A decorative graphic consisting of three blue circles of varying sizes and two thin blue lines. One large circle is at the top center, a smaller one is below it, and a very large one is at the bottom right. Two thin lines cross the page diagonally, one from the top left to the middle right, and another from the top right to the middle left.

# SLIM CD

## Secure Sessions & Hosted Payment Pages

This document describes how to create and use Secure Sessions to control a Hosted Payment Page , therein reducing your PA-DSS requirements and PCI Scope

**Frank Haggart**  
**2/8/2012**  
**Version 1.0**

# Secure Sessions & Hosted Payment Pages

---

## Table of Contents

Chapter 1: Secure Sessions .....	4
Session Overview .....	4
POS Session Diagram .....	5
WEB Session Diagram .....	5
Hosted Payment Pages .....	6
HTML Pages.....	7
Transaction Flow.....	7
Getting Started.....	7
Getting Payment Results.....	8
Mix and Match Approaches .....	8
Session Input Format Overview .....	8
Session Output Format Overview .....	9
CreateSession.....	9
CreateSession Input Values.....	9
CreateSession Output Values.....	10
CreateSession Examples .....	11
ShowSession.....	11
ShowSession Input Values .....	11
ShowSession Output Values .....	11
ShowSession Examples .....	12
MultiSession.....	12
MultiSession Input Values.....	12
MultiSession Output Values.....	12
MultiSession Examples.....	12
CheckSession.....	13
CheckSession Input Values.....	13
CheckSession Output Values.....	14

CheckSession Review .....	16
CheckSession Examples.....	17
CancelSession.....	18
CancelSession Input Values.....	18
CancelSession Output Values.....	18
CancelSession Examples.....	19
SendSession .....	19
SendSession Input Values .....	19
SendSession Output Values .....	19
SendSession Examples .....	20
Session Review .....	20
Best Practices .....	21
Session Summary .....	21
Chapter 2: Hosted Payment Pages.....	22
Parts of a Hosted Payment Page.....	22
Understanding Transaction Flow .....	22
Understanding Forms.....	23
Managing Forms .....	24
Edit Form.....	25
Page Options .....	26
Table Tag Information.....	27
Font Tag Information .....	28
Total Order Limits .....	29
Card Types Accepted.....	30
Cardnumber Entry Options .....	30
Form/Session Control Options.....	31
Hosted Paget Processing.....	32
Final Approval/Decline Screen Text .....	33
Email Options .....	34
Submit/Cancel Button Options .....	35
Edit Headers.....	36
HEAD Tag.....	36

BODY Tag.....	37
The Header Text.....	37
Footer Text.....	38
Table Header.....	39
Table Footer.....	39
Edit Templates.....	40
Pre Template HTML.....	40
Payment Text.....	41
Billing Text.....	41
Post Template HTML.....	42
Email Template.....	42
Mail-merge Field Options.....	43
List Fields.....	45
Edit Field.....	46
Field Definition.....	46
Display Position.....	47
Form Field.....	47
Display Attributes.....	48
Data values.....	49
Form/Data Actions.....	50
Processing Controls.....	51
Chapter 3: Manually Creating Sessions.....	53
Create Session Menu.....	53
Appendix A - SAMPLES.....	58
Appendix B - Form Generation Layout.....	63
SECTIONS.....	63
CONFIGURATION TOOLS.....	63
HTML Generation.....	64
Appendix C –State/Country ListBoxes.....	65
State ListBox.....	65
Country ListBox.....	65

# Chapter 1: Secure Sessions

## Session Overview

SLIM CD Secure Sessions offer the POS developer an easy integration that eliminates PA-DSS requirements. This is done by allowing the developer to create sessions on the SLIM CD servers whereby clerks perform payment data entry through standard Internet browsers. Developers then retrieve the results of clerk payment processing in real-time without ever touching cardholder information.



SLIM CD Secure Sessions offer four unique entry points:

1. CreateSession – This function allows the developer to create a session on the SLIM CD servers. Sessions created are identified by a unique SessionID. If desired, developers can provide additional data that pre-fills forms or enhances transactions.

URL: <https://stats.slimcd.com/soft/createsession.asp>

2. ShowSession – This function allows a web browser to display an HTML form and collect payment information from a clerk or consumer. The browser is launched by providing a URL containing the unique SessionID.

URL: [https://stats.slimcd.com/soft/showsession.asp?SessionID=\\_\\_\\_\\_\\_](https://stats.slimcd.com/soft/showsession.asp?SessionID=_____)

3. CheckSession – This function is used by the developer to monitor the session and identify the results as the credit card payment transaction is performed.

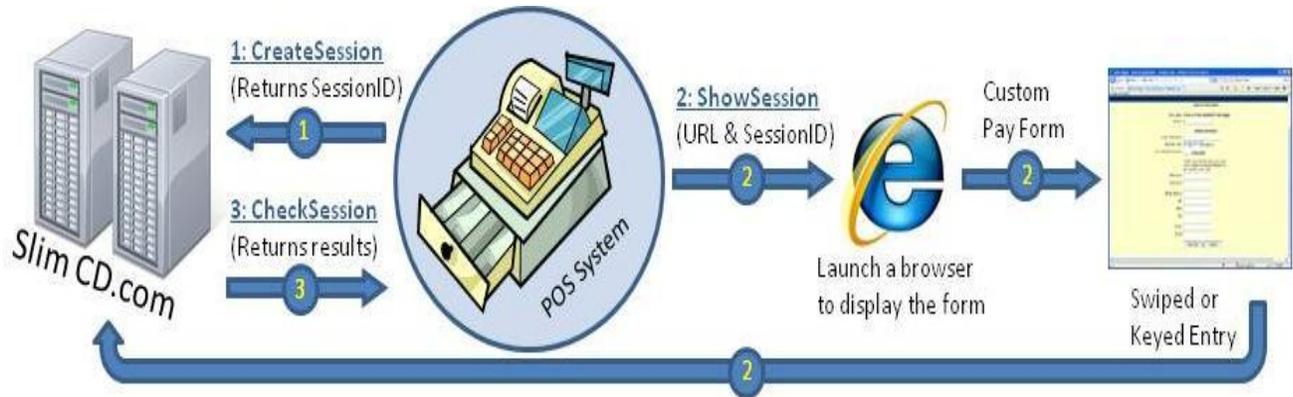
URL: <https://stats.slimcd.com/soft/checksession.asp>

4. CancelSession – This function allows the developer to cancel a session and block further processing.

URL: <https://stats.slimcd.com/soft/cancelsession.asp>

## POS Session Diagram

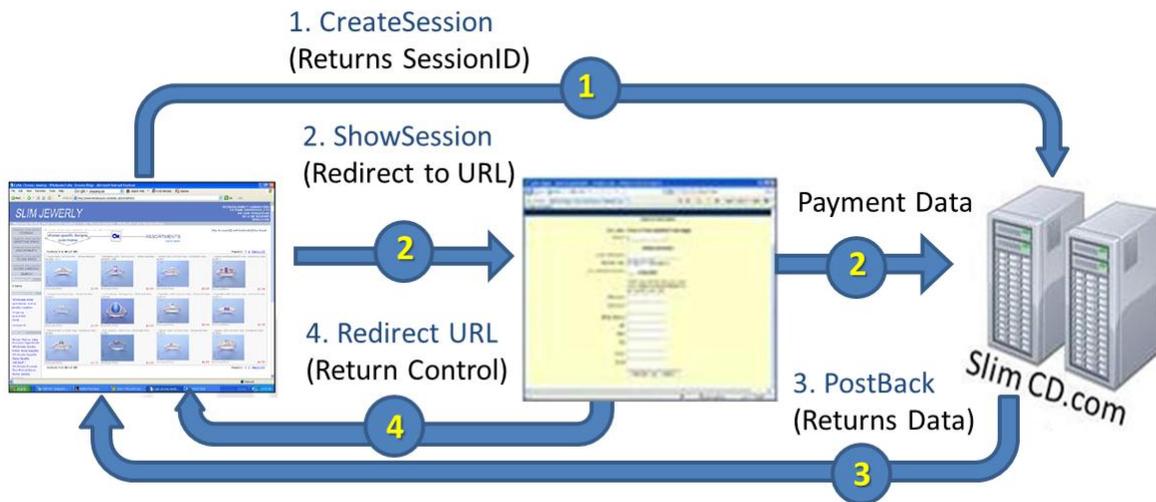
The following represents a simple diagram of CreateSession to start a session, ShowSession to launch a browser for the collection of payment information, and CheckSession to return the payment results.



*Example POS Flow*

## WEB Session Diagram

The following represents a simple diagram of CreateSession to start a session, ShowSession to launch a session and the PostBack URL/Redirect URL to update the web host.



*Example Web Flow*

## Hosted Payment Pages

The SLIM CD Hosted Payment Pages provide POS and WEB developers with the fastest and easiest way to attain PCI compliance. This is done by removing the collection of payment data from the POS or WEB application and placing the credit card information screens within a browser-based payment page that runs on SLIM CD's PCI-secured servers.



*Hosted Payment Page Example*

Once a developer's application avoids touching cardholder data within their applications, they are no longer required to obtain a PA-DSS certification. Developers circumvent the PA-DSS requirement altogether by directing the input of cardholder data to a SLIM CD hosted payment page. Developers will still have access to information required to print a receipt and can still perform normal credit card

operations such as processing refunds or performing recurring charges. These supplemental transactions are performed on transaction token data returned from the SLIM CD system.

The center of the system is the HTML payment page. This page can display information, relay information and can collect information. SLIM CD provides multiple ways to integrate information into the HTML page and retrieve results at the conclusion of payment processing.

The Hosted Payment Page System from SLIM CD offers multiple approaches to integration. Developers may elect to use individual approaches or may decide to combine approaches as desired to best fit a particular integration's requirements. SLIM CD's solutions offer traditional redirection-based responses and SLIM CD's proprietary LifeLine technology to allow seamless integration and separation of payment interfaces from merchant applications.

## HTML Pages

SLIM CD Hosted Payment Pages provide HTML-based display of forms for cardholder data input. Both keyed and swiped transactions are supported. Standard Internet browser-style applications are used to display the form and send data to our secure servers. These HTML-based forms can be pre-populated with consumer information from either a POS or WEB application. The operator or consumer can then process payments using traditional Internet browser software.

## Transaction Flow

In its simplest form, this SLIM CD hosted HTML page can be launched by web developers using an HTTP POST or web-based redirection. Upon completion of the payment, the HTML page can display receipts and/or redirect the browser to a location specified by the developer to complete the round-trip back to the originating system. This approach is popular among ecommerce developers as the payment request is launched from web-based applications and returning control to those applications from the web is easily programmed.

A second approach, popular among POS developers, is to create a session and send data to and from that session. In this implementation, an HTTP POST is performed by the POS or WEB application to initiate a session and acquire a unique session identifier. The payment page is then displayed separately using the unique session identifier. Finally, the POS or WEB application uses the SLIM CD LifeLine technology to retrieve synchronous or asynchronous results from the SLIM CD. This multi-step approach allows developers to send data outbound to the Hosted Payment Pages and retrieve results from those pages without requiring POS or WEB systems to accept inbound transmissions from SLIM CD's servers.

## Getting Started

To display the HTML web page, use the "Show Session" function. This function will accept merchant information and present the payment HTML page inside the browser. We recommend Web developers that can perform an HTTP POST from their current page use this approach.

Alternatively, the "Show Session" function can accept a session id that has previously been created using "Create Session". We recommend Web developers that prefer using redirection to HTTP POST use this method. We also recommend this method for POS merchants who need to launch a browser. Finally, a

URL containing session id information can be sent to mobile devices or included in emails so that recipients can use remote client devices to enter payment information independently of the POS system.

## Getting Payment Results

There are multiple ways to identify the results of the activity from the payment page and regain control of the user interaction. Each different approach is designed to work with different development environments.

The simplest way for Web-based applications to regain control is to have the hosted HTML payment page perform a web-based redirection back to the calling application. This can be used in conjunction with a “silent post”, where the payment page first posts the results back to the calling application and then redirects the user to a landing page within the calling application so results can be displayed for the user.

The recommended way for POS developers is to create a session prior to launching the payment page and monitor the session with the SLIM CD LifeLine to acquire the results of the payment page activity. Once the SLIM CD LifeLine returns payment results, the POS application can then resume other activities.

## Mix and Match Approaches

Web developers may prefer to use a combination of approaches, where initial data is sent silently by creating a session id, browser control is redirected using the session id, results are posted back using silent posts and control is regained with another browser redirect. Alternatively, Web developers may elect to monitor the session for results. Finally, mail-order or other applications that do not expect immediate payment information can create sessions send out payment requests via email or SMS-text using formatted URLs containing session ids, and schedule background tasks that poll for results at longer timed intervals using the session id. This approach is useful when delivering the payment request via email (or other mechanism) where the delay could be several hours or longer.

## Session Input Format Overview

SLIM CD Sessions accept traditional HTTP FORM name/value pairs with URL-Encoding format. Either HTTP GET or HTTP POST operations can be used, although SLIM CD highly recommends the use of POST when sending consumer information to pre-populate session data with the CreateSession function.

Notes:

- A. Please note that unless otherwise noted below, fields, fieldnames, data types and values are defined in the SLIM CD Transaction Gateway Manual, as part of the SLIM CD Developers’ Toolkits.
- B. SLIM CD Sessions are designed to communicate to and from the developer’s application using 40-byte SessionID values. These SessionID values are sent and received in HEX. Developers can elect to convert the values into 20-byte binary data for local storage; however the SLIM CD servers only send and receive the 40-byte hex version.

## Session Output Format Overview

The ShowSession function is designed to display an HTML form for user input, and is intended for use as a URL for a standard web browser (such as Internet Explorer).

All other SLIM CD Session functions, including CreateSession, CheckSession, and CancelSession, return standard XML, formatted in the traditional SLIM CD Gateway format. See the SLIM CD Transaction Gateway documentation for more details.

The SLIM CD standard XML reply document returned will contain a <reply> block. A standard set of fields can be found inside the <reply> block:

Field Name	Description
<response>	Text description of the outcome (Success, FAIL, Error, etc.)
<responsecode>	Numeric value of the response (0=Success)
<description>	Textual description of the FAIL, ERROR, or other non-Success responses
<responseurl>	Name of the URL called, to insure the response is from the intended service
<datablock>	A placeholder for service-specific information, including sub-fields as needed

If the <response> field contains the text "Success", then the service has been successful and the <datablock> will often contain additional information of value as returned from that specific service. All other <response> values represent errors, failures, or other conditions that need attention.

Below is sample XML returned from createsession.asp

```
<?xml version="1.0" encoding="utf-8"?>
<reply>
  <response>Success</response>
  <responsecode>0</responsecode>
  <description>OK</description>
  <responseurl>/soft/createsession.asp</responseurl>
  <datablock><sessionid>22ADCBF1F23F19D125DF76B00B6DC75B201CB6CF</sessionid></datablock>
</reply>
```

## CreateSession

CreateSession is used to create a SessionID. SessionID values are later used as part of a secure and tamper-resistant URL so that a browser-style application can be launched to accept cardholder information outside of the developer's software. CreateSession can also be used to send initial customer data to SLIM CD so that payment pages are pre-populated for the clerk, providing ease of data entry.

### CreateSession Input Values

CreateSession Input Values are broken into two sets, required and optional. The required fields match the normal SLIM CD fields created at account setup and used when sending credit card transactions to the SLIM CD Gateway.

CreateSession Required Fields:

Field Name	Description
ClientID	SLIM CD Client ID value to identify the merchant
SiteID	SLIM CD Site ID value used to identify the SLIM CD Site configuration to use

PriceID	SLIM CD Price ID value used to identify the pricing options
Password	Password for the Client ID

The above required fields are merchant data that identify which account to use when submitting the transaction for payment processing.

#### CreateSession Optional Fields:

Field Name	Description
FormID	SLIM CD numeric FormID for a Hosted Payment Page design
FormName	SLIM CD alpha-numeric Form name for a Hosted Payment Page design (recommended)
TransType	Type of transaction to perform (SALE, LOAD, AUTH, etc.)
Amount	Amount of the transaction (\$0.00, if not specified)

Optional Fields fall into a few categories. These include Form Data, Customer Data, Payment Data, Clerk Identification Data, and Custom Developer Data.

Form Data allows the developer to identify the pre-configured display form to use. Form configurations are created on the SLIM CD website and referenced by name or numerical value. This identification is sent when the session is created so that the proper form configuration will be used when displaying the form data to the clerk or consumer. (Note that at this time, specifying a form is optional. The system will use the lowest FormID available for the merchant if none is specified. This approach is subject to change and SLIM CD recommends supplying form names when sessions are created to insure the display functionality meets developer expectations)

Customer Data can include any of the payment data supported by the SLIM CD gateway. This includes First Name, Last Name, Address, City, State, Zip, etc.

Payment Data includes other fields supported by the SLIM CD, such as TransType and Amount. If the transaction type is not specified, it will default to SALE. Specialty fields can be provided, such as Hotel Checkin Date, etc. The “gateid” field can be provided to reference customer data from previous transactions without supplying that data or cardholder information.

Please note that CreateSession will not accept certain cardholder data information. This is to ensure that PA-DSS avoidance and PCI Scope Reduction is obtained by discouraging applications from storing or sending cardholder data. Previously processed cardholder data can be referenced using the SLIM CD “gateid” values from those previous transactions.

The following fields will not be accepted by the CreateSession function: Card Number, Track Data, CVV2, or PIN Block data fields. If the developer desires to provide this information to SLIM CD as part of a conversion that eliminates cardholder data storage, we recommend using the SLIM CD Developer’s Toolkit and perform the “LOAD” function.

### **CreateSession Output Values**

CreateSession returns standard XML, in the traditional SLIM CD Gateway format (see the SLIM CD Transaction Gateway documentation).

The XML document returned will contain a <reply> block. If the <response> field is Success, then the session has been created. All SLIM CD responses contain a <datablock> with additional fields of

information. Inside the datablock will be a <sessionid> that uniquely identifies the session created by CreateSession.

Below is sample XML returned from createsession.asp

```
<?xml version="1.0" encoding="utf-8"?>
<reply>
  <response>Success</response>
  <responsecode>0</responsecode>
  <description>OK</description>
  <responseurl>/soft/createsession.asp</responseurl>
  <datablock><sessionid>22ADCBF1F23F19D125DF76B00B6DC75B201CB6CF</sessionid></datablock>
</reply>
```

## CreateSession Examples

Sample HTTP GET:

[https://stats.slimcd.com/soft/createsession.asp?clientid=1032&siteid=249949344&priceid=74&password=289075&amount=20&first\\_name=Frank&var1=test2](https://stats.slimcd.com/soft/createsession.asp?clientid=1032&siteid=249949344&priceid=74&password=289075&amount=20&first_name=Frank&var1=test2)

(Note: it is not recommended to use GET for CreateSession as it sends data insecurely)

Sample HTTP POST:

URL: <https://stats.slimcd.com/soft/createsession.asp>

POSTDATA: clientid=1032&siteid=249949344&priceid=74&password=289075&amount=20&first\_name=Frank&var1=test2

Sample HTML Form with POST:

<https://stats.slimcd.com/session/create.htm>

## ShowSession

ShowSession is used to display a previously-created session. It is designed to accept limited input and display HTML output so that the SessionID value can be included in the URL that is provided as a command-line when launching in a browser. This also works well when redirecting from one web server to another.

### ShowSession Input Values

ShowSession requires a single input field, the SessionID. A form is displayed based on the configuration settings for the form on the SLIM CD server.

ShowSession Required Fields:

Field Name	Description
SessionID	40-Byte value produced by CreateSession

### ShowSession Output Values

The ShowSession service returns an HTML web page for display in a browser. The displayed form and the actions of that form can be managed using the SLIM CD HOSTED PAYMENT PAGES functionality in the SLIM CD website.

It is possible to configure HOSTED PAYMENT PAGES to perform a PostBack and/or Redirection at the end of the payment. Alternatively, the CheckSession and CancelSession functions can be used to monitor the outcome of the activity in a HOSTED PAYMENT PAGE.

## ShowSession Examples

Sample HTTP GET:

<https://stats.slimcd.com/soft/showsession.asp?sessionid=22ADCBF1F23F19D125DF76B00B6DC75B201CB6CF>

Sample HTTP POST:

URL: <https://stats.slimcd.com/soft/showsession.asp>  
POSTDATA: sessionid=22ADCBF1F23F19D125DF76B00B6DC75B201CB6CF

Sample HTML Form with POST:

<https://stats.slimcd.com/session/show.htm>

## MultiSession

MultiSession is used to display a previously-created session similarly to ShowSession. It is designed to accept limited input and display HTML output so that the SessionID value can be included in the URL that is provided as a command-line when launching in a browser. This also works well when redirecting from one web server to another. Unlike ShowSession, MultiSession is designed to be used by multiple recipients such as for charitable donations. MultiSession creates a unique session from the original template each time it is called.

### MultiSession Input Values

MultiSession requires a single input field, the SessionID. A form is displayed based on the configuration settings for the form on the SLIM CD server.

MultiSession Required Fields:

Field Name	Description
SessionID	40-Byte value produced by CreateSession

### MultiSession Output Values

The MultiSession service returns an HTML web page for display in a browser. The displayed form and the actions of that form can be managed using the SLIM CD HOSTED PAYMENT PAGES functionality in the SLIM CD website.

It is possible to configure HOSTED PAYMENT PAGES to perform a PostBack and/or Redirection at the end of the payment. Alternatively, the CheckSession and CancelSession functions can be used to monitor the outcome of the activity in a HOSTED PAYMENT PAGE.

## MultiSession Examples

Sample HTTP GET:

<https://stats.slimcd.com/soft/multisession.asp?sessionid=22ADCBF1F23F19D125DF76B00B6DC75B201CB6CF>

Sample HTTP POST:

URL: <https://stats.slimcd.com/soft/multisession.asp>

POSTDATA: sessionid=22ADCBF1F23F19D125DF76B00B6DC75B201CB6CF

## CheckSession

CheckSession is used to monitor the results of payment processing as it occurs in the browser. CheckSession is designed to allow behind-the-scenes monitoring of the clerk's payment activity without requiring the developer to touch cardholder data.

CheckSession can be used in either of two ways: Asynchronous "polling" performed by the client, or synchronous "wait" performed by CheckSession. The default for "wait" is 60 seconds. To perform asynchronous polling for a result, set the wait time to a lower value and call CheckSession repeatedly.

SLIM CD suggests that developers may find it easier to issue a single call to CheckSession instead of repeated polling of the CheckSession service. Other developers may prefer to implement polling to create a more interactive end-user experience, allowing other application activity between CheckSession calls.

## CheckSession Input Values

CheckSession Input Values are broken into two sets, required and optional. The required fields identify the session and authenticate access to the session information. The optional fields control the timeout for the session and allow additional data to be retrieved from the session or from the SLIM CD payment processing gateway.

CheckSession Required Fields:

Field Name	Description
ClientID	SLIM CD Client ID value to identify the merchant
Password	Password for the Client ID
SessionID	SLIM CD Session ID for the desired session

The above required fields are merchant data that identify which account to authenticate and the session to check.

CheckSession Optional Fields:

Field Name	Description
Wait	The maximum amount of time in seconds the CheckSession should consume while waiting for the session to be completed
waitforcompleted	Wait until the completed indicator is "yes", which means either the transaction is approved or all decline attempts have been complete.
(other fields)	Any other SLIM CD gateway field, Hosted Payment Page field, or session field.

The optional "wait" value is used to control the length of time CheckSession waits for the results of a completed payment transaction. This "wait" value identifies how long (in seconds) the CheckSession will run on the SLIM CD servers while waiting for the ShowSession to complete a transaction. As soon as the

payment transaction is successfully processed, the CheckSession function can return the results, which often occurs before the ShowSession browser is closed.

Note that if you are using HTTP libraries to perform the POST, you may have to specify a “connection timeout” value in your HTTP POST module such that your module’s timeout that exceeds the value you specify in the “wait” parameter for CheckSession. For example, if you are using MSXML to post data, make sure you set the MSXML timeout to greater than the value of the “wait” option to avoid synchronization problems where MSXML’s connection encounters a timeout before CheckSession can successfully return results.

The optional “waitforcompleted” flag allows developers to identify that the CheckSession call should not return until the final result is obtained, and ignore intermediate declines. If “waitforcompleted” is “yes”, then CheckSession will either return “Success” or “TIMEOUT” until either an approval occurs, or until the last allowed “DECLINE” count is reached. Once the last “DECLINE” is reached, a “FAIL” response will be returned.

If “waitforcompleted” is “no”, then CheckSession will return either “Success”, “TIMEOUT”, or “FAIL”. When using waitforcompleted=no, the developer will need to monitor the “completed” return value to identify if the user might still attempt additional cardnumbers, or if the session has reached it’s final completed conclusion.

Optional data fields allow CheckSession to retrieve additional information from the session, the form, or the SLIM CD gateway and return that data to the developer. Simply supply field names that identify additional data to be returned. If a value is supplied with the field name and the field is not found within the session, form or gateway data, the value supplied to CheckSession will be returned from CheckSession.

For example the developer can specify “first\_name” which will cause CheckSession to include the data captured for the first\_name field. Developers can also create custom input fields and values with CreateSession, and have these values returned by specifying the name of the custom fields when calling CheckSession.

## CheckSession Output Values

The CheckSession service returns an XML document in standard SLIM CD format. The developer should check the <response> field for “Success”, “Fail”, “TIMEOUT” or “Error”.

- The “Success” value indicates an approved transaction.
- The “FAIL” timeout indicates a declined transaction, check <description> for a reason.
- The “TIMEOUT” value indicates that ShowSession has not yet produced a result
- The “Cancel” value indicates that the session has been cancelled by the user or the application.
- The “Error” value indicates some sort of error (such as invalid session id, etc). Check the <description> for a reason.

The <datablock> section contains a standard set of fields, but can also contain additional data if requested by the developer.

The following fields are returned as part of the standard <datablock> and are designed to allow developers to print receipts and process future transactions such as voids and refunds. The gateid returned can be used to process voids, refunds, and sales without cardholder data or can be used to pre-populate cardholder data into forms using CreateSession.

Field Name	Description
Sessionid	40-byte session id
Completed	yes – means the session is concluded and no additional user interaction is permitted no – means the session is not concluded and the developer should continue monitoring the session for additional transactional results (such as the use of a different card to gain an approved response)
Declinecount	Numeric value that indicates the number of declined transactions which occurred for this record. If the hosted payment form allows multiple declines, this value will indicate how many attempts were made before the approval or final decline was reached.
Approved	Y, N , E or U Y=Yes, approved N=No, not approved E=Error processing transaction U=Unknown, not yet sent for processing or unknown status from gateway
Gateid	SLIM CD Ticket# (integer) representing transaction (can be used for tokenization)
Transtype	Transaction type for this transaction (useful if you let clerk pick transaction type)
Authcode	6-digit authcode on approvals
Avsreply	1-digit AVS reply (see SLIM CD documentation)
cvv2reply	1-digit CVV2 reply (see SLIM CD documentation)
Declinestr	String indicating the reason for the decline or error
Cardtype	V,M,A,D,C,B,T,G,X,Y
Cardname	Visa,MasterCard,American Express, Discover, Diners, PIN Debit, EBT, GiftCard, Paper Check, Electronic Check (respectively for above letters)
first6	First 6 digits of cardnumber
last4	Last 4 digits of cardnumber
Keyed	Yes if keyed. No if swiped.
Expmonth	Expiration month on card
Expyear	Expiration year for card
Approvedamt	The amount the transaction was actually approved for (could be less if you allow partial approvals)
Tip	Amount of the transaction that was considered a tip
Bal	The balance amount returned (to be printed on receipt)
Invoiceno	Originally the “client_transref” on input.
corporatecardindicator	If set, could indicate a corporate or business card. B=Business Card (also used if it is not determined which type of business card) R=Corporate Card S=Purchase Card D=Visa Commercial (reserved for future use, per VISA) 0, blank, or NULL= not a commercial card

In the example below, all fields represent the standard fields returned from CheckSession. The additional “var1” field (in red) was requested by the developer.

```
<?xml version="1.0" encoding="utf-8"?>
<reply>
  <response>Success</response>
  <responsecode>0</responsecode>
  <description>OK</description>
  <responseurl>/soft/checksession.asp</responseurl>
  <datablock>
    <approved>Y</approved>
    <gateid>163081</gateid>
    <transtype>Sale</transtype>
    <authcode>TAS527</authcode>
    <avsreply>N</avsreply>
    <cvv2reply> </cvv2reply>
    <cardtype>V</cardtype>
    <cardname>Visa</cardname>
    <first6>444455</first6>
    <last4>9999</last4>
    <expmonth>12</expmonth>
    <expyear>15</expyear>
    <approvedamt>10</approvedamt>
    <tip>0</tip>
    <bal></bal>
    <keyed>Yes</keyed>
    <invoiceno></invoiceno>
    <corporatecardindicator></corporatecardindicator>
    <sessionid>22ADCBF1F23F19D125DF76B00B6DC75B201CB6CF</sessionid>
    <var1>test</var1>
  </datablock>
</reply>
```

## CheckSession Review

Here is a brief outline of the two CheckSession approaches:

1. First approach – multiple calls to CHECKSESSION with a SHORT timeout.
  - a. Create Session to get the session ID
  - b. Use StartSession to create a URL and launch the browser
  - c. Loop, calling “CheckSession” with a short timeout. You can then run other events in between calls to CheckSession. This is a POLLED approach.
2. Second approach – a single call to CHECKSESSION with a long WAIT time and the calling HTTP code to set it’s “timeout” to even longer
  - a. Create Session to get the session ID
  - b. Use StartSession to create a URL and launch the browser

- c. Call "CheckSession" with a long timeout. (Make sure you set the WinINET , WinHTTP or MSXML timeout to even longer!)

In either case, you must refer to the <response> value to detect the results of CheckSession. If a "TIMEOUT" occurred, that means that this call to CHECKSESSION ran out of time before the clerk was able to finish typing in the cardnumber and get a response. Or, you can get a CANCEL, which means the clerk (or the form) cancelled the transaction. Or, you can get a FAIL, which means the transaction was declined, or a SUCCESS which means the transaction was approved.

Here are a few cases that highlight using the "waitforcompleted" and the "completed" values:

Case 1: The developer wishes to perform a single call to "CheckSession" with a long timeout value. In this case, the "waitforcompleted" value should be set to "yes". Intermediate cardholder attempts that cause "DECLINES" will not interrupt the CheckSession and only the final result will be returned. The "completed" value will be returned as "yes"

Case 2: The developer wishes to "poll" the session within a loop, calling "CheckSession" with short/medium timeout values. The developer is only interested in the final result of "Approved" or "Declined" once all attempts are complete. In this case, the "waitforcompleted" should be set to "yes", so that intermediate declines are not returned as "FAIL", but instead "TIMEOUT" is returned until the user interaction has been finalized. The "completed" value will be returned as "yes"

Case 3: The developer wishes to "poll" the session within a loop, calling "CheckSession" with short timeout values. The developer wishes to capture any card activity (intermediate approvals) so that the developer can maintain statistics on the number of declines before a card was approved, etc. In this case, set the "waitforcompleted" to "no", and a "FAIL" response will be returned with the "completed" set to "no" unless an APPROVAL, CANCEL, or final DECLINE occurs.

## CheckSession Examples

### Sample HTTP GET:

<https://stats.slimcd.com/soft/checksession.asp?clientid=1032&password=289075&wait=60&sessionid=22ADCBF1F23F19D125DF76B00B6DC75B201CB6CF>

### Sample HTTP POST:

URL: <https://stats.slimcd.com/soft/checksession.asp>  
POSTDATA:clientid=1032&password=289075&wait=60&sessionid=22ADCBF1F23F19D125DF76B00B6DC75B201CB6CF

### Sample HTML Form with POST:

<https://stats.slimcd.com/session/check.htm>

Note: The above examples request the “var1” field to be returned, if it exists from the user or the HOSTED FORM. The examples also specify a wait time of 60 seconds. This wait time can be increased to create a single call that lets the SLIM CD server loop waiting for a reply, or decreased if the client application wishes to loop and call CreateSession while performing other functions.

## CancelSession

CancelSession is used to cancel a session. Cancelling a session will prevent the clerk from performing payment processing. Cancelling a session may not be possible if the payment has already been completed (in which case a VOID transaction must be performed on the gateway), or if the payment is currently in-process for payment processing with the credit card companies.

## CancelSession Input Values

CancelSession Input Values identify the session and authenticate access to the session information.

CancelSession Required Fields:

Field Name	Description
ClientID	SLIM CD Client ID value to identify the merchant
Password	Password for the Client ID
SessionID	SLIM CD Session ID for the desired session

## CancelSession Output Values

The CheckSession service returns an XML document in standard SLIM CD format. The developer should check the <response> field for “Success”, “Fail” or “Error”.

- The “Success” value indicates the session has been successfully cancelled prior to payment processing.
- The “FAIL” timeout indicates a declined attempt to cancel. Check the <description> for one of two reasons
  - “Session already completed” – The session cannot be cancelled because payment has been performed. Use CheckSession to gather the results
  - “Session submitted for payment processing” – The session has been submitted for payment processing. Wait and use CheckSession, or wait and attempt to call CancelSession after payment processing finishes
- The “Error” value indicates some sort of error (such as invalid session id, etc). Check the <description> field for more information.

```
<?xml version="1.0" encoding="utf-8"?>
<reply>
  <response>Success</response>
  <responsecode>0</responsecode>
  <description>OK</description>
  <responseurl>/soft/checksession.asp</responseurl>
</datablock></datablock>
</reply>
```

## CancelSession Examples

Sample HTTP GET:

<https://stats.slimcd.com/soft/cancelsession.asp?clientid=1032&password=289075&sessionid=22ADCBF1F23F19D125DF76B00B6DC75B201CB6CF>

Sample HTTP POST:

URL: <https://stats.slimcd.com/soft/cancelsession.asp>

POSTDATA:clientid=1032&password=289075&sessionid=22ADCBF1F23F19D125DF76B00B6DC75B201CB6CF

Sample HTML Form with POST:

<https://stats.slimcd.com/session/cancel.htm>

## SendSession

SendSession is used to send a session URL via Email or SMS text message. The recipient can then follow this link on their PC or Mobile device to the previously created hosted payment page where they can enter their payment information and complete the transaction.

### SendSession Input Values

SendSession Input Values identify the session and authenticate access to the session information.

SendSession Required Fields:

Field Name	Description
clientid	SLIM CD Client ID value to identify the merchant
password	Password for the Client ID
sessionid	SLIM CD Session ID for the desired session
message	Message to send (precedes the session URL); Max Length: 20 for SMS Text, 4096 for Email
phone	Cellular Phone Number of the recipient
email	Email Address of the recipient
send_sms	yes or no
send_email	yes or no

### SendSession Output Values

The CheckSession service returns an XML document in standard SLIM CD format. The developer should check the <response> field for “Success”, “Fail” or “Error”.

- The “Success” value indicates the session has been successfully cancelled prior to payment processing.
- The “FAIL” timeout indicates a declined attempt to cancel. Check the <description> for one of three reasons
  - “Session already completed” – The session cannot be cancelled because payment has been performed. Use CheckSession to gather the results

- “Session submitted for payment processing” – The session has been submitted for payment processing. Wait and use CheckSession, or wait and attempt to call CancelSession after payment processing finishes
- “Session already canceled” – The session has been canceled via CancelSession. Session URL cannot be sent.
- The “Error” value indicates some sort of error (such as invalid session id, etc). Check the <description> field for more information.

```
<?xml version="1.0" encoding="utf-8"?>
<reply>
  <response>Success</response>
  <responsecode>0</responsecode>
  <description>OK</description>
  <responseurl>/soft/checksession.asp</responseurl>
  <datablock></datablock>
</reply>
```

## SendSession Examples

### Sample HTTP GET:

```
https://stats.slimcd.com/soft/sendsession.asp?clientid=1032&password=289075&sessionid=22
ADCBF1F23F19D125DF76B00B6DC75B201CB6CF&phone=&email=bsmith@gmail.com&send_e
mail=yes&send_sms=no
```

### Sample HTTP POST:

```
URL: https://stats.slimcd.com/soft/sendsession.asp
POSTDATA:clientid=1032&password=289075&sessionid=22ADCBF1F23F19D125DF76B00B6DC7
5B201CB6CF&phone=&email=bsmith@gmail.com&send_email=yes&send_sms=no
```

## Session Review

SLIM CD Secure Sessions provide a simple and effective way to separate cardholder data and payment acceptance from POS applications. Sessions can interact with hosted payment page forms previously created and configured on the SLIM CD servers. Developers can monitor the progress of sessions to identify their completion and can cancel sessions if needed.

Configuration of a hosted payment page occurs on the SLIM CD servers. SLIM CD provides interactive web pages that allow the developer to define the form look-and-feel, data fields to display and default values for those data fields. Fields can be display-only, hidden, or editable by clerks. Default values and display HTML form styles be specified.

Sessions can provide initial data to the hosted payment pages. Developers can provide data fields to be used by payment forms or to be used by the POS to manage the sessions. Developers can also perform payment processing without cardholder data by provide tokens for previously processed SLIM CD transactions.

## Best Practices

**Step 1:** Create a session and supply any data values you wish to be used by the form or returned at the end of the session.

**Step 2:** Display the form in a separate browser application. POS developers can accomplish this by launching the browser and supplying the SessionID in the URL. Web developers can simply use web redirection. It is also possible to email the URL to directly to a consumer so that payment can be collected directly from the consumer's computer or mobile phone.

**Step 3:** Retrieve the results. POS developers can accomplish this by using the CheckSession function to retrieve the payment processing results. Web developers can configure the hosted payment page to perform "silent postback" calls to update results and/or use browser redirection to transfer control back to their web application at the completion of payment processing.

**Step 4:** Finalize the transaction. Completed sessions provide developers with information they need to print receipts. Tokens provided by SLIM CD can be stored and re-used with the SLIM CD Gateway, allowing developers to process future payment transactions or perform refunds without storing cardholder data.

## Session Summary

SLIM CD Sessions allow developers to provide payment processing services to clients without touching cardholder data flow within their applications. Sessions are used to communicate information to and from SLM CD's Hosted Payment Pages. Hosted Payment Pages utilize SLIM CD Payment Gateway Services to process transactions and provide tokenization of cardholder data. Developers can use SLIM CD tokens retrieved from and SLIM CD Payment Gateway Services to process future transactions without touching or storing cardholder data.

## Chapter 2: Hosted Payment Pages

### Parts of a Hosted Payment Page

Payment pages are created/configured by developers and reside on the SLIM CD server. SLIM CD servers generate HTML for these hosted pages using a variety of options that can be configured. There are a few decisions to be made when creating the form.

1. How will the transaction flow operate?
2. What fields will be sent to the form, collected/updated by the form, and retrieved from the form?
3. How do you want the form fields to display/operate?
4. What will the graphics look like?

Developers use the forms configuration tools to specify all of the options listed above.

### Understanding Transaction Flow

Transaction flow to the Hosted Payment Page is performed with the `CreateSession` and the `ShowSession` functions. The `CreateSession` function stores data on the SLIM CD servers so that it can be displayed or utilized by the session and the Hosted Payment Page. A form is pre-populated by calling `CreateSession` with the data for the fields. The form is displayed by launching a browser or redirecting to `ShowSession`. There are no configuration options required to create a session or show a session, other than creating the form itself. Data fields can be created so their values can be displayed, edited, or returned.

Transaction flow from the Hosted Payment Page is performed in one of two ways. The calling application passes control to `ShowSession`. It can either call `CheckSession` to monitor results, or it can wait for a `PostBack` containing the results, followed by a returning `Redirect` to transfer display control to a pre-configured web page.

For POS merchants, the POS application should call `CheckSession` to poll for results, as there is no web server or page to use as a redirection target. `CheckSession` returns a standard set of values, plus any additional fields for which the “Pass back to checksession” checkbox has been marked.

For WEB merchants, the `PostBack` and redirect approach is the most popular. The pre-configured `PostBack` URL will be invoked for each attempted credit card transaction, providing intermediate results as the session progresses and allowing the developer to update their database with the most current progression. Upon an approved payment or a final declined attempt, the `PostBack` will be invoked with the “completed” flag set to “yes” and the `Redirect` URL will be invoked to transfer control back to the pre-configured URL. This allows the Hosted Payment Page to update the caller and transfer control upon completion.

Some web developers prefer a hybrid approach. They configure the `Redirect` URL and have that page call `CheckSession` to request approved/declined information. This approach works well, but does not provide any intermediate results in the event the user abandons a session by closing the browser.

## Understanding Forms

Forms have four main parts:

1. **The Form:** Configures the flow URLs, Receipt handling, page color, decline counter, card entry features, and more.
2. **The Headers:** These html elements are used to insert graphics on the page by surrounding the form table on top and bottom. There are two types of headers: Page headers and Table headers. Page headers/footers surround the outside of the table containing the form. Table headers/footers are placed inside the table containing the form.
3. **The Templates:** Similar to headers, but are additional rows of the table that are placed before or after the form's fields. Certain tags are also considered "templates" because they support merging
4. **The Fields:** The fields are items within the form where data is displayed or entered. Some fields can be hidden and only used to communicate to the caller. Others are only used to send options to the gateway for processing (such as `allow_duplicates`). Fields can display preset data, allow editing of data, or capture entirely new user entry and pass that information back to the calling application.

## Managing Forms

Forms are created and updated by selecting the *Manage Forms* item from the *Sessions* menu within slimcd.com. Forms can be exported and imported as a way to easily transfer the designs from one CLIENT ID to another. Clicking on the *Manage Forms* menu will list any available forms and let you create a new one. You can also import a form from this screen.

The screenshot displays the 'Configure Forms for 1032' interface. At the top, there are navigation tabs for Account, Config, Fraud Control, Sessions, Stats, Vops, ClientID: 1032, SiteID: All, and Logout. Below the navigation is a sidebar with a 'Sessions' menu containing 'Create Session', 'List Sessions', and 'Manage Forms'. The main content area is titled 'Configuration' and 'Configure Forms for 1032'. It includes buttons for 'Add a new form' and 'Import A New Form'. Below these is a table titled 'Edit an existing Form' with the following data:

Form ID	Form Name	Status	Field Count	Options
1	keypad	ACTIVE	0	Edit Form, Edit Headers, Export Form Design, Show Form, List Fields, Edit Templates, Import Form (replace), SHUTDOWN
2	sample_html	ACTIVE	1	Edit Form, Edit Headers, Export Form Design, Show Form, List Fields, Edit Templates, Import Form (replace), SHUTDOWN
3	sample_fields	ACTIVE	12	Edit Form, Edit Headers, Export Form Design, Show Form, List Fields, Edit Templates, Import Form (replace), SHUTDOWN
4	sample_validation	ACTIVE	17	Edit Form, Edit Headers, Export Form Design, Show Form, List Fields, Edit Templates, Import Form (replace), SHUTDOWN
5	template	ACTIVE	3	Edit Form, Edit Headers, Export Form Design, Show Form, List Fields, Edit Templates, Import Form (replace), SHUTDOWN
6	wholefoods	ACTIVE	6	Edit Form, Edit Headers, Export Form Design, Show Form, List Fields, Edit Templates, Import Form (replace), SHUTDOWN
8	TC	ACTIVE	0	Edit Form, Edit Headers, Export Form Design, Show Form, List Fields, Edit Templates, Import Form (replace), SHUTDOWN
10	tc_gifts	ACTIVE	7	Edit Form, Edit Headers, Export Form Design, Show Form, List Fields, Edit Templates, Import Form (replace), SHUTDOWN
12	WUG	ACTIVE	11	Edit Form, Edit Headers, Export Form Design, Show Form, List Fields, Edit Templates, Import Form (replace), SHUTDOWN
13	MeghansTestForm	ACTIVE	12	Edit Form, Edit Headers, Export Form Design, Show Form, List Fields, Edit Templates, Import Form (replace), SHUTDOWN

### Manage Forms

Managing forms displays all forms and allows you to manage them. You can create new ones and edit existing designs. You can directly access the different parts of the form, such as the Headers, Templates, and Fields. Click on a form to edit the form. Click on Show Form to show the form design once it has been edited.

## Edit Form

Select the Edit Form button or click on the name of the form to edit the form configuration. The Edit Form option has two pages, DISPLAY GRAPHICS and ADVANCED SETTINGS. The button at the top of each page toggles between the two, allowing easy access and editing.

The screenshot shows a web browser window with the URL [https://stats.slimcd.com/soft/interface/admin-forms.asp?action\\_formid=79&su](https://stats.slimcd.com/soft/interface/admin-forms.asp?action_formid=79&su). The page title is "Merchant Login Center". The navigation menu includes "Account", "Confia", "Fraud Control", "Sessions", "Stats", "Vops", "ClientID: 1032", "SiteID: All", and "Logout". The main content area is titled "Configuration Configure Forms for 1032".

**Edit a Form's DISPLAY GRAPHICS** [GO TO ADVANCED SETTINGS PAGE](#)

Form ID: 79  
ClientID: 1032  
Company: RETAIL TEST ACCOUNT  
Form Name:  Active:   
Created: 3/8/2012 10:01:00 AM  
Modified: 3/8/2012 11:39:00 AM  
Version:  (Your version number for the form)  
• Use this to identify the form and keep it in sync with your application.

**Page Options**

Page Title Text:   
Local Path to Company Logo:   
Page BgColor:   
Page Background Image:   
Page Link:   
Page VLink:   
Page ALink:   
Page NoMargins:   
Page Center:   
Page/Table Width:   
Display SessionID:   
Display Sitename:   
Display Privacy Policy:   
Privacy Policy URL:

### *Edit Form – DISPLAY GRAPHICS (a)*

The form configuration has a number of sections. Each section allows the user to specify different aspects of the form.

**Form Name** - This should be set to a unique value you can use to identify this form design. It is supplied when calling CreateSession.

*Active* – This setting identifies if the form can be used or not.

*Version* – This is your version number for the form so that you can keep track of your designs once you export them and save them in your source control system.

## Page Options

The *Page Options* control the HTML for each page.

*Page Title Text* – Text to appear in the browser’s title bar

*Local Path to Company Logo* – Provide an image to SLIM CD so it appears under or SSL (https) certificate, or provide a fully qualified https:// link to an image on your own server under SSL.

*Page BGColor* – Background color for the page, as a # with a 6-digit HEX code, or a short color name

*Page Background Image* – URL or local path to the background image for the

*Page Link* – BODY tag for LINK color

*Page VLink* - BODY tag for VLINK color

*Page ALink* - BODY tag for ALINK color

*Page NoMargins* – Set the margins as normal, or “slammed” where the color touches the edge of the browser window

- Normal page margins
- Eliminate browser page margins

*Page Center* – Control position of the form within the page

- Left-justify page
- Center page

*Page/Table Width* – Control the width of the form’s table in pixels or percent

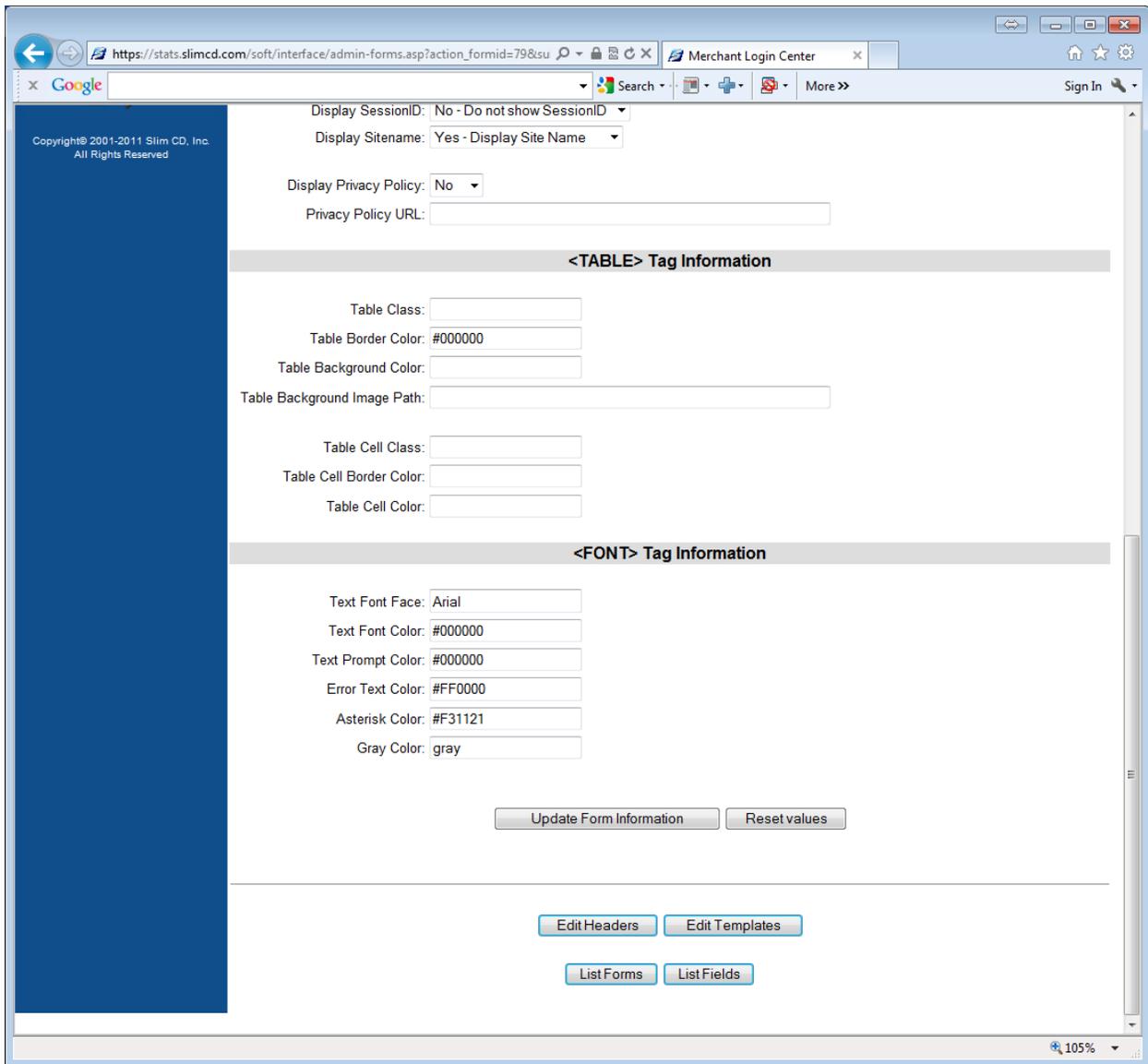
*Display SessionID* – Shows the session id on the screen

*Display Sitename* – Shows the site name on the screen (useful for one form for many merchants)

*Display Privacy Policy* – Display the privacy policy text/link on the bottom of the table/form.

*Privacy Policy URL* – Link to merchant’s privacy policy

**(Continued with next image)**



### *Edit Form – DISPLAY GRAPHICS (b)*

#### **Table Tag Information**

The *Table Tag Information* specifies the values to insert when generating HTML. (Note that the best way to judge the effectiveness of changes is to use SHOWSESSION to see the effects and VIEW SOURCE to see where the values are inserted.)

*Table Class* – CSS Class for the table tag

*Table Border Color* – Border color for the table

*Table Background Color* – HTML color code for the table background

*Table Background Image Path* – Image for the table background (either https:// url, or a locally hosted image)

*Table Cell Class* – CSS class for each cell block

*Table Cell Border Color* – Color code for border of cells

*Table Cell Color* – Color code for background of cells

## **Font Tag Information**

The *Font Tag Information* allows the developer to select font face and color for the page text, prompts, errors, and “grey” items.

*Text Font Face* – This is the HTML font face to be used inside the form

*Text Font Color* – This is the HTML color of the background text (instructions, etc)

*Text Prompt Color* – This is the HTML color for the prompts for user input.

*Error Text Color* – This is the color of error messages

*Asterisk Color* – This is the color of the asterisk that appears on “required” prompt items.

*Gray Color* – This is the color of instructional text that is smaller in font size and appearance (not currently used)

The *Navigation Buttons* appear at the bottom of this (and most) screens. These buttons allow you to quickly and easily switch between the different editors.

*Edit Headers* – Allows you to edit the HTML headers/footers for the page and tables.

*Edit Templates* – Allows you to edit the templates for the table, or for email receipts

*List Forms* – Lists all forms for your account

*List Fields* – Lists all fields for this form, and allows new fields to be created

**(Continued with next image)**

The screenshot shows a web browser window with the URL <https://stats.slimcd.com/soft/interface/admin-forms.asp>. The page title is "Configuration Configure Forms for 1032". The main content area is titled "Edit a Form's ADVANCED SETTINGS" and includes a button "GO TO DISPLAY GRAPHICS PAGE". The form details are as follows:

- Form ID: 79
- ClientID: 1032
- Company: RETAIL TEST ACCOUNT
- Form Name: POS\_demo\_form1
- Active: Yes
- Created: 3/8/2012 10:01:00 AM
- Modified: 3/8/2012 11:39:00 AM
- Version: 1.0 (Your version number for the form)
- Note: Use this to identify the form and keep it in sync with your application.

The "Total Order Limits" section contains two input fields:

- Minimum Total \$: 0
- Maximum Total \$: 999999.99

The "Card Types Accepted" section has the following checked options:

- Visa
- MasterCard
- Discover
- American Express
- Diners/Carte Blanche
- JCB

The "Cardnumber Entry Options" section includes the following dropdown menus:

- Card Entry Option: Show Both Keyed & Swiped Entry Fields
- On-Screen Keypad: None - Show a typeable HTML-Input Field for Card Number (for keyed entry)
- Card Type Option: None - Automatic Card Type Detection
- Expiration Entry Option: Show a single HTML-Input Field [MMYY]
- CVV2 / Seccode Option: None - Don't ask for CVV2 /Seccode information

### *Edit Form- ADVANCED SETTINGS (c)*

#### **Total Order Limits**

The *Total Order Limits* section allows the Hosted Page to automatically stop orders less than or more than the configured amounts.

*Minimum Total \$* - The minimum amount allowed for an order

*Maximum Total \$* - The maximum amount allowed for an order

## Card Types Accepted

The *Card Types Accepted* section identifies the types of cards to allow for this payment form. These include Visa, MasterCard, American Express, Discover, JCB, and Diners. (Note that JCB and Diners are now processed on the Discover network)

## Cardnumber Entry Options

The *Cardnumber Entry Options* specify how card numbers and expiration dates are entered

*Card Entry Option* – Card entry can be displayed as keyed, swiped, or both. The options are:

- Only Allow Keyed Entry
- Show both Keyed and Swiped entry fields
- Show swiped entry with a button to switch to keyed
- Show keyed entry with a button to switch to swiped

*On-Screen Keypad* – This option allows a keypad to be displayed instead of a normal edit field.

- None – Show a typable HTML edit field
- Table – Show a keypad as an HTML table
- Buttons – Show a keypad as HTML buttons

*Card Type Option* – Specifies if card types are identified automatically or must be selected

- None – Automatic card type detection
- Auto Detection with image display (shows the available card type logos)
- Show a user-selectable listbox of card types
- Show a user-selectable radio buttons of card types

*Expiration Entry Option* - Specifies how the expiration date will be entered

- Show a single HTML input field [MMYY]
- Show separate HTML pulldowns for [MM] and [YYYY]

*CVV2/Seccode Option* – Specifies how CVV2 information will be captured

- Show CVV2 Input Field (data entry optional)
- Show CVV2 Input Field (data entry required)
- Show CVV2 Input Field (data entry required for keyed transactions)

**(Continued with next image)**

The screenshot shows a web browser window with the URL <https://stats.slimcd.com/soft/interface/admin-forms.asp>. The page is titled "Form/Session Control Options" and contains several configuration sections:

- Form/Session Control Options:**
  - Decline Counter: Unlimited. (Allow any number of cards to be tried)
  - Duplicate Handling: DECLINE the DUPLICATE and display DUPLICATE\_DECLINED for the second transaction.
    - Specifies what to do with a DECLINE cause by a previously approved transaction.
    - Duplicate checking works best if you supply a "client\_transref" field that uniquely identifies each transaction.
    - Duplicate checking is disabled if the "allow\_duplicates" field is set to "yes".
  - Display Timer: No - Do not show onscreen timer
  - Timer Seconds: 0 Seconds to count down for a timer (0=No Timer)
  - Timer Action: Nothing - Do not do anything when timer expires
- Hosted-Page Post Processing Web Options:**
  - PostBack URL: [Empty text box]
    - This URL receives a "silent post" for each card transaction
    - There is no user interface (html code) expected or displayed from the silent post.
  - Redirect URL: [Empty text box]
    - This URL is visible in the browser and navigates the user away from the payment page.
    - This URL is used only at the end of the payment process.
    - If receipts are show, this URL appears as a "Continue" link
    - If receipts are not shown, the Redirect URL is invoked directly at the end of the payment.
    - This URL can display results saved from a previous PostBack or retrieved from CheckSession
- Final Approval/Decline Screen Text:**
  - Show Approved Receipts: Show receipt and print option upon approval.
    - Receipts are only shown for approved transactions
    - Receipts can contain "Print" links if specified.
    - Receipts can contain "Continue" links if a Redirect URL is specified.
    - Receipts can contain "Close" links if no Redirect URL is specified.
  - Approved Text: Payment Approved. Please close the browser. (if no receipt shown and not redirected)
  - Final Decline Text: Declined. Please start a new payment process. (if not redirected)
  - Close Link Text: [Empty text box] (<img> tag accepted)
  - Print Link Text: Print Receipt (<img> tag accepted)
  - Redirect Link Text: [Empty text box] (<img> tag accepted)
  - Browser Auto-close: No - Do not auto-close browser (show links instead) (Only auto-closes if receipts are not shown)

### *Edit Form – ADVANCED SETTINGS (d)*

#### **Form/Session Control Options**

The *Form/Session Control Options* section determines the way the form operates when processing transactions.

*Decline Counter* – indicates how many times a declined card will be allowed before the session returns control to the application. This can range from a single try to unlimited.

*Duplicate Handling* – indicates what to do if a DUPLICATE\_DECLINE message is returned from the gateway. If the form contains a "client\_transref" field and that field is pre-filled with a unique value by the developer for each transaction, we recommend that this be set to convert the DUPLICATE\_DECLINE into an approval. Otherwise, we suggest this be set to return the DUPLICATE\_DECLINE as a decline and let the calling application determine what to do. (Note

that if the developer creates an “allow\_duplicates” field and sets it to “yes”, then duplicates will never appear and this setting has no effect.)

*Display Timer* – Indicates an on-screen JavaScript timer should count down to encourage the clerk to complete the transaction within the specified countdown.

*Timer Seconds* –Identifies how many seconds to show in the countdown timer.

*Timer Action* – Identifies what to do at the end of the countdown. Choices are to do nothing, or to perform a CancelSession function and prevent the transaction from continuing.

## Hosted Paget Processing

The *Hosted-Page Post Processing* section allows the developer to specify the URLs to use when performing web-based server-to-server communication.

*PostBack URL* – Specifies the URL on the merchant’s server which will receive the responses for each credit card transaction as it is processed. Note that this URL can be called multiple times as multiple cards are attempted. The “completed” flag will indicate the final results. Below are the standard fields returned to the PostBack URL.

Field	Description
sessionid	The 40-byte session id from CreateSession
approved	Y, N, E (Y=Approved, N=Declined, E=Error)
transtype	The transaction type (usually SALE)
gateid	The SLIM CD gateid/Ticket# for this transaction. This uniquely identifies your transaction and can be used as a token on future transactions
authcode	6-Digit authcode for an approved transaction
avsreply	One-byte letter for the AVS (Address Verification Service). See the developer’s list of codes
cvv2reply	One-byte letter for the CVV2 response. See the developer’s list of codes
approvedamt	This will be the amount actually approved. If the “allow_partial” field is enabled, developers should check this value to identify when an approval does not cover the full requested amount. If the “allow_partial” field is not enabled, this value can be ignored.
invoiceno	This is the client_transref field as it will appear in SLIM CD reports (filtered for certain characters, etc)
bal	This is the balance remaining on the gift/credit card. It should be printed on the receipt whenever it is not empty (including if it is 0.00)
first6	The first 6 digits of the card
last4	The last 4 digits of the card
cardtype	The card type by letter (V,M,A,D,J,C)
cardname	The card name (Visa,MasterCard, American Express, Discover, JCB, Diners/Carte Blanche)
expmonth	The month of expiration for the card (MM format)
expyear	The year of expiration of the card
description	The description of the error or decline (if approved is not “Y”)

declinecount	The number of times a card has been declined
completed	“yes” means this is the final transaction.
(your field)	(user value)

Additional fields can be configured to return their contents when calling the PostBack URL by enabling the checkbox to send the field and its contents back with the PostBack (see FIELDS for more information)

*Redirect URL* – This URL is used to transfer the browser’s view back to the merchant’s website at the end of processing. The target of the redirection should use the previously acquired data from the PostBack URL, or should call CheckSession to determine the approved/declined status. Note that the “sessionid” is the only field passed back by default. Additional fields can be configured to be returned, but those fields will be viewable and editable by the end user. The PostBack is a much more secure way to pass data back to the caller, and then use the redirect page to display the previously sent information as an approval or a decline. If required, fields can be sent by checking the appropriate settings on the individual field configurations.

### Final Approval/Decline Screen Text

The *Final Approval/Decline Screen Text* controls the display after the credit card payment process has reached its conclusion. Note that the options for Receipt Display and the Redirect URL can cause this page to be skipped (if there is no receipt to display and there is a redirect link).

*Show Approved Receipts* – Shows the receipt template text on the screen for approved transactions

*Approved Text* – This text is shown instead of the receipt for approved transactions.

*Final Decline Text* – This text is displayed after the final transaction attempt is declined.

*Close Link Text* – This is the text to show in the link that closes the browser (if configured)

*Print Link Text* – This is the text for the link to run the JavaScript to print the receipt

*Redirect Link Text* – If a redirect link is configured AND a receipt is shown, this is the text for the link that allows the end user to continue back to the merchant’s website

*Browser Auto-close* – This option identifies if the ShowSession should attempt to close the browser after the last attempt.

- No – Do not auto-close browser (show links instead)
- Use JavaScript to close browser upon approval
- Use JavaScript to close browser upon approval or final decline

Note that the above options all work together, so that if you do not show receipts, you cannot print them and if you do not have a redirect URL, you will not see the redirect link text.

(Continued with next image)

The screenshot shows a web browser window with the URL <https://stats.slimcd.com/soft/interface/admin-forms.asp>. The page is titled "Merchant Login Center" and has a "Sign In" link in the top right corner. The main content area is divided into several sections:

- Link Texts:** Three input fields for "Close Link Text", "Print Link Text" (containing "Print Receipt"), and "Redirect Link Text". Each field has a note: "<img> tag accepted".
- Browser Auto-close:** A dropdown menu set to "No - Do not auto-close browser (show links instead)" with a note: "(Only auto-closes if receipts are not shown)".
- Email Options:** A section with a dropdown for "Email Receipt" (set to "No - Do not email receipt"), input fields for "Email From", "Email Subject" (with a note: "(this field's contents support data merge)"), "Email CC", and "Email Body" (with a note: "(Please edit the email template to create the receipt/email body)").
- Submit/Cancel Button Options:** A section with a dropdown for "Submit Button Type" (set to "<Submit> Button"), an input field for "Submit Button Text" (containing "Process Payment" with a note: "(Text to show for the 'Process Payment' button)"), a dropdown for "Submit Button ToolTips" (with a note: "(Tool Tips are shown when the mouse hovers over a field)"), input fields for "Submit Button Class", "Submit Button Style" (with a note: "(Only useful for Button Type of 'Button')"), "Submit Button Image" (with a note: "(Only useful for Button Type of 'Image')"), and "Submit Button Script" (containing "onclick='ProcessTransaction()'").
- Cancel Button:** A dropdown menu set to "No - Do not show the Cancel Transaction button".

At the bottom of the form, there are two buttons: "Update Form Information" and "Reset values". Below the form, there are four buttons: "Edit Headers", "Edit Templates", "List Forms", and "List Fields". The browser's status bar at the bottom right shows "105%".

### *Edit Form – ADVANCED SETTINGS (e)*

#### **Email Options**

The *Email Options* provide controls over emailing receipts to the consumer and/or the merchant.

*Email Receipt* – This setting allows receipts to be sent by email.

- No – Do not email receipt
- Yes – Email receipt upon approval
- Both – Email receipt and CC merchant upon approval
- CC Only – Only send receipt to merchant upon approval

*Email From* – Set the “from” email address here

*Email Subject* – Set the subject of the email. Note that this field operates like a “template” and supports mail-merge. This allows items like [sitename] or [gateid] to be merged into the subject line.

*Email CC* – This is the merchant’s email address to send receipts to when using the CC option. Multiple recipients can be specified by separating each entry with a semi-colon.

*Email Body* - (Please edit the email template to create the receipt/email body). See “Templates” for more information.

Note: When sending receipts to the consumer, you must edit the FIELD containing the user’s email address and set the checkbox that identifies that field as the recipient.

### **Submit/Cancel Button Options**

The *Submit/Cancel Button Options* allow configuration of the display and actions of these form buttons.

*Submit Button Type* – Specifies the HTML type as SUBMIT, BUTTON, or IMAGE tags.

*Submit Button Text* –Text that appears on the button

*Submit Button Tooltips* – Text that appears when the mouse is hovering over the button

*Submit Button Class* – CSS Class

*Submit Button Style* – CSS Style

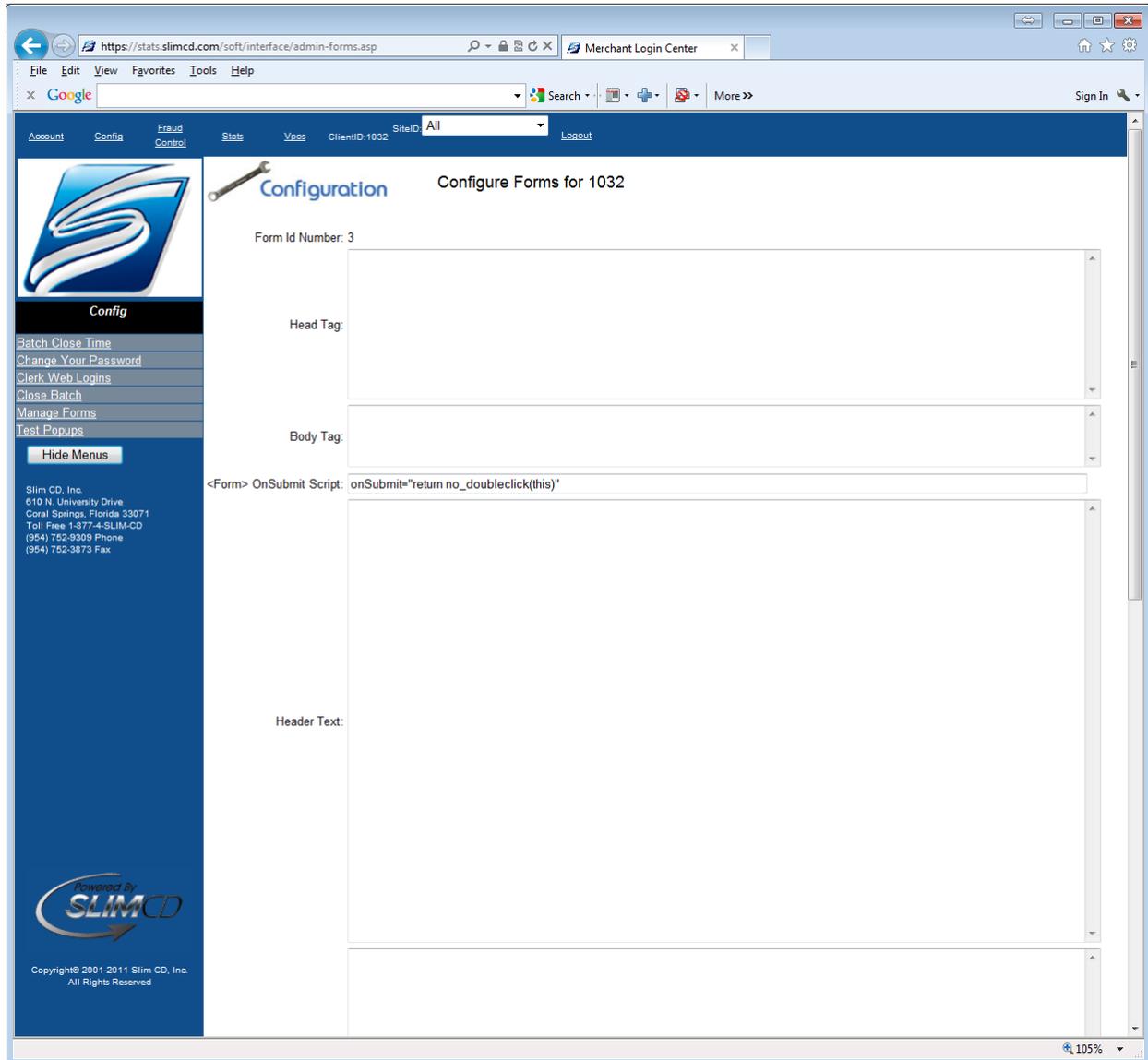
*Submit Button Image* – Image URL to display when the IMAGE option is selected

*Submit Button Script* – JavaScript commands for the button

*Cancel Button* – Allows a CancelSession to be invoked by the user. This option can hide or display the Cancel button.

## Edit Headers

The Headers are portions of the generated HTML. Some headers are actually part of the HTML, like the <head> and <body> tag. Others are simply HTML that is inserted into the page. Selecting EDIT HEADERS displays the following screen.



### *Edit Headers (a)*

## HEAD Tag

The *HEAD Tag* allows you to insert HTML/JavaScript between the <head> and </head> tags. Any code or script placed here will be included after the <head> and before the </head> tags.

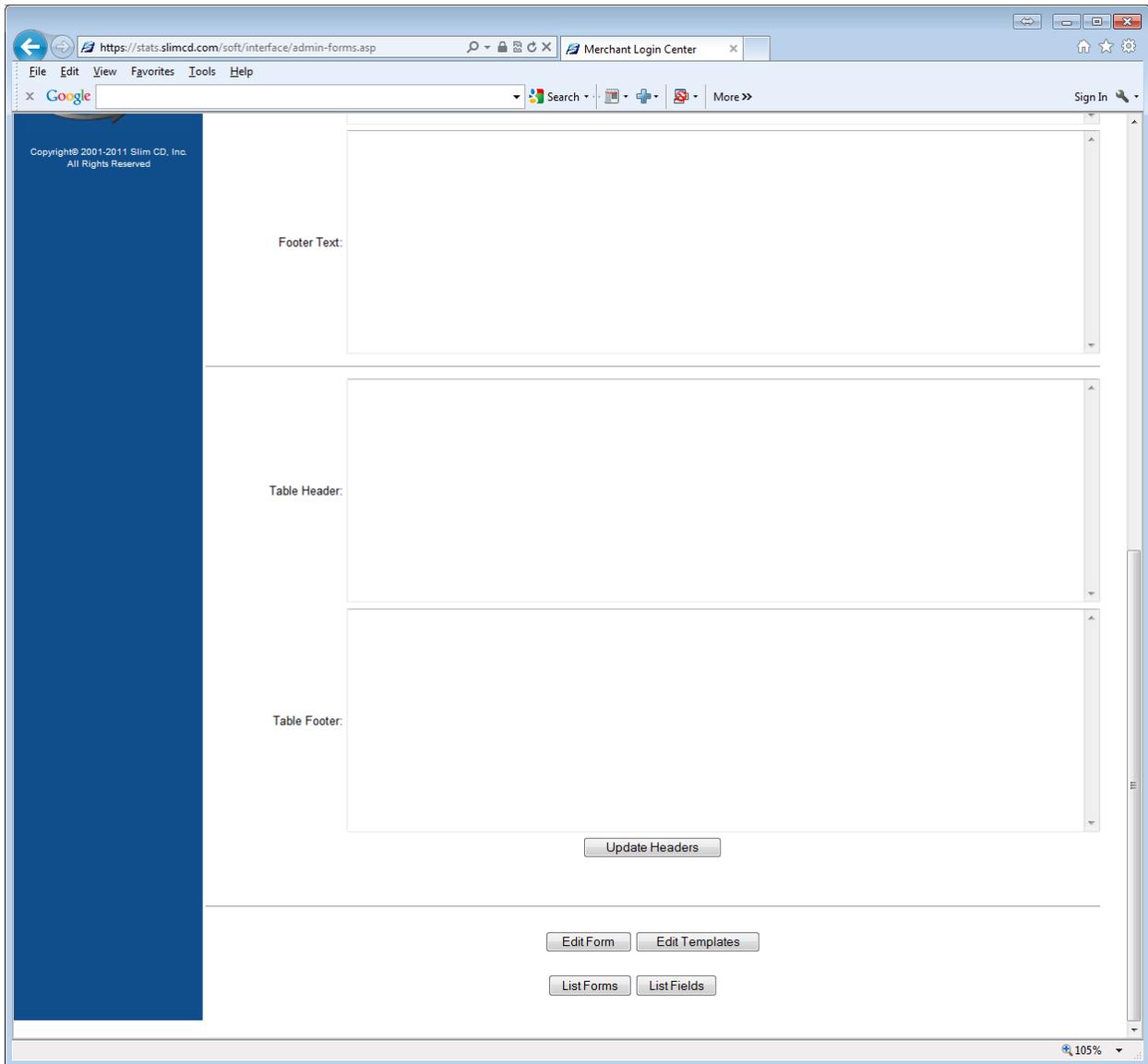
## **BODY Tag**

The *BODY Tag* allows actual code to be inserted in the middle of the <body> tag itself. The code inserted here will go after the <BODY, but before the trailing >. In other words, this will be inserted into the tag itself, so that JavaScript can be used.

## **The Header Text**

The *Header Text* is placed after the <body> and any page logos/images (see Edit Form). The Header Text can be used to display graphics on the top and left of the <table> used to hold the form fields. If graphics is displayed on the left, use either a CSS <div> tag, or create a table in the header and have the first column contain the leftmost graphics. If using the table/column approach, be sure to leave an open middle column for the table containing the form. Close any open columns and tables in the footer text.

**(Continued with next image)**



### Footer Text *Edit Headers (b)*

#### Footer Text

The *Footer Text* is inserted after the generation of the <table> containing the form. The Footer text should place graphics on the right and/or bottom of the page. If a 3-column table approach is used to create graphics on the left, the footer should close the middle column and create the rightmost column. Finally, any footer text (such as copyright notices, etc.) should be placed in the footer.

## **Table Header**

The *Table Header* is inserted after the outer table, before the table containing fields. This allows graphics to be placed inside the outermost table, but outside the fields themselves. This can also be used to split the field layout into two separate columns.

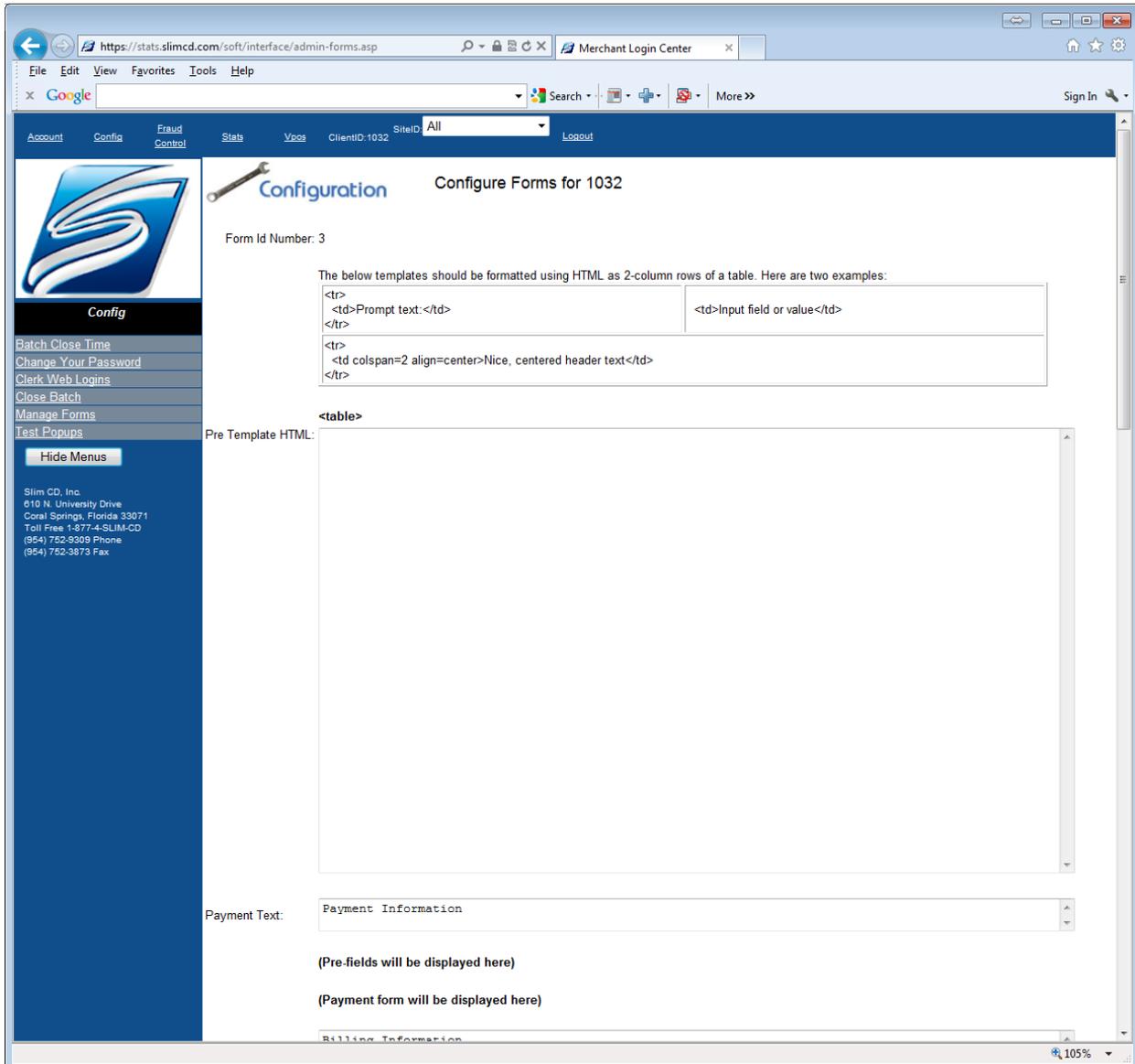
## **Table Footer**

The *Table Footer* is inserted after the table containing fields. This can have trailing graphics, or can close out a table that is split into multiple columns.

The *Navigation Buttons* allow quick access to the other form element editors, and appear at the bottom of each page.

## Edit Templates

Templates are different than headers because they appear as part of the innermost table of fields. They also support “Data Merge” where variables passed to the form can be displayed or utilized inside the page.



The screenshot shows a web browser window with the URL `https://stats.slimcd.com/soft/interface/admin-forms.asp`. The page title is "Merchant Login Center". The main content area is titled "Configuration Configure Forms for 1032". It shows a "Form Id Number: 3" and instructions: "The below templates should be formatted using HTML as 2-column rows of a table. Here are two examples:"

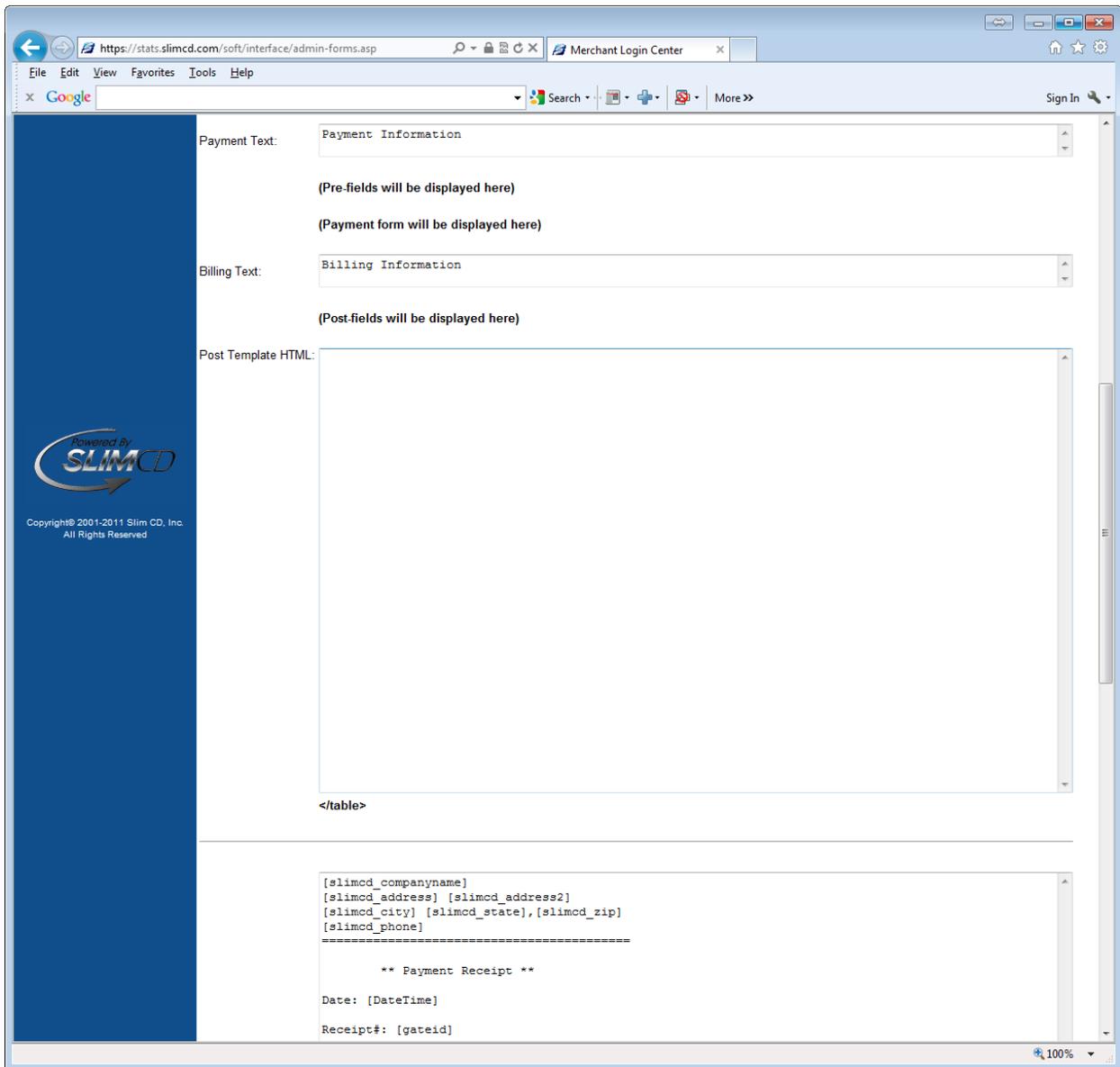
```
<tr>
  <td>Prompt text:</td>
  <td>Input field or value</td>
</tr>
<tr>
  <td colspan=2 align=center>Nice, centered header text</td>
</tr>
```

Below the examples is a large text area labeled "Pre Template HTML:" for editing. At the bottom, there are sections for "Payment Text:" and "Billing Information:".

### *Edit Templates(a)*

## Pre Template HTML

The *Pre Template HTML* is inserted before the actual fields. This occurs after the HTML `<table>` tag, so any template text here should be formatted as a row in a 2-column table. This can be done with two separate columns or with the use of `<td colspan=2>` to make one large area across the table. See below for a list of data merge fields.



### *Edit Templates (b)*

#### **Payment Text**

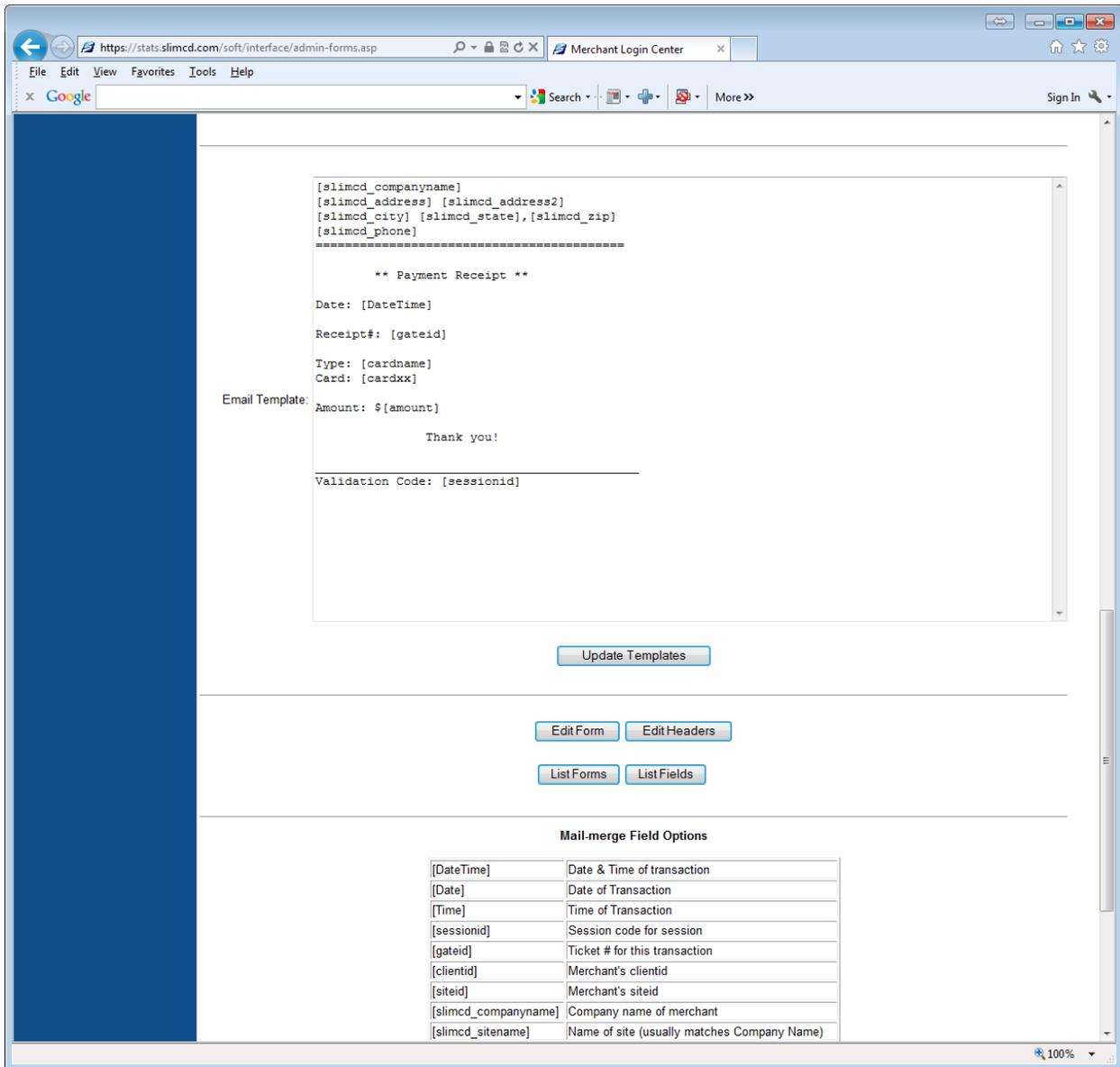
The *Payment Text* template controls the header text verbiage used before the “pre-fields”. Note that this can contain HTML, however this text will not support mail-merge values.

#### **Billing Text**

The *Billing Text* template controls the header text verbiage after the “pre-fields”. Note that this can contain HTML, however this text will not support mail-merge values.

## Post Template HTML

The *Post Template HTML* follows the same rules as the Pre Template. This is merged in as rows of the table, and can contain mail-merge fields. See below for a list of data merge fields.



```
[slimcd_companyname]
[slimcd_address] [slimcd_address2]
[slimcd_city] [slimcd_state],[slimcd_zip]
[slimcd_phone]
=====
** Payment Receipt **
Date: [DateTime]
Receipt#: [gateid]
Type: [cardname]
Card: [cardxx]
Email Template: Amount: ${amount}
                Thank you!
Validation Code: [sessionid]
```

Update Templates

Edit Form Edit Headers

List Forms List Fields

Mail-merge Field Options

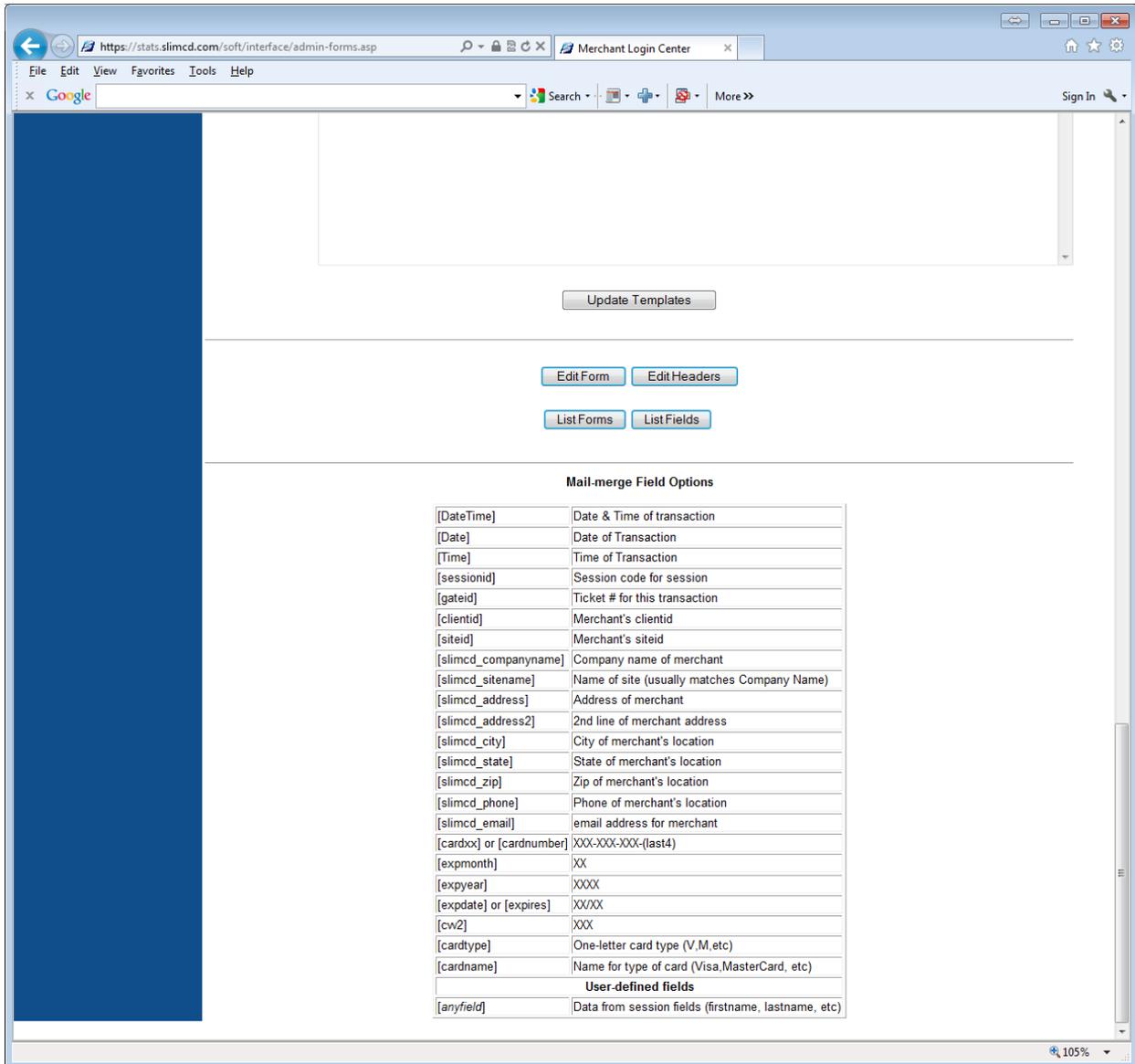
[DateTime]	Date & Time of transaction
[Date]	Date of Transaction
[Time]	Time of Transaction
[sessionid]	Session code for session
[gateid]	Ticket # for this transaction
[clientid]	Merchant's clientid
[siteid]	Merchant's siteid
[slimcd_companyname]	Company name of merchant
[slimcd_sitename]	Name of site (usually matches Company Name)

### *Edit Templates (c)*

## Email Template

The *Email Template* is used to format the receipt to be used for sending emails. A standard format will be included. You can observe the mail-merge field names in square brackets.

The *Navigation Buttons* allow quick access to the other form element editors, and appear at the bottom of each page. In this case, a legend of the available fieldnames for use with data merge and their definitions appears after the navigation buttons.



### *Edit Templates (d)*

#### **Mail-merge Field Options**

The *Mail-merge Field Options* section provides a quick legend to all of the built-in mail-merge fields. You can merge data from the merchant's setup (company name, site name, address, etc.). You can merge data for a receipt (x-filled card number, type of card, etc.). Finally, you can merge in ANY field you have posted over, even if it does not have a formal definition. In other words, if you call "CreateSession" and supply a field called "myCustomField", then you can reference "[myCustomField]" to place that value on

the page. This provides a great way to merge in custom information without having to create specific form fields. In addition, any formal “form field” can also be merged using this technique. Simply create the form field with the “Add Field” button and use its name to merge the value into any template.

## List Fields

Fields are used to configure and control data capture in a Hosted Payment Page. The List Fields function displays the fields and provides easy visibility to see which fields are above or below the card number entry. In addition, field order can easily be adjusted.

The screenshot shows the 'Configure Forms for 1032' interface. A table lists 12 fields with the following columns: Field Name, Active, Prompt, Data Type, HTML Type, Display Order, Payment, Call Back, Post Back, Redirect, Email, and Action. The 'Active' column is highlighted in green for all fields.

Field Name	Active	Prompt:	Data Type	HTML Type	Display Order	Payment	Call Back	Post Back	Redirect	Email	Action
<a href="#">checkbox</a>	ACTIVE	CheckBox Prompt:	bit	checkbox	1 Post	True	False	False	False	False	EDIT FIELD
<a href="#">disabled_field</a>	ACTIVE	Cannot enter:	char	input	1 Post	True	False	False	False	False	EDIT FIELD
<a href="#">Field1</a>	ACTIVE	Enter Value 1:	char	input	1 Post	True	False	False	False	False	EDIT FIELD
<a href="#">Field2</a>	ACTIVE	Enter Value 2:	char	input	1 Post	True	False	False	False	False	EDIT FIELD
<a href="#">ListColors</a>	ACTIVE	Enter Color:	char	listbox	1 Post	True	False	False	False	False	EDIT FIELD
<a href="#">Multiline</a>	ACTIVE	Enter Multiline Text:	char	textarea	1 Post	True	False	False	False	False	EDIT FIELD
<a href="#">passwd</a>	ACTIVE	Enter Pwd:	char	password	1 Post	True	False	False	False	False	EDIT FIELD
<a href="#">RadioButtons</a>	ACTIVE	Push A Radio:	char	radio	1 Post	True	False	False	False	False	EDIT FIELD
<a href="#">State</a>	ACTIVE	Enter State:	char	input	1 Post	True	False	False	False	False	EDIT FIELD
<a href="#">state_listbox</a>	ACTIVE	Select State:	char	state	1 Post	True	False	False	False	False	EDIT FIELD
<a href="#">Text_Checkbox</a>	ACTIVE	Enter Text:	char	checkbox	1 Post	True	False	False	False	False	EDIT FIELD
<a href="#">ZipCode</a>	ACTIVE	Enter Text:	char	input	1 Post	True	False	False	False	False	EDIT FIELD

12 fields listed.

*(List Fields screen)*

From this screen, you can easily change the order or pre/post position of a field. Click on the pulldown to change the information. Note that you can also see the HTML TYPE for a field and it's settings for gateway, postback and more. Click on the field name or the EDIT FIELD button to edit the details of a field.

## Edit Field

Add/Edit a field lets you configure data input or storage items on the form.

The screenshot shows the 'Configure Forms for 1032' interface. The left sidebar contains a 'Config' menu with options like 'Batch Close Time', 'Change Your Password', 'Clerk Web Logins', 'Close Batch', 'Manage Forms', and 'Test Popups'. The main content area is titled 'Configuration' and includes sections for 'Field Definition', 'Display Position', 'Form Fields', 'Display Attributes', and 'Data Values'. The 'Field Definition' section shows 'Field Name: Field1' and 'Active: Yes'. The 'Display Position' section shows 'Group Type: Display after payment input' and 'Display Order: 1'. The 'Form Fields' section includes 'Format: <HTML FIELD>', 'Prompt Text: Enter Value 1', 'HTML Type: <Input> TextField', 'Hint Text: This is the first value', 'Tool Tip Text: (empty)', 'Required: Always Required', and 'Description: Value 1'. The 'Display Attributes' section includes 'HTML Display Rows: 1', 'HTML Display Size: 0', 'HTML Maximum Length: 0', 'Disabled: unchecked', 'style: (empty)', and 'class: (empty)'. The 'Data Values' section is currently empty.

### *Edit Field (a)*

## Field Definition

The *Field Definition* identifies and activates the field.

*Field Name* - specifies the name of the field as it should appear in the HTML form, in CreateSession, and in CheckSession. The best practice is to create your field names such that they match to SLIM CD Gateway field names for any field which are utilized by the gateway (names, address, etc.). This is not required, but does reduce confusion.

*Active* - allows a field to be enabled or disabled, if desired. Note that disabling a field simply stops it from appearing on the form. Sending data to the field with `CreateSession` may be discouraged for disabled fields. Instead, use `HIDDEN` or `DATABASE-ONLY` fields for that purpose.

## Display Position

The *Display Position* settings determine where the field is displayed on the page

*Group Type* – Identifies if the field is positioned before or after the card number entry area.

*Display Order* – Specifies the order of the fields within the group type, with lowest number first

## Form Field

The *Form Field* information controls the look and text for the field when generated as HTML.

The first item shows a mock-up of how the field looks when generated on the page. It shows the Prompt, the HTML Field, and the “Hint Text”. This is a display-only legend.

*Prompt Text* – This text appears in the leftmost column of the table, providing a label for the HTML input field

*HTML Type* – This pull down selects the type of HTML (if any) to be generated. Options are:

- <Input> Text Field
- <Checkbox>
- <Radio> Buttons
- <Select> ListBox
- <Select> State Listbox
- <Select> Country Listbox
- <TextArea> (Multi-Line input)
- <Input> Password
- Display as HTML Text
- Hidden Form Field
- Database Only (not on form)
- <label> (2-Column Section header)
- <hr> (Horizontal Rule)

*Hint Text* – This text is displayed next to the HTML, providing instructions on how to enter data (such as `MMYY` for expiration date)

*Tool Tip Text* – This text is displayed when the mouse hovers over the input field

*Required* – Causes users to input values under specific conditions. Options are:

- Not required.
- Always Required.
- Required for keyed input.
- Required for swiped input.

*Description* – This text is displayed in error messages instead of the field’s name. For example, if the field is named “f\_name” and this value is set to “first name”, an error message would read “Enter first name” instead of “Enter f\_name”

## Display Attributes

The *Display Attributes* section controls the look of HTML items on the form.

*HTML Display Rows* – Number of rows for <textarea> multi-line fields

*HTML Display Size* – Width for <input> or <textarea> fields

Note:

For State listbox, enter "2" to display 2-letter abbreviations instead of state names.

For Country listbox, enter "3" to display 2-letter abbreviations instead of state names.

*HTML Maximum Length* – Sets the HTML “maxlength” value to limit typing.

*Disabled* – Starts the field as DISABLED (greyed). To enable the field requires custom JavaScript.

*style=* - Allows custom CSS Style attributes to be specified

*class=* - Allows custom CSS Class attributes to be specified

**Data Values**

Data Type:  (causes filtering by this value, not validation)

Default Value:  (default value for this field)

- Checkboxes: For a checked default, set this to match the value in the Value List
- List/Radio: To identify the default value set this to match ONE of the values in the Value List
- All Others: This value will be the default show/stored, unless overridden by session or user input

Value Lists:  (semicolon-separated list of options)

- Checkbox: value to return when checked
- List/Radio: Format Option 1 = value1,value2,value3
- List/Radio: Format Option 2 = value1=label1,value2=label2,value3=label3
- (not needed for State or Country ListBox types)

Case Sensitive:  (Field value matching will be case-sensitive)

---

**Form/Data Actions**

Validation Type:  ▾

Value Match field:  (other fieldname to match against)

- Only effective with "Matching Field Contents" Validation type
- Useful for email address or password matching
- Only set one of the two fields to validate a match (no need to set both)

Text Scrubbing:  ▾

Text Filtering:  ▾ (For user-typeable fields)

HTML/JavaScript Text:

(inserted before the closing > for the field)

---

**Processing Controls**

Payment Gateway:  (Will be passed to SLIM CD)

SLIM CD Field Name:  (overrides "Field Name" when sending data to SLIM CD gateway)

- Use this to map your fieldnames to SLIM CD fieldnames (if needed)

Callback:  (Will be passed back to checksession)

Postback:  (Will be passed to PostBack URL)

Redirect:  (Will be passed to Redirect URL)

Receipt Email Address:  (This field contains the "to" email address for receipts)

---

## *Edit Field (b)*

### Data values

The *Data values* specify the type of data and defaults for the fields.

*Data Type* - Select from Text, Integer, Money or Yes/No. The selection here controls basic filtering for input and return values from some list/checkboxes.

*Default Value* – Enter the default value to be used if not overridden by CreateSession data. If values for listboxes/radio buttons are different than names (see below), enter the value here.

*Value Lists* - Provide a list of values for ListBox or Radio Buttons, separated by semi-colons.

Exampe 1: Small;Medium;Large

For lists or radio buttons, item names can be different from values as follows:

Exampole 2: S=Small;M=Medium;L=Large

*Case Sensitive* - Check this if the value should be case-sensitive when comparing values to lists

## Form/Data Actions

The *Form/Data Actions* section specifies data validation or filtering actions to take on fields when SUBMIT is pressed.

*Validation Type* – Select from a variety of types that include name matching logic, email logic, etc.

- None - Performs no validation
- First/Last Name – Makes sure there are vowels and constantans in the text
- Email Address – Insures an @ and at least one period is in the text
- Numeric Street Address Information – Insures there are numeric values in the text
- Street Address Number & Text – Insures there are both numeric and alpha characters in the text
- State 2-letter Abbreviation – Matches against the built-in list of state abbreviations
- State Name – Matches against the built-in list of state names
- 5-Digit US Zip Code – Insures a 5 digit numeric value is entered
- 5 or 9 Digit US Zip Code – Allows a 5 or 9 digit numeric value to be entered
- 5 or 9 Digit US Zip Code, or Canada Postal Code– Allows a 5 digit numeric value, or a 6-digit Canadian Postal format text to be entered
- 10-Digit US Phone (Area Code & Number) – Insures 10 numeric digits are entered
- I Agree field (must be checked/filled) – The field must either be a checkbox that is checked or must be an INPUT box. The field must the value “1”, “YES”, “AGREE”, or “I AGREE”.
- Matching Field Contents (such as password, email, etc) – This checkbox indicates that the contents of one field must match the contents of another. When this is checked, fill in the name of the field to be matched against in the “Value Match Field” space below.

*Value Match Field* – The fieldname of another field to use when comparing the two field values for a match. For example, a form may contain a “password” and “reenter\_password” fields. If the ShowSession is to insure these data match, configure one of the two fields to match the other by selecting the appropriate filter and entering the second field’s name. A best practice

is to have the filtering rule on the first (such as email filtering) and a matching rule on the second (so the second field must match the first).

*Text Scrubbing* – This setting can cause ShowSession to reject input with unacceptable language. This is a useful feature if the form is consumer-facing and can be subject to general input rather than known customers.

- None
- Bad Words/Vulgar Language

*Text Filtering* – Text filtering strips input PRIOR to saving or further scrubbing. This is useful to help eliminate typographical extraneous characters (for example, quotes from phone numbers). Be careful to make sure your Validation rules and text filtering are in sync so that you are not stripping out required characters (like @ from email address).

- None (Determined by field data type)
- Natural Numbers: {[0-9]} (Positive Integers)
- Whole Numbers: {'-', [0-9]} (All Integers)
- Decimals: {[0-9], '.'} (Positive integer or decimal number)
- Any Number: {'-', [0-9], '.'} (Any integer or decimal number)
- Alphabetic {[A-Z], [a-z], space} (Letters and space)
- Alpha-numeric {[A-Z], [a-z], [0-9], space, '-'} (Letters, Number space and dash)
- Phone {[0-9], space, [(]-.+]} (Numbers and phone punctuation)
- Email {[A-Z], [a-z], [0-9], [ @ . \_ - ]} (Standard email address characters)
- Proper Name {[A-Z], [a-z], space, ['-']} (Letters in First/Last Name)
- Street {[0-9], [A-Z], [a-z], space, [-, #]} (Letters in Street Address)
- Sentence {[A-Z], [a-z], [0-9], space, [., - ! ? ( ) & % \$ @ # : / ']} (Typical punctuation)

*HTML/JavaScript Text* – Any HTML or JavaScript for an individual field can be placed here, such as an OnChange , OnBlur or other JavaScript or HTML structure.

## Processing Controls

### *Processing Controls*

*Payment Gateway* – Checkbox that indicates this field should be sent to the SLIM CD gateway as part of payment processing.

*SLIM CD Field Name* – Name for the field when sent to the SLIM CD gateway. Note that this is only filled if the field's name does not match the name used by SLIM CD's gateway. For example, the form may have "BillingZip", but the SLIM CD gateway expects "zip". Place the SLIM CD field name here so the value is correctly named when sending to SLIM CD.

*Callback* – This checkbox allows the value to be included in the returned data when CheckSession is called.

*Postback* - This checkbox allows the value to be included in the returned data when the PostBack URL is called.

*Redirect* - This checkbox allows the value to be included in the returned data when the REDIRECT URL is invoked. Please note that using this checkbox will cause the field and value to become part of the URL and visible to the end-user. It is recommended that this setting be avoided whenever possible. CheckSession and the PostBack URL provide alternative ways to retrieve the same data without exposing it to the end user or making it subject to user alterations.

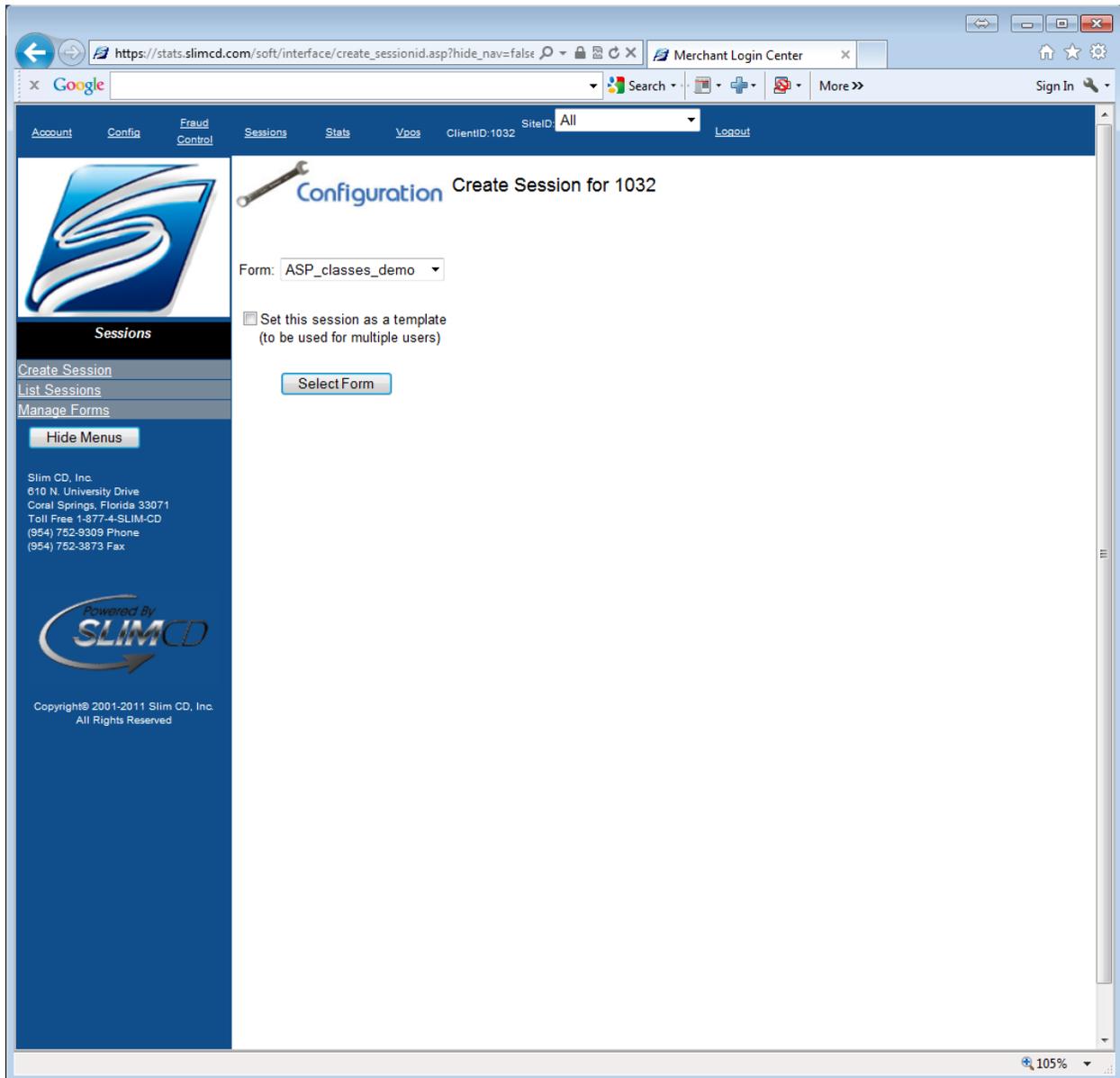
*Receipt Email Address* – This checkbox allows you to identify which field to use as the "TO" address when sending email receipts. Note that you can have more than one field selected, which allows multiple emails to be sent. In addition, hidden fields can contain values so that email receipts are sent to multiple merchant email addresses if desired.

The standard *Navigation buttons* appear at the bottom of the screen.

# Chapter 3: Manually Creating Sessions

## Create Session Menu

SLIM CD’s website contains a menu selection that allows sessions to be created for use in websites, email campaigns, via Facebook or Twitter.



### Create Session (a)

To create a session, first select the form to use. If this form is to be used by multiple recipients, such as a church donation page, check the box that indicates the session is a “template” for multiple users. When a session is created for a single user (without the “template” checkbox filled), the following page is displayed.

Merchant Login Center

Account Config Fraud Control Sessions Stats Vpos ClientID: 1032 SiteID: All Logout

**Configuration** Create Session for 1032

Order	Name	Data Type	HTML Type	Prompt	Value
1	Amount	char	input	Amount:	<input type="text"/>
2	first_name	char	text	First Name:	<input type="text"/>
3	last_name	char	text	Last Name:	<input type="text"/>
4	Address	char	input	Address:	<input type="text"/>
5	Zip	char	input	Enter Zip:	<input type="text"/>
*	transtype	char	none	TransType:	SALE

**Sessions**

[Create Session](#)  
[List Sessions](#)  
[Manage Forms](#)

Slim CD, Inc.  
 610 N. University Drive  
 Coral Springs, Florida 33071  
 Toll Free 1-877-4-SLIM-CD  
 (954) 752-9309 Phone  
 (954) 752-3873 Fax

Powered By **SLIMCD**

Copyright © 2001-2011 Slim CD, Inc.  
 All Rights Reserved

105%

### ***Create Session – Single Session (b)***

Creating a session displays all of the fields on the form so that each can be pre-filled with data as desired. Note that the AMOUNT and the TRANSTYPE must be filled. The table shows the fields in their display order, with their associated data type, HTML type and prompt text. Enter the values to be pre-filled on the form. Click SUBMIT when done to generate a list of session options.

The screenshot displays the Merchant Login Center interface for creating a session. The browser address bar shows the URL: [https://stats.slimcd.com/soft/interface/create\\_sessionid.asp](https://stats.slimcd.com/soft/interface/create_sessionid.asp). The page title is "Configuration Create Session for 1032".

**New Session Created**

Link/URL to Show the Session:  
<https://stats.slimcd.com/soft/showsession.asp?sessionid=DB08B51048D02A7065303FAD06C514ACD7E1DB21>

You may copy/paste this Link/URL into an email, or post it on your website.

[Test the above Link](#)

**Send Link/URL via SMS Text Message or Email**

Mobile Phone:

SMS Message:

[Send via SMS](#)

Email:

Message Body:

(Payment Link/URL is automatically included in the email)

[Send via EMAIL](#)

**Send Payment Link/URL via a Facebook Message**

[facebook](#)

[Send Message](#)

**Sessions**

[Create Session](#)

[List Sessions](#)

[Manage Forms](#)

[Hide Menus](#)

Slim CD, Inc.  
 610 N. University Drive  
 Coral Springs, Florida 33071  
 Toll Free 1-877-4-SLIM-CD  
 (954) 762-9309 Phone  
 (954) 762-3873 Fax

Powered By **SLIMCD**

Copyright © 2001-2011 Slim CD, Inc.  
 All Rights Reserved

### ***Create Session- Single Session (c)***

The above screen is displayed once the pre-filled form data entry is complete. There are three primary sections of this page. The first displays the URL to the new session and provides a link to test the session and see the pre-filled data. The link can be copied and used however you desire, but please be aware that the session may contain confidential customer data, like the name or other personal information.

The second section allows the URL to be sent via SMS-Text or via EMAIL. Note that the link itself is automatically added to the SMS-Text or the email. The third button allows the link to be sent via a private Facebook message. Clicking on this will display a pop-up window so the user can log into Facebook and select recipients from their "Friends" list.

Account Config **Fraud Control** Sessions Stats Vpos ClientID: 1032 SiteID: All Logout

**Configuration** Create Session for 1032

**Warning: This is a template form for use by multiple recipients. Do not enter individual or personal information!**

Order	Name	Data Type	HTML Type	Prompt	Value
1	Amount	char	input	Amount:	
2	first_name	char	text	First Name:	
3	last_name	char	text	Last Name:	
4	Address	char	input	Address:	
5	Zip	char	input	Enter Zip:	
*	transtype	char	none	TransType:	SALE

Submit

**Sessions**

Create Session  
List Sessions  
Manage Forms

Hide Menus

Slim CD, Inc.  
610 N. University Drive  
Coral Springs, Florida 33071  
Toll Free 1-877-4-SLIM-CD  
(954) 752-9309 Phone  
(954) 752-3873 Fax

Powered By **SLIMCD**

Copyright © 2001-2011 Slim CD, Inc.  
All Rights Reserved

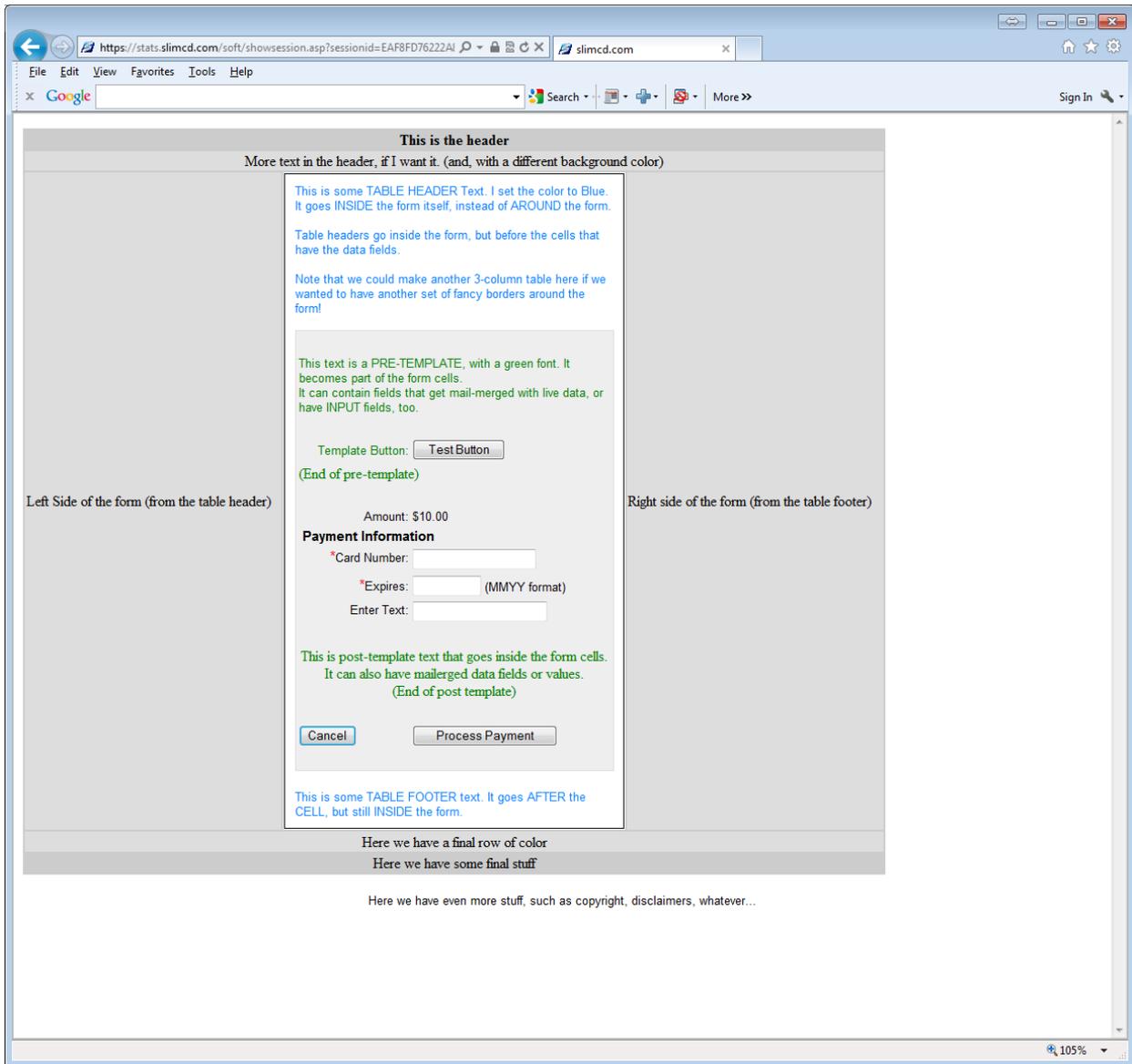
### ***Create Session – Template Session(d)***

The above screen is displayed if the user selects the “template” checkbox. In this case, the template to be created will be used by multiple consumers. Please note the large warning that advises against pre-filling the form with any personal information, such as name. Once the amount and transtype are provided, along with any information you wish to be pre-filled, click the SUBMIT button.

### ***Create Session – Template Session (e)***

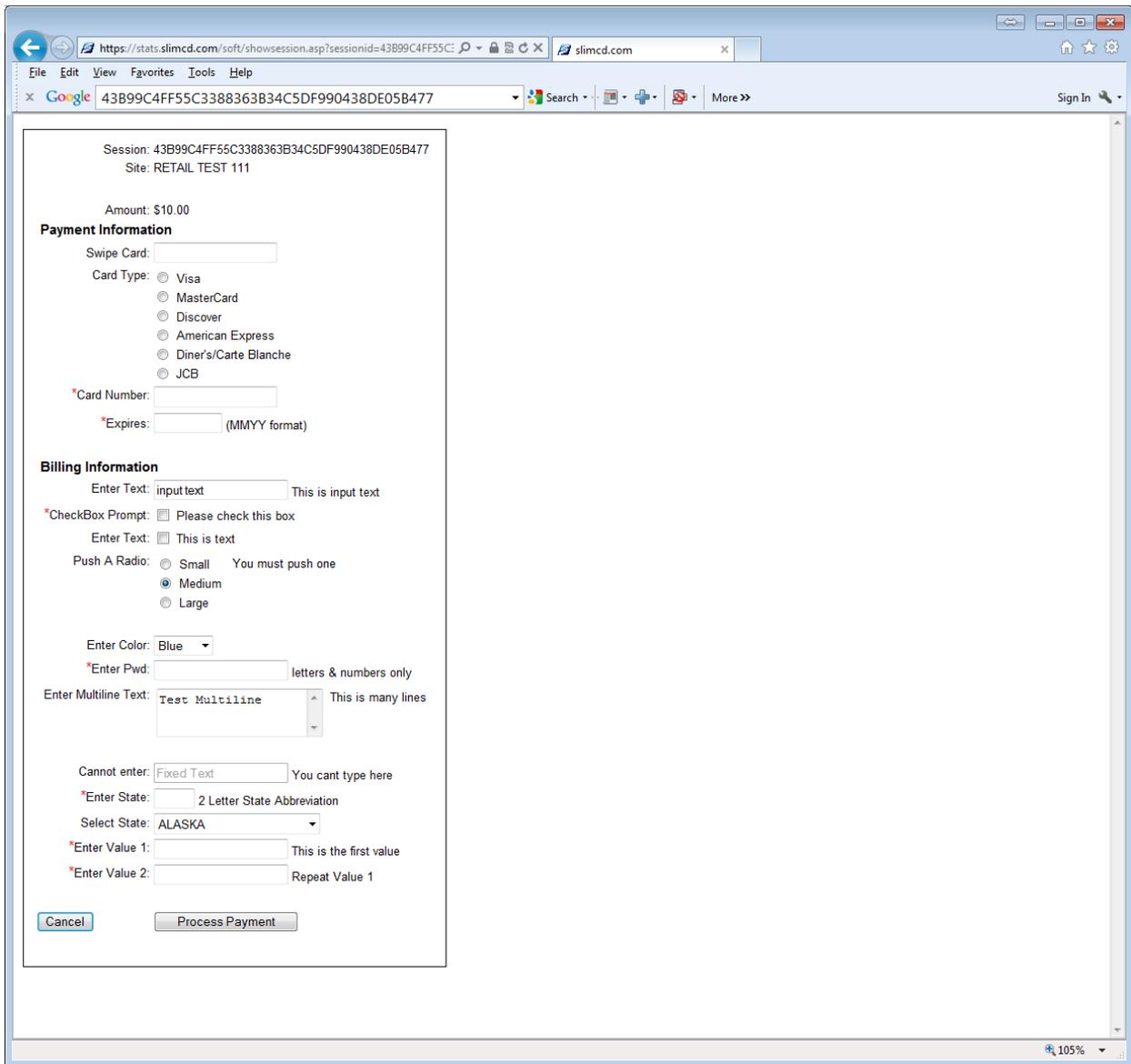
The above screen shows the options for a “template” session. Note that each template session will spawn individual sessions, so the options can vary from individual session. The first section of the above screen allows the user to cut/paste the link and test the link. The second section provides a simple way to create a “tinyurl.com” link to the session. Tinyurl.com is often used in print advertisement because the URL created is easily typed from printed media and tinyurl.com is a well-known service provider for these reduced-length URLs. Please note that tinyurl.com will retain the link it creates indefinitely, so please be sure you want to use this. The third section allows the URLs to be sent to multiple friends in Facebook, posted on your Facebook wall, or broadcast out using Twitter. If Twitter is used, Twitter will shorten the url using its own service that is similar to tinyurl.com. The above multi-user template services are great for non-profit donation pages and similar fund raisers.

# Appendix A - SAMPLES



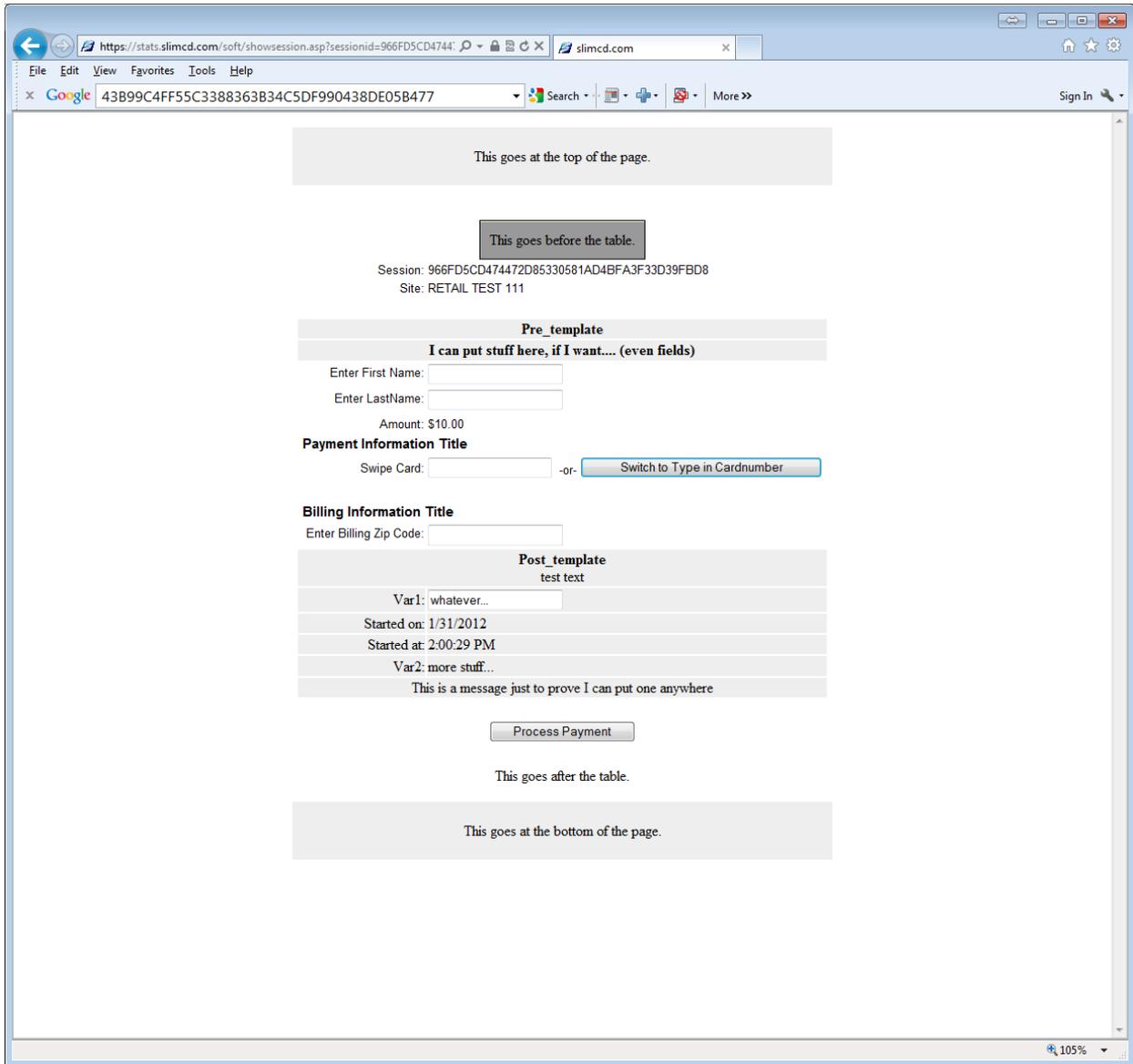
## HTML Placement Sample

This sample contains HTML examples of Headers, Footers, Table Headers, Table Footers, Pre-templates with buttons, post-templates.



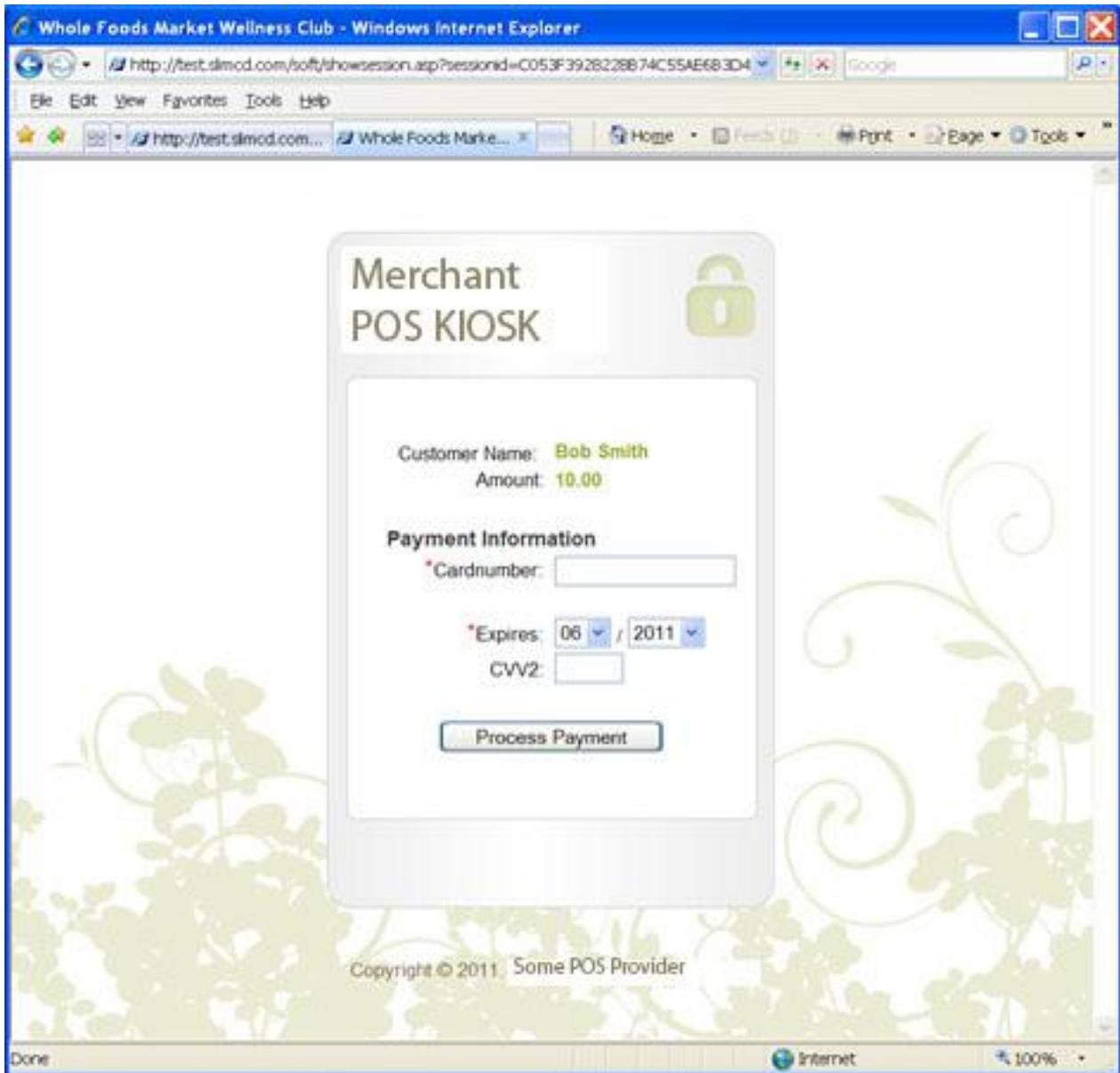
### *Field Types Sample*

This sample shows the different field types available.



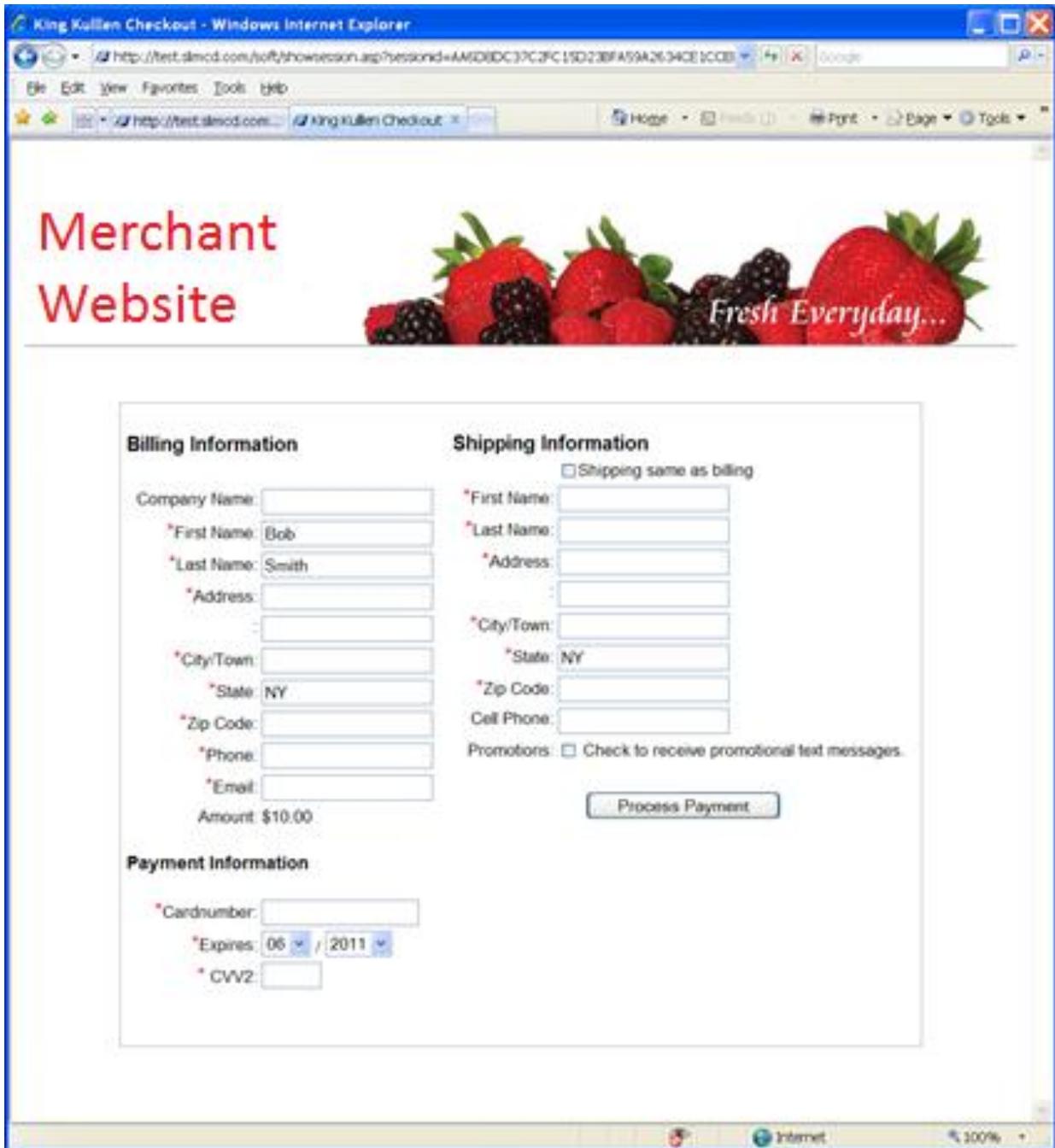
### *Templates Sample*

This sample shows data fields as part of templates, with merge.



*Sample KIOSK merchant page*

This sample shows a merchant's KIOSK that supports entry of card numbers.



*Sample WEB merchant page*

This merchant web page shows data split into two columns, with many required fields.

## Appendix B - Form Generation Layout

The form is generated in different sections. Most of these are merged into the form as it is created. The order of the generation for the configurable forms items is as follows:

### SECTIONS

*HEAD TAG* – Inserted between the <head> and </head> tags <sup>(A)</sup>

*BODY TAG* – Inserted into the <body> tag itself <sup>(A)</sup>

*HEADER TEXT* – In-line HTML inserted at the top of the page <sup>(A)</sup>

*IMAGE* - <img> tag created if an image is specified for the form <sup>(D)</sup>

*TABLE HEADERS* – In-line HTML inserted at the top of the table <sup>(A)</sup>

*PRE-TEMPLATES* – Mail-merged HTML, inside the field table but before the fields <sup>(B)</sup>

*PAYMENT TEXT LABEL* – Text string that precedes the fields before the card number entry <sup>(B)</sup>

*PRE-FIELDS* – Fields configured to display before the card number entry <sup>(C)</sup>

(CARDNUMBER ENTRY - automatically generated by the form, based on options selected) <sup>(D)</sup>

*BILLING TEXT LABEL* – Text string that precedes the fields after the card number entry <sup>(B)</sup>

*POST-FIELDS* - Fields configured to display after the card number entry <sup>(C)</sup>

*POST-TEMPLATE* – Mail-merged HTML inside the field table, but after the fields <sup>(B)</sup>

*TABLE FOOTER* – In-line HTML inserted at the bottom of the table <sup>(A)</sup>

*FOOTER TEXT* – In-line HTML inserted at the bottom of the table <sup>(A)</sup>

### CONFIGURATION TOOLS

- <sup>(A)</sup> HEADERS and TAGS are configured on the EDIT HEADERS screen
- <sup>(B)</sup> TEMPLATES are configured on the EDIT TEMPLATES screen
- <sup>(C)</sup> Fields and display order are configured on the LIST FIELDS screen
- <sup>(D)</sup> Image tag and card number entry are configured on the EDIT FORM screen

## HTML Generation

Form HTML is graphically displayed as follows, where [blue](#) or [\[square-brackets\]](#) represent inserted data from the configuration:

```
<HTML>
  <HEAD>
    [HEAD TAG, from EDIT HEADERS] (your "head" code)
  </HEAD>
  <BODY (link, vlink, alink, bgcolor, and background image)
    [BODY TAG, as configured]
  >

  <center> (if configured on the form)

  [HEADER TEXT, from EDIT HEADERS]

  <IMG tag> (if configured on the form)

  <table1 , a one-column table with TABLE border, color, bg image, class, width as configured)>

    <form with JavaScript, as configured)>

    [TABLE HEADER, from EDIT HEADERS]

    <center>

    <table2, a 2-column table with CELL colors, borders, images, etc.>

      [PRE-Template, from EDIT TEMPLATES]

      <tr> <td>PROMPT</td> <td>INPUTFIELD (hint text)</td> </tr>
      ... (one row for each field)

      [POST-Template, from EDIT TEMPLATES]

    </table2>

    [TABLE FOOTER, from EDIT HEADERS]

  </table1>

  [FOOTER TEXT, from EDIT HEADERS]

</BODY>
</HTML>
```

## Appendix C –State/Country ListBoxes

Below are the detailed descriptions of the built-in state and country listbox features.

### State ListBox

State list boxes automatically generate pulldowns with the states. You can produce full state names, 2-character abbreviations, or full names with 2-character values.

- The default displays state names. Leave display size and max length values as defaults.
- To display 2-letter state abbreviations, set the “display size” value to 2.
- To display state names but send abbreviations, leave the display size empty (or not 2) and set the “max length” value to 2.

### Country ListBox

Country list boxes automatically generate pulldowns with the countries. You can produce full country names, names, 3-character abbreviations, or ISO country codes.

- The default displays country names. Leave display size and max length values as defaults.
- To display country names but send 3-character abbreviations, leave the display size empty (or not 2) and set the “max length” value to 3.
- To display country names but send 3-digit ISO country codes, leave the display size empty (or not 2) and set the “data type” value to “integer”.