

CyberPay PCI Payment Page: Access and Testing for Release 3

Introduction

The new PCI-compliant web payment processor allows all CyberPay merchants to continue processing debit and credit card transactions in accordance with Payment Card Industry (PCI) guidelines. PCI is a series of industry security requirements implemented by Visa and Master Card (see Exhibit A for a view of the new process diagram).

With this new web application, CyberPay merchants can modify their current web front-end process to provide customer access to the new AIS-secured Debit/Credit Card Processing Page (see Exhibit B).

The page layout has been changed, and merchants can now customize the look and feel of the page by passing several new parameters.

Some of these new options include:

- URL for your logo.
- HTML for your header text.
- Background color or a background image.
- Font attributes, including: name, size, and color of the font.
- Post-transaction URL – allows you to determine whether customers should return to the payment page if the transaction declined.
- Double-blind card entry- lets you decide whether to mask credit card numbers to provide added security for public computers and card number protection.

We think these new features will improve your customers' overall experience when making a payment for the services you provide!

Required "POST" Fields

Before this new centrally controlled screen is presented to the customer, each merchant is required to do a **post** of the following fields to the AIS domain, including:

Field Name	Description	Field Size
MerchantID	Merchant ID assigned to you by the Student Financial Services during the initial setup. It is a 16 character number that consists of the following: <ol style="list-style-type: none"> 1. Location Code – one digit 2. Merchant ID – eleven digits (SFS provides) 3. Department Code – four digits 	16 char
OrderID	An OrderID is used to track the transaction throughout the whole process. You must create a unique Order ID for each transaction that is alphanumeric. The unique identifier must be 25 characters or less. The Order ID format is as follow: <ol style="list-style-type: none"> A. Your 4-digit NCR code followed by a dash "-" B. A unique identifier URSA and BruinCard only: <ol style="list-style-type: none"> A. UID followed by an underscore "_" 	40 char

	B. Your 4-digit NCR code followed by a dash “-“ C. A unique identifier	
MaxAmount	The maximum credit card amount that can be accepted.	7.2 num
FixedAmount	The transaction amount that must be authorized. The user will not be able to edit this field. The amount passed by the department is the amount that will be authorized.	7.2 num
Description	This corresponds to the FS description in the General Ledgers. This field must be in all CAPS and is provided by AMCO.	20 char
NCRCode	This four-character code is linked to your FS depository FAU.	4 char
ReturnURL	URL to return the transaction response. This field MUST have the Return URL only. No other parameters	255 char
CancelURL	URL to return the user to the merchant’s site. User never submitted the payment. The cancel button was hit. This field MUST have the Cancel URL only. No other parameters.	255 char
POSTURL	URL to post the transaction status. This field MUST have the POST URL only. No other parameters.	255 char
FAU (optional)	The FS FAU (Full Accounting Unit) to deposit all authorized transactions into. The format is as follow: 1. Location – 1 character 2. Dash – 1 character 3. Account Number – 6 characters 4. Dash – 1 character 5. Cost Center – 2 alphanumeric characters 6. Dash – 1 character 7. Fund – 5 characters 8. Dash – 1 character 9. Project – 6 alphanumeric characters 10. Dash – 1 character 11. Financial Class/Sub-Object – 6 characters 12. Dash – 1 character 13. Source – 6 alphanumeric characters If you are passing this field you must put ‘A999’ in the NCR Code field.	38 char
UserField1	Merchants may use this field to pass data back and forth from the request to the response. The information will be sent back to the merchant via query string.	200 char
BackgroundColor (optional)	Background color of the payment page.	15 char
BackgroundImageURL (optional)	Background image of the payment page. Applicable if Back Ground Color is not specified.	255 char

Header (optional)	HTML for your header text. Example: <h4 align=center>UCLA Parent Weekend 2006 January 23rd 2006 Online(Registration)</h4>	200 char
FontName (optional)	Font name to display the payment page labels. Example: Ariel, Verdana, or a font family; Ariel, Verdana, Times New Roman	60 char
FontSize (optional)	Font size to display the payment page labels. Example: 10 pt, 11 pt, 12 pt, 13 pt	6 char
FontColor (optional)	Font color to display the payment page labels.	15 char
ReturnToPaymentPageOnDecline (optional)	Boolean (Y/N) to return the customer to the payment page on a decline.	1 char
AccountDoubleBlind (optional)	Boolean (Y/N) to display a second credit card field for verification.	1 char

Return Codes/Fields

Following payment verification by CyberPay, the following fields of information are passed back to each merchant from AIS to include:

Field Name	Description	Field Size
OrderID	Same as above	
ReturnCode	Return code from CyberPay web service	4 char
ReturnCodeMessage	Return code message from CyberPay web service	40 char
AuthorizationResponseCode	Processor authorization response code Code Description A Approved C Call D Declined X Expired Card E Error	1 char
AuthorizationResponsesMessage	Processor authorization response message. If the transaction was not approved, check this message.	20 char
ApprovalCode	Processor approval code	9 char
ZipMatch	AVS result indicator Code Description Y Match N No match R,X Service unavailable	1 char
CVVResult	CVV result indicator Code Description M Match N No match I, P, S, U, X Unknown/Service unavailable	1 char

CardType	The field includes type of card used, a dash and last four digits of the card number. Example: MasterCard-1111	15 char
----------	---	---------

Additional Information

AIS and CyberPay Support do not provide programming support. We will try to answer any questions that you may have.

The following items are required for payment processing:

Preventing Duplicate Payments

Once your customers hit your "Pay Now" button, you should disable the button on your site to prevent duplicate requests.

The following JavaScript code prevents users from submitting more than one payment erroneously. In this example, it disables the "Pay Now" button when the customer clicks it.

Between **<HEAD>** tags:

```
<SCRIPT language="JavaScript">
  <!--
  function doSubmit() {
    document.form1.submitbtn.disabled = true;
  }
  // -->
</SCRIPT>
```

Between **<FORM>** tags:

```
<INPUT TYPE="BUTTON" name=submitbtn VALUE="Pay Now" ONCLICK="doSubmit()">
```

On the payment page, if the user cancels the transaction they will be routed to the cancel URL specified site. The OrderID and UserField1 data will be passed back via query string.

You must pass the parameter(s) you want included for each transaction. The parameters are not stored in your merchant configuration.

If you don't pass any parameters your customers will see the **default** payment page. It is the merchant's **responsibility to test** to make sure this is what they want.

You cannot pass a *Background Color* **and** a *Background Image* for the same transaction. These are mutually exclusive and only one can be selected.

Don't use **Red** as a background color. Error messages are displayed in red.

Use the *FixedAmount* field if your customers must pay a fixed amount. For example, if the customer has chosen items totaling \$250, you can populate this field and CyberPay will force the payment of this amount.

Use the *MaxAmount* field if your customers can pay up to but not over a certain amount. For example, if the student only owes \$200 and they cannot make an overpayment on their account, you can populate this field preventing them from over paying.

More about your Return Page:

A new credit card requirement requires all merchants to indicate on their Merchant Authorization/Confirmation Page the card type of all authorized transactions.

More about the Cancel URL:

The cancel URL you provide us will be used in the following ways:

When the user hit the cancel button on the payment page the CancelURL will be used to return the user to your cancel URL page.

It is the merchant's responsibility to write their code to allow their users to process another transaction or exit their site.

More about the Default Payment Page:

If your website calls the Payment Page without passing the following parameters, Background Color, Background Image URL, Header Text, Logo, Return to Payment Page on Decline, and Account Double Blind, the page will be displayed as follow:

- For UCLA merchants, the UCLA logo will appear in the upper left hand corner.
- For UCOP merchants, the UC seal will appear in the upper left hand corner.
- For UC Santa Cruz merchants, the UC Santa Cruz logo will appear in the upper left hand corner.

- The page will be displayed with a white background.
- The page will be displayed with no Header text.
- If you don't pass a font type, size, and color, the page will be displayed in **the user browser** default settings.
- The card number input will not be double blind.

Users will be returned to the Payment Page if:

- The card number or expiration date is invalid.
- The card type is not accepted by the merchant.
- The payment amount is invalid.

Return to Payment Page on Decline Parameter:

- If this parameter is set to "Y" and the transactions decline with an Authorization Response Code of "D" **only**, the user will be returned to the Payment Page.

When the user's **allotted time to process their transaction has expired**, the Payment Error Page will be displayed.

It is the merchant's responsibility to **test, and test again**, to make sure the Payment Page is displayed the way they want it to look.

Neither AIS nor CyberPay Support will give merchants any input on how to display the Payment Page to their users.

Code Samples

Below are two examples on how to access the PCI application for either a .NET or ASP application.

.NET Example:

Request Page:

```
Dim MerchantID As String = "12345..."
Dim OrderID As String = "1111-12345..."
Dim MaxAmount As decimal = 0
Dim FixedAmount As decimal = 0
Dim Description As String = "TESTING"
Dim NCRCode As String = "1111"
Dim ReturnURL As String = "HTTP://www..."
Dim CancelURL As String = "HTTP://www..."
Dim POSTURL As String = "HTTP://www..."
Dim FAU As String = "4-..."
Dim UserField1 As String = "userfield data"
Dim BackGroundColor As String = "Yellow"
Dim BackGroundImageURL As String = ""
Dim MerchantLogoURL As String = "HTTP://www..."
Dim Header As String = "<h4 align=center>UCLA Parent Weekend
    2006<br>January 23rd 2006<br>Online(Registration)</h4>"
Dim FontName As String = "Verdana"
Dim FontSize As String = "12 pt"
Dim FontColor As String = "Blue"
Dim AccountDoubleBlind As String = "Y"
Dim ReturnToPaymentPageOnDecline As String = "Y"

Dim StrPostData As String = ("MerchantID=" & MerchantID & "&OrderID=" &
OrderID & "&MaxAmount=" & MaxAmount & "&FixedAmount=" & FixedAmount &
"&Description=" & Description & "&NCRCode=" & NCRCode & "&ReturnURL=" &
ReturnURL & "&CancelURL=" & CancelURL & "&POSTURL=" & POSTURL & "&FAU="
& FAU & "&UserField1=" & UserField1 & "&BackGroundColor=" &
BackGroundColor & "&BackGroundImageURL=" & BackGroundImageURL &
"&MerchantLogoURL=" & MerchantLogoURL & "&Header=" & Header &
"&FontName=" & FontName & "&FontSize=" & FontSize & "&FontColor=" &
FontColor & "&AccountDoubleBlind=" & AccountDoubleBlind &
"&ReturnToPaymentPageOnDecline=" & ReturnToPaymentPageOnDecline)

Dim ObjRequest As HttpWebRequest =
CType(WebRequest.Create("http://cswt.ais.ucla.edu/PCIPaymentR2/Parameter.
aspx"), HttpWebRequest)
ObjRequest.Method = "POST"
ObjRequest.ContentType = "application/x-www-form-urlencoded"
ObjRequest.ContentLength = StrPostData.Length

Dim stOut As StreamWriter = New
StreamWriter(ObjRequest.GetRequestStream(), Encoding.ASCII)
stOut.Write(StrPostData)
stOut.Close()

Dim Status As String
Dim GUIDID As String 'Size can be up to 50 characters
Dim ErrorMessage As String
Try
    Dim ObjResponse As HttpWebResponse = ObjRequest.GetResponse()
```

```

Dim ResponseStream As New
    StreamReader(ObjResponse.GetResponseStream())
Status = ResponseStream.ReadLine()
If (Status = "GOOD") Then
    GUIDID = ResponseStream.ReadLine
Else
    ErrorMsg = ResponseStream.ReadLine
End If
ResponseStream.Close()
Catch Ex As Exception
    Status = "No Status"
    ErrorMsg = Ex.Message
End Try

If (Status = "GOOD") Then
    Dim StrReturnData As New System.Text.StringBuilder
StrReturnData.Append("http://cswt.ais.ucla.edu/PCIPaymentR2/LookUp.aspx")
    StrReturnData.Append("?")
    StrReturnData.Append("GUIDID=" & Server.UrlEncode(GUIDID))
    Response.Redirect(StrReturnData.ToString)
Else
    Check error message and do error handling
End If

```

Transaction Status Post Page:

Delete all the HTML from this page except for the page directives at the top.

```

Dim OrderID As string = Request.Form("OrderID")
Dim ReturnCode As string = Request.Form("ReturnCode")
Dim ReturnCodeMessage As string = Request.Form("ReturnCodeMessage")
Dim AuthResponseCode As string = Request.Form("AuthorizationResponseCode")
Dim AuthResponseMessage As string =
    Request.Form("AuthorizationResponseMessage")
Dim ApprovalCode As string = Request.Form("ApprovalCode")
Dim ZipMatch As string = Request.Form("ZipMatch")
Dim CVVResult As string = Request.Form("CVVResult")
Dim AmountApproved As string = Request.Form("AmountApproved")
Dim CardType As string = Request.Form("CardType")
Dim UserField1 As string = Request.Form("UserField1")

If (ReturnCode = "0") then
    If (AuthResponseCode = "A") then
        Transaction was successful
    else
        Transaction was declined
        Check AuthResponseCode
    End if
Else
    Check ReturnCodeMessage
End If

`Sending the response back to PCI.
Response.Clear
Response.Write("Y")
Response.Flush()

```

Transaction Response Page (Re-direct):

```
Dim OrderID As string = Request.QueryString("OrderID")
Dim ReturnCode As string = Request.QueryString("ReturnCode")
Dim ReturnCodeMessage As string =
Request.QueryString("ReturnCodeMessage")
Dim AuthResponseCode As string =
Request.QueryString("AuthorizationResponseCode")
Dim AuthResponseMessage As string =
Request.QueryString("AuthorizationResponseMessage")
Dim ApprovalCode As string = Request.QueryString("ApprovalCode")
Dim ZipMatch As string = Request.QueryString("ZipMatch")
Dim CVVResult As string = Request.QueryString("CVVResult")
Dim AmountApproved As string = Request.QueryString("AmountApproved")
Dim CardType As string = Request.QueryString("CardType")
Dim UserField1 As string = Request.QueryString("UserField1")

If (ReturnCode = "0") then
    If (AuthResponseCode = "A") then
        Transaction was successful
    else
        Transaction was declined
        Check AuthResponseCode
    End if
Else
    Check ReturnCodeMessage
End If
```

Click the link below to see an example page:

http://cswt.ais.ucla.edu/PCI_Merchant_Example_DOTNET_R2/MerchantRequest.aspx

ASP Example:

Request Page:

```
Dim MerchantID, OrderID, MaxAmount, FixedAmount, Description, NCRCode,
ReturnURL, CancelURL, POSTURL, FAU, UserField1, BackGroundColor,
BackGroundImageURL, MerchantLogoURL,
Header, FontName, FontSize, FontColor, AccountDoubleBlind,
ReturnToPaymentPageOnDecline, StrPostData
```

```
MerchantID = "12345..."
OrderID = "1111-12345..."
MaxAmount = 0
FixedAmount = 0
Description = "TESTING"
NCRCode = "1111"
ReturnURL = "HTTP://www..."
CancelURL = "HTTP://www..."
POSTURL = "HTTP://www..."
FAU = "4-..."
UserField1 = "userfield data"
BackGroundColor = "Yellow"
BackGroundImageURL = ""
MerchantLogoURL = "HTTP://www..."
Header = "<h4 align=center>UCLA Parent Weekend
2006<br>January 23rd 2006<br>Online(Registration)</h4>"
Rev. 7-25-2006 09:00 AM
```

```
FontName = "Verdana"  
FontSize = "12 pt"  
FontColor = "Blue"  
AccountDoubleBlind = "Y"  
ReturnToPaymentPageOnDecline = "Y"
```

```
StrPostData = ("MerchantID=" & MerchantID & "&OrderID=" & OrderID &  
"&MaxAmount=" & MaxAmount & "&FixedAmount=" & FixedAmount &  
"&Description=" & Description & "&NCRCode=" & NCRCode & "&ReturnURL=" &  
ReturnURL & "&CancelURL=" & CancelURL & "&POSTURL=" & POSTURL & "&FAU=" &  
FAU & "&UserField1=" & UserField1 & "&BackGroundColor=" & BackGroundColor  
& "&BackGroundImageURL=" & BackGroundImageURL & "&MerchantLogoURL=" &  
MerchantLogoURL & "&Header=" & Header & "&FontName=" & FontName &  
"&FontSize=" & FontSize & "&FontColor=" & FontColor &  
"&AccountDoubleBlind=" & AccountDoubleBlind &  
"&ReturnToPaymentPageOnDecline=" & ReturnToPaymentPageOnDecline)
```

```
set ObjRequest = Server.CreateObject("MsXml2.ServerXMLHTTP")  
ObjRequest.open "POST",  
"http://cswt.ais.ucla.edu/PCIPaymentR2/Parameter.aspx", false  
ObjRequest.setRequestHeader "Content-Type", "application/x-www-form-  
urlencoded"  
ObjRequest.Send StrPostData
```

```
Dim ObjResponse, Status, StrReturnData  
If (ObjRequest.Status <> 200) then  
    Problem communicating with the AIS server, do error handling  
Else  
    ObjResponse = Split(ObjRequest.responseText, vbCrLf)  
    Status = ObjResponse(0)  
    If (Status = "GOOD") Then  
        GUIDID = ObjResponse(1)  
        StrReturnData =  
            ("http://cswt.ais.ucla.edu/PCIPaymentR2/LookUp.aspx"  
            & "?" & "GUIDID=" & Server.UrlEncode(GUIDID))  
        Response.Redirect(StrReturnData)  
    Else  
        Check error message using ObjResponse(1) and do error  
        Handling  
    End If  
End If
```

Transaction Status Post Page:

```
Dim OrderID, ReturnCode, ReturnCodeMessage, AuthResponseCode,  
AuthResponseMessage, ApprovalCode, ZipMatch, CVVResult, AmountApproved,  
CardType, UserField1
```

```
OrderID = Request.Form("OrderID")  
ReturnCode = Request.Form("ReturnCode")  
ReturnCodeMessage = Request.Form("ReturnCodeMessage")  
AuthResponseCode = Request.Form("AuthorizationResponseCode")  
AuthResponseMessage = Request.Form("AuthorizationResponseMessage")  
ApprovalCode = Request.Form("ApprovalCode")  
ZipMatch = Request.Form("ZipMatch")  
CVVResult = Request.Form("CVVResult")  
AmountApproved = Request.Form("AmountApproved")  
CardType = Request.Form("CardType")
```

```
UserField1 = Request.Form("UserField1")
```

```
If (ReturnCode = "0") then  
    If (AuthResponseCode = "A") then  
        Transaction was successful  
    Else  
        Transaction was declined  
        Check AuthResponseCode  
    End if  
Else  
    Check ReturnCodeMessage  
End If
```

```
`Sending the response back to PCI.
```

```
Response.Clear  
Response.Write("Y")  
Response.Flush()
```

Transaction Response Page (Re-direct):

```
Dim OrderID, ReturnCode, ReturnCodeMessage, AuthResponseCode,  
AuthResponseMessage, ApprovalCode, ZipMatch, CVVResult,  
AmountApproved, CardType, UserField1
```

```
OrderID = Request.QueryString("OrderID")  
ReturnCode = Request.QueryString("ReturnCode")  
ReturnCodeMessage = Request.QueryString("ReturnCodeMessage")  
AuthResponseCode =  
    Request.QueryString("AuthorizationResponseCode")  
AuthResponseMessage = Request.QueryString("AuthorizationResponseMessage")  
ApprovalCode = Request.QueryString("ApprovalCode")  
ZipMatch = Request.QueryString("ZipMatch")  
CVVResult = Request.QueryString("CVVResult")  
AmountApproved = Request.QueryString("AmountApproved")  
CardType = Request.QueryString("CardType")  
UserField1 = Request.QueryString("UserField1")
```

```
If (ReturnCode = "0") then  
    If (AuthResponseCode = "A") then  
        Transaction was successful  
    Else  
        Transaction was declined  
        Check AuthResponseCode  
    End if  
Else  
    Check ReturnCodeMessage  
End If
```

Click the link below to see an example page:

http://cswt.ais.ucla.edu/PCI_Merchant_Example_ASP_R2/MerchantRequest.asp

Testing

How to Test

There are two steps involved in testing:

- 1) Test the process to ensure the merchant is properly accessing the application and sending the required data. Then, test to ensure they are receiving the proper response. Check to make sure the output functions correctly for reporting purposes. Check to make sure the look and feel is functioning as you want.
- 2) Complete a production end-to-end test that verifies the security setup and data to the bank. This step is required for NEW merchants only.

Contact the CyberPay Support Team at Cyberpay@finance.ucla.edu to arrange a time for your testing. We require this to ensure that someone can monitor the test and provide assistance interpreting the results.

Step One

Send a transaction from your web site to the PCI payment server, and receive a response.

For this test you will be accessing the PCI payment test server.

Please refer to the diagram in Exhibit A for all steps listed below.

1. In exhibit A step 3, set the PCI parameter page URL to:

<http://cswt.ais.ucla.edu/PCIPaymentR2/Parameter.aspx>

2. In exhibit A step 5, set the PCI lookup page URL to:

<http://cswt.ais.ucla.edu/PCIPaymentR2/Lookup.aspx>

3. Use the following test credit card numbers to test your transactions:

Visa =	4111111111111111
MasterCard =	5424000000000015
American Express =	370000000000002
Discover =	6011000000000004
JCB =	3530111333300000
PLD =	4031680000000004

Send any information you want to in your test cases, using the above credit card numbers. If you encounter UNEXPECTED errors, or unhandled exceptions, please forward a screen shot of the error, with the data you sent, and the date and time the error occurred to your Test Coordinator.

Authorization Approval testing is based on dollar amount. The following dollar amounts trigger the specified results from the test server.

Amount Value	AuthResponseCode Value	Description
0.00 to 100.00	A	Approval
100.01 to 200.00	C	Call
200.01 to 300.00	D	Decline
300.01 to 400.00	P	Pick up card

400.01 to 500.00	X	Expired card
500.01 to 600.00	E	Error
600.01 and up		Generates a random approval response

Step Two – **This step is ONLY required for NEW merchants.**

Sending a transaction from your web site, posting to the bank and capturing data for the general ledger.

This is the final test. For this test you will be accessing the PCI payment production server in production mode. This is the final end-to-end test to verify that everything is in place.

You will now be receiving 'real' Authorization Response Code, Authorization Response Message and Approval code.

Please refer to the diagram in Exhibit A for all steps listed below.

1. In exhibit A step 3, set the PCI parameter page URL to:

<https://cswv.ais.ucla.edu/PCIPaymentR2/Parameter.aspx>

2. In exhibit A step 5, set the PCI lookup page URL to:

<https://cswv.ais.ucla.edu/PCIPaymentR2/Lookup.aspx>

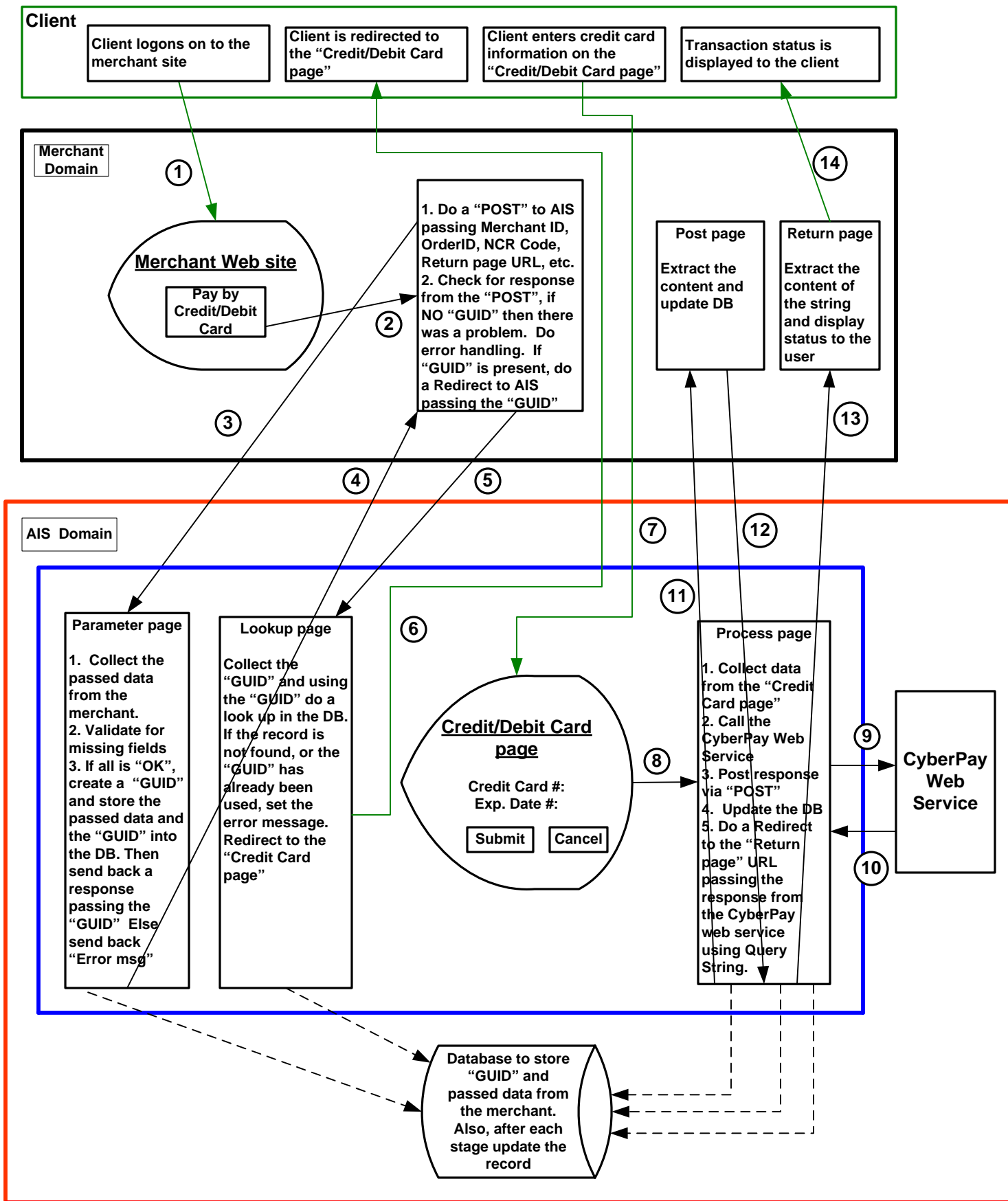
Create and send a transaction in the amount of \$1.00 using a **real credit** card (the CyberPay Support Team will reverse the charge as soon as the test is complete.)

AIS will notify you and the CyberPay Support Team when the transaction successfully reaches the CyberPay database. You are done for the day. The CyberPay Support Team will notify you of the final result within two business days. When the test is successful the CyberPay Support Team will coordinate a time to move you permanently into production.

The PCI application is available to accept credit card transactions 24-hours a day, 7-days a week. If you have any problems (system is not available, unusual return code, etc.) please contact the **AIS Help Desk at (310) 206-6951.**

For any questions about procedures please contact the **CyberPay Support Team:**
[Cyberpay@finance.ucla.edu.](mailto:Cyberpay@finance.ucla.edu)






CyberPay PCI Flow Diagram



PCI Online Payment Screen – Exhibit B

Each merchant web application will be calling the new centrally controlled online payment screen.

Test Payment Site
UCLA Parent Weekend 2006
January 23rd 2006
Online(Registration)

We accept:     

Type of Card: U.S. Credit/Debit Card
 Foreign Credit/Debit Card

Cardholder's Name: First:
Last:

Credit/Debit Card Number:

Expiration Date: (MMYY)

Security Code: [What is this?](#)

Billing Zip Code:

Amount of Payment: \$

The new PCI-compliant front-end includes the following features:

1. Logos of the acceptable credit/debit cards are displayed for each merchant.
2. The US Debit/Credit card button is the default setting. Customers using a foreign card click the Foreign Debit/Credit card instead. This automatically grays out the zip code box.
3. Both first and last names must be entered.
4. The card number must be entered with the option of double-blind entry for verification and added cardholder protection.
5. Zip Code must be entered for US cards only. The field is grayed out for foreign cards.
6. Card expiration date and security code must be entered for all cards. The “*What is this*” question is linked to an explanation security codes.
7. An amount must be entered. It may be entered in whole dollars or in dollars and cents.
 - a. This can be filled with a pre-determined fixed amount. The customer and the merchant must both know the final price. Sending this amount forces the customer to pay the amount they have agreed to on the merchant’s site.
 - b. Some merchants may desire a maximum amount due which the customer can’t exceed when entering the amount.
8. The “*Submit*” button activates the CyberPay web service card validation.
9. The “*Cancel*” button terminates the transaction and returns the customer to the merchant application screen.
10. Error messages are displayed in red next to the field with the error.