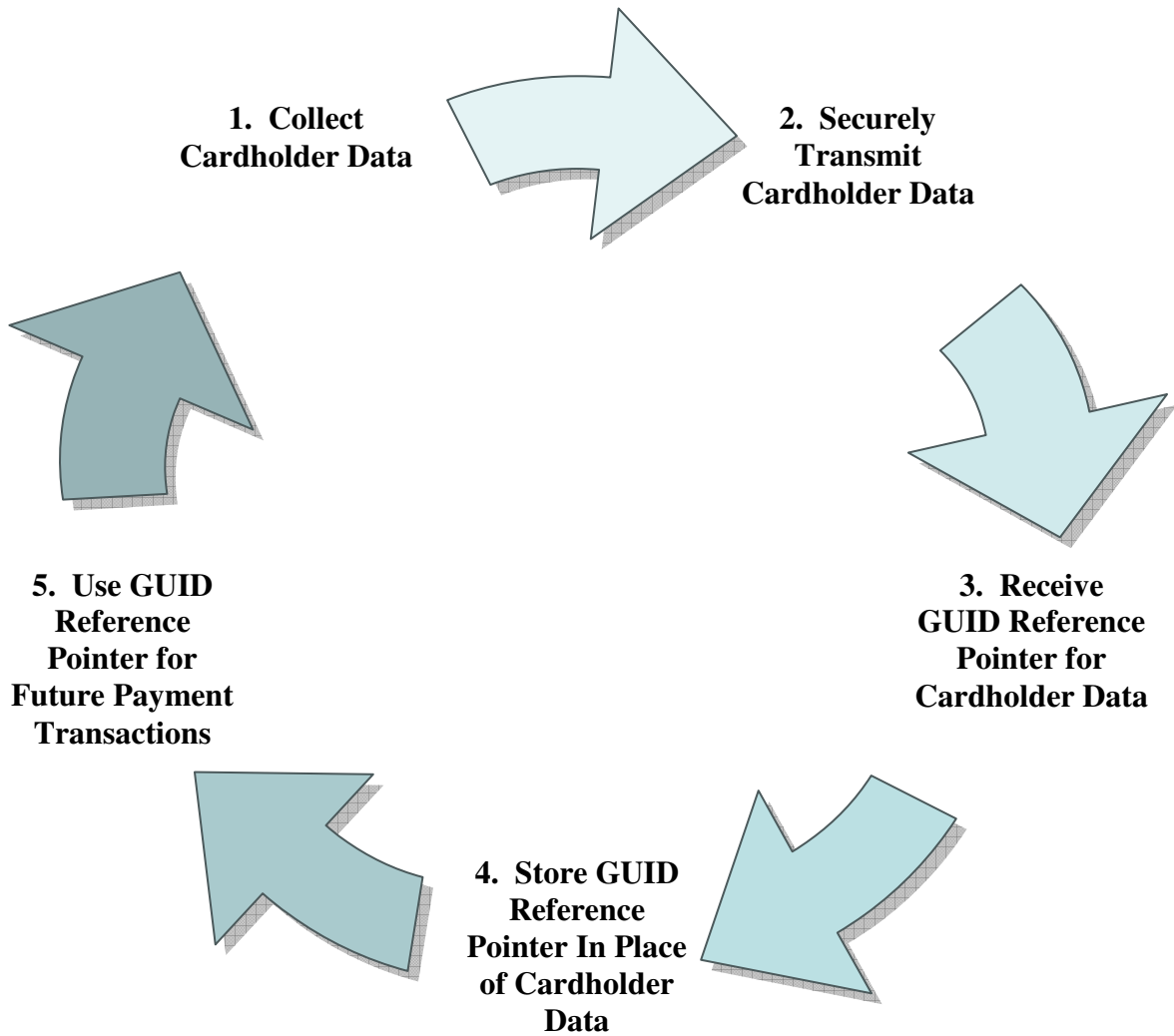
This objective is met by not changing any of the existing underlying payment processing patterns, but changing the location where sensitive cardholder data is stored. Please note: PASS only stores sensitive cardholder data (i.e. account numbers). PASS does not store non-sensitive cardholder data (i.e. Name, Address, etc.). Truncated account numbers and non-sensitive cardholder data can be stored, if necessary, to satisfy any customer service or other business need that may exist.

---

## 5.1 “How PASS Works”

---

PASS follows these five simple steps:



For a detailed description on how PASS works please refer to section [7.1](#) of the Appendix.

---

### **5.1.1 Step 1 – Collect Cardholder data**

---

Merchants and service providers accept cardholder data as they always have; examples include, but are not limited to, track data readers and manual data entry.

---

### **5.1.2 Step 2 - Securely Transmit Cardholder Data**

---

The merchant's payment system transmits sensitive cardholder data over a secure 128-bit encrypted SSL connection to Element Payment Services', Inc. PCI DSS compliant storage facility.

---

### **5.1.3 Step 3 - Receive Reference Pointer to Cardholder Data**

---

The merchant payment system receives a globally unique identifier (GUID) as a reference to be used as a pointer to cardholder data stored remotely.

---

### **5.1.4 Step 4 - Store Reference Pointer In Place of Cardholder Data**

---

The Merchant stores the GUID reference pointer instead of the sensitive cardholder data. The reference pointer is of no value to hackers because the cardholder information can never be obtained by using the reference pointer. By design, extraction of full card account number is not allowed through PASS to any merchant or service provider.

---

### **5.1.5 Step 5 - Use Reference Pointer for Future Payment Transactions**

---

The merchant transmits the GUID reference pointer to process future payment transactions instead of transmitting the actual cardholder data. Likewise, the GUID reference pointer can be used to update, delete, and query cardholder accounts. The query functionality only returns truncated account numbers, never the entire account number.

---

## 6 Summary

---

The time consuming complexity and the inordinate expense inherent with PCI DSS compliance can be an overwhelming burden to most merchants and service providers. In addition, the enormous financial risk inherent with storing sensitive cardholder data can be enough to bankrupt most merchants and service providers.

Element Payment Services, Inc. Payment Account Storage Service (PASS) dramatically reduces the complexity and expense related to PCI compliance. At the same time, PASS reduces the financial risk associated with storing sensitive cardholder data simply by removing the need to store the data all together. By not storing sensitive cardholder data, merchants and service providers can save a substantial amount of time and money; likewise they can reduce an enormous financial risk.

---

## 7 Appendix

---

---

### 7.1 Detailed Description of PASS

---

PASS is a unique combination of the following technologies.

First, PASS is designed as a secure Web Service comprised of four basic methods 1) PaymentAccountCreate, 2) PaymentAccountUpdate, 3) PaymentAccountDelete, and 4) PaymentAccountQuery (see section 6.2). These methods are used to create, update, delete, and query payment data storage accounts.

Second, PASS produces a Globally Unique Identifier (GUID) for use as a reference pointer to maintain and access payment data storage accounts.

Third, PASS uses HyperText Transport Protocol Secure (HTTPS) along with Secure Socket Layer (SSL) connections to transport and encrypt all requests. Every request is therefore protected by 128-bit level of encryption.

Forth, every request is authenticated through use of an Element Payment Services, Inc. assigned end user AccountID and AccountToken. AccountID is a unique account identifier. AccountToken is an alpha numeric string produced using a proprietary hash algorithm.

---














## 7.2 Methods

---








### 7.2.1 PaymentAccountCreate

---

#### 7.2.1.1 Input Fields

Name	Class	Required	Description
 <a href="#">ApplicationID</a>	<a href="#">Application</a>	Required	Unique application identifier
 <a href="#">ApplicationName</a>	<a href="#">Application</a>	Required	Name of application
 <a href="#">ApplicationVersion</a>	<a href="#">Application</a>	Required	Version of application
 <a href="#">AccountID</a>	<a href="#">Credential</a>	Required	Unique account identifier
 <a href="#">AccountToken</a>	<a href="#">Credential</a>	Required	Secret token used for authentication
 <a href="#">AcceptorID</a>	<a href="#">Credential</a>	Required	MerchantID
 <a href="#">PaymentAccountType</a>	<a href="#">PaymentAccount</a>	Required	Payment Account Type
 <a href="#">PaymentAccountReferenceNumber</a>	<a href="#">PaymentAccount</a>	Required	Payment Account Reference Number
 <a href="#">CardNumber</a>	<a href="#">Card</a>	Conditional -(Required if PaymentAccountType is CreditCard)	Card Number
 <a href="#">ExpirationMonth</a>	<a href="#">Card</a>	Conditional -(Required if PaymentAccountType is CreditCard)	Expiration Month (MM)
 <a href="#">ExpirationYear</a>	<a href="#">Card</a>	Conditional -(Required if PaymentAccountType is CreditCard)	Expiration Year (YY)
 <a href="#">AccountNumber</a>	<a href="#">DemandDepositAccount</a>	Conditional -(Required if PaymentAccountType is a DDA account)	Account Number
 <a href="#">RoutingNumber</a>	<a href="#">DemandDepositAccount</a>	Conditional -(Required if PaymentAccountType is a DDA account)	Routing Number

#### 7.2.1.2 Output Fields

Name	Class	Returned	Description
 <a href="#">ExpressResponseCode</a>	<a href="#">Response</a>	Returned	Express Response Code
 <a href="#">ExpressResponseMessage</a>	<a href="#">Response</a>	Returned	Express Response Message
 <a href="#">ExpressTransactionDate</a>	<a href="#">Response</a>	Returned	Express transaction date formatted [YYYYMMDD]
 <a href="#">ExpressTransactionTime</a>	<a href="#">Response</a>	Returned	Express transaction time formatted [HHMMSS]
 <a href="#">ExpressTransactionTimeZone</a>	<a href="#">Response</a>	Returned	Express transaction UTC time zone
 <a href="#">PaymentAccountID</a>	<a href="#">Response.PaymentAccount</a>	Returned	Unique GUID that identifies the Payment Account.
 <a href="#">PaymentAccountReferenceNumber</a>	<a href="#">Response.PaymentAccount</a>	Returned	Payment Account Reference Number

## 7.2.1.3 SOAP 1.2 Web Service Request

```
POST /express.asmx HTTP/1.1
Host: 207.138.56.52
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <PaymentAccountCreate xmlns="https://services.elementexpress.com">
      <credentials>
        <AccountID>string</AccountID>
        <AccountToken>string</AccountToken>
        <AcceptorID>string</AcceptorID>
        <NewAccountToken>string</NewAccountToken>
      </credentials>
      <application>
        <ApplicationID>string</ApplicationID>
        <ApplicationName>string</ApplicationName>
        <ApplicationVersion>string</ApplicationVersion>
      </application>
      <paymentAccount>
        <PaymentAccountID>string</PaymentAccountID>
        <PaymentAccountType>CreditCard or Checking or Savings or ACH or Other</PaymentAccountType>
        <PaymentAccountReferenceNumber>string</PaymentAccountReferenceNumber>
      </paymentAccount>
      <card>
        <Track1Data>string</Track1Data>
        <Track2Data>string</Track2Data>
        <CardNumber>string</CardNumber>
        <TruncatedCardNumber>string</TruncatedCardNumber>
        <ExpirationMonth>string</ExpirationMonth>
        <ExpirationYear>string</ExpirationYear>
        <CVV>string</CVV>
        <CAVV>string</CAVV>
        <XID>string</XID>
        <PINBlock>string</PINBlock>
        <KeySerialNumber>string</KeySerialNumber>
        <AVSResponseCode>string</AVSResponseCode>
        <CVVResponseCode>string</CVVResponseCode>
        <CAVVResponseCode>string</CAVVResponseCode>
      </card>
      <demandDepositAccount>
        <AccountNumber>string</AccountNumber>
        <CheckNumber>string</CheckNumber>
        <RoutingNumber>string</RoutingNumber>
        <TruncatedAccountNumber>string</TruncatedAccountNumber>
        <TruncatedRoutingNumber>string</TruncatedRoutingNumber>
      </demandDepositAccount>
      <extendedParameters>
        <ExtendedParameters>
          <Key>string</Key>
          <Value />
        </ExtendedParameters>
        <ExtendedParameters>
          <Key>string</Key>
          <Value />
        </ExtendedParameters>
      </extendedParameters>
    </PaymentAccountCreate>
  </soap12:Body>
</soap12:Envelope>
```














## 7.2.1.4 SOAP 1.2 Web Service Response

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length








<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <PaymentAccountCreateResponse xmlns="https://services.elementexpress.com">
      <response>
        <Card>
          <Track1Data>string</Track1Data>
          <Track2Data>string</Track2Data>
          <CardNumber>string</CardNumber>
          <TruncatedCardNumber>string</TruncatedCardNumber>
          <ExpirationMonth>string</ExpirationMonth>
          <ExpirationYear>string</ExpirationYear>
          <CVV>string</CVV>
          <CAVV>string</CAVV>
          <XID>string</XID>
          <PINBlock>string</PINBlock>
          <KeySerialNumber>string</KeySerialNumber>
          <AVSResponseCode>string</AVSResponseCode>
          <CVVResponseCode>string</CVVResponseCode>
          <CAVVResponseCode>string</CAVVResponseCode>
        </Card>
        <DemandDepositAccount>
          <AccountNumber>string</AccountNumber>
          <CheckNumber>string</CheckNumber>
          <RoutingNumber>string</RoutingNumber>
          <TruncatedAccountNumber>string</TruncatedAccountNumber>
          <TruncatedRoutingNumber>string</TruncatedRoutingNumber>
        </DemandDepositAccount>
        <PaymentAccount>
          <PaymentAccountID>string</PaymentAccountID>
          <PaymentAccountType>CreditCard or Checking or Savings or ACH or Other</PaymentAccountType>
          <PaymentAccountReferenceNumber>string</PaymentAccountReferenceNumber>
        </PaymentAccount>
      </response>
    </PaymentAccountCreateResponse>
  </soap12:Body>
</soap12:Envelope>
```

## 7.2.2 PaymentAccountUpdate

### 7.2.2.1 Input Fields

Name	Class	Required	Description
 <a href="#">ApplicationID</a>	<a href="#">Application</a>	Required	Unique application identifier
 <a href="#">ApplicationName</a>	<a href="#">Application</a>	Required	Name of application
 <a href="#">ApplicationVersion</a>	<a href="#">Application</a>	Required	Version of application
 <a href="#">AccountID</a>	<a href="#">Credential</a>	Required	Unique account identifier
 <a href="#">AccountToken</a>	<a href="#">Credential</a>	Required	Secret token used for authentication
 <a href="#">AcceptorID</a>	<a href="#">Credential</a>	Required	MerchantID
 <a href="#">PaymentAccountID</a>	<a href="#">PaymentAccount</a>	Required	Unique GUID that identifies the Payment Account.
 <a href="#">PaymentAccountReferenceNumber</a>	<a href="#">PaymentAccount</a>	Required	Payment Account Reference Number
 <a href="#">PaymentAccountType</a>	<a href="#">PaymentAccount</a>	Required	Payment Account Type
 <a href="#">CardNumber</a>	<a href="#">Card</a>	Conditional -(Required if PaymentAccountType is CreditCard)	Card Number
 <a href="#">ExpirationMonth</a>	<a href="#">Card</a>	Conditional -(Required if PaymentAccountType is CreditCard)	Expiration Month (MM)
 <a href="#">ExpirationYear</a>	<a href="#">Card</a>	Conditional -(Required if PaymentAccountType is CreditCard)	Expiration Year (YY)
 <a href="#">AccountNumber</a>	<a href="#">DemandDepositAccount</a>	Conditional -(Required if PaymentAccountType is a DDA account)	Account Number
 <a href="#">RoutingNumber</a>	<a href="#">DemandDepositAccount</a>	Conditional -(Required if PaymentAccountType is a DDA account)	Routing Number

### 7.2.2.2 Output Fields

Name	Class	Returned	Description
 <a href="#">ExpressResponseCode</a>	<a href="#">Response</a>	Returned	Express Response Code
 <a href="#">ExpressResponseMessage</a>	<a href="#">Response</a>	Returned	Express Response Message
 <a href="#">ExpressTransactionDate</a>	<a href="#">Response</a>	Returned	Express transaction date formatted [YYYYMMDD]
 <a href="#">ExpressTransactionTime</a>	<a href="#">Response</a>	Returned	Express transaction time formatted [HHMMSS]
 <a href="#">ExpressTransactionTimeZone</a>	<a href="#">Response</a>	Returned	Express transaction UTC time zone
 <a href="#">PaymentAccountID</a>	<a href="#">Response.PaymentAccount</a>	Returned	Unique GUID that identifies the Payment Account.
 <a href="#">PaymentAccountReferenceNumber</a>	<a href="#">Response.PaymentAccount</a>	Returned	Payment Account Reference Number

## 7.2.2.3 SOAP 1.2 Web Service Request

```
POST /express.asmx HTTP/1.1
Host: 207.138.56.52
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <PaymentAccountUpdate xmlns="https://services.elementexpress.com">
      <credentials>
        <AccountID>string</AccountID>
        <AccountToken>string</AccountToken>
        <AcceptorID>string</AcceptorID>
        <NewAccountToken>string</NewAccountToken>
      </credentials>
      <application>
        <ApplicationID>string</ApplicationID>
        <ApplicationName>string</ApplicationName>
        <ApplicationVersion>string</ApplicationVersion>
      </application>
      <paymentAccount>
        <PaymentAccountID>string</PaymentAccountID>
        <PaymentAccountType>CreditCard or Checking or Savings or ACH or Other</PaymentAccountType>
        <PaymentAccountReferenceNumber>string</PaymentAccountReferenceNumber>
      </paymentAccount>
      <card>
        <Track1Data>string</Track1Data>
        <Track2Data>string</Track2Data>
        <CardNumber>string</CardNumber>
        <TruncatedCardNumber>string</TruncatedCardNumber>
        <ExpirationMonth>string</ExpirationMonth>
        <ExpirationYear>string</ExpirationYear>
        <CVV>string</CVV>
        <CAVV>string</CAVV>
        <XID>string</XID>
        <PINBlock>string</PINBlock>
        <KeySerialNumber>string</KeySerialNumber>
        <AVSResponseCode>string</AVSResponseCode>
        <CVVResponseCode>string</CVVResponseCode>
        <CAVVResponseCode>string</CAVVResponseCode>
      </card>
      <demandDepositAccount>
        <AccountNumber>string</AccountNumber>
        <CheckNumber>string</CheckNumber>
        <RoutingNumber>string</RoutingNumber>
        <TruncatedAccountNumber>string</TruncatedAccountNumber>
        <TruncatedRoutingNumber>string</TruncatedRoutingNumber>
      </demandDepositAccount>
      <extendedParameters>
        <ExtendedParameters>
          <Key>string</Key>
          <Value />
        </ExtendedParameters>
        <ExtendedParameters>
          <Key>string</Key>
          <Value />
        </ExtendedParameters>
      </extendedParameters>
    </PaymentAccountUpdate>
  </soap12:Body>
</soap12:Envelope>
```

## 7.2.2.4 SOAP 1.2 Web Service Response

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length









<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <PaymentAccountUpdateResponse xmlns="https://services.elementexpress.com">
      <response>
        <Card>
          <Track1Data>string</Track1Data>
          <Track2Data>string</Track2Data>
          <CardNumber>string</CardNumber>
          <TruncatedCardNumber>string</TruncatedCardNumber>
          <ExpirationMonth>string</ExpirationMonth>
          <ExpirationYear>string</ExpirationYear>
          <CVV>string</CVV>
          <CAVV>string</CAVV>
          <XID>string</XID>
          <PINBlock>string</PINBlock>
          <KeySerialNumber>string</KeySerialNumber>
          <AVSResponseCode>string</AVSResponseCode>
          <CVVResponseCode>string</CVVResponseCode>
          <CAVVResponseCode>string</CAVVResponseCode>
        </Card>
        <DemandDepositAccount>
          <AccountNumber>string</AccountNumber>
          <CheckNumber>string</CheckNumber>
          <RoutingNumber>string</RoutingNumber>
          <TruncatedAccountNumber>string</TruncatedAccountNumber>
          <TruncatedRoutingNumber>string</TruncatedRoutingNumber>
        </DemandDepositAccount>
        <PaymentAccount>
          <PaymentAccountID>string</PaymentAccountID>
          <PaymentAccountType>CreditCard or Checking or Savings or ACH or Other</PaymentAccountType>
          <PaymentAccountReferenceNumber>string</PaymentAccountReferenceNumber>
        </PaymentAccount>
      </response>
    </PaymentAccountUpdateResponse>
  </soap12:Body>
</soap12:Envelope>
```

---






## 7.2.3 PaymentAccountDelete

---

### 7.2.3.1 Input Fields

Name	Class	Required	Description
 <a href="#">ApplicationID</a>	<a href="#">Application</a>	Required	Unique application identifier
 <a href="#">ApplicationName</a>	<a href="#">Application</a>	Required	Name of application
 <a href="#">ApplicationVersion</a>	<a href="#">Application</a>	Required	Version of application
 <a href="#">AccountID</a>	<a href="#">Credential</a>	Required	Unique account identifier
 <a href="#">AccountToken</a>	<a href="#">Credential</a>	Required	Secret token used for authentication
 <a href="#">AcceptorID</a>	<a href="#">Credential</a>	Required	MerchantID
 <a href="#">PaymentAccountID</a>	<a href="#">PaymentAccount</a>	Required	Unique GUID that identifies the Payment Account.
 <a href="#">PaymentAccountReferenceNumber</a>	<a href="#">PaymentAccount</a>	Required	Payment Account Reference Number

### 7.2.3.2 Output Fields

Name	Class	Returned	Description
 <a href="#">ExpressResponseCode</a>	<a href="#">Response</a>	Returned	Express Response Code
 <a href="#">ExpressResponseMessage</a>	<a href="#">Response</a>	Returned	Express Response Message
 <a href="#">ExpressTransactionDate</a>	<a href="#">Response</a>	Returned	Express transaction date formatted [YYYYMMDD]
 <a href="#">ExpressTransactionTime</a>	<a href="#">Response</a>	Returned	Express transaction time formatted [HHMMSS]
 <a href="#">ExpressTransactionTimeZone</a>	<a href="#">Response</a>	Returned	Express transaction UTC time zone

## 7.2.3.3 SOAP 1.2 Web Service Request

```
POST /express.asmx HTTP/1.1
Host: 207.138.56.52
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <PaymentAccountDelete xmlns="https://services.elementexpress.com">
      <credentials>
        <AccountID>string</AccountID>
        <AccountToken>string</AccountToken>
        <AcceptorID>string</AcceptorID>
        <NewAccountToken>string</NewAccountToken>
      </credentials>
      <application>
        <ApplicationID>string</ApplicationID>
        <ApplicationName>string</ApplicationName>
        <ApplicationVersion>string</ApplicationVersion>
      </application>
      <paymentAccount>
        <PaymentAccountID>string</PaymentAccountID>
        <PaymentAccountType>CreditCard or Checking or Savings or ACH or Other</PaymentAccountType>
        <PaymentAccountReferenceNumber>string</PaymentAccountReferenceNumber>
      </paymentAccount>
      <extendedParameters>
        <ExtendedParameters>
          <Key>string</Key>
          <Value />
        </ExtendedParameters>
        <ExtendedParameters>
          <Key>string</Key>
          <Value />
        </ExtendedParameters>
      </extendedParameters>
    </PaymentAccountDelete>
  </soap12:Body>
</soap12:Envelope>
```

## 7.2.3.4 SOAP 1.2 Web Service Request

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length









<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <PaymentAccountDeleteResponse xmlns="https://services.elementexpress.com">
      <response>
        <Card>
          <Track1Data>string</Track1Data>
          <Track2Data>string</Track2Data>
          <CardNumber>string</CardNumber>
          <TruncatedCardNumber>string</TruncatedCardNumber>
          <ExpirationMonth>string</ExpirationMonth>
          <ExpirationYear>string</ExpirationYear>
          <CVV>string</CVV>
          <CAVV>string</CAVV>
          <XID>string</XID>
          <PINBlock>string</PINBlock>
          <KeySerialNumber>string</KeySerialNumber>
          <AVSResponseCode>string</AVSResponseCode>
          <CVVResponseCode>string</CVVResponseCode>
          <CAVVResponseCode>string</CAVVResponseCode>
        </Card>
        <DemandDepositAccount>
          <AccountNumber>string</AccountNumber>
          <CheckNumber>string</CheckNumber>
          <RoutingNumber>string</RoutingNumber>
          <TruncatedAccountNumber>string</TruncatedAccountNumber>
          <TruncatedRoutingNumber>string</TruncatedRoutingNumber>
        </DemandDepositAccount>
        <PaymentAccount>
          <PaymentAccountID>string</PaymentAccountID>
          <PaymentAccountType>CreditCard or Checking or Savings or ACH or Other</PaymentAccountType>
          <PaymentAccountReferenceNumber>string</PaymentAccountReferenceNumber>
        </PaymentAccount>
      </response>
    </PaymentAccountDeleteResponse>
  </soap12:Body>
</soap12:Envelope>
```

---








## 7.2.4 PaymentAccountQuery

---

### 7.2.4.1 Input Fields

Name	Class	Required	Description
 <a href="#">ApplicationID</a>	<a href="#">Application</a>	Required	Unique application identifier
 <a href="#">ApplicationName</a>	<a href="#">Application</a>	Required	Name of application
 <a href="#">ApplicationVersion</a>	<a href="#">Application</a>	Required	Version of application
 <a href="#">AccountID</a>	<a href="#">Credential</a>	Required	Unique account identifier
 <a href="#">AccountToken</a>	<a href="#">Credential</a>	Required	Secret token used for authentication
 <a href="#">AcceptorID</a>	<a href="#">Credential</a>	Required	MerchantID
 <a href="#">PaymentAccountID</a>	<a href="#">PaymentAccount</a>	Required	Unique GUID that identifies the Payment Account.
 <a href="#">PaymentAccountReferenceNumber</a>	<a href="#">PaymentAccount</a>	Required	Payment Account Reference Number

### 7.2.4.2 Output Fields

Name	Class	Returned	Description
 <a href="#">ExpressResponseCode</a>	<a href="#">Response</a>	Returned	Express Response Code
 <a href="#">ExpressResponseMessage</a>	<a href="#">Response</a>	Returned	Express Response Message
 <a href="#">ExpressTransactionDate</a>	<a href="#">Response</a>	Returned	Express transaction date formatted [YYYYMMDD]
 <a href="#">ExpressTransactionTime</a>	<a href="#">Response</a>	Returned	Express transaction time formatted [HHMMSS]
 <a href="#">ExpressTransactionTimeZone</a>	<a href="#">Response</a>	Returned	Express transaction UTC time zone
 <a href="#">PaymentAccountID</a>	<a href="#">Response.PaymentAccount</a>	Returned	Unique GUID that identifies the Payment Account.
 <a href="#">PaymentAccountReferenceNumber</a>	<a href="#">Response.PaymentAccount</a>	Returned	Payment Account Reference Number

## 7.2.4.3 SOAP 1.2 Web Service Request

```
POST /express.asmx HTTP/1.1
Host: 207.138.56.52
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

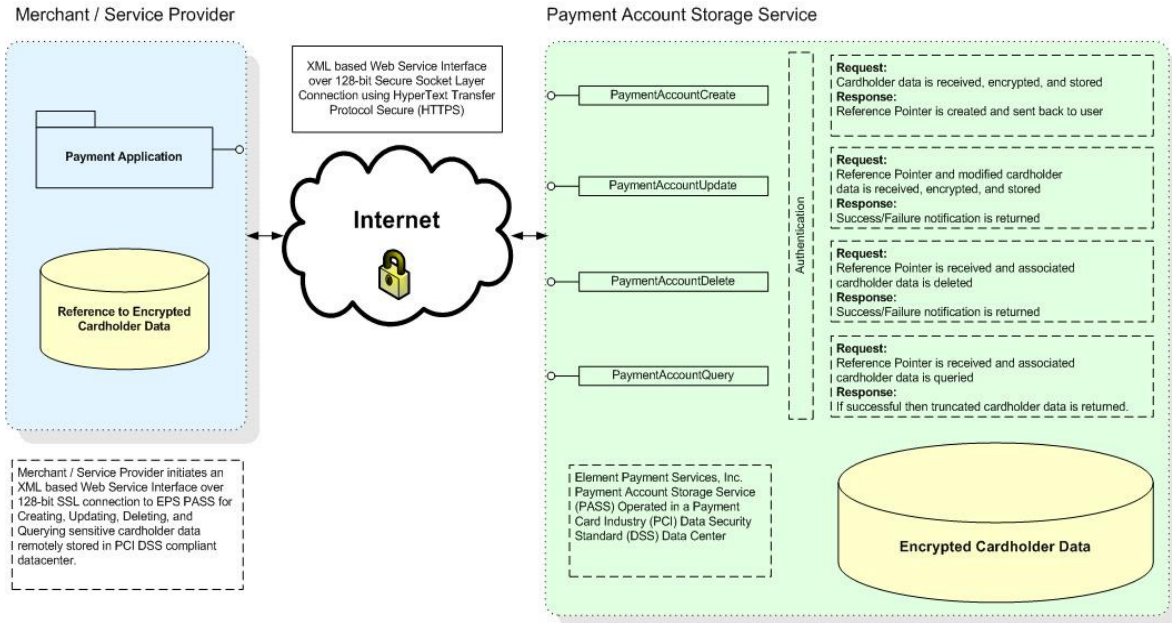
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <PaymentAccountQuery xmlns="https://services.elementexpress.com">
      <credentials>
        <AccountID>string</AccountID>
        <AccountToken>string</AccountToken>
        <AcceptorID>string</AcceptorID>
        <NewAccountToken>string</NewAccountToken>
      </credentials>
      <application>
        <ApplicationID>string</ApplicationID>
        <ApplicationName>string</ApplicationName>
        <ApplicationVersion>string</ApplicationVersion>
      </application>
      <paymentAccount>
        <PaymentAccountID>string</PaymentAccountID>
        <PaymentAccountType>CreditCard or Checking or Savings or ACH or Other</PaymentAccountType>
        <PaymentAccountReferenceNumber>string</PaymentAccountReferenceNumber>
      </paymentAccount>
      <extendedParameters>
        <ExtendedParameters>
          <Key>string</Key>
          <Value />
        </ExtendedParameters>
        <ExtendedParameters>
          <Key>string</Key>
          <Value />
        </ExtendedParameters>
      </extendedParameters>
    </PaymentAccountQuery>
  </soap12:Body>
</soap12:Envelope>
```

## 7.2.4.4 SOAP 1.2 Web Service Response

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <PaymentAccountQueryResponse xmlns="https://services.elementexpress.com">
      <response>
        <Card>
          <Track1Data>string</Track1Data>
          <Track2Data>string</Track2Data>
          <CardNumber>string</CardNumber>
          <TruncatedCardNumber>string</TruncatedCardNumber>
          <ExpirationMonth>string</ExpirationMonth>
          <ExpirationYear>string</ExpirationYear>
          <CVV>string</CVV>
          <CAVV>string</CAVV>
          <XID>string</XID>
          <PINBlock>string</PINBlock>
          <KeySerialNumber>string</KeySerialNumber>
          <AVSResponseCode>string</AVSResponseCode>
          <CVVResponseCode>string</CVVResponseCode>
          <CAVVResponseCode>string</CAVVResponseCode>
        </Card>
        <DemandDepositAccount>
          <AccountNumber>string</AccountNumber>
          <CheckNumber>string</CheckNumber>
          <RoutingNumber>string</RoutingNumber>
          <TruncatedAccountNumber>string</TruncatedAccountNumber>
          <TruncatedRoutingNumber>string</TruncatedRoutingNumber>
        </DemandDepositAccount>
        <PaymentAccount>
          <PaymentAccountID>string</PaymentAccountID>
          <PaymentAccountType>CreditCard or Checking or Savings or ACH or Other</PaymentAccountType>
          <PaymentAccountReferenceNumber>string</PaymentAccountReferenceNumber>
        </PaymentAccount>
      </response>
    </PaymentAccountQueryResponse>
  </soap12:Body>
</soap12:Envelope>
```

## 7.3 Design Diagram



---

## 7.4 Cardholder and Sensitive Authentication Data

---

	Data Element	Storage Permitted	Protection Required	PCI DSS Req. 3.4
<b>Cardholder Data</b>	Primary Account Number (PAN)	YES	YES	YES
	Cardholder Name*	YES	YES*	NO
	Service Code*	YES	YES*	NO
	Expiration Date*	YES	YES*	NO
<b>Sensitive Authentication Data**</b>	Full Magnetic Stripe	NO	N/A	N/A
	CVC2/CVV2/CID	NO	N/A	N/A
	PIN / PIN Block	NO	N/A	N/A

\* These data elements must be protected if stored in conjunction with the PAN. This protection must be consistent with PCI DSS requirements for general protection of the cardholder environment. Additionally, other legislation (for example, related to consumer personal data protection, privacy, identity theft, or data security) may require specific protection of this data, or proper disclosure of a company's practices if consumer-related personal data is being collected during the course of business. PCI DSS, however, does not apply if PANs are not stored, processed, or transmitted.

\*\* Sensitive authentication data must not be stored subsequent to authorization (even if encrypted).