



CyberSource Silent Order POST

User's Guide

March 2008

CyberSource Contact Information

For technical support questions, go to the Home page in the Business Center to see the contact information appropriate for your account.

To manage your online payment transactions, visit the Business Center at <https://businesscenter.cybersource.com>.

For general information about our company, products, and services, go to <http://www.cybersource.com>.

For sales questions about any CyberSource Service, email sales@cybersource.com, or call 650-965-6000 or 888-330-2300 (toll-free in the United States).

Copyright

© 2008 CyberSource Corporation. All rights reserved. CyberSource Corporation ("CyberSource") furnishes this document and the software described in this document under the applicable agreement between the reader of this document ("You") and CyberSource ("Agreement"). You may use this document and/or software only in accordance with the terms of the Agreement. Except as expressly set forth in the Agreement, the information contained in this document is subject to change without notice and therefore should not be interpreted in any way as a guarantee or warranty by CyberSource. CyberSource assumes no responsibility or liability for any errors that may appear in this document. The copyrighted software that accompanies this document is licensed to You for use only in strict accordance with the Agreement. You should read the Agreement carefully before using the software. Except as permitted by the Agreement, You may not reproduce any part of this document, store this document in a retrieval system, or transmit this document, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written consent of CyberSource.

Restricted Rights Legends

For Government or defense agencies. Use, duplication, or disclosure by the Government or defense agencies is subject to restrictions as set forth the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 and in similar clauses in the FAR and NASA FAR Supplement.

For civilian agencies. Use, reproduction, or disclosure is subject to restrictions set forth in subparagraphs (a) through (d) of the Commercial Computer Software Restricted Rights clause at 52.227-19 and the limitations set forth in CyberSource Corporation's standard commercial agreement for this software. Unpublished rights reserved under the copyright laws of the United States.

Trademarks

CyberSource, the CyberSource logo, SmartCert, and PaylinX are registered trademarks of CyberSource Corporation in the U.S. and other countries. The Power of Payment, CyberSource Payment Manager, CyberSource Risk Manager, CyberSource Decision Manager, and CyberSource Connect are trademarks and/or service marks of CyberSource Corporation. All other brands and product names are trademarks or registered trademarks of their respective owners.

Contents

Documentation Changes and Enhancements	vii
Chapter 1	
Introduction.....	1
Introduction to the Silent Order POST.....	1
Processing an Order.....	2
Using this Guide.....	3
Additional Resources.....	3
Introduction to Payer Authentication Services.....	4
Preparing to Use Payer Authentication	4
Using and Storing Payer Authentication Data	4
API Results.....	4
Standard Transaction and Payer Authentication Searches.....	7
Payer Authentication Reports.....	7
Storing Payer Authentication Data	7
Customer Experience with Payer Authentication.....	8
Successful Order with Payer Authentication.....	8
Unsuccessful Order with Payer Authentication.....	9
Chapter 2	
Configuring the Business Center Settings	11
Choosing Your Default Settings.....	11
Payer Authentication.....	11
Reply Pages	12
Receipt Page.....	12
Decline Page.....	14
Notification Fields.....	15
Transaction Receipts.....	15
Specific Settings.....	15
Downloading Security Scripts.....	17
Chapter 3	
Creating and Customizing Checkout Pages.....	21
Gathering the Information for the Checkout Page.....	21
Product Information	21

URL for Sending Your Orders	22
Credit Card Information	22
Maestro (UK Domestic) and Solo Cards.....	22
MasterCard and Diners Club Alliance	22
Card Verification Number.....	23
Electronic Checks Information.....	23
Custom Fields.....	24
Creating Checkout Pages	25
Customizing the Silent Order POST.....	26
Order Information Fields.....	26
Customer Information Fields	28
Payment Information Fields.....	30
Order Confirmation Fields	32
Custom Fields.....	34

Chapter 4

Interpreting and Customizing Receipt Pages.....	35
Parsing and Interpreting the Reply	35
Decisions	36
Reason Codes.....	36
Missing or Invalid Fields	36
Tracking Orders	37
Order Number.....	37
Request ID.....	37
Request Token.....	37
Determining the Receipt Information.....	38
Creating Receipt Pages.....	38
Customizing the Receipt Page.....	39
Order Result Fields	39
Reason Codes.....	43

Chapter 5

Testing the Silent Order POST	47
Using the Test Version of the Silent Order POST.....	47
Using the Debug Page	48

Appendix A

Creating Web Pages With PHP.....	51
Checkout Page	51
Receipt Page.....	53
Verifying the Authenticity of an Order	53
Sample Script.....	54

Appendix B

Creating Web Pages With ASP.....	55
Checkout Page	55
Receipt Page.....	57

Verifying the Authenticity of an Order	57
Sample Script	58
Appendix C	
Creating Web Pages With Perl.....	59
Checkout Page	59
Creating the Page	59
Uploading a Web Page	61
Receipt Page	62
Verifying the Authenticity of an Order	62
Sample Script	63
Appendix D	
Creating Web Pages With JSP.....	65
Checkout Page	65
Receipt Page	67
Verifying the Authenticity of an Order	67
Sample Script	68
Appendix E	
Creating Web Pages with ColdFusion	69
Checkout Page	69
Receipt Page	71
Verifying the Authenticity of an Order	71
Sample Script	71
Appendix F	
Creating Web Pages with ASP.NET	73
Checkout Page	73
Receipt Page	74
Appendix G	
Sample Reply Data and Receipts	75
Reply Data	75
Successful Order.....	75
Order to Review	77
Request	77
Reply	78
Failed Order	78
Transaction Receipts	79
Customer's Receipt	79
Merchant's Receipt.....	79
Appendix H	
Description of Return Codes.....	81
AVS Codes.....	81
Card Verification Codes	83

Smart Authorization Factor Codes	84
Index	85

Documentation Changes and Enhancements

The following table lists changes made in the last releases of this document:

Release	Changes
March 2008	<ul style="list-style-type: none">• Added ColdFusion and ASP.NET (C# and Visual Basic) to the types of scripting languages that you can use to create Web pages: “Creating Web Pages with ColdFusion” on page 69 and “Creating Web Pages with ASP.NET” on page 73.• Redesigned the guide as follows:<ul style="list-style-type: none">• Combined the introduction to the Hosted Order Page and the Payer Authentication services into one chapter: “Introduction” on page 1.• Combined the chapters on creating and customizing the order form: “Creating and Customizing Checkout Pages” on page 21.• Moved the sections on scripting languages to the appendices (one for each language): “Creating Web Pages With PHP” on page 51, “Creating Web Pages With ASP” on page 55, “Creating Web Pages With Perl” on page 59, “Creating Web Pages With JSP” on page 65, “Creating Web Pages with ColdFusion” on page 69, and “Creating Web Pages with ASP.NET” on page 73.• Removed the section on deprecated signatures.• Changed how orderPage_transactionType is used: Because the type of transaction is included in the signature, this field is no longer required. For more information, see “orderPage_transactionType” on page 27 (checkout page) and “orderPage_transactionType” on page 42 (receipt page).
February 2008	<ul style="list-style-type: none">• Combined the two appendices of examples into one and eliminated redundant examples. For more information, see “Sample Reply Data and Receipts” on page 75.• Moved the section on deprecated signatures to “Deprecated Signatures” on page 87.
January 2008	<ul style="list-style-type: none">• Corrected the JSP receipt page example. See “Receipt Page” on page 67
November 2007	<ul style="list-style-type: none">• Corrected an error in the JSP receipt page example: the verifySignature procedure should not contain a semicolon (;). See “Receipt Page” on page 67.
September 2007	<ul style="list-style-type: none">• In the Business Center, added Payer Authentication data to the transaction details page and the ability to store and retrieve Payer Authentication data for up to 12 months after the transaction. For more information, see “Using and Storing Payer Authentication Data” on page 4.

Release	Changes
----------------	----------------

- | | |
|-------------|--|
| August 2007 | <ul style="list-style-type: none">• Added a transaction signature that you can use to verify the content of every field in the order, which ensures that the order is not fraudulent. For more information, see the appendix appropriate for your script.• Corrected an error in “Missing or Invalid Fields” on page 36: InvalidField_0...N and MissingField_0...N should be InvalidField0...N and MissingField0...N (no underscore). |
|-------------|--|
-

Chapter 1

Introduction

This guide describes how to use the Silent Order POST, a customized implementation of the Hosted Order Page. With this method, you do not use CyberSource's Hosted Order Page. Instead, you use the Hosted Order Page as an HTTP POST API.

To use the Silent Order POST, you must be able to create Web pages that will gather customer and order information for the services that you want to use into requests, and you must be able to process the reply information to fulfill the customer's order. You must meet these requirements:

- Your Web site has shopping-cart software.
- You have programming skills in ASP, PHP, Perl, JSP, ColdFusion, or ASP.NET.
- Your ISP supports these languages and SSL security.

With the Silent Order POST, you can use Simple Order API fields without having to install a client application. However, if you would like more information about the Simple Order API, which uses a client, see the [Business Center Simple Order API User's Guide](#).

This chapter provides an introduction to the Silent Order POST and to Payer Authentication services. With Payer Authentication services, you can quickly and easily add support for Verified by Visa and MasterCard SecureCode to your Web store without running additional software on your own server. For more information, see "[Introduction to Payer Authentication Services](#)" on page 4.

Introduction to the Silent Order POST

The Silent Order POST is a convenient way to customize part or all of your Web site to accept credit cards and check payments. After your customers place orders through the Hosted Order Page, you can view the order information in the Business Center, a central location for managing your online payment transactions. If you want to process checks in addition to credit cards, see *Processing Electronic Checks* in the [Business Center User's Guide](#).

To implement the Silent Order POST, you need to make changes in the Business Center and to your Web site. When done with your changes, test your implementation to make sure that it works completely before you start accepting your customers' orders.

This chapter comprises these sections:

Processing an Order

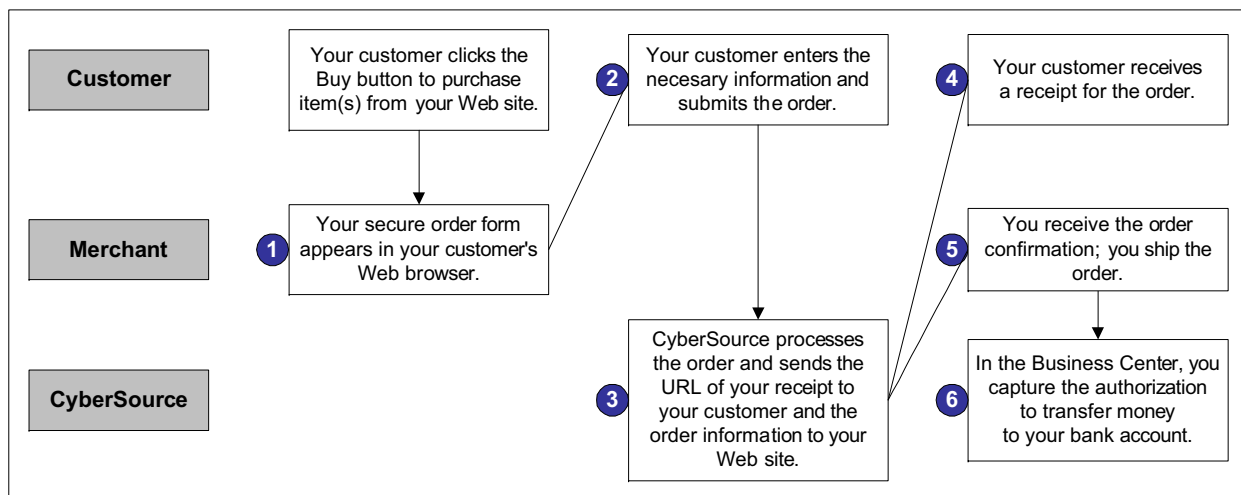
[Using this Guide](#)

[Additional Resources](#)

Processing an Order

The figure below shows how the components of the Silent Order POST function: your order is processed by CyberSource, which sends the appropriate reply to your customer and sends email notifications to you and to your customer if you chose to do so.

This figure shows what occurs when a customer places an order on your Web site. This sample order describes a credit card authorization.



When your customer clicks the **Buy** button, these events occur:

- 1 Your secure order form appears in the customer's browser.
- 2 The customer completes and submits the form.
- 3 CyberSource processes the authorization by verifying that the information is complete and valid and by processing any other optional services, such as credit card capture. CyberSource sends you the order information and notifies you and your customer by email if you have configured the Business Center settings accordingly.
- 4 CyberSource directs the customer's browser to your receipt or decline page. Optionally the customer receives email notification of the successful transaction if you configured this option.
- 5 A script on your Web site receives and converts the HTTP POST reply data. You store the order information in your database. You ship the order.
- 6 After you ship the order, you go to the Business Center to capture the authorization to have money transferred to your bank account. You receive payment within two to four days. If you do not capture the authorization, you do not receive payment.

Using this Guide

This guide contains these chapters and appendices.

Reference Chapters. The chapters discuss concepts and contain all necessary information for configuration.

Chapter 2	Configuring the Business Center Settings	How to configure the required elements of the Silent Order POST
Chapter 3	Creating and Customizing Checkout Pages	How to design and create your checkout pages
Chapter 4	Interpreting and Customizing Receipt Pages	How to configure a receipt for your customers
Chapter 5	Testing the Silent Order POST	How to test the Silent Order POST before accepting your customers' orders

Appendices. The appendices provide information to assist you in creating Web pages and interpreting the results of the replies:

Appendix G	Creating Web Pages With PHP	How to create order and receipt pages with PHP.
Appendix H	Creating Web Pages With ASP	How to create order and receipt pages with ASP.
Appendix C	Creating Web Pages With Perl	How to create order and receipt pages with Perl.
Appendix D	Creating Web Pages With JSP	How to create order and receipt pages with JSP.
Appendix E	Creating Web Pages with ColdFusion	How to create order and receipt pages with ColdFusion.
Appendix F	Creating Web Pages with ASP.NET	How to create order and receipt pages with ASP.NET (C# or Visual Basic).
Appendix G	Sample Reply Data and Receipts	How to interpret all types of replies
Appendix H	Description of Return Codes	Codes that you may see in the reply and in the Business Center reports

Additional Resources

You can obtain more information about processing orders in these documents:

- [Business Center User's Guide](#)— This guide describes the features of the Business Center, such as configuring your settings and searching for orders. We recommend that you read this guide before you start using the Silent Order POST.
- [Business Center Reporting User's Guide](#)—This guide describes the reports available in the Business Center.
- In the Business Center, the online help provides detailed instructions for performing all tasks.

Introduction to Payer Authentication Services

CyberSource Payer Authentication enables you to quickly and easily add support for Verified by Visa and MasterCard SecureCode to your Web store without running additional software on your own server. The Verified by Visa and MasterCard SecureCode cardholder authentication services deter unauthorized card use. In addition, participating merchants receive added protection from fraudulent chargeback activity. For more information, see the [Payer Authentication Implementation Guide](#).

This section comprises these sections:

- Preparing to Use Payer Authentication
- Using and Storing Payer Authentication Data
- [Customer Experience with Payer Authentication](#)

Preparing to Use Payer Authentication

Before using CyberSource Payer Authentication, you must contact Customer Support and provide information about your company and your acquiring bank so that CyberSource may configure your account to implement this service.

After your account is configured, check the option at the bottom of the Hosted Order Page settings page. For more information, see the "[Payer Authentication](#)" on page 11.

Using and Storing Payer Authentication Data

For each transaction with the Payer Authentication services, you receive detailed information in the replies and in the transaction details page of the Business Center. You can store this data for up to 12 months so that you can retrieve it later if necessary.

- API Results
- [Standard Transaction and Payer Authentication Searches](#)
- [Payer Authentication Reports](#)
- [Storing Payer Authentication Data](#)

API Results

With each transaction processed with Payer Authentication, you receive specific results that you need to analyze to ensure that only safe and valid orders are placed by your customers.

Result Data. When using Payer Authentication with the Hosted Order Page, you receive result data in all the replies that you have selected:

- Merchant POST URL and email: reply fields only
- Merchant email receipt: same email as the customer but with specific additional merchant data

For sample result data, see “[Sample Reply Data and Receipts](#)” on page 75.

Reply Fields. The Hosted Order Page processes Payer Authentication slightly differently from the API described in the implementation guide mentioned above:

- With the API implementation, the enrollment check, validation, and authorization are usually requested separately so that the results are received (and shown in that guide) in separate replies.
- With the Hosted Order Page, the enrollment check and authentication validation services are requested for you together when you request an authorization (or sale) so that your transaction results show all services. Therefore, the sample transaction results that you see in this guide are slightly different from those in the implementation guide.

In the Hosted Order Page reply data, you can see these Payer Authentication fields, depending on the card type:

Card Type	Payer Authentication Services	
	Enrollment	Validation
Visa	eci, proofXML, and xid	cavv, eci, xid
MasterCard	proofXML and xid	ucafAuthenticationData, ucafCollectionIndicator, and xid

This table describes the fields mentioned above. For detailed information about all the fields and how to interpret the results, see the [Payer Authentication Implementation Guide](#).

Table 1 Reply Fields

Field Name	Description	Used By	Data Type and Length
cavv	Transaction identifier generated by the issuing bank for Verified by Visa transactions.	validation	String (50)
eci	Numeric electronic commerce indicator (ECI) returned only for Verified by Visa transactions. This field applies only to non U.S.-issued cards. You receive this result when enrollment is checked, and the card is not enrolled. This field contains one of these values: <ul style="list-style-type: none"> • 06: The card can be enrolled. You are protected. • 07: The card cannot be enrolled. You are not protected. 	enrollment	String (3)

Table 1 Reply Fields (Continued)

Field Name	Description	Used By	Data Type and Length
	<p>This field contains one of these values:</p> <ul style="list-style-type: none"> • 00: Failed authentication. • 05: Successful authentication • 06: Authentication attempted • 07: Failed authentication (No response from the merchant because of a problem.) 	validation	
proofxml	<p>Data that includes the date and time of the enrollment check and the <code>VEReq</code> and <code>VERes</code> elements. Store the complete proof whenever it is returned. If you ever need to show proof of enrollment checking, you will need to parse the string for the information required by the card association.</p> <ul style="list-style-type: none"> • For cards not issued in the U.S. or Canada, your bank may require this data as proof of enrollment checking for any payer authentication transaction that you re-present because of a chargeback. • For cards issued in the U.S. or Canada, Visa may require this data for specific merchant category codes. 	enrollment	String (1024)
ucafAuthenticationData	<p>MasterCard SecureCode UCAF authentication data. Returned only for MasterCard SecureCode transactions.</p>	validation	String (32)
ucafCollectionIndicator	<p>Returned only for MasterCard SecureCode transactions to indicate whether the authentication data is collected at your Web site. This field contain one of these values:</p> <ul style="list-style-type: none"> • 0: Authentication data not collected and customer authentication was not completed. • 1: Authentication data not collected because customer authentication was not completed. • 2: Authentication data collected because customer completed authentication. 	validation	Non-negative integer (1)

Table 1 Reply Fields (Continued)

Field Name	Description	Used By	Data Type and Length
xid	Transaction identifier generated by CyberSource Payer Authentication. Used to match an outgoing PAREq with an incoming PAREs.		String (28)
		enrollment	
	CyberSource forwards the XID with the credit card authorization service only if the payment processor is Barclays.	validation	

Standard Transaction and Payer Authentication Searches

You can find in the Business Center all the Payer Authentication data that you receive in your reply either with the standard transaction search or with the specific Payer Authentication search.

With the second option, you can easily search for transactions processed with the Payer Authentication and card authorization services. You use the XID as search parameter, and the list of applications is simplified to facilitate searching for the relevant service requests. The table of results include the XID and the customer's account number (PAN). You can use the XID to find all parts of a transaction. The transaction details include all Payer Authentication data in the proof XML, VEReq, VERes, PAREq, and PAREs.

Payer Authentication Reports

From the Payer Authentication details page, you can export the data to the Payer Authentication Detail report, which is in XML format. This report is available for 12 months. Each report contains a single transaction.

The previous Payer Authentication report remains as a summary report. The summary report is available for six months.

Storing Payer Authentication Data

Card associations allow a certain number of days between the Payer Authentication (enrollment check and/or customer authentication) and the authorization requests. If you settle transactions older than the pre-determined number of days, card associations may require that you send them the AAV or CAVV and/or the XID. The specific requirements depend on the card type and the region. For more information, see your agreement with your card association. After your transactions are settled, you can also use this data to update the statistics of your business.

CyberSource recommends that you store the Payer Authentication because you may be required to show this data as proof of enrollment checking for any payer authentication transaction that you re-present because of a chargeback. Because your account provider

may require that you provide all data in human-readable format, make sure that you can decode the PAREq and PAREs.

Visa and MasterCard have implemented the 3D Secure protocol differently throughout the world. The following paragraphs provide general information about the Visa and MasterCard implementations throughout the world. To find out the specific requirements for your business and for more information on decrypting and providing the PAREs, contact your account manager.

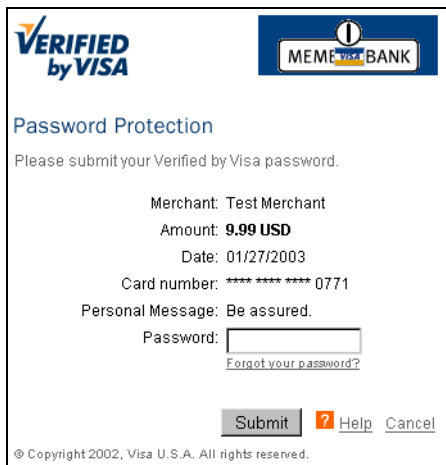
Customer Experience with Payer Authentication

This section describes what your customer sees when placing a successful and an unsuccessful order.

Successful Order with Payer Authentication

This example shows what happens if the customer is enrolled in Verified by Visa. A similar process occurs with MasterCard SecureCode.

- 1 When the customer clicks **Buy Now**, the order page appears.
- 2 The customer completes the order page and clicks **Buy**. The Verified by Visa page appears and requests a password for the credit card.



The screenshot shows a web page for 'Verified by Visa' with the 'MEMPHIS BANK' logo. The page title is 'Password Protection' and it asks the user to submit their Verified by Visa password. The page displays the following information: Merchant: Test Merchant, Amount: 9.99 USD, Date: 01/27/2003, Card number: **** * 0771, and Personal Message: Be assured. There is a password input field with a 'Forgot your password?' link below it. At the bottom, there are 'Submit', 'Help', and 'Cancel' buttons. A copyright notice at the bottom left reads: © Copyright 2002, Visa U.S.A. All rights reserved.

- 3 The customer enters the password and clicks **Submit**.
- 4 The browser displays a receipt for the order.

The following table describes your purchase.

Purchase Description	
Toy Giraffe	
Payment Details	
Order Number	1046134306294
Subtotal	9.99
Tax	0.00
Total	9.99

We sincerely appreciate your business.

As with other orders, the customer receives an email receipt for the order.

Unsuccessful Order with Payer Authentication

In this example, a fraudulent customer has stolen a credit card and is trying to use it on your Web site. The example begins after Nancy has clicked the Buy Now button and opened the order page.

- 1 The fraudulent customer completes the order page and clicks **Buy**. The Verified by Visa page appears and requests a password for the credit card.
- 2 The fraudulent customer enters the wrong password and clicks **Submit**. An error message tells the fraudulent customer that the password is incorrect and requests a password again.

VERIFIED
by VISA

MEME BANK

Password Protection

The password you submitted is incorrect. Please check that this is the right password for this Visa card and re-submit.

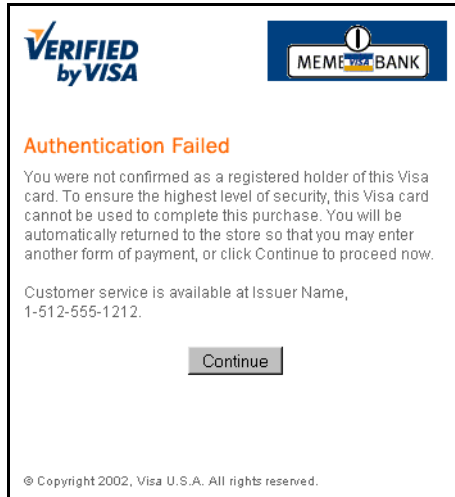
Merchant: Test Merchant
Amount: **9.99 USD**
Date: 01/27/2003
Card number: **** * 0771
Personal Message: Be assured.

*Password:
[Forgot your password?](#)

Submit [?](#) Help Cancel

© Copyright 2002, Visa U.S.A. All rights reserved.

- 3 The fraudulent customer keeps trying but cannot guess the real password. After several tries, a final error message says that authentication failed.



- 4 The fraudulent customer clicks **Continue**.
The Web browser displays a message saying that the order failed.

Chapter 2

Configuring the Business Center Settings

This chapter provides detailed instructions to configure the default settings for the Silent Order POST and downloading the security script:

Choosing Your Default Settings

[Downloading Security Scripts](#)

Choosing Your Default Settings

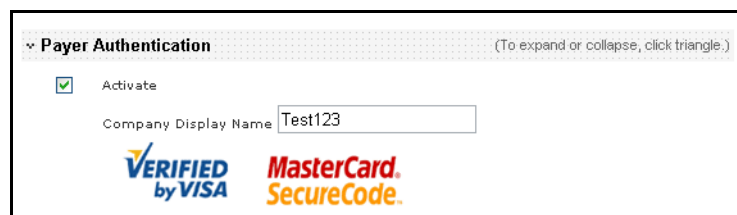
This section is required for all users because the Hosted Order Page does not work unless you have chosen and saved your default settings. Afterward, you can return to the settings page at any time to make changes if you wish.

Important To choose or modify the default settings, you must be an administrator user.

You must first choose the default settings for the Silent Order POST. After logging into the Business Center with your merchant user name and password, browse to **Tools & Settings > Hosted Order Page > Settings**. Configure the reply pages and the notification fields as described below, and scroll to the bottom of the page to click **Update**.

Payer Authentication

This option appears only if you are configured to use CyberSource Payer Authentication.



The screenshot shows a settings panel for Payer Authentication. At the top, there is a header "Payer Authentication" with a small triangle icon and the text "(To expand or collapse, click triangle.)". Below the header, there is a checked checkbox labeled "Activate". Underneath, there is a text input field labeled "Company Display Name" with the value "Test123". At the bottom of the panel, there are two logos: "VERIFIED by VISA" and "MasterCard SecureCode".

CyberSource Payer Authentication enables you to quickly and easily add support to your Web store for the Verified by VisaSM and MasterCard[®] SecureCode[™] programs without

running additional software on your own server. These cardholder authentication services deter unauthorized card use by fraudulent customers and allows you to receive added protection from fraudulent chargeback activity. For more information, see "[Introduction to Payer Authentication Services](#)" on page 4.

Important This note applies to merchants who have the Essentials or Essentials Plus package. If you check the box while you are in test mode, the Go Live button is grayed out (inactive) for about five days while your configuration is being updated.

Activation Option. Even if you are configured for Payer Authentication, you need to check the box to use the service.

Company Display Name. You can see and use this field only if you are configured to use Payer Authentication. Enter the business name to appear on the order page. To customize this field, see "[orderPage_merchantDisplayName](#)" on page 27.

Reply Pages

In this section, you choose the reply pages that CyberSource sends to your customers.

If you create your own receipt pages, CyberSource recommends that you enter the same URL for the success and failure pages so that all replies coming from CyberSource go to a single location on your Web site. You can later parse the reply results from that URL according to the reason code and redirect your customer to the appropriate receipt page.

Receipt Page

In this section, you enter either the URL for *your* custom receipt page or the text for *CyberSource's* receipt page.

Receipt Response URL. This field is required when you check the box below. If you check this box, you indicate that you want to use your own receipt page, not CyberSource's receipt page. If you use your own receipt page, you do not need to enter any data in the Receipt Link Text Header and Footer fields of this section. Enter a URL for your receipt page in the format: `http://www.example.com/`.

Receipt Link Text. If you plan to use CyberSource's receipt page, enter text for the link to be displayed at the bottom of the receipt page, for example: `Thank you for your order or Return to Example.com.`

Important Your customers receive this page for successful orders *and* for orders declined by Smart Authorization. You should review these declined orders as soon as possible because you may be able to correct problems related to address or card verification, or you may be able to obtain a verbal authorization. For complete information about Smart Authorization and reviewing orders, see the [Business Center User's Guide](#).

This text links the user to a receipt page on your Web site that lists the following contact information: name, phone number, and email address. Customers receive this contact

information in the email confirmation of their order if you have configured appropriately the “[Notification Fields](#)” on page 15.

- Header: Optional text to be displayed at the top of the receipt page. You can include HTML code, for example: ``
- Footer: Optional text to be displayed at the bottom of the receipt page. You can include HTML code, for example: `<p>Thank you for your order.</p>`

In addition, you can add specific order information to the header and/or the footer with the smart tags:

Account Number*	Billing Street 1	Shipping City
Amount	Billing Street 2	Shipping Company
Billing City	Comments	Shipping Country
Billing Company	Customer ID	Shipping First Name
Billing Country	Logo	Shipping Last Name
Billing First Name	Merchant-Defined Data 1	Shipping Postal Code
Billing Email Address	Merchant-Defined Data 2	Shipping State
Billing Last Name	Merchant-Defined Data 3	Shipping Street 1
Billing Phone Number	Merchant-Defined Data 4	Shipping Street 2
Billing Postal Code	Order Number	Tax Amount
Billing State		

* Account Number refers to the customer's payment account number with only the last four digits shown.

You can one or more element of the order listed in the table above to the header and/or the footer of the order form. When you select an option in the list, the item will be inserted in the field *where the cursor is located if the data is present in the order*. If no data exists for the tag that you choose, nothing will appear.

The screenshot shows a configuration window titled "Receipt Page" with a collapse/expand triangle. It contains the following fields:

- Receipt Response URL:** A text input field.
- Receipt Link Text:** A text input field with a checkbox below it labeled "This URL is a custom receipt page."
- Header:** A text area containing the HTML code: `##LOGO##
Receipt
##ORDER_NU
MBER##`. Below the text area is a "Smart Tags" dropdown menu with "Order Number" selected and an "Insert" button.
- Footer:** A text area containing the HTML code: `Thank you!
##ORDER_NUMBER##`. Below the text area is a "Smart Tags" dropdown menu with "Order Number" selected and an "Insert" button.

In the [Smart Tags](#) figure above, the header for the default (CyberSource's) receipt page will contain the logo and the order number. The footer will contain the text *Thank you!* and the order number.

Decline Page

In this section, you can enter either the URL for *your* custom decline page or the text for *CyberSource's* decline page.

The screenshot shows a configuration form titled "Decline Page" with a collapse/expand triangle and the instruction "(To expand or collapse, click triangle.)". The form contains the following fields and controls:

- Decline Response URL:** A text input field containing "http://www.example.com/decline.jsp".
- This URL is a custom decline page.
- Decline Link Text:** An empty text input field.
- Header:** A large text area for the page header.
- Smart Tags:** A dropdown menu set to "Comments" and an "Insert" button.
- Footer:** A large text area for the page footer.
- Smart Tags:** A dropdown menu set to "Comments" and an "Insert" button.

Decline Response URL. This field is required when you check the box immediately below. Checking the box indicates that you want to use your own decline page instead of CyberSource's decline page. If you use your own decline page, do not enter data in the Decline Link Text Header and Footer fields. Enter a URL for your decline page, such as <http://www.example.com/>. The figure above shows a custom decline page: the URL is for the page that the customer will see.

Decline Link Text. If you plan to use CyberSource's decline page, enter text for the link to be displayed at the bottom of the receipt page, for example: *Questions about your order?, Try another payment type or card.*

Important Your customers receive this page for all rejected orders. You should review these declined orders as soon as possible because you may be able to correct some of the problems.

This text links the user to a decline page on your Web site that lists the following contact information: name, phone number, and email address. Customers receive this contact information in the email confirmation of their order if you have configured appropriately the "[Notification Fields](#)" on page 15. In addition, you can use the same smart tags available for the receipt page.

Depending on the reason for the decline, if the customer returns to the previous page, you can display an error message such as this one: To resubmit your order, return to the previous page, and click the appropriate link.

- **Header:** Optional text to be displayed at the top of the receipt page. You can include HTML code, for example: ``
- **Footer:** Optional text to be displayed at the bottom of the receipt page. You can include HTML code, for example: `<p>We are sorry for the problem with your order.</p>`

Notification Fields

CyberSource can send to you and to your customer a receipt for each new transaction.

Transaction Receipts

These email receipts are formatted as memos. You cannot specify the content, but you can specify how you want the header and the footer to appear. For sample receipts, see [“Transaction Receipts”](#) on page 79.

Customer’s Receipt. The customer’s receipt shows the payment information with only the last four digits of the credit card number. CyberSource sends a receipt only when customers make a payment.

Merchant’s Receipt. CyberSource sends a receipt for each type of transactions, and you can specify the type of message(s) that you want to receive:

- **POST:** string of the transaction results sent in two ways, which are optional. In both cases, the content is the same as what you receive in the API reply:
 - URL to a URL that you specify.
 - POST to an email address that you specify.
- **Email:** standard email receipt as in the example below, which contains the payment information, the return codes, and all the information relevant to the order. With this receipt, you can fulfill the order without having to search the Business Center to see the details.

Specific Settings

These fields are optional.

The default email message contains only the description of the product, the cost, and the order number, but you can customize the messages by adding a header and/or a footer and your customized body if you wish. To customize this section, see [“Order Confirmation Fields”](#) on page 32. To use a custom email body, you need to contact Customer Support.

The figure below shows the Email section of the settings page.

Addresses and Options (To expand or collapse, click triangle.)

Merchant POST URL

Merchant POST Email

Customer Receipt Email

Merchant Receipt Email

Sender's Email Address

Sender's Name

Email Receipt Header

Smart Tags

Email Receipt Footer

Smart Tags

Merchant POST URL. Enter the URL where you want to send the transaction results from CyberSource to your business.

Email Messages. Enter the appropriate information for the email messages that you want to use.

Table 2 Types of Email Messages

Type	Description
Merchant POST Email	Email address for sending the transaction results from CyberSource to your business, for example: <code>orders@example.com</code> .
Customer Receipt Email	Email receipt sent to the customer. You also need to complete the four fields below.
Merchant Receipt Email	Email receipt sent to your business email address. You also need to complete the four fields below.

Default Email Information. These four fields apply to all merchant and customer email messages.

Table 3 Email Information

Type	Description
Sender's Email Address	Email address to appear on the email messages, for example: <code>service@example.com</code> . This field is required if you checked any or all of the email message boxes above. Important Because some customers will reply to the email receipt to ask questions about their orders, make sure to use a real email address and to check it regularly.
Sender's Name	Name to use on the email messages, for example: Widgets Incorporated.
Email Receipt Header	Text to use on the email messages, for example: Widgets Incorporated. In addition, you can use for the receipt headers and footers the same smart tags available for the order page, except for the logo to the receipt, which you cannot add to a receipt. For more information, see Smart Tags on page 13.
Email Receipt Footer	Text to display at the bottom of the email messages, for example: <code>If you have questions about your order, please contact service@example.com.</code>

Downloading Security Scripts

The Silent Order POST uses a security key to prevent fraudulent customers from using your account. The security key is contained in a security script that you download from the Business Center. Only merchant users with SK access (Security Keys) can download a key.

When a customer is ready to make a purchase, your checkout page uses the security script to generate a signature for the order. The signature uses the order data (amount, number, and decision for the order), the script's version and serial number, your merchant ID, and the time stamp to verify that the order came from your Web site and has not been modified by anyone.

The unique serial number embedded in each new script is sent in every Hosted Order Page request and is returned to you in the reply so that you can verify its validity. If you think that one or more transactions may be fraudulent, you can keep track of the serial number in the script and delete any serial number that may be compromised. You can have many security scripts, and because you can name each script uniquely, you can assign each script to a specific group or type of orders.

Important Because your merchant ID is included in the script, if you do not protect your security script, your CyberSource account may be compromised. Therefore, do not allow anyone to make copies of your scripts, and if you lose one, you should immediately delete it.

The download page, located under **Tools & Settings > Security**, contains these sections:

- The top section shows the serial number of all your current active keys and gives you the option to delete one or more of them as necessary to protect your business.
- The bottom section shows the types of scripts that you can create.

Download a security script of a type supported by your ISP: ASP, PHP, Perl, JSP, ColdFusion, or ASP.NET. Because Perl is more difficult to use than the other languages, most hosting providers that support Perl CGI scripts also support the others. For more information, contact your hosting provider. If you choose the JSP script, Java™ version 1.4.2 must be installed on your computer.

The figure below shows a sample page with many active keys.

Hosted Order Page Security

Security scripts ensure that each transaction is protected during its transmission across the Internet and prevent fraudulent customers from using your account. If you do not protect the security scripts, your CyberSource account may be compromised. Therefore, do not allow anyone to make copies of your security scripts. If you feel that a security script has been compromised, delete it immediately and generate a new one.

Existing Security Keys

Each new security script contains a unique serial number that is transmitted with your transaction requests. You can deactivate a script by deleting its serial number. Upon deletion, transactions submitted with this serial number will be rejected.

(Note: Scripts created before May 2006 do not contain or transmit a serial number.)

Serial Number	Delete?
1483971064430167904065	<input type="checkbox"/>
1484202762320167904065	<input type="checkbox"/>

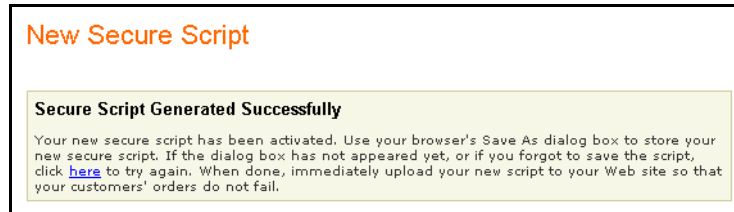
Generate Security Script ?

To process transactions with the Hosted Order Page, you must create at least one security script.

To generate a security script, select the scripting language supported by your hosting provider, click **Submit**, and follow the instructions on the screen.

ASP Classic
 C# (.NET)
 VB (.NET)
 ColdFusion
 JSP
 Perl
 PHP

After you submit your request for a security script, this page appears when your script is created.



All you need to do is to save the file either when your Web browser prompts you or when you click [here](#). As soon as you have saved your script in a safe location, you need to upload the script in ASCII mode to the root directory of your Web server, for example:

```
\opt\jakarta-tomcat-4.1.24\webapps\ROOT
```

Important For your scripts to work properly, you must upload them in ASCII mode, not in binary mode, and you must do so immediately so that your customers' orders do not fail.

Chapter 3

Creating and Customizing Checkout Pages

With a simple HTML page, you can describe your product to your customer, but the customer cannot purchase your product. By adding to the product page a button that leads to a checkout page, your customer can purchase your product through the Silent Order POST.

When the customer clicks the button, your order form appears in the customer's Web browser. After the customer completes and submits the form, the order information is sent to CyberSource to be processed.

This chapter describes in these sections how to create checkout pages for your Web site:

Gathering the Information for the Checkout Page

[Creating Checkout Pages](#)

[Customizing the Silent Order POST](#)

Gathering the Information for the Checkout Page

Before you start to create the checkout page, prepare this information.

Product Information

The Silent Order POST cannot process line items, only total amounts. If your site uses shopping-cart technology, you need to collect information about all the products and send the total amount of all products to CyberSource. For each of your products, you need this information:

Name	<i>sample_product_name</i>
Description	Here, you enter the description of your product.
Price	Price of product, such as 10.00 (without the currency sign)

URL for Sending Your Orders

You need to send your order requests to this CyberSource URL:

Test	<pre><form action="https://orderpagetest.ic3.com/hop/ProcessOrder.do" method="post"></pre>
Production	<pre><form action="https://orderpage.ic3.com/hop/ProcessOrder.do" method="post"></pre>

Add this URL to the `form` tag of your checkout page.

Credit Card Information

The Silent Order POST supports these card types:

Visa	Diners Club*
MasterCard	JCB
American Express	Maestro (UK Domestic) and Solo**
Discover	

* Although the Switch card is now called Maestro (UK Domestic), you may still see references to Switch in the documentation or in the Business Center.

** Diners Club does not appear on the order form if your default currency is U.S. dollars.

Maestro (UK Domestic) and Solo Cards

These popular debit cards are available only in the United Kingdom for payments in GBP. The cards are sometimes embossed with an issue number and/or a start date consisting of a month and year. The issue number may consist of one or two digits, and the first digit can be a zero. An issue number of 2 is different from an issue number of 02. If the card has an issue number and/or a start date, you must supply them when requesting an authorization. Make sure to supply exactly what is printed on the card.

MasterCard and Diners Club Alliance

This alliance enables merchants who accept MasterCard to automatically process Diners Club as MasterCard cards.

MasterCard cards have a 16-digit number that begins with 5. Diners Club have a 14-digit number that begins with either 30 or 38 if issued in North America by Diners Club North America or with 36 if issued outside of North America by Diners Club International. Process these cards as follows:

- If you are a merchant outside North America, you do not need to change how you process MasterCard or Diners Club cards.
- If you are a North American merchant, you must process these cards as MasterCard cards by setting the card type to MasterCard. You are responsible for indicating the

correct card type to CyberSource so that a Diners Club card is processed correctly either as a Diners Club or a MasterCard card:

- If you explicitly set the card type in your request, use the card number to determine the card type instead of the card type indicated by the customer.
- If you do not set the card type in the request, CyberSource determines the card type based on the card number.

Although setting the card type in your request is optional, it is required for Diners Club International cards (14-digit number that begins with 36) to indicate a MasterCard card.

Card Verification Number

The Card Verification Number is a three- or four-digit number printed on the back or front of credit cards. This number ensures that your customer has physical possession of the card at the time of the order.

Visa	Card Verification Value (CVV2)	Back of the card: if present, 3 digits in the signature area to the right of the credit card number
MasterCard	Card Verification Code (CVC2)	Back of the card: if present, 3 digits in the signature area to the right of the credit card number
American Express	Card Identification Number (CID)	Front of the card: 4 digits on the right above the card number
Discover	Cardmember ID (CID)	Back of the card: if present, 3 digits in the signature area to the right of the credit card number
Diners Club	Card Verification Value (CVV)	Back of the card: 3 digits in the signature area
JCB	Card Identification Number (CVN)	Front of the card: 4 digits on the left below the card number
Maestro (UK Domestic) and Solo	Card Verification Value (CVV)	Back of the card: if present, 3 digits in the signature area to the right of the credit card number

Electronic Checks Information

Note Check payments apply only to U.S. transactions.

The fields that you need to show on the order form vary according to your payment processor. If your processor is AmeriNet, check with AmeriNet to see if you are required to provide the date of birth.

If you want to offer electronic checks as payment option, you need to add a consent statement and a link to your Web site.

Check Statement. You must add to the checkout process of your Web site a consent statement for the check authorization. Your customer *must accept* this statement before submitting the order:

*By clicking **Authorize**, you authorize an electronic debit from your checking account that will be processed through the regular banking system. If your full order is not available at the same time, you authorize partial debits to your account, not to exceed the total authorized amount. The partial debits will take place upon each shipment of partial goods. If any of your payments are returned unpaid, you will be charged a returned item fee up to the maximum allowed by law. To exit without authorizing, click **Cancel**.*

Click [here](#) to view your state's returned item fee.

For the content of the [here](#) link, see the next section. You may omit the Cancel button as long as you provide a method for the customer to choose another form of payment.

Link to Returned Fees. You must add a link to the table of returned check fees. The table lists by state the amount that your customer has to pay if the check is returned. Because this table is updated regularly, we recommend that you create a link directly to it:

http://www.achex.com/html/NSF_pop.jsp

You can display the state fees table in a pop-up window, a full browser window, or directly on the checkout page.

Custom Fields

You can use two types of custom fields:

- Name-value pairs: you can assign any name that you wish.
- Merchant-defined fields: one to four fields to add other information to the order, such as an order number, additional customer information, or a special comment or request from the customer: merchantDefinedData1 through merchantDefinedData4.

To prevent your customers from seeing these fields, add them to your Web page as hidden fields as follows:

Example Name-value pair:

```
Special Order: <input type="hidden"
                  name="special_order"
                  value="Special Order">
```

Example Merchant-defined field:

```
merchantDefinedData1: <input type="hidden"
                            name="merchantDefinedData1"
                            value="sample_field_content">
```


The content of the fields appears in your receipt, the transaction search detail page, the exportable search results, and the Order Detail report. When using these fields, you must follow these rules:

Warning Merchant-Defined Data fields are not intended to and MUST NOT be used to capture personally identifying information. Accordingly, Merchant is prohibited from capturing, obtaining, and/or transmitting any personally identifying information in or via the Merchant-Defined Data fields. Personally identifying information includes, but is not limited to, name, address, credit card number, social security number, driver's license number, state-issued identification number, passport number, and card verification numbers (CVV, CVC2, CVV2, CID, CVN). In the event CyberSource discovers that Merchant is capturing and/or transmitting personally identifying information via the Merchant-Defined Data fields, whether or not intentionally, CyberSource WILL immediately suspend Merchant's account, which will result in a rejection of any and all transaction requests submitted by Merchant after the point of suspension.

Creating Checkout Pages

The pages look identical regardless of the type of scripting language that you use. Choose a scripting language that corresponds to the security script that you downloaded, and check with your ISP about any requirements for uploading new scripts to your Web site. The instructions for creating Web pages are located in appendices at the end of this guide.

[Creating Web Pages With PHP](#)

[Creating Web Pages With JSP](#)

[Creating Web Pages With ASP](#)

[Creating Web Pages with ColdFusion](#)

[Creating Web Pages With Perl](#)

[Creating Web Pages with ASP.NET](#)

The examples in the appendices show only the required fields that you need to include in your checkout page, but you can customize your page as you wish with HTML and with the request fields in the following section.

Many of the fields that you add to your checkout pages require that you decide whether they should be `hidden` or `text`. Choose `hidden` if you do not want customers to change the information on the screen.

After you have created and uploaded your Web pages, proceed to "[Testing the Silent Order POST](#)" on page 47.

Customizing the Silent Order POST

This chapter explains how to customize the Silent Order POST by using the different types of fields that you can include in your HTML form:

[Order Information Fields](#)

[Order Confirmation Fields](#)

[Customer Information Fields](#)

[Custom Fields](#)

[Payment Information Fields](#)

Make sure that you send at least the required fields. Some fields are always required (**amount** and **merchantID**), some fields may be required, but most are optional. Evaluate your business requirements to determine what fields to send in your requests. All field names listed are case sensitive. The values of all request fields must not contain new lines or carriage returns. However, they can contain embedded spaces and any other printable characters. All leading and trailing spaces will be removed during processing.

Order Information Fields

The order information fields contain information that the customer is never allowed to change, such as the description of the order. These fields allow you to manage your orders more effectively. Make sure that you set the type of these fields to `hidden`. For example, to set the order number, use a line such as this one:

```
<input type="hidden" name="orderNumber" value="87593">
```

Table 4 Order Information Fields

Field Name	Description	Required or Optional	Data Type (Length)
comments	Additional general information about the order. Do not insert sensitive information in this field. This information can appear in the order confirmation e-mail that you send to your customers.	Optional	String (255)
merchantID	Your CyberSource merchant ID.	Required	String (30)
orderNumber	Reference number unique to each order; equivalent to the merchant reference number. If you do not send an order number, the Hosted Order Page creates one for you. Use this number to keep track of the orders in your system and in the Business Center.	Optional	String

Table 4 Order Information Fields (Continued)

Field Name	Description	Required or Optional	Data Type (Length)
orderPage_ignoreAVS	<p>Indicates how to use the result of the address verification test. Use one of these values:</p> <ul style="list-style-type: none"> <code>true</code>: Ignore the result of the test. In this case, no transaction will be declined if the test fails (reason code 200). Depending on the type of transaction, the order is processed as follows: <ul style="list-style-type: none"> authorization: The authorization proceeds normally. Because you will review the order before capturing it, you can decide at that time whether to capture it or not. sale: The authorization and the capture proceed normally. <code>false</code>: Do not ignore the result of the test. In this case, the authorization is declined if the test fails, and the order is not processed. 	Optional	String (3)
orderPage_ignoreCVN	<p>Indicates how to use the result of the card verification test. Use one of these values:</p> <ul style="list-style-type: none"> <code>true</code>: Ignore the result of the test. In this case, no transaction will be declined if the test fails (reason code 230). Depending on the type of transaction, the order is processed as follows: <ul style="list-style-type: none"> authorization: The authorization proceeds normally. Because you will review the order before capturing it, you can decide at that time whether to capture it or not. sale: The authorization and the capture proceed normally. <code>false</code>: Do not ignore the result of the test. In this case, the authorization is declined if the test fails (D or N codes), and the order is not processed. 	Optional	String (3)
orderPage_merchantDisplayName	<p>Business name displayed on the order page. This field is available only with Payer Authentication. For more information, see "Introduction to Payer Authentication Services" on page 4.</p>	Optional	String
orderPage_transactionType	<p>Field that is populated automatically when the insertSignature function is called. Do not assign a value to this field</p>		String

Table 4 Order Information Fields (Continued)

Field Name	Description	Required or Optional	Data Type (Length)
taxAmount	Amount of tax on the order that is displayed only for the customer's reference. This amount is not added to the cost of the order. This field is required if you check the Display Tax Amount box in the Hosted Order Page settings.	Optional	String (15)

Customer Information Fields

Customer information fields comprise the billing and shipping address fields. Customers may have already entered their personal information on your Web site. The fields in [Table 5](#) are used to transfer the customer's billing, shipping, and account information from your Web site to CyberSource. If you do not use these fields, customer may need to provide this information again.

Table 5 Customer Information Fields

Field Name	Description	Required/ Optional	Data Type (Length)
billTo_city	City of the billing address. Make sure to send only valid and well-formed data in this field.	Required	String (50)
billTo_company	Name of the customer's company. Used only for TeleCheck corporate checks.	Optional	String (40)
billTo_country	Billing country for the account. Use the two-character country codes located in the Support Center . Make sure to send only valid and well-formed data in this field.	Required	String (2)
billTo_customerID	Optional customer's account ID, tracking number, reward number or other unique number that you assign to the customer for the purpose that you choose.	Optional	String (50)
billTo_dateOfBirth	Date of birth of the customer. Use format YYYY-MM-DD or YYYYMMDD. If your processor is AmeriNet, check with AmeriNet to see if you are required to provide the date of birth, and for AmeriNet corporate checks, use the value 1970-01-01.	Optional; Required for AmeriNet	String (10)
billTo_dateOfBirthMonth	Month of birth of the customer.	Optional; These three fields are accepted but not used by the Silent Order POST.	Integer (2)
billTo_dateOfBirthDate	Date of birth of the customer.		Integer (2)
billTo_dateOfBirthYear	Year of birth of the customer.		Integer (4)

Table 5 Customer Information Fields (Continued)

Field Name	Description	Required/ Optional	Data Type (Length)
billTo_driversLicenseNumber	Driver's license number of the customer.	Optional; Required for TeleCheck	String (30)
billTo_driversLicenseState	State or province where the customer's driver's license was issued. Use the two-character country codes located in the Support Center .	Optional; Required for TeleCheck	String (2)
billTo_email	Customer's email address, including the domain name, for example: <code>jdoe@example.com</code> . Make sure to send only valid and well-formed data in this field.	Required for checks	String (100)
billTo_firstName	Customer's first name as it appears on the card.	Required	String (60)
billTo_lastName	Customer's last name as it appears on the card.	Required	String (60)
billTo_phoneNumber	Customer's phone number.	Optional; Required for checks	String (15)
billTo_postalCode	Billing postal code for the credit card. Make sure to send only valid and well-formed data in this field. This field must contain between five and nine digits. For Canada, format the postal code as follows: <ul style="list-style-type: none"> • If the postal code has more than three characters, the first three must be <alpha><numeric><alpha>. • If the postal code has seven characters, the last three must be <numeric><alpha><numeric> 	Required for U.S. and Canada	String (10)
billTo_state	State or province of the customer. Make sure to send only valid and well formed data in this field. Use the two-character codes located in the Support Center .	Required for U.S. and Canada	String (2)
billTo_street1	Street address of the customer as it appears in the account issuer's records. Make sure to send only valid and well-formed data in this field.	Required	String (60)
billTo_street2	Used for additional address information, for example: <i>Attention: Accounts Payable</i>	Optional	String (60)
shipTo_city	City of the shipping address.	Optional	String (50)
shipTo_company	Company name of the shipping address.	Optional	String (60)
shipTo_country	Country where to ship the product. Use the two-character country codes located in the Support Center . Required if any shipping address information is included.	Optional	String (60)

Table 5 Customer Information Fields (Continued)

Field Name	Description	Required/ Optional	Data Type (Length)
shipTo_firstName	First name of the customer who receives the order.	Optional	String (60)
shipTo_lastName	Last name of the customer who receives the order.	Optional	String (10)
shipTo_postalCode	Postal code of the shipping address. Required for U.S. and Canada.	Optional	String (2)
shipTo_state	State, province, or territory of the shipping address. See the Support Center for a list of codes. Required for U.S. and Canada.	Optional	String (60)
shipTo_street1	First line of the shipping address.	Optional	String (50)
shipTo_street2	Second line of the shipping address.	Optional	String (50)

Payment Information Fields

Depending on the payment method chosen, either the credit card fields or the check fields are required. By default, the credit card fields are required.

Table 6 Payment Information Fields

Field Name	Description	Required or Optional	Data Type (Length)
amount	Amount of the order that is passed in the signature.	Required	Decimal (15)
card_accountNumber	Card account number. Make sure to send only valid and well-formed data for this field. Non-numeric values are ignored.	Required for cards	String w/ numbers only (20)
card_cardType	Type of card. CyberSource strongly recommends that you send the card type even if it is optional for your processor. Omitting the card type can cause the transaction to be processed with the wrong card type. Use one of these values: <ul style="list-style-type: none"> • 001: Visa • 002: MasterCard • 003: American Express • 004: Discover • 005: Diners Club • 007: JCB • 024: Maestro (UK Domestic) and Solo 	Required for cards	String (3)

Table 6 Payment Information Fields (Continued)

Field Name	Description	Required or Optional	Data Type (Length)
card_cvNumber	A three- or four-digit number printed on the back or front of credit cards that ensures that your customer has physical possession of the card. For more information, see " Card Verification Number " on page 23. Do not send this field if you do not want to use this test.	Optional for cards	String with numbers only (4)
card_expirationMonth	Expiration month (<i>MM</i>) of the credit card. For example, June is 06. The leading 0 is required.	Required for cards	Integer (2)
card_expirationYear	Expiration year (<i>YYYY</i>) of the credit card.	Required for cards	Integer (4)
card_issueNumber	Indicates how many times a Maestro (UK Domestic) or Solo card has been issued to the account holder. The card may or may not have an issue number. The field is required if the card has an issue number. The number can consist of one or two digits, and the first digit may be a zero. Include exactly what is printed on the card (a value of 2 is different from a value of 02). Do not include the field (even with a blank value) if the card is not a Maestro (UK Domestic) or Solo card.	Required for authorization only	String (5)
card_startMonth	Starting month (<i>MM</i>) of the validity period for the Maestro (UK Domestic) or Solo card. The card may or may not have a start date. The field is required if the card has a start date. Do not include the field (even with a blank value) if the card is not a Maestro (UK Domestic) or Solo card. The valid values are 01 to 12, inclusive.	Required for authorization only	String (2)
card_startYear	Year (<i>YYYY</i>) of the start of the Maestro (UK Domestic) or Solo card validity period. The card may or may not have a start date printed on it; the field is required if the card has a start date. Do not include the field (even with a blank value) if the card is not a Maestro (UK Domestic) or Solo card.	Required for authorization only	String (4)
check_accountNumber	Checking account number.	Required for checks	String with numbers only (17)
check_accountType	Checking account type. This field can contain one of these values: <ul style="list-style-type: none"> • C: Checking • S: Savings • X: Corporate checking 	Required for checks	String (1)

Table 6 Payment Information Fields (Continued)

Field Name	Description	Required or Optional	Data Type (Length)
check_bankTransitNumber	Bank routing or transit number.	Required for checks	String with numbers only (9)
check_checkNumber	Check number. If using AmeriNet, check with them to see if they require you to provide this field.	Optional	String with numbers only (8)
currency	Currency for the order. If this field is absent, the default currency is U.S. dollars.	Optional	String (5)
ecDebitService_referenceNumber	TeleCheck: reference number that you generate and that is used to track the transaction in the payment processor system. Use this number for reference with all your orders. This field is optional, but if you do not send this number, CyberSource creates one for you. The field also links a credit to the original debit request. AmeriNet: not used	Optional	String (25)
paymentOption	Method of payment. The field can contain one of the following values: <ul style="list-style-type: none"> • <code>card</code> (default) • <code>check</code> When the value is <code>check</code> , the contents of the credit card fields are ignored, and the fields appropriate for your check processor become required. Check payments apply only to U.S. transactions.	Optional; Required for checks	String (6)

Order Confirmation Fields

After customers complete their order, they receive a confirmation message indicating whether the order was successful or declined. This page includes a link back to your Web site. You can add text to the top and bottom of the receipt page. You can use these fields to override any of the data that you entered in the Hosted Order Page settings.

Important To ensure that no one captures your order form and overrides some of the data, such as the amount of a product, retain a copy of the order and compare it with the confirmation on the order.

Table 7 Choosing How Orders Are Confirmed

Field Name	Description	Required or Optional	Data Type (Length)
orderPage_declineLinkText	Text for the link at the bottom of the decline page, such as <code>Return to Example.com</code> .	Optional	String
orderPage_declineResponseURL	URL for the link at the bottom of the decline page, such as <code>http://www.example.com/</code> .	Optional	String
orderPage_emailFromAddress	Sender's email address to use on the customer's email receipt, such as <code>service@example.com</code> . Note Because some customers may reply to the email receipt to ask questions about their orders, make sure to use a real email address and to check it frequently.	Optional	String
orderPage_emailFromName	Sender's name to use on the customer's email receipt, such as <code>Customer Service</code> .	Optional	String
orderPage_merchantEmailAddress	Email address for sending an order receipt to your business, such as <code>orders@example.com</code> .	Optional	String
orderPage_receiptLinkText	Text for the link at the bottom of the receipt page, such as <code>Return to Example.com</code> .	Optional	String
orderPage_receiptResponseURL	URL for the link at the bottom of the receipt page, such as <code>http://www.example.com/</code> .	Optional	String
orderPage_sendCustomerReceiptEmail	Indicates whether to send an email receipt to the customer. Use one of these values: <ul style="list-style-type: none"> <code>true</code>: Send an email receipt to the customer. <code>false</code>: Do not send an email receipt to the customer. 	Optional	String (5)
orderPage_sendMerchantEmailPost	Indicates whether to send the transaction results to the address specified in the field orderPage_merchantEmailPostAddress . Use one of these values: <ul style="list-style-type: none"> <code>true</code>: Send the transaction results to your business. <code>false</code>: Do not send the transaction results to your business. 	Optional; Required to send a POST to an email address	String
orderPage_merchantEmailPostAddress	Address for sending the transaction results to your business, such as <code>orders@example.com</code> .	Optional	String

Table 7 Choosing How Orders Are Confirmed (Continued)

Field Name	Description	Required or Optional	Data Type (Length)
orderPage_sendMerchantReceiptEmail	Indicates whether to send an email receipt to the address specified in the field orderPage_merchantEmailAddress . Use one of these values: <ul style="list-style-type: none"> <code>true</code>: Send an email receipt to your business. <code>false</code>: Do not send an email receipt to your business. 	Optional; Required to send a receipt to an email address	String (5)
orderPage_sendMerchantURLPost	Indicates whether to send the transaction results to the URL specified in the field orderPage_merchantURLPostAddress . Use one of these values: <ul style="list-style-type: none"> <code>true</code>: Send the transaction results to your business. <code>false</code>: Do not send the transaction results to your business. 	Optional but required to send a POST to a URL	String
orderPage_merchantURLPostAddress	URL for sending the transaction results to your business, such as <code>www.example.com</code> .	Optional	String

Custom Fields

In addition to the fields listed in this chapter, you can send custom fields, which are returned to you unchanged in the reply. Do not include sensitive information in these fields.

Table 8 Sending Custom Fields

Field Name	Description	Required/Optional	Data Type (Length)
	Any name-value pair that you wish.	Optional	String (255)
merchantDefinedData1 merchantDefinedData2 merchantDefinedData3 merchantDefinedData4	Optional information that you may want to add to the order. You can use one or more of these fields to add other information, such as an order number, additional customer information, or a special comment or request from the customer. The content of the field will appear in the receipt, the transaction search detail page, the exportable search results, and the Order Detail report.	Optional	String (64)

Chapter 4

Interpreting and Customizing Receipt Pages

Every time a customer places an order, CyberSource sends information about both successful and failed orders to a specific URL on your Web site. You receive the reply in the form of an encoded string that you need to verify, convert, and store in your database.

This chapter provides all the information necessary to write a script that parses the reply and shows the appropriate response page to the customer (success or failure):

[Parsing the Reply](#)

[Creating Receipt Pages](#)

[Order Result Fields](#)

Parsing and Interpreting the Reply

The reply message includes general information for the entire request and information relevant to the results of the type of transaction that you requested. After verifying that the order is authentic, you need to convert and store the reply information.

If you create your own receipt pages and use the same URL for the success and failure pages, you need to parse the reply results from that URL and to redirect your customer to the appropriate receipt page. For more information about receipt pages, see "[Determining the Receipt Information](#)" on page 38.

Important Because CyberSource may add reply fields and reason codes at any time, proceed as follows:

- You should parse the reply data according to the names of the fields instead of their order in the reply. For more information on parsing reply fields, see the documentation for your scripting language.
 - Your error handler should use the **decision** field to determine the result if it receives a reason code that it does not recognize.
-

To use the reply information, you must integrate it into your system by storing the data and passing it to any other system that needs the information. Do not show the reply information directly to customers. Instead, write an error handler to interpret the information that you receive, and present an appropriate response that tells customers the result.

Decisions

In the reply, you receive the **decision** field, which summarizes the overall result of your request. To determine your course of action, look at this field first. The decision can be one of these values:

- **ACCEPT:** The request succeeded.
- **REVIEW:** Smart Authorization was triggered. You should review this order.
- **ERROR:** There was a system error.
- **REJECT:** One or more of the services was declined.

Errors are due to system issues unrelated to the content of the request itself.

Requests can be rejected by CyberSource, the payment processor, or the issuing bank. For example, CyberSource will reject a request if required data is missing or invalid. Also, the issuing bank will reject a request if the card limit has been reached and funds are not available. To determine the reason for the reject decision, use the **reasonCode** field.

You are charged for all accepted and rejected requests but not for requests that result in errors.

Reason Codes

Two fields provide further information about the reason for the decision.

reasonCode. This field gives you the overall result of the request. You can use this field to determine the reason for the decision for the order and decide if you want to take further action:

- If the decision was **ERROR**, the **reasonCode** tells you the type of error.
- If the decision was **REJECT**, the **reasonCode** tells you the reason for the reject and whether you can take action that might result in a successful order. For a description of the reason codes, see "[Reason Codes](#)" on page 43.

<service>_reasonCode. This field gives you the result of the transaction type that you requested. You can use this field to debug your system.

Missing and Invalid Information. In the other API services, you receive reason code 101 for missing fields and 102 for invalid fields. With the Hosted Order Page, you receive the reason code 102 for all missing or invalid information.

Missing or Invalid Fields

You are responsible for ensuring that the data that you send to CyberSource is complete (no missing fields) and correct (no invalid data). To do this, verify the data entered on your Web site and point-of-sale applications before sending the information to CyberSource.

If you send a request with missing or invalid information, you will receive the appropriate reason code(s) and one or more reply fields, **InvalidField0...N** or **MissingField0...N**, which list the fields that you need to correct. The service(s) that you requested and the nature of the missing or invalid information will be used to determine the number and the content of the reply fields. For example, if three required fields are missing from your request, you will receive at least one and up to three reply fields named **MissingField0**, **MissingField1**, and **MissingField2**. You should correct these fields and resubmit the request.

Because the API behavior pertaining to these reply fields is always subject to change, do not use these fields to communicate with your customers.

Tracking Orders

The API provides different fields that act as identifiers for tracking orders. You need to use these fields in follow-on requests if you use the Simple Order API in addition of the Silent Order POST.

Order Number

This reference number is unique to each order. If you do not send an order number in the request, the Silent Order POST creates one for you. Use this number to keep track of the orders in your system and in the Business Center. This field contains the same information as the merchant reference number or code that is returned by the API services.

Request ID

The **requestID** is a unique identifier that CyberSource assigns in each request and returns to you in each reply. You receive a different **requestID** in each type of reply: API, email, and receipts.

You use the **requestID** to link a follow-on request to the original request. CyberSource uses the value to locate the capture information, reducing the amount of information that you must provide in the credit request. Also, you can use the **requestID** to discuss a specific request with Customer Support.

Request Token

The **orderPage_requestToken** is a unique identifier that CyberSource assigns to each request and returns to you in each reply. This field is an encoded string that contains no confidential information, such as account or card verification number. The string *may* contain up to 256 characters. Depending on how you process follow-on requests, such as captures or credits, you should store the content of this field because you may later need to retrieve and send the data in your follow-on service requests:

- If you process follow-on requests through the Business Center, you do not need to change how you use the Business Center or provide the request token data at this time.

- If you process follow-on requests through the API, you need to store the content of this field so that you can retrieve and send the request token data in the follow-on service requests. For more information on this subject, see the [Simple Order API User's Guide](#) and the [Request Token Migration Guide](#).

Determining the Receipt Information

Depending on the outcome of a request, you see different reply information, and your customer is forwarded to one of CyberSource's default or to your receipt pages. You and your customer may receive an email receipt. [Table 9](#) shows the results that you can receive, the corresponding receipt page that your customer can see, and the type of email message that CyberSource sends after a successful order.

Note You and your customer can receive email messages only if you have configured your Hosted Order Page settings accordingly. For more information, see "[Notification Fields](#)" on page 15.

Table 9 Mapping the Reply Information to the Customer's Receipt Page

Decision	Reason Code	Customer's Receipt Page	Email Messages Sent
ACCEPT	100	Success	Merchant POST
REVIEW	200, 201, 230, 520	Success	Customer Receipt Merchant Receipt
REJECT	102, 202, 203, 204, 205, 207, 208, 210, 211, 220, 221, 222, 231, 232, 233, 234, 236, 240, 476	Failure	No POST or receipt
REJECT	475	Payer Authentication	
ERROR	150, 151, 152, 250	System error	

Creating Receipt Pages

Although verification is optional, CyberSource strongly recommends that before you process an order, you verify that the information is not fraudulent and that it reflects the order placed by your customer. To assist in the verification process, retain a copy of your orders and compare these copies with the confirmation of the orders.

Important Verifying that an order is authentic does not mean that the order is approved.

[Creating Web Pages With PHP](#)

[Creating Web Pages With ASP](#)

[Creating Web Pages With Perl](#)

[Creating Web Pages With JSP](#)

[Creating Web Pages with ColdFusion](#)

[Creating Web Pages with ASP.NET](#)

After you have created and uploaded your Web pages, proceed to “[Testing the Silent Order POST](#)” on page 47.

Customizing the Receipt Page

This section describes the fields and [Reason Codes](#) that you can receive in the reply.

Order Result Fields

Table 10 Order Result Fields

Field Name	Description	Data Type (Length)
billTo_city	City of the billing address.	String (50)
billTo_company	Company name of the billing address.	String (40)
billTo_country	Country of the billing address. See the Support Center for a list of codes.	String (2)
billTo_customerID	Optional customer’s account ID, tracking number, reward number or other unique number that you assign to the customer for the purpose that you choose.	String (50)
billTo_email	Email address of the customer who pays for the order.	String (255)
billTo_firstName	First name of the customer who pays for the order.	String (60)
billTo_lastName	Last name of the customer who pays for the order.	String (60)
billTo_phoneNumber	Phone number of the customer who pays for the order.	String (15)
billTo_postalCode	Postal code of the billing address.	String (10)
billTo_state	State, province, or territory of the billing address. For a list of codes, see the Support Center .	String (2)
billTo_street1	First line of the billing address.	String (60)
billTo_street2	Second line of the billing address.	String (60)
card_accountNumber	Card account number, with all but the last four digits hidden. For example, if the card number is 4111111111111111, you will receive the value #####1111.	String (20)

Table 10 Order Result Fields (Continued)

Field Name	Description	Data Type (Length)
card_cardType	Type of credit card. This field can contain one of these values: <ul style="list-style-type: none"> • 001: Visa • 002: MasterCard • 003: American Express • 004: Discover • 005: Diners Club • 007: JCB • 024: Maestro (UK Domestic) and Solo 	String (3)
card_expirationMonth	Expiration month (<i>MM</i>) of the credit card. For example, June is 06.	Integer (2)
card_expirationYear	Expiration year (<i>YYYY</i>) of the credit card.	Integer (4)
ccAuthReply_amount	Total amount of the authorization.	Decimal (15)
ccAuthReply_authFactorCode	Smart Authorization factor code(s) that are returned if you use Smart Authorization. This field contain one or more codes, separated by carets (^): M^N^O^U. For a list of the possible values, see “Smart Authorization Factor Codes” on page 84.	String (100)
ccAuthReply_authorizationCode	Authorization code. Returned only if a value if returned by the processor.	String (6)
ccAuthReply_authorizedDateTime	Time of authorization in the format is YYYY-MMDDThh:mm:ssZ where T separates the date and the time, and the Z indicates GMT. For example, 2003-08-11T22:47:57Z equals August 11, 2003, at 10:47:57 P.M.	Date and time (20)
ccAuthReply_avsCode	Results of address verification. For a list of the possible values, see “AVS Codes” on page 81.	String (1)
ccAuthReply_avsCodeRaw	AVS result code sent directly from the processor. Returned only if a value is returned by the processor. Note Do not use this field to interpret the result of AVS. Use for debugging purposes only.	String (10)
ccAuthReply_cvCode	Result of processing the card verification number. For a list of the possible values, see “Card Verification Codes” on page 83.	String (1)
ccAuthReply_cvCodeRaw	Card verification result code sent directly from the processor. Note Returned only if a value is returned by the processor. Use for debugging purposes only.	String (10)

Table 10 Order Result Fields (Continued)

Field Name	Description	Data Type (Length)
ccAuthReply_processorResponse	Processor's response code sent by the processor. Note Returned only if a value is returned by the processor. Do not use this field to interpret the result of the authorization.	String (10)
ccAuthReply_reasonCode	Numeric value corresponding to the result of the overall request. For a list of possible values, see " Reason Codes " on page 43.	Integer (5)
check_accountNumber	Checking account number, with all but the last four digits hidden, for example: #####2345.	String with numbers only (17)
comments	Optional text that you included in your request.	String (255)
custom fields	Custom field(s) that you created with your own name-value pair names.	String (255)
decision	Summarizes the result of the overall request. The field can contain one of these values: <ul style="list-style-type: none"> • ACCEPT: The request succeeded. • REVIEW: Smart Authorization was triggered. You should review this order. • ERROR: A system error occurred. • REJECT: One or more of the services requested was declined. 	String (6)
decision_publicSignature	Validation result of the reply from the CyberSource that uses decision to verify that the order data is not fraudulent. This field can contain a string that you need to decrypt.	String
InvalidField0...N	Fields in the request that contained invalid data. Important In the other API services, this field is spelled invalidField_0...N . Note the initial lowercase i .	String (100)
merchantDefinedData1-4	Information that you included in your request.	String (64)
merchantID	Your CyberSource merchant ID.	String (30)
MissingField0...N	Required fields that were missing from the request. Important In the other API services, this field is spelled missingField_0...N . Note the initial lowercase m .	String (100)
orderAmount	Total amount of the authorization.	Decimal (15)

Table 10 Order Result Fields (Continued)

Field Name	Description	Data Type (Length)
orderAmount_publicSignature	Validation result of the reply from CyberSource that uses orderAmount to verify that the order data is not fraudulent. This field can contain a string that you need to decrypt.	String
orderCurrency	Currency used for the order.	String (5)
orderCurrency_publicSignature	Validation result that uses orderCurrency to verify that the order data is not fraudulent. This field can contain a string that you need to decrypt.	String (28)
orderNumber	Reference number that you provided for the order and that you can use to reconcile your orders.	String (50)
orderNumber_publicSignature	Validation result of the reply from CyberSource that uses orderNumber to verify that the order data is not fraudulent. This field can contain a string that you need to decrypt.	String (28)
orderPage_requestToken	Unique string that identifies the request.	String (256)
orderPage_serialNumber	Serial number of the security script that you used for the request. You can compare this value with that contained in the script that you used in the request to verify that your script is not compromised.	String (25)
orderPage_transactionType	Type of transaction used in the request script. You can compare this value with that contained in the request script to verify that your script is not compromised.	String (60)
paymentOption	Method of payment. The field contains one of these values: <ul style="list-style-type: none"> • card • check 	String (6)
reasonCode	Numeric value corresponding to the result of the overall request. For a list of possible values, see “Reason Codes” on page 43.	Integer (5)
reconciliationID	Reference number that you use to reconcile your CyberSource reports with your processor reports.	String (60)
requestID	Unique number that identifies the payment request.	String (26)
shipTo_city	City of the shipping address.	String (50)
shipTo_company	Company name of the shipping address.	String (40)
shipTo_country	Country of the shipping address. See the Support Center for a list of codes.	String (2)
shipTo_firstName	First name of the customer who receives the order.	String (60)

Table 10 Order Result Fields (Continued)

Field Name	Description	Data Type (Length)
shipTo_lastName	Last name of the customer who receives the order.	String (60)
shipTo_postalCode	Postal code of the shipping address.	String (10)
shipTo_state	State, province, or territory of the shipping address. See the Support Center for a list of codes.	String (2)
shipTo_street1	First line of the shipping address.	String (60)
shipTo_street2	Second line of the shipping address.	String (60)
signedFields	Comma-delimited list of the field names whose values are included in the encoded transaction signature. This field is returned in the POST data (URL and email).	String
taxAmount	Amount of tax on the order.	String (15)
transactionSignature	Validation result that uses all fields to verify that the order data is not fraudulent. This field is returned in the POST data (URL and email).	String (60)

Reason Codes

This table lists the reason codes returned by the API.

Table 11 Reason Codes

Reason Code	Description
100	Successful transaction.
102	One or more fields in the request are missing or invalid. Possible action: See the reply fields InvalidField0...N and MissingField0...N for the fields that are invalid or missing. Resend the request with the correct information.
150	Error: General system failure. Possible action: Wait a few minutes and resend the request.
151	Error: The request was received, but a server time-out occurred. This error does not include time-outs between the client and the server. Possible action: To avoid duplicating the order, do not resend the request until you have reviewed the order status in the Business Center.
152	Error: The request was received, but a service did not finish running in time. Possible action: To avoid duplicating the order, do not resend the request until you have reviewed the order status in the Business Center.

Table 11 Reason Codes (Continued)

Reason Code	Description
200	The authorization request was approved by the issuing bank but declined by CyberSource because it did not pass the Address Verification Service (AVS) check. Possible action: You can capture the authorization, but consider reviewing the order for the possibility of fraud.
201	The issuing bank has questions about the request. You cannot receive an authorization code in the API reply, but you may receive one verbally by calling the processor. Possible action: Call your processor or the issuing bank to obtain a verbal authorization code. For contact phone numbers, refer to your merchant bank information.
202	Expired card. Possible action: Request a different card or other form of payment.
203	The card was declined. No other information provided by the issuing bank. Possible action: Request a different card or other form of payment.
204	Insufficient funds in the account. Possible action: Request a different card or other form of payment.
205	The card was stolen or lost. Possible action: Review the customer's information and determine if you want to request a different card from the customer.
207	The issuing bank was unavailable. Possible action: Wait a few minutes and resend the request.
208	The card is inactive or not authorized for card-not-present transactions. Possible action: Request a different card or other form of payment.
210	The credit limit for the card has been reached. Possible action: Request a different card or other form of payment.
211	The card verification number is invalid. Possible action: Request a different card or other form of payment.
220	The processor declined the request based on a general issue with the customer's account. Possible action: Request a different form of payment.
221	The customer matched an entry on the processor's negative file. Possible action: Review the order and contact the payment processor.
222	The customer's bank account is frozen. Possible action: Review the order or request a different form of payment.

Table 11 Reason Codes (Continued)

Reason Code	Description
230	The authorization request was approved by the issuing bank but declined by CyberSource because it did not pass the card verification number check. Possible action: You can capture the authorization, but consider reviewing the order for the possibility of fraud.
231	The account number is invalid. Possible action: Request a different card or other form of payment.
232	The card type is not accepted by the payment processor. Possible action: Request a different card or other form of payment. Also, check with CyberSource Customer Support to make sure that your account is configured correctly.
233	The processor declined the request based on an issue with the request itself. Possible action: Request a different form of payment.
234	There is a problem with your CyberSource merchant configuration. Possible action: Do not resend the request. Contact Customer Support to correct the configuration problem.
236	A processor failure occurred. Possible action: Possible action: Wait a few minutes and resend the request.
240	The card type is invalid or does not correlate with the credit card number. Possible action: Ask your customer to verify that the card is really the type indicated in your Web store, then resend the request.
250	The request was received, but a time-out occurred with the payment processor. Possible action: To avoid duplicating the transaction, do not resend the request until you have reviewed the transaction status in the Business Center.
475	The customer is enrolled in payer authentication. Possible action: Authenticate the cardholder before continuing with the transaction.
476	The customer cannot be authenticated. Possible action: Review the customer's order.
520	The authorization request was approved by the issuing bank but declined by CyberSource based on your Smart Authorization settings. Possible action: Do not capture the authorization without further review. Review the ccAuthReply_avsCode , ccAuthReply_cvCode , and ccAuthReply_authFactorCode fields to determine why CyberSource rejected the request.

Chapter 5

Testing the Silent Order POST

Before you allow customers to place orders, test the Silent Order POST to make sure that your checkout pages work correctly.

[Using the Test Version of the Silent Order POST](#)
[Using the Debug Page](#)

Using the Test Version of the Silent Order POST

To use the test version of the Silent Order POST, follow these steps:

- 1 Modify the `action` attribute of the HTML form to the test URL:

```
<form action="https://orderpagetest.ic3.com/hop/ProcessOrder.do"
method="post">
```

- 2 To verify that your implementation is correct, request test transactions, including authorizations, captures, and credits, with your own credit card. If you prefer to use test credit card numbers, use those provided below with any future expiration date.

Table 12 Test Credit Card Numbers

Credit Card Type	Test Account Number (remove spaces when sending to CyberSource)
Visa	4111 1111 1111 1111
MasterCard	5555 5555 5555 4444
American Express	3782 8224 6310 005
Discover	6011 1111 1111 1117
JCB	3566 1111 1111 1113
Diners Club	3800 000000 0006
Maestro (UK Domestic)	6759 4111 0000 0008 (Issue number not required)
	6759 5600 4500 5727 054 (One-digit issue number required)
	5641 8211 1116 6669 (Two-digit issue number required)

Table 12 Test Credit Card Numbers (Continued)

Credit Card Type	Test Account Number (remove spaces when sending to CyberSource)
Solo	6334 5898 9800 0001 (Issue number not required)
	6767 8200 9988 0077 06 (One-digit issue number required)
	6334 9711 1111 1114 (Two-digit issue number required)

To easily tell what cards are processed successfully, use varying amounts for each type of credit card that you accept, such as \$1.00 for Visa, \$1.03 for MasterCard, and \$1.06 for American Express.

If the transaction fails, verify your implementation, including your payment processor information.

Note If your test order uses an amount from \$1,000–\$4,000, you may receive an error message that is a response configured specifically for the test system. This error does not occur when your customers place real orders.

- When you are finished with your testing, change the `action` attribute back to the original content (production URL) if necessary.

```
<form action="https://orderpage.ic3.com/hop/ProcessOrder.do"
method="post">
```

Using the Debug Page

This page simulates a customer placing an order at your Web store. All the fields in the request are shown, including the signature and the name-value pairs that you created. Use this page to verify that all the fields in the request are passed correctly from the customer to your Web store before the request is forwarded to CyberSource.

The debug page shows the fields that are required or optional depending on the method of payment. By default, credit card fields are required, but if the customer chooses to pay by check, check fields become required and appear on the debug page.

To produce this page, you need to modify some of the information in the fields (at least in the required fields), such as the amount or the signature, or use letters instead of numbers to simulate nonsensical text.

Note To see all the content, your browser must be enabled for JavaScript. When you perform this test repeatedly and with different values, you may notice that the results are not refreshed because the previous results are cached. To avoid this issue, you may need to close your browser and re-open it, or you can empty your browser's cache.

When you are in test mode, and the action on the previous page is `ProcessOrder.do`, this page never appears if all the fields are valid; however, the page always appears if you

send invalid fields. If you want to see the page even if all the fields are valid, you can temporarily change the action on the previous page to `CheckOrderData.do`.

```
<form action="https://orderpagetest.ic3.com/hop/CheckOrderData.do"
method="post">
```

When you are in live mode, your customers may see this page if they enter invalid value(s) in some of the fields directly on your Web store, and you do not validate the fields before the POST action.

```
<form action="https://orderpage.ic3.com/hop/CheckOrderData.do"
method="post">
```

To prevent your customers from seeing this page, you must either validate the fields before the POST, or you must not allow customers to enter information in any fields that the Silent Order POST makes available to them after the POST. In this second case, the customer sees the appropriate standard validation form.

The figure below shows a sample debug page in which the signature is incorrect. The validation column shows the valid fields in green and the invalid ones in red. In the second table at the bottom of the figure, you can see the extra fields that you included in your request. These fields will be returned unchanged in the reply. To complete your test, you can correct the errors and re-send the information.

Note If you see this page when all the fields are valid, most likely your system's time differs from that of CyberSource by more than 15 minutes. In this case, you see the **orderPage_timestamp** field with the value `INVALID`. CyberSource recommends that you verify and, if necessary, adjust your system's time. In the figure on the next page, the time stamp is valid.

An error occurred while you submitted your order. If you are trying to make a purchase, please contact the merchant.

Here are the API fields that were sent in the request:

Field Name	Required for HOP	Required for SOP	Present	Validation	Field Value
amount	X	X			
billTo_city		X	X	VALID	mountain view
billTo_country		X			
billTo_customerID					
billTo_email					
billTo_firstName		X	X	VALID	John
billTo_lastName		X	X	VALID	Doe
billTo_phoneNumber					
billTo_postalCode			X	VALID	94043
billTo_state			X	VALID	ca
billTo_street1		X	X	VALID	234 Elm St
billTo_street2					
card_accountNumber		X			
card_cardType		X	X	VALID	001
card_cvNumber		X			
card_expirationMonth		X			
card_expirationYear		X			
comments					
currency	X	X	X	VALID	usd
merchantDefinedData1					
merchantDefinedData2					
merchantDefinedData3					
merchantDefinedData4					
merchantID	X	X	X	VALID	ubcyp1_1
orderNumber					
orderPage_signaturePublic	X	X	X	INVALID	PcpgRnM0Ydmn7FVKIHm1oDVdP9U=
orderPage_timestamp	X	X	X	VALID	1185489041871
orderPage_transactionType	X	X	X	VALID	authorization
reconciliationID					
shipTo_city					
shipTo_country					
shipTo_firstName					
shipTo_lastName					
shipTo_postalCode					
shipTo_state					
shipTo_street1					
shipTo_street2					
taxAmount					

Here are the other pass-through fields that were sent in the request:

Field Name	Field Value
------------	-------------

Appendix A

Creating Web Pages With PHP

With a simple HTML page, you can describe your product to your customer, but the customer cannot purchase your product. By adding a Buy Now button to the page, your customer can purchase your product through the Hosted Order Page.

When the customer clicks the button, the Hosted Order Page appears in the customer's browser. After the customer completes and submits the form, the order information is sent to CyberSource to be processed.

Checkout Page

Create and upload a Web page for each product that you want to sell. The instructions below take you through the steps necessary to create a checkout page for your Web store.

- 1 Save a text file with the file extension `.php`.
- 2 Describe your product, for example:

```
<html><body>
  <h1>sample_product_name</h1>
  <p>Here, you enter the description of your product.</p>
```

Below the description of the product, start constructing the form that will contain all the information to process the order: customer and payment information, type of transaction, and any additional information that you want to collect about the customer or the order.

- 3 Start creating your page with this line.
If necessary, replace `HOP.php` with the name of your script.

```
<?php include("HOP.php") ?>
```

- 4 Start the order form with this line, which will send the order information to the correct CyberSource location for the Hosted Order Page:

```
<form action="https://orderpagetest.ic3.com/hop/ProcessOrder.do"
  method="POST">
```

- 5 Add the signature method, replacing the sample values with your own:

```
insertSignature3("10.00", "usd", "sale");
```

- Amount: In this example, **10.00** is the sample cost of an order.
- Currency: If you do not include the currency, the default is USD. For more information on the currency, see [“Payment Information Fields”](#) on page 30.
- Transaction type: This option enables CyberSource to verify that no fraudulent customer intercepts and modifies the transaction type when you transmit the transaction to CyberSource. In this example, the type of transaction is `sale`. The options are as follows: `authorization` or `sale`.

- 6 Add the payment information.

- 7 Add the customer’s billing and shipping information:

```
<input type="hidden" name="billTo_firstName" value="John">
<input type="hidden" name="billTo_lastName" value="Doe">
<input type="hidden" name="billTo_email"
      value="jdoe@cybersource.com">
```

- 8 Add any optional fields that you want, such as page configuration or a comment:

```
<input type="hidden" name="merchantDefinedData1" value="ship">
```

Note Note For each field, choose the hidden type for fields that you do not want the customer to see the content and text if you want the customer to modify the field.

- 9 Insert the button that the customer must click to complete the purchasing process:

```
<input type="submit" value="Buy Now">
```

- 10 Finish the order form with the closing tags:

```
</form>
</body>
</html>
```

This finished sample Web page appears as follows:

```
<html><body>
<h1>sample_product_name</h1>
<p>Here, you enter the description of your product.</p>

<?php include("HOP.php") ?>
<form action="https://orderpagetest.ic3.com/hop/ProcessOrder.do" method="post">

<?php InsertSignature3("10.00", "usd", "sale") ?>

  <input type="hidden" name="billTo_firstName" value="John">
  <input type="hidden" name="billTo_lastName" value="Doe">
  <input type="hidden" name="billTo_email" value="jdoe@cybersource.com">
  <input type="hidden" name="merchantDefinedData1" value="ship">
  <input type="submit" name="submit" value="Buy Now">

</form>
</body></html>
```

- 11 Upload the Web page and the security script to your Web server, making sure to store the Web page and the security script in the same directory.

Check with your ISP to learn how to upload new scripts for your Web site. Make sure that the Web page and the security script must be in the same directory.

You can now use the Hosted Order Page on your Web site. To make sure your customers' orders are successful, test the order page before you use it to accept orders. For more information, see "[Testing the Silent Order POST](#)" on page 47.

Receipt Page

You need to write a script that will perform two main functions: the transaction signature uses the order data to verify the authenticity of the order whereas the second parses the reply and shows the appropriate response page to the customer (success or failure). After you have created your script, upload it to your Web server as you did for your Web pages.

Verifying the Authenticity of an Order

You can verify the authenticity of an order by using the signature, which uses the transaction data. After you are satisfied with the authenticity of the data, you can parse the rest of the reply.

- 1 Choose a secure (HTTPS) and appropriate URL for CyberSource to send the order information.
- 2 Add the `VerifyTransactionSignature` function.

This function verifies that the contents of the reply fields were not altered during transmission from your Web site to CyberSource. You receive a boolean response that you need to parse.

- 3 Write a script to capture the order information and store it on your system.
- 4 Upload the program to your Web server as you did for your Web pages.

Sample Script

This example shows a PHP script with the minimum number of fields required to process the data that CyberSource returns after a order is placed. The script tells only whether the order was accepted, rejected, or invalid. This example shows `decision = ACCEPT`.

```
<html><body>
<?php
if (VerifyTransactionSignature($_POST))
{
    if ($_POST['decision'] == "ACCEPT")
    {
        echo "<h1>Transaction Receipt</h1>\n";
        echo "<p>Your order has been accepted.<p>\n";
        echo "<p>Your order number is: " . $_POST['orderNumber']. "</p>\n";
    }
    else
    {
        echo "<h1>Decline Page</h1>\n";
        echo "<p>Your order has been rejected.</p>\n\n";
    }
}
else
{
    echo "<h1>Transaction Invalid</h1>\n";
    echo "<p>The signature does not match the order.</p>\n";
}
?>
</body></html>
```

Appendix B

Creating Web Pages With ASP

With a simple HTML page, you can describe your product to your customer, but the customer cannot purchase your product. By adding a Buy Now button to the page, your customer can purchase your product through the Hosted Order Page.

When the customer clicks the button, the Hosted Order Page appears in the customer's browser. After the customer completes and submits the form, the order information is sent to CyberSource to be processed.

Checkout Page

Create and upload a Web page for each product that you want to sell. The instructions below take you through the steps necessary to create a checkout page for your Web store.

- 1 Save a text file with the file extension `.asp`.
- 2 Start creating your page with this line:
If necessary, replace `HOP.asp` with the name of your security script.

```
<!--#include file="HOP.asp"-->
```

- 3 Describe your product, for example:

```
<html><body>  
  <h1>sample_product_name</h1>  
  <p>Here, you enter the description of your product.</p>
```

Below the description of the product, start constructing the form that will contain all the information to process the order: customer and payment information, type of transaction, and any additional information that you want to collect about the customer or the order.

- 4 Start the order form with this line, which will send the order information to the correct CyberSource location for the Silent Order POST:

```
<form action="https://orderpagetest.ic3.com/hop/ProcessOrder.do"  
method="POST">
```

- 5 Add the required signature field, replacing the sample value with your own:

```
<% call InsertSignature3("10.00", "usd", "sale") %>
```

- Amount. In this example, **10.00** is the sample cost of an order.
- Currency: If you do not include the currency, the default is USD. For more information on the currency, see [“Payment Information Fields”](#) on page 30.
- Transaction type: This option enables CyberSource to verify that no fraudulent customer intercepts and modifies the transaction type when you transmit the transaction to CyberSource. In this example, the type of transaction is `sale`. The options are as follows: `authorization` or `sale`.

- 6 Add the payment information.

- 7 Add the customer’s billing and shipping information:

```
<input type="hidden" name="billTo_firstName" value="John">
<input type="hidden" name="billTo_lastName" value="Doe">
<input type="hidden" name="billTo_email"
value="jdoe@cybersource.com">
```

- 8 Add any optional fields that you want, such as page configuration or a comment:

```
<input type="hidden" name="merchantDefinedData1" value="ship">
```

- 9 Insert the button that the customer must click to complete the purchasing process:

```
<input type="submit" value="Buy Now">
```

- 10 Finish the order form with the closing tags:

```
</form>
</body>
</html>
```

This finished sample Web page appears as follows:

```
<!--#include file="HOP.asp"-->
<html><body>
<h1>sample_product_name</h1>
<p>Here, you enter the description of your product.</p>

<form action="https://orderpagetest.ic3.com/hop/ProcessOrder.do" method="post">
<% call InsertSignature3("10.00", "usd", "sale") %>
  <input type="hidden" name="billTo_firstName" value="John">
  <input type="hidden" name="billTo_lastName" value="Doe">
  <input type="hidden" name="billTo_email" value="jdoe@cybersource.com">
  <input type="hidden" name="merchantDefinedData1" value="ship">
  <input type="submit" name="submit" value="Buy Now">
</form>
</body></html>
```


- 11 Upload the Web page and the security script to your Web server, making sure that the Web page and the security script are in the same directory.

Check with your ISP to learn how to upload new scripts for your Web site.

You can now use the Hosted Order Page on your Web site. To make sure your customers' orders are successful, test the order page before you use it to accept orders. See "[Testing the Silent Order POST](#)" on page 47 for more information.

Receipt Page

You need to write a script that will perform two main functions: the transaction signature uses the order data to verify the authenticity of the order whereas the second parses the reply and shows the appropriate response page to the customer (success or failure). After you have created your script, upload it to your Web server as you did for your Web pages.

Verifying the Authenticity of an Order

You can verify the authenticity of an order by using the signature, which uses the transaction data. After you are satisfied with the authenticity of the data, you can parse the rest of the reply.

- 1 Choose a secure (HTTPS) and appropriate URL for CyberSource to send the order information.
- 2 Add the `VerifyTransactionSignature` function:
This function verifies that the contents of the reply fields were not altered during transmission from your Web site to CyberSource. You receive a boolean response, which you need to parse.
- 3 Write a script to capture the order information and store it on your system.
- 4 Upload the program to your Web server as you did for your Web pages.

Sample Script

This example shows a script with the minimum number of fields required to process the data that CyberSource returns after an order is placed. The script tells only whether the order was accepted, rejected, or invalid. This example shows `decision = ACCEPT`

```
<html><body>
<%
var map = new Array();
for (var x = 1; x <= Request.Form.count(); x++) {
    map[Request.Form.key(x)] = Request.Form.item(x);
}
if (VerifyTransactionSignature(map)) {
if (Request.Form("decision") == "ACCEPT") {
    Response.Write("<h1>Transaction Receipt</h1>");
    Response.Write("<p>");
    Response.Write("Your order has been accepted.");
    Response.Write("<p>");
    Response.Write("Your order number is: ");
    Response.Write(Request.Form("orderNumber"));
} else {
    Response.Write("<h1>Decline Page</h1>");
    Response.Write("<p>");
    Response.Write("Your order has been rejected.");
}
} else {
    Response.Write("<h1>Invalid Transaction</h1>");
    Response.Write("<p>The signature does not match the order.</p>");
}
%>
</body></html>
```

Appendix C

Creating Web Pages With Perl

With a simple HTML page, you can describe your product to your customer, but the customer cannot purchase your product. By adding a Buy Now button to the page, your customer can purchase your product through the Hosted Order Page.

When the customer clicks the button, the Hosted Order Page appears in the customer's browser. After the customer completes and submits the form, the order information is sent to CyberSource to be processed.

Checkout Page

Create and upload a Web page for each product that you want to sell.

Creating the Page

Note Although the code below is shown with skipped lines and aligned text for readability, the Perl language ignores the white space.

- 1 Save a text file with the file extension `.cgi` or `.pl` depending on your server requirement.
- 2 At the top of your Web page, above the `<html>` tag, add the following text:

```
#!/usr/bin/perl -I.  
use CGI qw(:standard);  
use strict;  
use HOP qw(insert_signature3);  
  
print new CGI->header(-type=>"text/html");  
print<<EOF;
```

- 3 Below the text that you added in the previous step, describe your product:

```
<html><body>  
  <h1>sample_product_name</h1>  
  <p>Here, you enter the description of your product.</p>
```

Below the description of the product, start constructing the form that will contain all the information to process the order: customer and payment information, type of transaction, and any additional information that you want to collect about the customer or the order.

- 4 Start the order form with this line, which will send the order information to the correct CyberSource location for the Silent Order POST:

```
<form action="https://orderpagetest.ic3.com/hop/ProcessOrder.do"
method="POST">
```

- 5 Add the beginning Perl statement for your form:

```
EOF
;
```

Note Do not indent EOF and always follow with a semicolon on the next line.

- 6 Add the required signature field, replacing the sample values with your own:

`insert_signature3` requires the amount, the currency, and the transaction type:

```
insert_signature3("10.00", "usd", "sale");
```

- Amount: In this example, **10.00** is the sample cost of an order.
- Currency: If you do not include the currency, the default is USD. For more information on the currency, see [“Payment Information Fields”](#) on page 30.
- Transaction type: This option enables CyberSource to verify that no fraudulent customer intercepts and modifies the transaction type when you transmit the transaction to CyberSource. In this example, the type of transaction is `sale`. The options are as follows: `authorization` or `sale`.

- 7 Add the ending Perl statement for your form:

```
print<<EOF;
```

- 8 Add the payment information.

- 9 Add the customer’s billing and shipping information:

```
<input type="hidden" name="billTo_firstName" value="John">
<input type="hidden" name="billTo_lastName" value="Doe">
```

- 10 Add any optional fields that you want, such as page configuration or a comment:

```
<input type="hidden" name="merchantDefinedData1" value="ship">
```

- 11 Insert the button that the customer must click to complete the purchasing process:

```
<input type="submit" value="Buy Now">
```

- 12** Finish the order form with the closing tags:

```
</form>
</body>
</html>
```

- 13** Add the Perl CGI wrapper to your Web page below the </html> tag:

```
EOF
;
```

This finished sample Web page appears as follows:

```
#!/usr/bin/perl -I.
use CGI qw(:standard);
use strict;
use HOP qw(insert_signature3);

print new CGI->header(-type=>"text/html");
print<<EOF;

<html><body>
<h1>sample_product_name</h1>
<p>Here, you enter the description of your product.</p>

<form action="https://orderpagetest.ic3.com/hop/ProcessOrder.do" method="post">
EOF
;
insert_signature3("10.00", "usd", "sale");
print<<EOF;
  <input type="hidden" name="billTo_firstName" value="John">
  <input type="hidden" name="billTo_lastName" value="Doe">
  <input type="hidden" name="merchantDefinedData1" value="ship">
  <input type="submit" name="submit" value="Buy Now">

</form>
</body></html>
EOF
;
```

Uploading a Web Page

To upload the Web page and the security script to the `cgi-bin` directory of your Web server, you use an FTP program.

Important You must upload your files in ASCII mode, not in binary mode.

- 1** Enter the name of the FTP host to which you upload files for your Web site as follows:

```
ftp.name_of_your_web_server.com
```

- 2 Enter the name of the host directory of your Web server where documents visible to the public are stored.
For example, the files of your Web server may be located in a directory called `public_html`. If you do not know where to place the files, contact the help desk of your ISP.
- 3 Enter the user name and password that you use to connect to the FTP server.
- 4 Select appropriate firewall options for your site:
 - Select **Use Firewall** if you connect to the remote server from behind a firewall.
 - Select **Use Passive FTP** if your firewall configuration requires it. Some firewalls require using passive FTP, which lets your local software set up the FTP connection instead of requesting the remote server to set it up.
- 5 Click **OK**.

You can now use the Hosted Order Page on your Web site. To make sure your customers' orders are successful, test the order page before you start using it to accept orders. See "[Testing the Silent Order POST](#)" on page 47.

Receipt Page

You need to write a script that contains two main sections: the first section verifies the authenticity of the order, and the second parses the reply and shows the appropriate response page to the customer (success or failure). After you have created your script, upload it to your Web server as you did for your Web pages.

Verifying the Authenticity of an Order

You can verify the authenticity of an order by using the signature, which uses the transaction data. After you are satisfied with the authenticity of the data, you can parse the rest of the reply.

- 1 Choose a secure (HTTPS) and appropriate URL for CyberSource to send the order information, for example:

```
order_result.cgi  
https://www.example.com/cgi-bin/order_result.cgi
```
- 2 Add the `verify_transaction_signature` function:
This function verifies that the contents of the reply fields were not altered during transmission from your Web site to CyberSource. You receive a boolean response that you need to parse.
- 3 Write a script to capture the order information and store it on your system.
- 4 Upload the program to your Web server as you did for your Web pages.

Sample Script

This example shows a Perl script with the minimum number of fields required to process the data that CyberSource returns after an order is placed. The script tells only whether the order was accepted, rejected, or invalid. This example shows `decision = ACCEPT`.

Important For your script to work properly, you must upload your file in ASCII mode, not in binary mode.

```
#!/usr/bin/perl -I.
use CGI qw(:standard Vars);
use HOP qw(verify_transaction_signature);
my $query = CGI->new;
print new CGI->header(-type=>"text/html");
print "<html><body>";
if (verify_transaction_signature(CGI::Vars())) {
    if ($query->param('decision') eq "ACCEPT") {
        print "<h1>Transaction Receipt</h1>";
        print "<P>";
        print "Your order has been accepted.";
        print "<P>";
        $number=$query->param('orderNumber');
        print "Your order number is:$number";
    } else {
        print "<h1>Decline Page</h1>";
        print "<P>";
        print "Your order has been rejected.";
    }
} else {
    print "<h1>Transaction Invalid</h1>";
    print "<p>The signature does not match the order.</p>";
}
print "</body></html>";
```


Appendix D

Creating Web Pages With JSP

With a simple HTML page, you can describe your product to your customer, but the customer cannot purchase your product. By adding a Buy Now button to the page, your customer can purchase your product through the Hosted Order Page.

When the customer clicks the button, the Hosted Order Page appears in the customer's browser. After the customer completes and submits the form, the order information is sent to CyberSource to be processed.

When choosing a Web server, such as Apache Tomcat®, make sure that it can support JSP (Java Server Pages). You can find the software and the installation and configuration instructions at [The Apache Software Foundation](#). In addition, to use JSP, you must have Java™ version 1.4.2 or later installed. If you need more information on this subject, contact your ISP.

Checkout Page

The instructions below take you through the steps necessary to create a checkout page for your Web store.

- 1 Save a text file with the file extension `.jsp`.
- 2 Start creating your page with these two lines:

If necessary, replace `HOP.jsp` with the name of your security script.

```
<%@ page contentType="text/html; charset=UTF-8" language="java"%>
<%@ include file="HOP.jsp"%>
```

Important Make sure that the character encoding is UTF-8.

- 3 Describe your product, for example:

```
<html><body>
  <h1>sample_product_name</h1>
  <p>Here, you enter the description of your product.</p>
```

Below the description of the product, start constructing the form that will contain all the information to process the order: customer information, payment information,

type of transaction, and any additional information that you would like to collect about the customer or the order.

- 4** Start the order form with this line, which will send the order information to the correct CyberSource location for the Hosted Order Page:

```
<form action="https://orderpagetest.ic3.com/hop/ProcessOrder.do"
method="POST">
```

- 5** Add the required signature field, replacing the sample values with your own.

```
<%= insertSignature("30.00", "usd", "sale") %>
```

- Amount: In this example, **30.00** is the sample cost of an order.
- Currency: If you do not include the currency, the default is USD. For more information on the currency, see [“Payment Information Fields”](#) on page 30.
- Transaction type: This option enables CyberSource to verify that no fraudulent customer intercepts and modifies the transaction type when you transmit the transaction to CyberSource. The options are as follows: `authorization` or `sale`.

- 6** Add the payment information.

- 7** Add the customer’s billing and shipping information:

```
<input type="hidden" name="billTo_firstName" value="John">
<input type="hidden" name="billTo_lastName" value="Doe">
<input type="hidden" name="billTo_email"
value="jdoe@cybersource.com">
```

- 8** Add any optional additional fields that you want, such as page configuration or a comment:

```
<input type="hidden" name="merchantDefinedData1" value="ship">
```

- 9** Insert the button that the customer must click to complete the purchasing process:

```
<input type="submit" value="Buy Now">
```

- 10** Finish the order form with the closing tags:

```
</form>
</body></html>
```

The finished order form appears as follows:

```
<%@ page contentType="text/html; charset=UTF-8" language="java"%>
<%@ include file="HOP.jsp"%>
<html><body>
<h1>sample_product_name</h1>
<p>Here, you enter the description of your product.</p>

<form action="https://orderpagetest.ic3.com/hop/ProcessOrder.do" method="POST">

<%= insertSignature("30.00", "usd", "sale") %>

<input type="hidden" name="billTo_firstName" value="John">
<input type="hidden" name="billTo_lastName" value="Doe">
<input type="hidden" name="billTo_email" value="jdoe@cybersource.com">
<input type="hidden" name="merchantDefinedData1" value="ship">
<input type="submit" value="Buy Now">
</form>
</body></html>
```

- 11 Upload the Web page and the security script to the `ROOT` directory of your Web server, making sure that the Web page and the security script are in the same directory.
- 12 Stop and restart your Web server so that it recognizes your Web store.

When done, to make sure that your customers' orders are successful, test the order page before you use it to accept orders. For testing information, see "[Testing the Silent Order POST](#)" on page 47.

Receipt Page

You need to write a script that contains two main sections: the first section verifies the authenticity of the order, and the second parses the reply and shows the appropriate response page to the customer (success or failure). After you have created your script, upload it to your Web server as you did for your Web pages.

Verifying the Authenticity of an Order

You can verify the authenticity of an order by using the signature, which uses the transaction data. After you are satisfied with the authenticity of the data, you can parse the rest of the reply.

- 1 Choose a secure (HTTPS) and appropriate URL for CyberSource to send the order information.
- 2 Add the `verifyTransactionSignature` function:

This function verifies that the contents of the reply fields were not altered during transmission from your Web site to CyberSource. You receive a boolean response that you need to parse.

- 3 Write a script to capture the order information and store it on your system.
- 4 Upload the program to your Web server as you did for your Web pages.

Sample Script

This receipt page is an example of the data that an order can contain the signatures that you need to verify. You need to write a script to extract all of the fields from the request parameters and insert them into your database.

```
<%@ page contentType="text/html; charset=UTF-8" language="java" import="java.util.*" %>
<%@ include file="HOP.jsp"%>
<html><body>
  <h1>Custom Decline/Receipt Page</h1>
  <table border="1">
    <tr>
      <th>Field Name</th>
      <th>Field Value</th>
    </tr>
    <%
      HashMap map = new HashMap();
      Enumeration myEnum = request.getParameterNames();
      while(myEnum.hasMoreElements()){
        String paramName = (String)myEnum.nextElement();
        String paramValue = request.getParameter(paramName);
        out.print("<tr><td>" + paramName + "</td><td>" + paramValue + "</td></tr>");
        map.put(paramName, paramValue);
      }
      out.print("<tr><td>VerifyTransactionSignature()</td><td>" +
        verifyTransactionSignature(map) + "</td></tr>");
    %>
  </table>
</body></html>
```

Appendix E

Creating Web Pages with ColdFusion

With a simple HTML page, you can describe your product to your customer, but the customer cannot purchase your product. By adding a Buy Now button to the page, your customer can purchase your product through the Hosted Order Page.

When the customer clicks the button, the Hosted Order Page appears in the customer's browser. After the customer completes and submits the form, the order information is sent to CyberSource to be processed.

For examples of the security, request, and receipt scripts for this project, download [HOP_CFM.zip](#).

Checkout Page

These instructions describe how to create a checkout page for your Web store.

- 1 Save a text file with the file extension `.cfm`.
- 2 Start creating your page with this line.

If necessary, replace `HOP.cfm` with the name of your script.

```
<cfinclude template = "HOP.cfm">
```

- 3 Describe your product, for example:

```
<h1>sample_product_name</h1>
<p>Here, you enter the description of your product.</p>
```

Below the description of the product, start constructing the form that will contain all the information to process the order: customer and payment information, type of transaction, and any additional information that you want to collect about the customer or the order.

- 4 Start the order form with this line, which will send the order information to the correct CyberSource location for the Hosted Order Page:

```
<form action="https://orderpagetest.ic3.com/hop/ProcessOrder.do"
method="POST">
```

5 Add the signature method, replacing the sample values with your own:

`insertSignature3` requires the amount, the currency, and the transaction type:

```
<cfscript>
writeOutput(#insertSignature3("10.00", "usd", "sale")#);
</cfscript>
```

- Amount: In this example, **10.00** is the sample cost of an order.
- Currency: If you do not include the currency, the default is USD. For more information on the currency, see [“Payment Information Fields”](#) on page 30.
- Transaction type: This option enables CyberSource to verify that no fraudulent customer intercepts and modifies the transaction type when you transmit the transaction to CyberSource. In this example, the type of transaction is `sale`. The options are as follows: `authorization` or `sale`.

6 Optionally, add payment and customer’s billing and shipping information:

```
<input type="hidden" name="billTo_firstName" value="John">
<input type="hidden" name="billTo_lastName" value="Doe">
<input type="hidden" name="billTo_email"
value="jdoe@cybersource.com">
```

7 Insert the button that the customer must click to complete the purchasing process:

```
<input type="submit" value="Buy Now">
```

This finished sample Web page appears as follows:

```
<cfinclude template = "HOP.cfm">

<h1>sample_product_name</h1>
<p>Here, you enter the description of your product.</p>

<form action="https://orderpagetest.ic3.com/hop/ProcessOrder.do" method="POST">
  <cfscript>
    writeOutput(#insertSignature3("10.00", "usd", "sale")#);
  </cfscript>
  <input type="hidden" name="billTo_firstName" value="John">
  <input type="hidden" name="billTo_lastName" value="Doe">
  <input type="hidden" name="billTo_email" value="null@cybersource.com">
  <input type="submit" value="Buy Now">
</form>
```

8 Upload your checkout page and your security script (`HOP.cfm`) to the same directory of your Web server.

You can now use the Hosted Order Page on your Web site. To make sure your customers' orders are successful, test the order page before you use it to accept orders. For more information, see [“Testing the Silent Order POST”](#) on page 47.

Receipt Page

You need to write a script that will perform two main functions: the transaction signature uses the order data to verify the authenticity of the order whereas the second parses the reply and shows the appropriate response page to the customer (success or failure). After you have created your script, upload it to your Web server as you did for your Web pages.

Verifying the Authenticity of an Order

You can verify the authenticity of an order by using the signature, which uses the transaction data. After you are satisfied with the authenticity of the data, you can parse the rest of the reply.

- 1 Choose a secure (HTTPS) and appropriate URL for CyberSource to send the order information.
- 2 Add the `verifyTransactionSignature` function:
This function verifies that the contents of the reply fields were not altered during transmission from your Web site to CyberSource. You receive a boolean response that you need to parse.
- 3 Write a script to capture the order information and store it on your system.
- 4 Upload the program to your Web server as you did for your Web pages.

Sample Script

This example shows a script that prints the complete response.

```
<cfinclude template = "HOP.cfm">
<cfscript>
    str1 = FORM.toString();
    if(!str1.equals("{}")){
        writeOutput("orderNumber: " & #FORM.orderNumber#);
        writeOutput("<br>reasonCode: " & #FORM.reasonCode#);
        writeOutput("<br>requestID: " & #FORM.requestID#);
        if(verifyTransactionSignature(FORM))
            writeOutput("<br>The order signature was validated.");
        else
            writeOutput("<br>The order signature was rejected.");
    }
    else{
        writeOutput("The response form was blank.");
    }
</cfscript>
<cfdump var="#FORM#">
```


Appendix F

Creating Web Pages with ASP.NET

With a simple HTML page, you can describe your product to your customer, but the customer cannot purchase your product. By adding a Buy Now button to the page, your customer can purchase your product through the Hosted Order Page.


When the customer clicks the button, the Hosted Order Page appears in the customer's browser. After the customer completes and submits the form, the order information is sent to CyberSource to be processed.

These instructions apply to C# and Visual Basic.

Checkout Page

You need to create a checkout page for each product that you want to sell.

- 1 Download [HOP_ASPNET.zip](#) and extract the contents on your Web server. This archive contains sample security, request, and receipt scripts.



CSharp	File Folder
VB	File Folder
HOP_ASPNET.sln	3 KB SLN File
HOP_ASPNET.suo	15 KB SUO File

- 2 Replace the security script provided with the script that you generated in the Business Center:

C#	Replace CSharp/HOP.cs with your generated HOP.cs.
Visual Basic	Replace VB/HOP.vb with your generated HOP.vb.

- 3 Open HOP_ASPNET.sln with Visual Studio 2005.
- 4 Proceed as follows:

C# Request.aspx is the default startup page. To run the solution file, press Ctrl + F5.

Visual Basic

1. Right-click the VB project.
2. Select **Set as StartUp Project**.
3. To run the solution file, press Ctrl + F5.

- 5 To send the order information to the correct CyberSource location for the Silent Order POST, change the action of the POST from `orderform.jsp` to `ProcessOrder.do`:

```
<form action="https://orderpagetest.ic3.com/hop/ProcessOrder.do"
method="POST">
```

- 6 Customize the rest of your form as desired.

You can now use the Hosted Order Page on your Web site. To make sure your customers' orders are successful, test the order page before you use it to accept orders. For more information, see "[Testing the Silent Order POST](#)" on page 47.

Receipt Page

You need to write a script that will perform two main functions: the transaction signature uses the order data to verify the authenticity of the order whereas the second parses the reply and shows the appropriate response page to the customer (success or failure). After you have created your script, upload it to your Web server as you did for your Web pages. To use the custom receipt page (`Response.aspx`), enter the URL in the HOP Settings in the Business Center ("[Receipt Page](#)" on page 12).

- 1 Choose a secure (HTTPS) and appropriate URL for CyberSource to send the order information.
- 2 Add the `VerifyTransactionSignature` function:
This function verifies that the contents of the reply fields were not altered during transmission from CyberSource to your server. You receive a true or false value confirming the authenticity of the response data.
- 3 Capture the order information to store it on your system.
You can see sample scripts in the `.zip` file that you downloaded.
- 4 Upload the script to your Web server as you did for your checkout pages.

Appendix G

Sample Reply Data and Receipts

This appendix shows samples of requests and replies.

Reply Data

[Transaction Receipts](#)

Reply Data

[Successful Order](#)

[Order to Review](#)

[Failed Order](#)

When reviewing orders, these fields provide the information that you need to decide whether to accept, review, or reject an order:

- The `decision` field provides the overall decision for the order: accept, review, or reject.
- The `reasonCode` field provides the overall reason code for the order and the recommended action associated with the code. Reason codes are described in “[Reason Codes](#)” on page 43.

You can receive at a URL or email address the same data that you receive in the reply.

Successful Order

This reply shows a successful card transaction. In this case, the customer receives the reply page for a successful order. These elements are highlighted:

- **Factor code U** means that the address cannot be verified.
- **AVS code X** means that the street address and the 9-digit code match.
- **Reason code 100** and the **decision** mean that the order is accepted.
- **transactionSignature** and **signedFields** are used by the verify transaction signature function to verify the contents of the order.

```

merchantID=example
requestID=1117778322620167904543
orderPage_requestToken=q4wQLRVsPcwYXi99FER23fGxk3jRK4
billTo_firstName=John
billTo_lastName=Doe
billTo_street1=123 HAPPY LANE
billTo_city=San Francisco
billTo_state=CA
billTo_country=us
billTo_postalCode=94133
billTo_email=jdoe@example.com
ccAuthReply_authFactorCode=U
ccAuthReply_avsCode=X
ccAuthReply_authorizationCode=888888
ccAuthReply_authorizedDateTime=2005-03-25T191034Z
ccAuthReply_amount=10.00
ccAuthReply_reasonCode=100
reasonCode=100
decision=ACCEPT
decision_publicSignature=aEFRvKdMUCwa5JguMZ8LGfwGcZE=
orderAmount=10.00
orderAmount_publicSignature=VpqNi+pZpwqpGztckBfunTQwLT0=
orderNumber=1111777830935
orderNumber_publicSignature=piWfSvHR1gGgfteZVgjbMfGSe8U=
orderPage_transactionType=authorization
paymentOption=credit
card_cardType=001
card_cardNumber=xxxx
card_expirationMonth=11
card_expirationYear=2008
transactionSignature= APoom70Mk115OaeOeNmSc280Tts=
signedFields=merchantID,requestID,orderPage_requestToken,billTo_firstName,billTo_lastName,
billTo_street1,billTo_city,billTo_state,billTo_country,billTo_postalCode,billTo_email,
ccAuthReply_authFactorCode,ccAuthReply_amount,ccAuthReply_authorizationCode,
ccAuthReply_authorizedDateTime,ccAuthReply_avsCode,ccAuthReply_reasonCode,reasonCode,
decision,decision_publicSignature,orderAmount,orderAmount_publicSignature,orderNumber,
orderNumber_publicSignature,orderPage_transactionType,paymentOption,card_cardType,
card_cardNumber,card_expirationMonth,card_expirationYear

```

With Payer Authentication with an enrolled Visa card (subsequently validated), the reply for a successful authorization includes data similar to this:

```

proofXML=<AuthProof><Time>2007 Jul 26 09:04:53</Time><DSUrl>111.111.36:443/DSMsgServlet<DSUrl>
<VEReqProof><Message id="xfm5_3_0.269"><VEReq><version>1.0.2</version><pan>XXXXXXXXXXXX0771</pan>
<Merchant><acqBIN>123456</acqBIN><merID>8768516</merID></Merchant><Browser><accept>text/xml,
application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;
q=0.5<accept ><userAgent>Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.5) Gecko/
20070713 Firefox/2.0.0.5</userAgent></Browser></VEReq></Message></
VEReqProof><VEResProof><Message id="xfm5_3_0.269"><VERes><version>1.0.2</
version><CH><enrolled>Y</enrolled><acctID> NDAxMjsocTAxMTAwMDc3MQ==</acctID></CH><url>https://
www.littletooth.com/verification/appl/pareq.php?vaa=a</url><protocol>ThreeDSecure</protocol></
VERes></Message></VEResProof></AuthProof>

```

```
xid=8okfADuVn038vn7MOPfIzMwAHBwE=
eci=05
cavv=AAAAAAAAAAAAAAAAAAAAAAAAAAAA=
```

Order to Review

CyberSource directs the customer to the receipt page for accepted orders for all successful orders and for orders declined by Smart Authorization. In the search results page and in the reply, these orders are marked Review. By default, replies that contain these reason codes are marked for review:

- 200: The authorization request was approved by the issuing bank but declined by CyberSource because it did not pass the Address Verification Service (AVS) check.
- 230: The credit card was accepted by the bank but refused by CyberSource because it did not pass the card verification number check. The card verification result is N.
- 520: The authorization request was approved by the issuing bank but declined by CyberSource based on your Smart Authorization settings.

However, if you customized your implementation to choose the page that your customers receive, the reason code(s) that trigger a review for your orders may differ from those that are described above.

Request

In this example, CyberSource receives this transaction information from the your Web site:

Merchant ID	Example
Customer's billing information	
first name	John
last name	Doe
street address 1	123 Happy Lane
city	San Francisco
state	CA
country	US
postal code	94133
email address	jdoe@example.com
Shipping information	
first name	Jane
last name	Doe
street address 1	456 Elm Street
city	Mountain View
state	CA
country	us
postal code	94066

(The rest of the order information follows.)

Reply

Because the billing and shipping addresses do not match, the merchant would receive this reply from CyberSource:

```
merchantID=example
requestID=9996668322620167904543
orderPage_requestToken=BlIBFvErZ7cymqd0hFF177R2TNSy8UhiZ
ccAuthReply_authFactorCode=J^U
ccAuthReply_amount=10.00
ccAuthReply_authorizationCode=888888
ccAuthReply_authorizedDateTime=2005-03-25T191034Z
ccAuthReply_avsCode=Y
ccAuthReply_reasonCode=520
reasonCode=520
decision=REVIEW
decision_publicSignature=aEFRvKdMUCwa5JguMZ8LGfwGczE=
orderAmount=10.00
orderAmount_publicSignature=VpqNi+pZpwqpGztckBfunTQwLT0=
orderNumber=1111777830935
orderNumber_publicSignature=piWfSvHR1gGgfteZVgjBMfGSe8U=
orderPage_transactionType=authorization
paymentOption=credit
card_cardType=001
card_cardNumber=xxxx
card_expirationMonth=11
card_expirationYear=2008
transactionSignature=APooM70Mk1150AeOeNMsc280Tts=
signedFields=<comma-separated list of all the data fields in the order>
```

- Factor codes U and J mean that the billing and shipping addresses do not match.
- Reason code 520 means that the order failed the Smart Authorization tests.
- The decision field indicates that you should review the order.

Failed Order

Most transactions that cannot be accepted are rejected, and the customer receives the reply page for a failed order. However, if you customized your implementation to choose the page that your customers receive, the reason code(s) that trigger a rejection for your orders may differ from the CyberSource defaults.

The samples below show excerpts for transactions in which the credit card is either declined (203) or missing (102).

Invalid field reply

```
decision=REJECT
reasonCode=203
```

Missing field reply

```
decision=REJECT
reasonCode=102
MissingField0=card_expirationYear
```

Transaction Receipts

The two following email receipts, customer and merchant, are formatted as memos. You do not specify the content, except for the header and the footer.

Customer's Receipt

The customer's receipt shows the payment information with only the last four digits of the credit card number.

From: merchant@sample.com [<mailto:merchant@sample.com>]

Sent: Wednesday, May 25, 2005 12:42 PM

To: Doe, Jane

Subject: Purchase Confirmation

Purchase Description

Widget

Payment Details

Order Number: 12345

Subtotal: 30.00

Tax: 0.00

Total: 30.00

Order Details

Payment Option: card

Card Description: Visa #####1111

Customer ID:

Bill Address:

123 Anystreet

Mountain View, California 94043

If you have questions about your order, please contact service@example.com.

Merchant's Receipt

Your receipt contains the payment information, the return codes, and all the information relevant to the order. This receipt shows an enrolled card transaction with Visa and Payer Authentication. Because the card is not enrolled, validation does not occur, and the enrollment check is followed by the authorization. With this receipt, you can fulfill the order without having to search the Business Center to see the details.

From: Cybersource [<mailto:sample@cybersource.com>]

Sent: Friday, June 10, 2005 3:13 PM

To: Merchant, Sample

Subject: Purchase Confirmation

Purchase Description

Widget

Payment Details

Transaction Receipts

Order Number: 12345
Subtotal: 30.00
Tax: 0.00
Total: 30.00

Return Codes

Result Code: Request was processed successfully.
Auth Code: 888888
AVS Code: X
CVN Code:

Transaction Details

Transaction Type: sale
Transaction Source: Silent Order POST
Payment Option: card
Card Description: Visa #####1111
Request Token: H7v9JeVSuzD3YtvP6hIPeU3wN0dvJy16RC9GRH8Y

Customer Information

Customer ID:

Bill Address:
123 Anystreet
Mountain View, California 94043

Payer Authentication

```
proofXML=<AuthProof><Time>2007 Jul 26 09:04:53</Time><DSUrl>https:111. 111.111.36:443/
DSMsgServlet</DSUrl><VEReqProof><Message id="xfm5_3_0.269"><VEReq><version>1.0.2</version><pan>
XXXXXXXXXXXX0771</pan><Merchant><acqBIN>123456</acqBIN><merID>876851</merID></Merchant><Browser>
<accept>text/xml,application/xml,application/xhtml+xml,text/html;q=0.9, text/plain;q=0.8,image/
png,*/*;q=0.5</accept><userAgent>Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.5)
Gecko/20070713 Firefox/2.0.0.5</userAgent></Browser></VEReq></Message></VEReqProof><VEResProof>
<Message id="xfm5_3_0.269"><VERes><version>1.0.2</version><CH><enrolled>Y</enrolled><acctID>
NDAxMjsocTAxMTAwMdc3MQ==</acctID></CH><url>https://www.littletooth.com/verification/appl/pareq.
php?vaa=a</url><protocol>ThreeDSecure</protocol></VERes></Message></VEResProof></AuthProof>
eci=06
```

Merchant Defined Data

Field 1: Ref No.: 12345
Field 2: Brand: Mackie
Field 3: Express shipping

If you have questions about your order, please contact service@example.com.

Appendix H

Description of Return Codes

This appendix describes result codes for the Address Verification Service (AVS), card verification numbers (CVN), and the factor codes returned by Smart Authorization. You can see these results in the Transaction Detail page. For more information on searching for orders, see the [Business Center User's Guide](#).

AVS Codes

[Card Verification Codes](#)

[Smart Authorization Factor Codes](#)

AVS Codes

When you request a credit card authorization, the customer's issuing bank may use the Address Verification Service (AVS) to confirm that your customer has provided the correct billing address. If the customer provides incorrect information, the transaction might be fraudulent.

AVS is requested for the following payment processors and card types:

Payment Processors	Credit Card Types
Concord EFS	Visa, MasterCard, American Express, Discover, Diners Club
First Data Merchant Services - Nashville	Visa, MasterCard, American Express, Discover
First Data Merchant Services - South	Visa, MasterCard, American Express, Discover, Diners Club
Paymentech - New Hampshire	<ul style="list-style-type: none">• Visa (billing country must be U.S., Canada, or Great Britain)• American Express (billing country must be U.S. or Canada)• MasterCard, Discover, Diners Club (billing country must be U.S.)
Vital Processing Services	Visa, MasterCard, American Express, Diners Club (billing country must be U.S.)

The following table describes each AVS code.

Table 13 Address Verification Service Codes

Code	Summary	Description
A	Partial match	Street address matches, but 5- and 9-digit postal codes do not match.
B	Partial match	Street address matches, but postal code not verified. Returned only for non U.S.-issued Visa cards.
C	No match	Street address and postal code do not match. Returned only for non U.S.-issued Visa cards.
D	Match	Street address and postal code match. Returned only for non U.S.-issued Visa cards.
E	Invalid	AVS data is invalid or AVS is not allowed for this card type.
F	Partial match	Card member's name does not match, but postal code matches. Returned only for the American Express card type.
G	Not supported	Non-U.S. issuing bank does not support AVS.
H	Partial match	Card member's name does not match. Street address and postal code match. Returned only for the American Express card type.
I	No match	Address not verified. Returned only for non U.S.-issued Visa cards.
K	Partial match	Card member's name matches but billing address and billing postal code do not match. Returned only for Vital for the American Express card type.
L	Partial match	Card member's name and billing postal code match, but billing address does not match. Returned only for Vital for the American Express card type.
N	No match	Street address and postal code do not match. <i>or</i> Card member's name, street address and postal code do not match. Returned only for the American Express card type.
O	Partial match	Card member's name and billing address match, but billing postal code does not match. Returned only for Vital for the American Express card type.
P	Partial match	Postal code matches, but street address not verified. Returned only for non-U.S.-issued Visa cards.
R	System unavailable	System unavailable.

Table 13 Address Verification Service Codes (Continued)

Code	Summary	Description
S	Not supported	U.S.-issuing bank does not support AVS.
T	Partial match	Card member's name does not match, but street address matches. Returned only for the American Express card type.
U	System unavailable	Address information unavailable. Returned if non-U.S. AVS is not available or if the AVS in a U.S. bank is not functioning properly.
V	Partial match	Card member's name, billing address, and billing postal code match. Returned only for Vital for the American Express card type.
W	Partial match	Street address does not match, but 9-digit postal code matches.
X	Match	Exact match. Street address and 9-digit postal code match.
Y	Match	Exact match. Street address and 5-digit postal code match.
Z	Partial Match	Street address does not match, but 5-digit postal code matches.
1	Not supported	CyberSource AVS code. AVS is not supported for this processor or card type.
2	Invalid	CyberSource AVS code. The processor returned an unrecognized value for the AVS response.

Card Verification Codes

When you request a credit card authorization, you can include the customer's card verification code, a three- or four-digit number printed on the back or front of credit cards. If the customer cannot provide the correct number, the transaction may be fraudulent.

The following payment processors support card verification codes for Visa and MasterCard:

FDMS Nashville Paymentech New Hampshire
 FDMS South Vital

The following table describes each card verification result code.

Table 14 Card Verification Codes

Code	Description
D	Transaction determined suspicious by issuing bank.
I	Card verification number failed processor's data validation check.
M	Card verification number matched.
N	Card verification number not matched.
P	Card verification number not processed by processor for unspecified reason.
S	Card verification number is on the card but was not included in the request.
U	Card verification is not supported by the issuing bank.
X	Card verification is not supported by the card association.
1	Card verification is not supported for this processor or card type.
2	Unrecognized result code returned by processor for card verification response.
3	No result code returned by processor.

Smart Authorization Factor Codes

If you use Smart Authorization to evaluate the risk of your orders, you receive factor codes that show which parts of an order appeared to be risky. You receive factor codes for any order that shows risk, even if Smart Authorization does not decline the order.

The following table describes each factor code that Smart Authorization can return. To use Smart Authorization, define your settings in the Business Center. For detailed information about how to choose which factor codes result in a Smart Authorization decline, see the [Business Center User's Guide](#).

Table 15 Smart Authorization Codes

Code	Description
J	Billing and shipping address do not match.
M	Cost of the order exceeds the maximum transaction amount.
N	Nonsensical input in the customer name or address fields.
O	Obscenities in the order form.
U	Unverifiable billing or shipping address.
X	Order does not comply with the USA PATRIOT Act.

Index

A

ACCEPT decision [36](#)
action attribute [47](#)
Address Verification Service (AVS) [81](#)
AmeriNet, date of birth for checks [23](#), [28](#)
Apache Web server [65](#)
ASCII mode
 uploading Perl script [63](#)
 uploading script [19](#), [61](#)
Authenticity of order, verifying
 ASP [57](#)
 ASP.NET [74](#)
 ColdFusion [71](#)
 JSP [67](#)
 Perl [62](#)
 PHP [53](#)
Authorization [2](#)
AVS [81](#)

B

Barclays, MasterCard SecureCode [7](#)
Billing address [28](#)
Binary mode, uploading Perl script [63](#)
Binary mode, uploading script [19](#), [61](#)

C

C# [73](#)
Capturing authorizations [2](#)
Card verification number (CVN) [23](#), [83](#)
cgi-bin directory [61](#)
Chargeback protection with Payer
 Authentication [7](#)
Checks

 Silent Order POST requirements [23](#)
 state return fees [24](#)
Credit card numbers, test [47](#)
Credit cards supported [22](#)
Currency [32](#)
Custom fields [34](#)
Customer information fields [28](#)
CVN [23](#), [83](#)

D

decision field [36](#), [41](#)
Decisions [36](#)
Decline page [14](#)
Diners Club-MasterCard alliance [22](#)

E

Email receipts [15](#), [79](#)
ERROR decision [36](#)
Errors [36](#)
Example
 failed order [78](#)
 order to review [77](#)
 successful order [75](#)

F

Factor codes [84](#)
Failed order, example [78](#)
Fields, customizing
 customer information [28](#)
 order information [26](#), [34](#)
 payment information [30](#)

I

Invalid fields [36](#)
Invalid fields, reason code [43](#)
Issue number (Maestro (UK Domestic) & Solo) [22](#)

L

Line items [21](#)

M

Maestro (UK Domestic) and Solo [22](#)
MasterCard SecureCode [1](#), [4](#), [11](#)
MasterCard-Diners Club alliance [22](#)
Merchant reference number [37](#)
Merchant-defined fields [34](#)
Missing fields [36](#)
 reason code [43](#)

O

Order information
 fields [26](#)
 receiving [35](#)
Order number [37](#)
Order tracking [37](#)

P

Parsing a reply
 ASP [57](#)
 ASP.NET [74](#)
 ColdFusion [71](#)
 JSP [67](#)
 Perl [62](#)
 PHP [53](#)
Payer Authentication [4](#), [11](#)
 finding data in Business Center [7](#)
 reports [7](#)
 result fields [4](#)
 storing and retrieving data [7](#)
Payment information [30](#)
Payment method field [32](#), [42](#)
Processing checks [1](#)
publicSignature fields [41](#)

R

Reason codes [36](#), [43](#)
Receipt page [12](#)
Receipt page, redirecting customer [12](#),
 [35](#), [38](#)
Receipts
 configuring [32](#)
 customer & merchant [15](#)
Reference number for tracking orders [37](#)
REJECT decision [36](#)
Reply information to the receipt page,
 mapping [38](#)
Reply, parsing
 ASP [57](#)
 ASP.NET [74](#)
 ColdFusion [71](#)
 JSP [67](#)
 Perl [62](#)
 PHP [53](#)
Request ID [37](#)
Request token [37](#)
REVIEW decision [36](#)

S

Security script [17](#)
Settings page, URLs for receipt pages [12](#)
Settling transactions, time limit [7](#)
Shipping address [28](#)
Shopping cart [21](#)
Signature [17](#)
Signature, authenticating orders
 ASP [57](#)
 ASP.NET [74](#)
 ColdFusion [71](#)
 JSP [67](#)
 Perl [62](#)
 PHP [53](#)
Silent Order POST
 configuring [11](#)
 testing [47](#)
Smart Authorization [84](#)
Smart tags [13](#)
Soft decline, example [77](#)
Start date (Maestro (UK Domestic) & Solo) [22](#)
Successful order, example [75](#)

T

Tax amount, customizing [28](#)
Testing Silent Order POST [47](#)
Tracking number [37](#)

U

URL for Silent Order POST [22](#)
URLs for receipt pages, configuring [12](#)

V

Validating orders
 ASP [57](#)
 ASP.NET [74](#)
 ColdFusion [71](#)
 JSP [67](#)
 Perl [62](#)
 PHP [53](#)
Verified by Visa [1](#), [4](#), [11](#)
Visual Basic [73](#)

W

Web pages, creating
 ASP pages [55](#)
 ASP.NET [73](#)
 ColdFusion [69](#)
 JSP pages [65](#)
 Perl CGI pages [59](#)
 PHP pages [51](#)

