Payment WorkSuite CardVault Silent Post Interface

This technical specification defines the CardVault® Silent Post interface, a Payment WorkSuite® application offered by 3Delta Systems, Inc.® (3DSI).

Last Updated: 04/14/2017

14151 Newbrook Drive, Suite 200
Chantilly, VA, 20151
P: 703.234.6030 | F: 877.408.7968
integration@3DSI.com | www.3DSI.com

Table of Contents

Preface

3Delta System's CardVault Silent Post Interface is an optional method of capturing credit card data without it going into a merchant's environment. A CardVault client can incorporate HTML and ASP code into a form presented by their web application to add a credit card to CardVault via Silent Post. This code will call 3DSI's CardVault web server directly from the web application user's browser to add a card account to CardVault. Because this call is made directly from the user's browser to 3DSI's server, the CardVault client's web server is bypassed. Card account data travels directly from the user's browser to 3DSI's server over a secure SSL channel. The user's browser must support TLS 1.1 or greater for CardVault Silent Post to function properly. The Silent Post call is made via a web POST to 3DSI's CardVault web server. Silent Post connects via SSL to an approval URL predetermined by the client when certifying to use Silent Post. 3DSI will POST a collection of approval data to this URL, expecting a response value indicating acceptance or denial of the card data submission.

Once approved, the current card data submission is stored in CardVault. 3DSI provides the client's web server with the results and optionally redirects the user's browser back to the client's web server, using URLs provided by the client in the initiating POST. When 3DSI redirects the user's browser to the client's URL, a series of name/value pairs will be provided in the URL's query string to indicate the result of adding the card data to CardVault. The primary account number of the card is not included in these result fields. This structure gives the client control over how their web server will handle processing the result of adding a card account to CardVault. It also gives the user information that may be used in customer support cases. The client can choose which of the result fields, if any, to display to the user. The implementation of the fields returned by 3DSI when a card is added to CardVault is decided by the client.

A client may additionally choose to verify the validity of a card account prior to adding it to CardVault via Silent Post. This is achieved by performing a card verification request with the bank against the card account during the Silent Post process. If the card verification request does not succeed, the card is not stored. If the card verification request does succeed, the card is stored. A client account may optionally be configured to disallow storage of cards where certain desired Address Verification or Card Security Code conditions are not met. Please refer to your account manager for more information about account configuration options.

The Silent Post form also allows the client to provide any number of "pass-through" fields. These fields are not stored by 3DSI. They are simply sent back unaltered via POST as name/value pairs to the client's ResultURL after the card is added to CardVault. They can be used by the client in any number of ways as dictated by client requirements.

Form Variable Definitions

Variables passed from the user's browser to CardVault Silent Post

Card Addition Only

The following is a list of form variables that may be passed from the user's browser to CardVault Silent Post for a Card Addition only request. Some form variables are required while others are not.

Field Name	Required	Description
ResultURL	Yes	The URL that 3DSI will POST result messages to. This URL must use SSL.
BrowserRedirectURL	No	The URL that 3DSI will redirect the user's browser to after submitting data to CardVault. This URL must use SSL.
ResponseType	No	A value identifying whether result data should be sent in a response to the POST from the user's browser in the form of an XML fragment, or, should be sent in a querystring via a BrowserRedirectURL.
SubmissionID	Yes	A unique identifier associated with the current attempt to submit data to CardVault. The client should store this value to be used later in both the approval and result stages of Silent Post. This value is transient within the CardVault system and is not stored for later use.
APCount	Yes	The number of approval values supplied with this submission.
AP#	No	An individual approval value, with "#" replaced with an integer between 1 and *APCount*. These values will be passed back as form variables to the *ApprovalURL* that was specified by the client. Code at the *ApprovalURL* should confirm that the data provided matches the data sent to the user's browser and either approve or deny the request based upon this data.
PTCount	Yes	The number of pass-through values supplied with this submission.
PT#	No	An individual pass-through value, with "#" replaced with an integer between 1 and *PTCount*. These values will be passed back as form variables to the *ResultURL* to be utilized by custom logic on the client's web server as dictated by client requirements.

CustomerIdentifier_MerchantCode	Yes	The code assigned to the Merchant by 3DSI.
CustomerIdentifier_LocationCode	Yes	The code assigned to the Location by 3DSI. If an account is configured to store customers at the client level, this property is not required and is ignored if provided.
CustomerIdentifier_CustomerCode	Yes	The code assigned to the Customer where the credit card will be stored.
CreditCard_CardAccountNumber	Yes	The credit card primary account number.
CreditCard_ExpirationMonth	No	The credit card expiration month.
CreditCard_ExpirationYear	No	The credit card expiration year.
CreditCard_Notes	No	This field contains text that is securely stored and may be retrieved with the credit card. These notes can optionally be displayed to a user when a transaction is processed via the Payment WorkSuite's EC-Zone® web application. The *CreditCard_DisplayNotesDuringTransaction* property determines this behavior.
CreditCard_DisplayNotesDuringTransaction	No	This field determines whether or not the *CreditCard_Notes* value will be displayed to a user when processing a transaction using the Payment WorkSuite's EC-Zone web application.
CreditCard_Token	No	A token uniquely identifies a specific credit card that is associated with the customer identified by the specified *CustomerIdentifier_CustomerCode*. If no token is passed in with the request, 3DSI will assign one and return it in the response. A client may choose to assign tokens by passing them in with requests.
CreditCard_FriendlyName	No	A human-readable identifier that a client may assign to the credit card.
CreditCard_CardType	No	The type of the credit card.
CreditCard_NameOnCard	No	The name of the cardholder as it appears on the credit card.
CreditCard_Cardholder_Title	No	The title of the cardholder.
CreditCard_Cardholder_FirstName	No	The first name of the cardholder.
CreditCard_Cardholder_MiddleName	No	The middle name the cardholder.
CreditCard_Cardholder_LastName	No	The last name of the cardholder.
CreditCard_Cardholder_Suffix	No	The name suffix of the cardholder.
CreditCard_Cardholder_Phone	No	The telephone number for the cardholder.

CreditCard_Cardholder_Fax	No	The fax number for the cardholder.
CreditCard_Cardholder_Email_DisplayAs	No	This field specifies a human-readable name to associate with the email address.
CreditCard_Cardholder_Email_Address	No	This field specifies the email address.
CreditCard_BillingAddress_AddressLine1	No	The credit card billing address line 1.
CreditCard_BillingAddress_AddressLine2	No	The credit card billing address line 2.
CreditCard_BillingAddress_City	No	The credit card billing address city.
CreditCard_BillingAddress_PostalCode	No	The credit card billing address postal code.
CreditCard_BillingAddress_CountryCode	No	The credit card billing address country code.
CreditCard_BillingAddress_StateProvinceCode	No	The credit card billing address state or province code. This field is only used if UnitedStates, Canada or Mexico is set as the CountryCode. Otherwise, this field should be omitted.
CreditCard_BillingAddress_ForeignStateProvinceRegion	No	The credit card billing address foreign state, province or region. This field is only used if a value other than UnitedStates, Canada or Mexico is set as the CountryCode. Otherwise, this field should be omitted.
CreditCard_ShippingAddress_AddressLine1	No	The credit card shipping address line 1.
CreditCard_ShippingAddress_AddressLine2	No	The credit card shipping address line 2.
CreditCard_ShippingAddress_City	No	The credit card shipping address city.
CreditCard_ShippingAddress_PostalCode	No	The credit card shipping address postal code.
CreditCard_ShippingAddress_CountryCode	No	The credit card shipping address country code.
CreditCard_ShippingAddress_StateProvinceCode	No	The credit card shipping address state or province code. This field is only used if UnitedStates, Canada or Mexico is set as the CountryCode. Otherwise, this field should be omitted.
CreditCard_ShippingAddress_ForeignStateProvinceRegion	No	The credit card shipping address foreign state, province or region. This field is only used if a value other than UnitedStates, Canada or Mexico is set as the CountryCode. Otherwise, this field should be omitted.

Notes

AddressLine1 and PostalCode data are needed for Address Verification where it is available.

Please review the Secure Approval Suggestions for guidelines on SubmissionID generation and handling.

If UnitedStates, Canada or Mexico is specified as the CountryCode, then an appropriate StateProvinceCode should be set for that CountryCode.

If anything other than UnitedStates, Canada or Mexico is specified as the CountryCode, then StateProvinceCode should be omitted and the ForeignStateProvinceRegion field may optionally be used to provide the appropriate foreign state, province or region.

If UnitedStates, Canada or Mexico is specified as the CountryCode, then the PostalCode will be validated based on the country specified.
If anything other than UnitedStates, Canada or Mexico is specified as the CountryCode, including None, then the PostalCode will be considered valid if it is valid for any country.

Card Verification and Addition

The following is a list of form variables that may be passed from the user's browser to CardVault Silent Post in addition to those used for a card addition only request in order to perform a card verification and addition request. Some form variables are required while others are not.

Field Name	Required	Description
CreditCard_CSCIndicator	No	Indicates the status of the card security code. This field must be properly set if providing a card security code value.
CreditCard_CSCValue	No	The card security code value. While this field is not required in the Silent Post API, it is strongly recommended that a client-side requirement is imposed in the originating application.
TerminalIdentifier_MerchantCode	Yes	The code assigned to the Merchant by 3DSI which houses the location and terminal where the credit card transaction will be processed.
TerminalIdentifier_LocationCode	Yes	The code assigned to the Location by 3DSI which houses the terminal where the credit card transaction will be processed.
TerminalIdentifier_TerminalCode	Yes	The code assigned to the Terminal by 3DSI where the credit card transaction will be processed.
Transaction_CardVerificationCurrencyCode	No	Specifies the currency in which the card verification is performed. If this field is not provided, the default currency will be USD.

Notes

The data in the above tables must be provided as form variables in the POST, not as query string variables in a GET.

Please review the Secure Approval Suggestions for guidelines on SubmissionID generation and handling.

Variables passed from CardVault Silent Post to the Approval URL

Field Name	Description
SilentPostType	A constant value sent from 3DSI.
SubmissionID	A unique identifier associated with the current attempt to submit data to CardVault. The client should use this value during the approval stage to look up approval data values provided to the user.
APCount	The number of approval values supplied with this submission.
AP#	An individual approval value, with "#" replaced with an integer between 1 and *APCount*. Code at the *ApprovalURL* should confirm that data provided matches data sent to the user's browser and either approve or deny the request based upon this data. These values are not sent to the *ResultURL*.
CardBIN	The BIN of the credit card submitted to CardVault. *This value is only sent to the Approval URL if the merchant is configured to have it sent.*

Notes

The data in the above table will be provided as form variables in the POST.

Please review the Secure Approval Suggestions for guidelines on SubmissionID handling by the Approval URL.

Silent Post Submission Approval Response

In order to ensure a secure end-to-end submission of data via the CardVault Silent Post process, the CardVault system must obtain approval of each submission from the client’s web server. The server can process the approval using other technologies; it must simply accept an HTTP POST, process the data provided in the POST, and send an HTTP response value in the below format.

Example of Approve and Deny Submission Responses

Approve Submission

```
<ApprovalResponse>
  <Succeeded>1</Succeeded>
  <ResponseStatus>001</ResponseStatus>
  <ResponseMessage>Submission approved.</ResponseMessage>
</ApprovalResponse>
```

Deny Submission

```
<ApprovalResponse>
  <Succeeded>0</Succeeded>
  <ResponseStatus>000</ResponseStatus>
  <ResponseMessage>Submission denied.</ResponseMessage>
</ApprovalResponse>
```

Variables passed from CardVault Silent Post to the Result URL

Card Addition Only

The following is a list of form variables that may be passed from CardVault Silent Post to the merchant's Result URL.

Field Name	Description
SilentPostType	A constant value sent from 3DSI.
SubmissionID	A unique identifier associated with the current attempt to submit data to CardVault.
Approved	A boolean value (0 for Denied or 1 for Approved) indicating whether approval of storage of the card succeeded.
ApprovalResponseStatus	The ResponseStatus value returned from the client's *ApprovalURL*.
ApprovalResponseMessage	The ResponseMessage value returned from the client's *ApprovalURL*.
CardStored	A boolean value (0 for Failure or 1 for Success) indicating whether storage of the card succeeded.
SubmissionStatusMessage	A message describing the overall status of the submission and any card verification transaction validation failures.
Token	The value of the token representing the stored card.
PTCount	The number of pass-through values supplied with this submission.
PT#	An individual pass-through value, with "#" replaced with an integer between 1 and *PTCount*. These values will be passed back as form variables to the *ResultURL* to be utilized by custom logic on the client's web server as dictated by client requirements. These values are not sent to the Approval URL.
CardType	The type of the stored credit card. *This value is only sent to the Result URL if the merchant is configured to have it sent.*
CardEnd	The last 4 digits of the stored credit card. *This value is only sent to the Result URL if the merchant is configured to have it sent.*
ExpirationDate	The expiration date of the stored credit card. *This value is only sent to the Result URL if the merchant is configured to have it sent.*
CardFundingType	Indicates whether the card is a debit card, a credit card or a prepaid card. *This value is only sent to the Result URL if the merchant is configured to have it sent.*
CardBIN	The BIN of the credit card submitted to CardVault. *This value is only sent to the Result URL if the merchant is configured to have it sent.*

Card Verification and Addition

In addition to those found in the above table, the following is a list of form variables that are passed from CardVault Silent Post to the merchant’s Result URL for a card verification and addition request.

Field Name	Description
AddressAVSResponse	A value indicating the results of the Address Line 1 Address Verification check.
PostalCodeAVSResponse	A value indicating the results of the Postal Code Address Verification check.
CardSecurityCodeResponse	A value indicating the results of the Card Security Code check.
CardVerified	A Boolean value (0 for Failure or 1 for Success) indicating whether the card verification succeeded.
ProcessorAVSResponse	A value indicating the results of Address Verification returned from the processor.
ProcessorCardSecurityCodeResponse	A value indicating the results of the Card Security Code check returned from the processor.
TransactionKey	A value uniquely identifying the card verification transaction.
ProcessorResponse	The response, if any, returned from the processor.
CreditCardTransactionFailedSubstatus	Results, if any, indicating why a card verification transaction failed.

Notes

This data will be provided as form variables in the POST.

Silent Post Results Response

Upon completion of processing of a submission, the CardVault Silent Post process will send the submission's status information to the client's web server. The server can process the result using other technologies; it must simply accept an HTTP POST, process the data provided in the POST, and send an HTTP response value in the below format.

Example of Silent Post Results Responses

Success Response

```
<ResultResponse>
  <Succeeded>1</Succeeded>
  <ResponseStatus>001</ResponseStatus>
  <ResponseMessage>Result approved.</ResponseMessage>
</ResultResponse>
```

Failure Response

```
<ResultResponse>
  <Succeeded>0</Succeeded>
  <ResponseStatus>000</ResponseStatus>
  <ResponseMessage>Result failed.</ResponseMessage>
</ResultResponse>
```

QueryString variables passed from CardVault Silent Post to the browser in the browser redirect

Card Addition Only

The following is a list of form variables that may be passed from CardVault Silent Post to the merchant's browser.

Field Name	Description
SubmissionID	A unique identifier associated with the current attempt to submit data to CardVault.
Approved	A boolean value (0 for Denied or 1 for Approved) indicating whether approval of storage of the card succeeded.
ApprovalResponseStatus	The ResponseStatus value returned from the *ApprovalURL*.
ApprovalResponseMessage	The ResponseMessage value returned from the *ApprovalURL*.
CardStored	A boolean value (0 for Failure or 1 for Success) indicating whether storage of the card succeeded.
SubmissionStatusMessage	A message describing the overall status of the submission.
ResultPosted	A boolean value (0 for Failure or 1 for Success) indicating whether post to the *ResultURL* succeeded.
ResultPostMessage	A message describing the overall status of the post to the *ResultURL*.
ResultPostResponseStatus	The ResponseStatus value returned from the *ResultURL*.
ResultPostResponseMessage	The ResponseMessage value returned from the *ResultURL*.
Token	The value of the token representing the stored card. *This value is only sent to the browser if the merchant is configured to have it sent.*
CardFundingType	Indicates whether the card is a debit card, a credit card or a prepaid card. *This value is only sent to the browser if the merchant is configured to have it sent.*

Card Verification and Addition

In addition to those found in the above table, the following is a list of form variables that are passed from CardVault Silent Post to the merchant's browser for a card verification and addition request.

Field Name	Description
AddressAVSResponse	A value indicating the results of the Address Line 1 Address Verification check.
PostalCodeAVSResponse	A value indicating the results of the Postal Code Address Verification check.
CardSecurityCodeResponse	A value indicating the results of the Card Security Code check.
CardVerified	A Boolean value (0 for Failure or 1 for Success) indicating whether the card verification succeeded.
ProcessorAVSResponse	A value indicating the results of Address Verification returned from the processor.
ProcessorCardSecurityCodeResponse	A value indicating the results of the Card Security Code check returned from the processor.
TransactionKey	A value uniquely identifying the card verification transaction.
ProcessorResponse	The response, if any, returned from the processor.
CreditCardTransactionFailedSubstatus	Results, if any, indicating why a card verification transaction failed.

ResponseType = "Return" XML Response Structure

If a value of "Return" is provided in the ResponseType form variable in the POST from the user's browser, result data will be provided in a response to the POST in the below format. A browser redirect will not occur.

ResponseType Value	Behavior
Return	Result Data is returned to the user's browser in the POST response as an XML fragment.
Redirect (default)	Result Data is returned in the QueryString via the BrowserRedirectURL.

Example of Success and Failure Submission Responses for a Card Addition Only web service call

Success Submission

```
<SilentPostResults>
  <SubmissionId>Test1</SubmissionId>
  <Approved>1</Approved>
  <ApprovalResponseStatus>001</ApprovalResponseStatus>
  <ApprovalResponseMessage>Submission approved.</ApprovalResponseMessage>
  <CardStored>1</CardStored>
  <ResultPosted>1</ResultPosted>
  <ResultPostMessage>Success.</ResultPostMessage>
  <ResultPostResponseStatus>001</ResultPostResponseStatus>
  <ResultPostResponseMessage>Result approved.</ResultPostResponseMessage>
  <SubmissionStatusMessage>Success.</SubmissionStatusMessage>
  <Token />
</SilentPostResults>
```

Failure Submission

```
<SilentPostResults>
  <SubmissionId>Test1</SubmissionId>
  <Approved>1</Approved>
  <ApprovalResponseStatus>001</ApprovalResponseStatus>
  <ApprovalResponseMessage>Submission approved.</ApprovalResponseMessage>
  <CardStored>0</CardStored>
  <ResultPosted>1</ResultPosted>
  <ResultPostMessage>Success.</ResultPostMessage>
  <ResultPostResponseStatus>001</ResultPostResponseStatus>
  <ResultPostResponseMessage>Result approved.</ResultPostResponseMessage>
  <SubmissionStatusMessage>The credit card ending in 1111 is already being stored within this location.</SubmissionStatusMessage>
  <Token />
</SilentPostResults>
```

Silent Post Secure Approval Guidelines

The following is a list of guidelines to implement a secure Silent Post approval process to prevent misuse of your merchant account.

It is imperative for the Silent Post process that only valid requests are approved at the approval URL. Invalid requests should be denied to prevent third-parties from unauthorized card verification transactions via your merchant account; as well as to prevent non-customer cards from being stored on your 3DSI account. These guidelines will ensure a secure implementation of the Silent Post approval process:

1. The SubmissionID value should be unique for each Silent Post request and should be generated using a degree of randomness to prevent a third-party from guessing valid values. The SubmissionID should be generated and stored on the merchant's web server for each Silent Post request.

2. Each call to the merchant's Silent Post approval URL should look-up the provided SubmissionID to verify that it was generated by the merchant's web site; confirming that it exists in the merchant's SubmissionID storage. This is in addition to the look-up and verification of the provided AP# values.

3. Each SubmissionID should only be valid for a single approval submission. Therefore, each SubmissionID should be stored with a flag to indicate whether or not it has been used for an approval request. Upon successful validation and approval of a Silent Post approval request, the SubmissionID should be flagged as 'used' and future Silent Post approval requests with the same SubmissionID should be denied.

Silent Post URLs

The following URLs are used in the form that is rendered to the user's browser depending on whether one is using the Card Addition Only or Card Verification and Addition feature of Silent Post. PWSDemo is 3DSI's Certification/Test platform, whereas Production is the live production platform:

Card Addition Only

PWSDemo

https://silentpost.pwsdemo.com/SilentPostService.svc/SilentPostAddCard

Production

https://silentpost.paymentworksuite.com/SilentPostService.svc/SilentPostAddCard

Card Verification and Addition

PWSDemo

https://silentpost.pwsdemo.com/SilentPostService.svc/SilentPostVerifyAndAddCard

Production

https://silentpost.paymentworksuite.com/SilentPostService.svc/SilentPostVerifyAndAddCard

Code Samples and Notes

Silent Post HTML Examples

Summary

These examples are intended to show only the base HTML elements required to perform a silent post. It is to each client's discretion to create an aesthetically-pleasing web page that contains such a form.

Example Form to be Rendered to the User's Browser For Adding a Card without Bank Verification

```
<html>
  <body>
    <form action="https://silentpost.pwsdemo.com/SilentPostService.svc/SilentPostAddCard" method="POST" onsubmit="return validate();">

      <!-- // mandatory Silent Post fields -->

      <input type="hidden" name="CustomerIdentifier_MerchantCode" value="MyMerchantCode"/><br>
      <input type="hidden" name="CustomerIdentifier_LocationCode" value="MyLocationCode"/><br>
      <input type="hidden" name="CustomerIdentifier_CustomerCode" value="MyCustomerCode"/><br>

      <input type="hidden"
             name="ResultURL"
             value="https://www.{place valid domain here}.com/SilentPostResult.asp"/><br>

      <input type="hidden"
             name="BrowserRedirectURL"
             value="https://www.{place valid domain here}.com/MyBrowserRedirectPage.asp"/><br>

      <input type="hidden"
             name="SubmissionID"
             value="put a unique number for each submission here"/><br>

      <!-- // optional fields passed all the way through (round tripped) -->

      Token: <input type="text" name="Token" value="TestCard1"/><br><br>

      PTCount: <input type="text" name="PTCount" value="3"               /><br>
      PT1:     <input type="text" name="PT1"     value="TestPassThrough1"/><br>
      PT2:     <input type="text" name="PT2"     value="TestPassThrough2"/><br>
      PT3:     <input type="text" name="PT3"     value="TestPassThrough3"/><br>

      <!-- // optional approval fields passed to browser to 3DSI to merchant for submission approval -->

      APCount: <input type="text" name="APCount" value="2"             /><br>
      AP1:     <input type="text" name="AP1"     value="ApprovalValue1"/><br>
      AP2:     <input type="text" name="AP2"     value="ApprovalValue2"/><br>

      <!-- // mandatory card fields -->

      CardAccountNumber: <input type="text" name="CreditCard_CardAccountNumber" value="4111111111111111"/><br>
```

```
<!-- // optional card fields -->
ExpirationMonth:
<input type="text"
       id="CreditCard_ExpirationMonth"
       name="CreditCard_ExpirationMonth"
       value="01"
       onchange='toggleHighlight(this);'/><br>

ExpirationYear:
<input type="text"
       id="CreditCard_ExpirationYear"
       name="CreditCard_ExpirationYear"
       value="18"
       onchange='toggleHighlight(this);'/><br>

Notes:
<input type="text"
       id="CreditCard_Notes"
       name="CreditCard_Notes"
       value="Test Note"
       onchange='toggleHighlight(this);'/><br>

DisplayNotesDuringTransaction:
<input type="text"
       id="CreditCard_DisplayNotesDuringTransaction"
       name="CreditCard_DisplayNotesDuringTransaction"
       value="1"
       onchange='toggleHighlight(this);'/><br>

Token:
<input type="text"
       id="CreditCard_Token"
       name="CreditCard_Token"
       value="TestToken147"
       onchange='toggleHighlight(this);'/><br>

FriendlyName:
<input type="text"
       id="CreditCard_FriendlyName"
       name="CreditCard_FriendlyName"
       value="Test Card 1111"
       onchange='toggleHighlight(this);'/><br>

CardType:
<input type="text"
       id="CreditCard_CardType"
       name="CreditCard_CardType"
       value="VISA"
       onchange='toggleHighlight(this);'/><br>
```

```
NameOnCard:
<input type="text"
       id="CreditCard_NameOnCard"
       name="CreditCard_NameOnCard"
       value="John Q Smith"
       onchange='toggleHighlight(this);'/><br>

Cardholder_Title:
<input type="text"
       id="CreditCard_Cardholder_Title"
       name="CreditCard_Cardholder_Title"
       value="Mr."
       onchange='toggleHighlight(this);'/><br>

Cardholder_FirstName:
<input type="text"
       id="CreditCard_Cardholder_FirstName"
       name="CreditCard_Cardholder_FirstName"
       value="John"
       onchange='toggleHighlight(this);'/><br>

Cardholder_MiddleName:
<input type="text"
       id="CreditCard_Cardholder_MiddleName"
       name="CreditCard_Cardholder_MiddleName"
       value="Quentin"
       onchange='toggleHighlight(this);'/><br>

Cardholder_LastName:
<input type="text"
       id="CreditCard_Cardholder_LastName"
       name="CreditCard_Cardholder_LastName"
       value="Smith"
       onchange='toggleHighlight(this);'/><br>

Cardholder_Suffix:
<input type="text"
       id="CreditCard_Cardholder_Suffix"
       name="CreditCard_Cardholder_Suffix"
       value="Jr."
       onchange='toggleHighlight(this);'/><br>

Cardholder_Phone:
<input type="text"
       id="CreditCard_Cardholder_Phone"
       name="CreditCard_Cardholder_Phone"
       value="800-555-1212"
       onchange='toggleHighlight(this);'/><br>

Cardholder_Fax:
<input type="text"
       id="CreditCard_Cardholder_Fax"
       name="CreditCard_Cardholder_Fax"
       value="800-555-1234"
       onchange='toggleHighlight(this);'/><br>

Cardholder_Email_DisplayAs:
```

```
<input type="text"
       id="CreditCard_Cardholder_Email_DisplayAs"
       name="CreditCard_Cardholder_Email_DisplayAs"
       value="J Q Smith"
       onchange='toggleHighlight(this);'/><br>

Cardholder_Email_Address:
<input type="text"
       id="CreditCard_Cardholder_Email_Address"
       name="CreditCard_Cardholder_Email_Address"
       value=”j.smith@a.com”
       onchange='toggleHighlight(this);'/><br>

BillingAddress_AddressLine1:
<input type="text"
       id="CreditCard_BillingAddress_AddressLine1"
       name="CreditCard_BillingAddress_AddressLine1"
       value="12 Main St"
       onchange='toggleHighlight(this);'/><br>

BillingAddress_AddressLine2:
<input type="text"
       id="CreditCard_BillingAddress_AddressLine2"
       name="CreditCard_BillingAddress_AddressLine2"
       value="Suite 100"
       onchange='toggleHighlight(this);'/><br>

BillingAddress_City:
<input type="text"
       id="CreditCard_BillingAddress_City"
       name="CreditCard_BillingAddress_City"
       value="New City"
       onchange='toggleHighlight(this);'/><br>

BillingAddress_PostalCode:
<input type="text"
       id="CreditCard_BillingAddress_PostalCode"
       name="CreditCard_BillingAddress_PostalCode"
       value="32543"
       onchange='toggleHighlight(this);'/><br>

BillingAddress_CountryCode:
<input type="text"
       id="CreditCard_BillingAddress_CountryCode"
       name="CreditCard_BillingAddress_CountryCode"
       value="USA"
       onchange='toggleHighlight(this);'/><br>

BillingAddress_StateProvinceCode:
<input type="text"
       id="CreditCard_BillingAddress_StateProvinceCode"
       name="CreditCard_BillingAddress_StateProvinceCode"
       value="FL"
       onchange='toggleHighlight(this);'/><br>

ShippingAddress_AddressLine1:
<input type="text"
```

```
         id="CreditCard_ShippingAddress_AddressLine1"
         name="CreditCard_ShippingAddress_AddressLine1"
         value="34 Second St"
         onchange='toggleHighlight(this);'/><br>

  ShippingAddress_AddressLine2:
  <input type="text"
         id="CreditCard_ShippingAddress_AddressLine2"
         name="CreditCard_ShippingAddress_AddressLine2"
         value="Suite 200"
         onchange='toggleHighlight(this);'/><br>

  ShippingAddress_City:
  <input type="text"
         id="CreditCard_ShippingAddress_City"
         name="CreditCard_ShippingAddress_City"
         value="Old City"
         onchange='toggleHighlight(this);'/><br>

  ShippingAddress_PostalCode:
  <input type="text"
         id="CreditCard_ShippingAddress_PostalCode"
         name="CreditCard_ShippingAddress_PostalCode"
         value="31301"
         onchange='toggleHighlight(this);'/><br>

  ShippingAddress_CountryCode:
  <input type="text"
         id="CreditCard_ShippingAddress_CountryCode"
         name="CreditCard_ShippingAddress_CountryCode"
         value="USA"
         onchange='toggleHighlight(this);'/><br>

  ShippingAddress_StateProvinceCode:
  <input type="text"
         id="CreditCard_ShippingAddress_StateProvinceCode"
         name="CreditCard_ShippingAddress_StateProvinceCode"
         value="GA"
         onchange='toggleHighlight(this);'/><br>

  <br>

  <input type="submit" value="Submit"/>

</form>

<script type="text/javascript">
  function validate()
  {
    var message = '';

    var billingAddressPostalCode  = document.getElementById('CreditCard_BillingAddress_PostalCode');
    var shippingAddressPostalCode = document.getElementById('CreditCard_ShippingAddress_PostalCode');

    if(!validateMultipleLengths(billingAddressPostalCode.value, 5, 9))
    {
      message += 'Please enter a valid Billing Address Postal Code.\n';
```

```
          toggleHighlight(billingAddressPostalCode, "#EE0000");
        }

        if(!validateMultipleLengths(shippingAddressPostalCode.value, 5, 9))
        {
          message += 'Please enter a valid Shipping Address Postal Code.\n';
          toggleHighlight(shippingAddressPostalCode, "#EE0000");
        }

        if(message.length != 0)
        {
          return displayMessage(message);
        }

        return true;
      }

      function validateNumber(val, len)
      {
        var isCorrectLength = (val.length != len) ? false : true;
        var isNumber        = !isNaN(parseInt(val)) && isFinite(val);

        return isCorrectLength && isNumber;
      }

      function validateMultipleLengths(val, len1, len2)
      {
        var resultNumber1 = validateNumber(val, len1);
        var resultNumber2 = validateNumber(val, len2);

        return resultNumber1 || resultNumber2;
      }

      function toggleHighlight(ref, color)
      {
        if(color == null)
        {
          ref.style.borderColor = '';
        }
        else
        {
         ref.style.borderColor = color;
        }
      }

      function displayMessage(message)
      {
        alert(message);
        return false;
      }
    </script>
  </body>
</html>
```

Example Form to be Rendered to the User’s Browser For Card Verification and Addition

```
<html>
  <body>
    <form action=="https://silentpost.pwsdemo.com/SilentPostService.svc/SilentPostVerifyAndAddCard" method="POST" onsubmit="return validate();">

      <!-- // mandatory Silent Post fields -->

      <input type="hidden" name="CustomerIdentifier_MerchantCode" value="MyMerchantCode"/><br>
      <input type="hidden" name="CustomerIdentifier_LocationCode" value="MyLocationCode"/><br>
      <input type="hidden" name="CustomerIdentifier_CustomerCode" value="MyCustomerCode"/><br>
      <input type="hidden" name="TerminalIdentifier_MerchantCode" value="MyTerminalMerchantCode"/><br>
      <input type="hidden" name="TerminalIdentifier_LocationCode" value="MyTerminalLocationCode"/><br>
      <input type="hidden" name="TerminalIdentifier_TerminalCode" value="MyTerminalTerminalCode"/><br>
      <input type="hidden"
             name="ResultURL"
             value="https://www.{place valid domain here}.com/SilentPostResult.asp"/><br>

      <input type="hidden"
             name="BrowserRedirectURL"
             value="https://www.{place valid domain here}.com/MyBrowserRedirectPage.asp"/><br>

      <input type="hidden"
             name="SubmissionID"
             value="put a unique number for each submission here"/><br>

      <!-- // optional fields passed all the way through (round tripped) -->

      Token: <input type="text" name="Token" value="TestCard1"/><br><br>

      PTCount: <input type="text" name="PTCount" value="3"               /><br>
      PT1:     <input type="text" name="PT1"     value="TestPassThrough1"/><br>
      PT2:     <input type="text" name="PT2"     value="TestPassThrough2"/><br>
      PT3:     <input type="text" name="PT3"     value="TestPassThrough3"/><br>

      <!-- // optional approval fields passed to browser to 3DSI to merchant for submission approval -->

      APCount: <input type="text" name="APCount" value="2"             /><br>
      AP1:     <input type="text" name="AP1"     value="ApprovalValue1"/><br>
      AP2:     <input type="text" name="AP2"     value="ApprovalValue2"/><br>

      <!-- // mandatory card fields -->

      CardAccountNumber: <input type="text" name="CreditCard_CardAccountNumber” value="4111111111111111"/><br>

      <!-- // optional card fields -->

      CSCIndicator:
      <input type="text"
             id="CreditCard_CSCIndicatorCreditCard_CSCIndicator"
             name="CreditCard_CSCIndicatorCreditCard_CSCIndicator"
             value="Provided"
             onchange="toggleHighlight(this);"/><br>


      CSCValue:
      <input type="text"
```

```
           id="CreditCard_CSCIndicatorCreditCard_CSCValue"
           name="CreditCard_CSCIndicatorCreditCard_CSCValue"
           value="124"
           onchange="toggleHighlight(this);"/><br>

    ExpirationMonth:
    <input type="text"
           id="CreditCard_ExpirationMonth"
           name="CreditCard_ExpirationMonth"
           value="01"
           onchange="toggleHighlight(this);"/><br>

    ExpirationYear:
    <input type="text"
           id="CreditCard_ExpirationYear"
           name="CreditCard_ExpirationYear"
           value="18"
           onchange="toggleHighlight(this);"/><br>

    Notes:
    <input type="text"
           id="CreditCard_Notes"
           name="CreditCard_Notes"
           value="Test Note"
           onchange="toggleHighlight(this);"/><br>

    DisplayNotesDuringTransaction:
    <input type="text"
           id="CreditCard_DisplayNotesDuringTransaction"
           name="CreditCard_DisplayNotesDuringTransaction"
           value="1"
           onchange="toggleHighlight(this);"/><br>

    Token:
    <input type="text"
           id="CreditCard_Token"
           name="CreditCard_Token"
           value="TestToken147"
           onchange="toggleHighlight(this);"/><br>

    FriendlyName:
    <input type="text"
           id="CreditCard_FriendlyName"
           name="CreditCard_FriendlyName"
           value="Test Card 1111"
           onchange="toggleHighlight(this);"/><br>

    CardType:
    <input type="text"
           id="CreditCard_CardType"
           name="CreditCard_CardType"
           value="VISA"
           onchange="toggleHighlight(this);"/><br>

    NameOnCard:
    <input type="text"
           id="CreditCard_NameOnCard"
```

```
               name="CreditCard_NameOnCard"
               value="John Q Smith"
               onchange="toggleHighlight(this);"/><br>

        Cardholder_Title:
        <input type="text"
               id="CreditCard_Cardholder_Title"
               name="CreditCard_Cardholder_Title"
               value="Mr."
               onchange="toggleHighlight(this);"/><br>

        Cardholder_FirstName:
        <input type="text"
               id="CreditCard_Cardholder_FirstName"
               name="CreditCard_Cardholder_FirstName"
               value="John"
               onchange="toggleHighlight(this);"/><br>

        Cardholder_MiddleName:
        <input type="text"
               id="CreditCard_Cardholder_MiddleName"
               name="CreditCard_Cardholder_MiddleName"
               value="Quentin"
               onchange="toggleHighlight(this);"/><br>

        Cardholder_LastName:
        <input type="text"
               id="CreditCard_Cardholder_LastName"
               name="CreditCard_Cardholder_LastName"
               value="Smith"
               onchange="toggleHighlight(this);"/><br>

        Cardholder_Suffix:
        <input type="text"
               id="CreditCard_Cardholder_Suffix"
               name="CreditCard_Cardholder_Suffix"
               value="Jr."
               onchange="toggleHighlight(this);"/><br>

        Cardholder_Phone:
        <input type="text"
               id="CreditCard_Cardholder_Phone"
               name="CreditCard_Cardholder_Phone"
               value="800-555-1212"
               onchange="toggleHighlight(this);"/><br>

        Cardholder_Fax:
        <input type="text"
               id="CreditCard_Cardholder_Fax"
               name="CreditCard_Cardholder_Fax"
               value="800-555-1234"
               onchange="toggleHighlight(this);"/><br>

        Cardholder_Email_DisplayAs:
        <input type="text"
               id="CreditCard_Cardholder_Email_DisplayAs"
               name="CreditCard_Cardholder_Email_DisplayAs"
```

```
       value="J Q Smith"
       onchange="toggleHighlight(this);"/><br>

Cardholder_Email_Address:
<input type="text"
       id="CreditCard_Cardholder_Email_Address"
       name="CreditCard_Cardholder_Email_Address"
       value=”j.smith@a.com”
       onchange="toggleHighlight(this);"/><br>

BillingAddress_AddressLine1:
<input type="text"
       id="CreditCard_BillingAddress_AddressLine1"
       name="CreditCard_BillingAddress_AddressLine1"
       value="12 Main St"
       onchange="toggleHighlight(this);"/><br>

BillingAddress_AddressLine2:
<input type="text"
       id="CreditCard_BillingAddress_AddressLine2"
       name="CreditCard_BillingAddress_AddressLine2"
       value="Suite 100"
       onchange="toggleHighlight(this);"/><br>

BillingAddress_City:
<input type="text"
       id="CreditCard_BillingAddress_City"
       name="CreditCard_BillingAddress_City"
       value="New City"
       onchange="toggleHighlight(this);"/><br>

BillingAddress_PostalCode:
<input type="text"
       id="CreditCard_BillingAddress_PostalCode"
       name="CreditCard_BillingAddress_PostalCode"
       value="32543"
       onchange="toggleHighlight(this);"/><br>

BillingAddress_CountryCode:
<input type="text"
       id="CreditCard_BillingAddress_CountryCode"
       name="CreditCard_BillingAddress_CountryCode"
       value="USA"
       onchange="toggleHighlight(this);"/><br>

BillingAddress_StateProvinceCode:
<input type="text"
       id="CreditCard_BillingAddress_StateProvinceCode"
       name="CreditCard_BillingAddress_StateProvinceCode"
       value="FL"
       onchange="toggleHighlight(this);"/><br>

ShippingAddress_AddressLine1:
<input type="text"
       id="CreditCard_ShippingAddress_AddressLine1"
       name="CreditCard_ShippingAddress_AddressLine1"
       value="34 Second St"
```

```
         onchange="toggleHighlight(this);"/><br>

  ShippingAddress_AddressLine2:
  <input type="text"
         id="CreditCard_ShippingAddress_AddressLine2"
         name="CreditCard_ShippingAddress_AddressLine2"
         value="Suite 200"
         onchange="toggleHighlight(this);"/><br>

  ShippingAddress_City:
  <input type="text"
         id="CreditCard_ShippingAddress_City"
         name="CreditCard_ShippingAddress_City"
         value="Old City"
         onchange="toggleHighlight(this);"/><br>

  ShippingAddress_PostalCode:
  <input type="text"
         id="CreditCard_ShippingAddress_PostalCode"
         name="CreditCard_ShippingAddress_PostalCode"
         value="31301"
         onchange="toggleHighlight(this);"/><br>

  ShippingAddress_CountryCode:
  <input type="text"
         id="CreditCard_ShippingAddress_CountryCode"
         name="CreditCard_ShippingAddress_CountryCode"
         value="USA"
         onchange="toggleHighlight(this);"/><br>

  ShippingAddress_StateProvinceCode:
  <input type="text"
         id="CreditCard_ShippingAddress_StateProvinceCode"
         name="CreditCard_ShippingAddress_StateProvinceCode"
         value="GA"
         onchange="toggleHighlight(this);"/><br>

  <br>

  <input type="submit" value="Submit"/>


</form>

<script type="text/javascript">
  function validate()
  {
    var message = '';

    var cardCSCMapping = new Array();
    cardCSCMapping["AMEX"]     = 4;
    cardCSCMapping["VISA"]     = 3;
    cardCSCMapping["DISCOVER"] = 3;

    var creditCardTye             = document.getElementById('CreditCard_CardType');
    var cscValue                  = document.getElementById('CreditCard_CSCIndicatorCreditCard_CSCValue');
    var billingAddressPostalCode  = document.getElementById('CreditCard_BillingAddress_PostalCode');
    var shippingAddressPostalCode = document.getElementById('CreditCard_ShippingAddress_PostalCode');

    if(!validateNumber(cscValue.value, cardCSCMapping[creditCardTye.value.toUpperCase()]))
    {
      message += 'Please enter a valid Security Code for ' + creditCardTye.value.toUpperCase() + '.\n';
      toggleHighlight(cscValue, "#EE0000");
    }

    if(!validateMultipleLengths(billingAddressPostalCode.value, 5, 9))
    {
      message += 'Please enter a valid Billing Address Postal Code.\n';
      toggleHighlight(billingAddressPostalCode, "#EE0000");
    }

    if(!validateMultipleLengths(shippingAddressPostalCode.value, 5, 9))
    {
      message += 'Please enter a valid Shipping Address Postal Code.\n';
      toggleHighlight(shippingAddressPostalCode, "#EE0000");
    }

    if(message.length != 0)
    {
      return displayMessage(message);
    }

    return true;
  }

  function validateNumber(val, len)
  {
    var isCorrectLength = (val.length != len) ? false : true;
    var isNumber        = !isNaN(parseInt(val)) && isFinite(val);

    return isCorrectLength && isNumber;
  }

  function validateMultipleLengths(val, len1, len2)
  {
    var resultNumber1 = validateNumber(val, len1);
    var resultNumber2 = validateNumber(val, len2);

    return resultNumber1 || resultNumber2;
  }

  function toggleHighlight(ref, color)
  {
    if(color == null)
    {
      ref.style.borderColor = '';
    }
    else
    {
     ref.style.borderColor = color;
    }
  }

  function displayMessage(message)
```

```
      {
        alert(message);
        return false;
      }
    </script>
  </body>
</html>
```

Sample HTML Page Demonstrating the Fundamentals of a CardVault Silent Post Add Using AJAX

```
<html>
  <script>
  function htmlEscape(str)
  {
    return String(str).replace(/&/g, '&amp;')
                      .replace(/"/g, '&quot;')
                      .replace(/'/g, '&#39;')
                      .replace(/</g, '&lt;')
                      .replace(/>/g, '&gt;');
  }

  function createRequest(method, url)
  {
    var xhr = new XMLHttpRequest();

    if ('withCredentials' in xhr)
    {
      // XHR for Chrome/FireFox/Opera/Safari
      xhr.open(method, url, true);
    }
    else if (typeof XDomainRequest != 'undefined')
    {
      // XDomainRequest for IE
      xhr = new XDomainRequest();
      xhr.open(method, url);
    }
    else
    {
      // CORS is not supported
      xhr = null;
    }

    return xhr;
  }

  function sendRequest()
  {
    var addCardUrl = 'https://silentpost.pwsdemo.com/SilentPostService.svc/SilentPostAddCard';
    var resultUrl  = 'https://www.YourDomain.com/YourSilentPostResultPage.aspx';

    document.getElementById("resultDiv").innerHTML = '&nbsp;';
    document.getElementById("statusDiv").innerHTML = 'Sending request. Please wait...';

    var xhr = createRequest('POST', addCardUrl);

    if (!xhr)
    {
      alert('CORS is not supported.');
      return;
    }

    xhr.setRequestHeader('Content-type', 'application/x-www-form-urlencoded');

    xhr.onreadystatechange=function()
    {
```

```
    if (xhr.readyState == 4 && xhr.status == 200)
    {
      var responseText = htmlEscape(xhr.responseText);
      responseText = responseText.replace(/&gt;/g, '&gt;<br/>');
      document.getElementById("resultDiv").innerHTML = responseText;
    }

    document.getElementById("statusDiv").innerHTML  = 'Complete.';
  }

  xhr.onerror = function()
  {
    alert('An error occurred making the request.');
  };

  var data = 'CustomerIdentifier_MerchantCode=YourMerchantCode' +
            '&CustomerIdentifier_LocationCode=YourLocationCode' +
            '&CustomerIdentifier_CustomerCode=YourCustomerCode' +
            '&ResponseType=Return' +
            '&BrowserRedirectUrl=' +
            '&ResultUrl=' + htmlEscape(resultUrl) +
            '&APCount=1' +
            '&AP1=Approve' +
            '&PTCount=1' +
            '&PT1=Approve' +
            '&SubmissionID=Test1' +
            '&CreditCard_Notes=Test notes.' +
            '&CreditCard_DisplayNotesDuringTransaction=True' +
            '&CreditCard_Token=' +
            '&CreditCard_FriendlyName=SilentPost Test' +
            '&CreditCard_CardType=VISA' +
            '&CreditCard_NameOnCard=John A Smith' +
            '&CreditCard_CardAccountNumber=4111111111111111' +
            '&CreditCard_ExpirationMonth=12' +
            '&CreditCard_ExpirationYear=13' +
            '&CreditCard_BillingAddress_AddressLine1=' +
            '&CreditCard_BillingAddress_AddressLine2=' +
            '&CreditCard_BillingAddress_City=' +
            '&CreditCard_BillingAddress_PostalCode=' +
            '&CreditCard_BillingAddress_CountryCode=' +
            '&CreditCard_BillingAddress_StateProvinceCode=' +
            '&CreditCard_ShippingAddress_AddressLine1=' +
            '&CreditCard_ShippingAddress_AddressLine2=' +
            '&CreditCard_ShippingAddress_City=' +
            '&CreditCard_ShippingAddress_PostalCode=' +
            '&CreditCard_ShippingAddress_CountryCode=' +
            '&CreditCard_ShippingAddress_StateProvinceCode=';

  xhr.send(data);
}
</script>
<body>
  <table width=700>
    <tr><td colspan="2"><u>Response</u></td></tr>
    <tr><td colspan="2"><div id="resultDiv">&nbsp;</div></td></tr>
    <tr><td colspan="2">&nbsp;</td></tr>
    <tr><td colspan="2"><u>Status</u></td></tr>
```

```
      <tr><td colspan="2"><div id="statusDiv">Ready.</div></td></tr>
      <tr><td colspan="2">&nbsp;</td></tr>
      <tr align="center"><td colspan="2"><button type="button" onclick="JavaScript:sendRequest()">Send Silent POST Request</button></td></tr>
    </table>
  </body>
</html>
```

Silent Post Submission Approval

Summary

The ASP code below is just one example of how Silent Post approval may be implemented by the client. The server can process the approval using other technologies; it must simply accept an HTTP POST, process the data provided in the POST, and send an HTTP response value in the appropriate format.

Example Active Server Page to Approve a User's CardVault Silent Post Submission

```
<%@Language=VBScript%>
<%Option Explicit%>
<%
'*******************************************************************************************************
Response.Buffer = True
Response.Expires = 0

On Error Resume Next

Response.Write(ApproveSilentPost())
'*******************************************************************************************************
Function ApproveSilentPost()
  On Error Resume Next

  Dim bSuccess
  Dim sSubmissionID
  Dim sStoredAP1
  Dim sStoredAP2

  bSuccess = False

  sSubmissionID = Request.Form("SubmissionID")(1)  'Using "(1)" to prevent errors if two instances
                                                   'of the form variable exist

  If (Len(sSubmissionID) > 0) Then
    'TODO: Make a call to your local database to look up the details associated with this SubmissionID
    '      and confirm that it is an expected value. Consider limiting its validity time window.

    sStoredAP1 = "AP1GoodVal"  'Use test value for now
    sStoredAP2 = "AP2GoodVal"  'Use test value for now

    'After retrieving data for this SubmissionID...
    If (      Request.Form("AP1")(1) = sStoredAP1 _
          And Request.Form("AP2")(1) = sStoredAP2  ) Then
      bSuccess = True  'This is a valid Silent Post submission from the user
    End If
  End If

  If (bSuccess = True) Then
    'Approve this submission
    ApproveSilentPost = "<ApprovalResponse>" _
```

```
                          "<Succeeded>1</Succeeded>" _
                          "<ResponseStatus>001</ResponseStatus>" _
                          "<ResponseMessage>Submission approved.</ResponseMessage>" _
                        "</ApprovalResponse>"

  Else
    'Decline this submission
    ApproveSilentPost = "<ApprovalResponse>" _
                          "<Succeeded>0</Succeeded>" _
                          "<ResponseStatus>000</ResponseStatus>" _
                          "<ResponseMessage>Submission denied.</ResponseMessage>" _
                        "</ApprovalResponse>"
  End If
End Function
'*********************************************************************************************************
%>
```

Silent Post Results

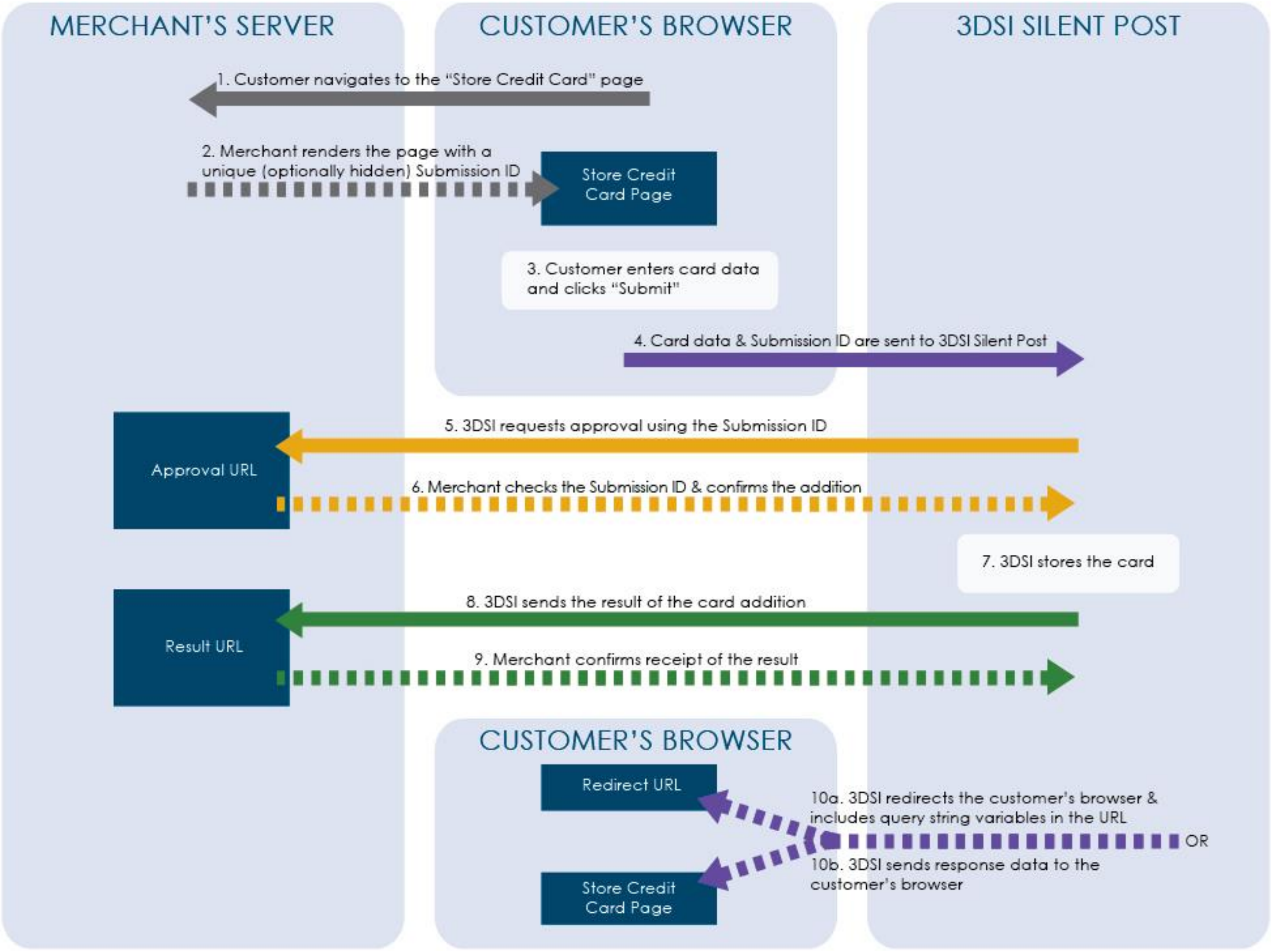
Summary

Upon completion of processing of a submission, the CardVault Silent Post process will send the submission's status information to the client's web server. The ASP code below is just one example of how the card submission's status information may be received by the client at the ResultURL. The server can process the result using other technologies; it must simply accept an HTTP POST, process the data provided in the POST, and send an HTTP response.

Example Active Server Page to be used to process CardVault Silent Post results

```
<%@Language=VBScript%>
<%Option Explicit%>
<%
'********************************************************************************************************
Response.Buffer = True
Response.Expires = 0

On Error Resume Next

Response.Write(SilentPostResult())
'********************************************************************************************************
Function SilentPostResult()
  On Error Resume Next

  Dim sSubmissionID
  Dim sResponseCode
  Dim sResponseStatus
  Dim sResponseMsg
  Dim sCardRefVal
  Dim iFieldCount
  Dim iCurrFieldNum
  Dim sCurrFieldName
  Dim sCurrFieldValue

  sSubmissionID = Request.Form("SubmissionID")(1)  'Using "(1)" to prevent errors if two instances
                                                   'of the form variable exist

  'Store off CardVault response information...
  sSucceeded      = Request.Form("Succeeded")(1)
  sResponseStatus = Request.Form("ResponseStatus")(1)
  sResponseMsg    = Request.Form("ResponseMsg")(1)
  sCardRefVal     = Request.Form("CardRefVal")(1)

  'TODO: Add logic to handle CardVault results

  'Get Pass-through data...

  iFieldCount = CInt(Request.Form("PTCount")(1))
  If (iFieldCount > 0) Then
    For iCurrFieldNum = 1 To iFieldCount  'Attempt to retrieve data for each
                                          'of the specified number of form variables
      sCurrFieldName = "AP" & iCurrFieldNum  'Build name of current field

      sCurrFieldValue = Trim("" & Request.Form(sCurrFieldName)(1))  'Get value for current field name

      'TODO: store each pass-through field value
    Next
  End If

  'TODO: Add logic to set SilentPostResult value below based upon result data.

  'Tell 3DSI that the result data was received
  SilentPostResult = "<ApprovalResponse>" _
                       "<Succeeded>1</Succeeded>" _
                       "<ResponseStatus>001</ResponseStatus>" _
                       "<ResponseMessage>Result approved.</ResponseMessage>" _
                     "</ApprovalResponse>"
End Function
'*******************************************************************************************************%>
```

Appendix A – Silent Post Sequence Diagram



MERCHANT'S SERVER

CUSTOMER'S BROWSER

3DSI SILENT POST

1. Customer navigates to the "Store Credit Card" page

2. Merchant renders the page with a unique (optionally hidden) Submission ID

Store Credit Card Page

3. Customer enters card data and clicks "Submit"

4. Card data & Submission ID are sent to 3DSI Silent Post

Approval URL

5. 3DSI requests approval using the Submission ID

6. Merchant checks the Submission ID & confirms the addition

7. 3DSI stores the card

Result URL

8. 3DSI sends the result of the card addition

9. Merchant confirms receipt of the result

CUSTOMER'S BROWSER

Redirect URL

Store Credit Card Page

10a. 3DSI redirects the customer's browser & includes query string variables in the URL

OR

10b. 3DSI sends response data to the customer's browser

▬▬▬

All *solid-line* data transmissions between servers are HTTP POSTs over SSL (i.e., HTTPS).

▮▮▮▮▮▮▮▮▮▮

Dashed-lines indicate HTTP responses from POSTs, also over SSL.

Appendix B – Expected Values

AddressAVSResponse

None
Matched
NotMatched
Unavailable
Error

CardSecurityCodeResponse

None
Matched
NotMatched
ShouldBeOnCard
Unavailable

CardFundingType

Unknown
Debit
Credit
Prepaid

CardType

Visa
MasterCard
AmericanExpress
Discover
Diners
JCB

PrivateLabel
Unknown

CardVerificationCurrencyCode

Algerian Dinar	DZD
Argentine Peso	ARS
Armenian Dram	AMD
Aruban Florin	AWG
Australian Dollar	AUD
Azerbaijanian Manat	AZN
Bahamian Dollar	BSD
Bangladeshi Taka	BDT
Barbados Dollar	BBD
Belize Dollar	BZD
Bermudian Dollar	BMD
Boliviano	BOB
Bosnia Herzegovina Convertible Marks	BAM
Botswana Pula	BWP
Brazilian Real	BRL
Brunei Dollar	BND
Bulgarian Lev	BGN
Cambodian Riel	KHR
Canadian Dollars	CAD
Cape Verde Escudo	CVE
Cayman Islands Dollar	KYD
Chinese Yuan Renminbi	CNY
Colombian Peso	COP
Costa Rican Colon	CRC
Czech Koruna	CZK
Danish Krone	DKK
Dominican Peso	DOP
East Caribbean Dollar	XCD

Egyptian Pound	EGP
Ethiopian Birr	ETB
Euro	EUR
Falkland Islands Pound	FKP
Fiji Dollar	FJD
Gambian Dalasi	GMD
Georgian Lari	GEL
Ghanaian Cedi	GHS
Gibraltar Pound	GIP
Guatemala Quetzal	GTQ
Guyanese Dollar	GYD
Haitian Gourde	HTG
Honduras Limpera	HNL
Hong Kong Dollar	HKD
Hungarian Forint	HUF
Indian Rupee	INR
Indonesian Rupiah	IDR
Israeli New Shekel	ILS
Jamaican Dollar	JMD
Kazakhstan Tenge	KZT
Kenyan Shilling	KES
Kyrgyzstan Som	KGS
Lebanese Pound	LBP
Macanese Pataca	MOP
Malawi Kwacha	MWK
Malaysian Ringgit	MYR
Maldivian Rufiyaa	MVR
Mauritania Ouguiya	MRO
Mauritian Rupee	MUR
Mexican Peso	MXN
Moldovan Leu	MDL
Mongolian Togrog	MNT

Moroccan Dirham	MAD
Mozambique Metical	MZN
Namibian Dollar	NAD
Nepalese Rupee	NPR
Netherlands Antillean Guilder	ANG
New Taiwan Dollar	TWD
New Zealand Dollar	NZD
Nicaraguan Cordoba	NIO
Nigerian Naira	NGN
Norwegian Krone	NOK
Pakistani Rupee	PKR
Panamanian Balboa	PAB
Papua New Guinean Kina	PGK
Peruvian Sol	PEN
Philippine Peso	PHP
Polish Zloty	PLN
Pound Sterling	GBP
Qatari Rial	QAR
Romanian Leu	RON
Russian Ruble	RUB
Saint Helena Pound	SHP
Salvadoran Colon	SVC
Samoan Tala	WST
Sao Tome Principe Dobra	STD
Saudi Riyal	SAR
Seychelles Rupee	SCR
Sierra Leonean Leone	SLL
Singapore Dollar	SGD
Solomon Islands Dollar	SBD
Somali Shilling	SOS
South African Rand	ZAR
Sri Lankan Rupee	LKR

Swazi Lilangeni	SZL
Swedish Krona	SEK
Swiss Franc	CHF
Tanzanian Shilling	TZS
Thai Baht	THB
Tongan Paanga	TOP
Trinidad Tobago Dollar	TTD
Turkish Lira	TRY
US Dollars	USD
Ukrainian Hryvnia	UAH
United Arab Emirates Dirham	AED
Uruguayan Peso	UYU
Uzbekistan Som	UZS
Yemeni Rial	YER
Zambian Kwacha	ZMW
Zimbabwean Dollar	ZWD

CountryCode

Afghanistan	AFG
Aland Islands	ALA
Albania	ALB
Algeria	DZA
American Samoa	ASM
Andorra	AND
Angola	AGO
Anguilla	AIA
Antarctica	ATA
Antigua And Barbuda	ATG
Argentina	ARG
Armenia	ARM
Aruba	ABW
Australia	AUS

Austria	AUT
Azerbaijan	AZE
Bahamas	BHS
Bahrain	BHR
Bangladesh	BGD
Barbados	BRB
Belarus	BLR
Belgium	BEL
Belize	BLZ
Benin	BEN
Bermuda	BMU
Bhutan	BTN
Bolivia	BOL
Bonaire	BES
Bosnia And Herzegovina	BIH
Botswana	BWA
Bouvet Island	BVT
Brazil	BRA
British Indian Ocean Territory	IOT
British Virgin Islands	VGB
Brunei Darussalam	BRN
Bulgaria	BGR
Burkina Faso	BFA
Burundi	BDI
Cambodia	KHM
Cameroon	CMR
Canada	CAN
Cape Verde	CPV
Cayman Islands	CYM
Central African Republic	CAF
Chad	TCD
Chile	CHL

China	CHN
Christmas Island	CXR
Cocos Keeling Islands	CCK
Colombia	COL
Comoros	COM
Congo	COG
Cook Islands	COK
Costa Rica	CRI
Croatia	HRV
Curacao	CUW
Cyprus	CYP
Czech Republic	CZE
Democratic Republicofthe Congo	COD
Denmark	DNK
Djibouti	DJI
Dominica	DMA
Dominican Republic	DOM
Ecuador	ECU
Egypt	EGY
El Salvador	SLV
Equatorial Guinea	GNQ
Eritrea	ERI
Estonia	EST
Ethiopia	ETH
Falkland Islands	FLK
Faroe Islands	FRO
Federated States Of Micronesia	FSM
Fiji	FJI
Finland	FIN
France	FRA
French Guiana	GUF
French Polynesia	PYF

French Southern Territories	ATF
Gabon	GAB
Gambia	GMB
Georgia	GEO
Germany	DEU
Ghana	GHA
Gibraltar	GIB
Greece	GRC
Greenland	GRL
Grenada	GRD
Guadeloupe	GLP
Guam	GUM
Guatemala	GTM
Guernsey	GGY
Guinea	GIN
Guinea Bissau	GNB
Guyana	GUY
Haiti	HTI
Heard Island And Mc Donald Islands	HMD
Honduras	HND
Hong Kong	HKG
Hungary	HUN
Iceland	ISL
India	IND
Indonesia	IDN
Iraq	IRQ
Ireland	IRL
Isle Of Man	IMN
Israel	ISR
Italy	ITA
Ivory Coast	CIV
Jamaica	JAM

Japan	JPN
Jersey	JEY
Jordan	JOR
Kazakhstan	KAZ
Kenya	KEN
Kiribati	KIR
Kuwait	KWT
Kyrgyzstan	KGZ
Laos	LAO
Latvia	LVA
Lebanon	LBN
Lesotho	LSO
Liberia	LBR
Libya	LBY
Liechtenstein	LIE
Lithuania	LTU
Luxembourg	LUX
Macao	MAC
Macedonia	MKD
Madagascar	MDG
Malawi	MWI
Malaysia	MYS
Maldives	MDV
Mali	MLI
Malta	MLT
Marshall Islands	MHL
Martinique	MTQ
Mauritania	MRT
Mauritius	MUS
Mayotte	MYT
Mexico	MEX
Moldova	MDA

Monaco	MCO
Mongolia	MNG
Montenegro	MNE
Montserrat	MSR
Morocco	MAR
Mozambique	MOZ
Namibia	NAM
Nauru	NRU
Nepal	NPL
Netherlands	NLD
New Caledonia	NCL
New Zealand	NZL
Nicaragua	NIC
Niger	NER
Nigeria	NGA
Niue	NIU
Norfolk Island	NFK
Northern Mariana Islands	MNP
Norway	NOR
Oman	OMN
Pakistan	PAK
Palau	PLW
Palestine	PSE
Panama	PAN
Papua New Guinea	PNG
Paraguay	PRY
Peru	PER
Philippines	PHL
Pitcairn	PCN
Poland	POL
Portugal	PRT
Puerto Rico	PRI

Qatar	QAT
Reunion	REU
Romania	ROU
Russian Federation	RUS
Rwanda	RWA
Saint Barthelemy	BLM
Saint Helena	SHN
Saint Kitts And Nevis	KNA
Saint Lucia	LCA
Saint Martin	MAF
Saint Pierre And Miquelon	SPM
Saint Vincent And The Grenadines	VCT
Samoa	WSM
San Marino	SMR
Sao Tome And Principe	STP
Saudi Arabia	SAU
Senegal	SEN
Serbia	SRB
Seychelles	SYC
Sierra Leone	SLE
Singapore	SGP
Sint Maarten	SXM
Slovakia	SVK
Slovenia	SVN
Solomon Islands	SLB
Somalia	SOM
South Africa	ZAF
South Georgia And The South Sandwich Islands	SGS
South Korea	KOR
Spain	ESP
Sri Lanka	LKA
Suriname	SUR

Svalbard And Jan Mayen	SJM
Swaziland	SWZ
Sweden	SWE
Switzerland	CHE
Taiwan	TWN
Tajikistan	TJK
Tanzania	TZA
Thailand	THA
Timor Leste	TLS
Togo	TGO
Tokelau	TKL
Tonga	TON
Trinidad And Tobago	TTO
Tunisia	TUN
Turkey	TUR
Turkmenistan	TKM
Turks And Caicos Islands	TCA
Tuvalu	TUV
US Virgin Islands	VIR
Uganda	UGA
Ukraine	UKR
United Arab Emirates	ARE
United Kingdom	GBR
United States	USA
United States Minor Outlying Islands	UMI
Uruguay	URY
Uzbekistan	UZB
Vanuatu	VUT
Vatican City	VAT
Venezuela	VEN
Vietnam	VNM
Wallis And Futuna	WLF

Western Sahara	ESH
Yemen	YEM
Zambia	ZMB
Zimbabwe	ZWE

CreditCard_CSCIndicator

None
Provided
NotProvided
Illegible
NotOnCard

CreditCardTransactionFailedSubstatus

None
ProcessingFailed
BusinessRulesFailed
AuthorizedAmountLessThanRequested
ProcessorResponse
ProcessorCommunicationError

PostalCodeAVSResponse

None
Matched
NotMatched
Unavailable
Error

SilentPostType

AddStoredCreditCard
VerifyAndAddStoredCreditCard

StateProvinceCode

None

Valid values for the USA CountryCode:
AL
AK
AS
AZ
AR
CA
CO
CT
DE
DC
FM
FL
GA
GU
HI
ID
IL
IN
IA
KS
KY
LA
ME
MH
MD
MA
MI
MN
MS
MO
MT
NE

NV
NH
NJ
NM
NY
NC
ND
MP
OH
OK
OR
PW
PA
PR
RI
SC
SD
TN
TX
UT
VT
VI
VA
WA
WV
WI
WY

Valid values for the CAN CountryCode:
AB
BC
MB
NB
NL
NT
NS
NU

ON
PE
QC
SK
YK

Valid values for the MEX CountryCode:
AG
BJ
BS
CP
CH
CI
CU
CL
DF
DG
GJ
GR
HG
JA
EM
MH
MR
NA
NL
OA
PU
QA
QR
SL
SI
SO
TA
TM
TL
VZ

YC
ZT
AE
AP
AA

Document Revision History

Revision Date	Summary of Changes
3/22/2016	Updated CardType expected values with new value.
3/30/2016	Updated Preface to include mention of TLS 1.1 or greater requirement.
4/14/16	Updated CountryCode expected values.
4/14/16	Added new CreditCard_BillingAddress_ForeignStateProvinceRegion and CreditCard_ShippingAddress_ForeignStateProvinceRegion form variables to Card Addition Only form variable definitions.
4/14/16	Updated copyright notice in the footer of the document.
4/19/16	Added new card verification and addition field Transaction_CardVerificationCurrencyCode as well as expected values for it.
5/11/16	Removed Cuba from list of valid CountryCodes.
5/13/16	Removed Ugandan Shilling and Icelandic Krona from list of valid CardVerificationCurrencyCodes.
5/20/16	Added new CardBIN field to both Approval and Result URL form variable definitions.
4/3/17	Removed Guinean Franc as a valid currency code.
4/14/17	Removed Syrian Arab Republic from list of Country Codes.