

CyberSource Electronic Check Services

Implementation Guide

Simple Order API
SCMP API

May 2010

CyberSource Contact Information

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).

For support information about any CyberSource Service, visit the Support Center at <http://www.cybersource.com/support>.

For Customer Support, see the [CyberSource Knowledgebase](#).

Copyright

© 2010 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 in 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 Unpublished rights reserved under the copyright laws of the United States.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

<u>Documentation Changes and Enhancements</u>	<u>vii</u>
Chapter 1	
<u>Introduction to the Electronic Check Services</u>	<u>1</u>
<u>Payment Processors</u>	<u>1</u>
<u>Changes to Your Web Site for Electronic Checks</u>	<u>2</u>
<u>Determining if a Check Has Cleared</u>	<u>3</u>
<u>Order Tracking</u>	<u>4</u>
<u>Reconciliation IDs and Transaction Reference Numbers</u>	<u>4</u>
<u>Request IDs</u>	<u>4</u>
<u>Check Reference Numbers</u>	<u>5</u>
<u>Processor Transaction Identifiers</u>	<u>6</u>
Chapter 2	
<u>Electronic Check Processing with the Simple Order API</u>	<u>7</u>
<u>Electronic Check Debits</u>	<u>7</u>
<u>Requesting a Debit with the Simple Order API</u>	<u>7</u>
<u>Handling Customer Account Information</u>	<u>8</u>
<u>Verification and Validation</u>	<u>12</u>
<u>Optional Features for Debits</u>	<u>17</u>
<u>Requirements for the Debit Request Fields</u>	<u>17</u>
<u>Electronic Check Credits</u>	<u>21</u>
<u>Requesting a Credit with the Simple Order API</u>	<u>21</u>
<u>Follow-on Credits and Stand-Alone Credits</u>	<u>22</u>
<u>Deciding What Kind of Credit to Request</u>	<u>22</u>
<u>ACH Verification</u>	<u>23</u>
<u>Notifications of Change (NOC)</u>	<u>25</u>
<u>Optional Features for Credits</u>	<u>25</u>
<u>Requirements for the Credit Request Fields</u>	<u>26</u>
<u>Authenticate</u>	<u>29</u>
<u>VOIDs</u>	<u>30</u>
<u>Requesting a Void with the Simple Order API</u>	<u>30</u>
<u>Requirements for the Void Request Fields</u>	<u>32</u>

Chapter 3

Electronic Check Processing with the SCMP API **33**

Electronic Check Debits 33

Requesting a Debit with the SCMP API 33

Handling Customer Account Information 34

Verification and Validation 38

Optional Features for Debits 42

Requirements for the Debit Request Fields 43

Electronic Check Credits 46

Requesting a Credit with the SCMP API 46

Follow-on Credits and Stand-Alone Credits 47

Deciding What Kind of Credit to Request 47

ACH Verification 48

Notifications of Change (NOC) 50

Optional Features for Credits 50

Requirements for the Credit Request Fields 51

Authenticate 54

VOIDs 55

Requesting a Void with the SCMP API 55

Requirements for the Void Request Fields 56

Chapter 4

Optional Features for the Simple Order API **57**

Corporate Checks 57

Deferred and Partial Payments 58

Chase Paymentech Solutions 58

TeleCheck 59

Encoded Account Numbers 59

Merchant Descriptors 60

Multiple Partial Credits 61

Non-Sufficient Funds (NSF) Service 61

Payment Tokenization 62

Recurring Billing 63

Settlement Delivery Methods 64

Chapter 5	
<u>Optional Features for the SCMP API</u>	<u>65</u>
<u>Corporate Checks</u>	<u>65</u>
<u>Deferred and Partial Payments</u>	<u>66</u>
<u>Chase Paymentech Solutions</u>	<u>66</u>
<u>TeleCheck</u>	<u>67</u>
<u>Encoded Account Numbers</u>	<u>67</u>
<u>Merchant Descriptors</u>	<u>68</u>
<u>Multiple Partial Credits</u>	<u>69</u>
<u>Non-Sufficient Funds (NSF) Service</u>	<u>69</u>
<u>Payment Tokenization</u>	<u>70</u>
<u>Recurring Billing</u>	<u>71</u>
<u>Settlement Delivery Methods</u>	<u>72</u>
Chapter 6	
<u>Testing Electronic Check Services</u>	<u>73</u>
<u>Requirements for Testing</u>	<u>73</u>
<u>Testing Chase Paymentech Solutions Transactions</u>	<u>74</u>
<u>Successful Transactions</u>	<u>74</u>
<u>Testing Chase Paymentech Solutions Declines</u>	<u>75</u>
<u>Testing CyberSource ACH Service Transactions</u>	<u>75</u>
<u>Testing RBS WorldPay Atlanta</u>	<u>76</u>
<u>Going Live</u>	<u>76</u>
Appendix A	
<u>Fields for the Simple Order API</u>	<u>79</u>
<u>Formatting Restrictions</u>	<u>79</u>
<u>Data Type Definitions</u>	<u>79</u>
<u>Request Fields</u>	<u>80</u>
<u>Reply Fields</u>	<u>88</u>
Appendix B	
<u>Fields for the SCMP API</u>	<u>99</u>
<u>Formatting Restrictions</u>	<u>99</u>
<u>Data Type Definitions</u>	<u>99</u>
<u>Request-Level Fields</u>	<u>100</u>
<u>Offer-Level Fields</u>	<u>107</u>
<u>Reply Fields</u>	<u>108</u>
Appendix C	
<u>Examples for the Simple Order API</u>	<u>119</u>
<u>Name-Value Pair Examples</u>	<u>119</u>
<u>XML Examples</u>	<u>121</u>

Appendix D	
<u>Examples for the SCMP API</u>	<u>123</u>
Appendix E	
<u>Reason Codes for the Simple Order API</u>	<u>125</u>
Appendix F	
<u>Reply Flags for the SCMP API</u>	<u>129</u>
Appendix G	
<u>Product Codes</u>	<u>131</u>
Appendix H	
<u>NOC Codes</u>	<u>133</u>
Appendix I	
<u>Check Point Summary Codes</u>	<u>135</u>
<u>Primary Result Codes</u>	<u>136</u>
<u>Address Result Codes</u>	<u>137</u>
<u>Phone Codes</u>	<u>139</u>
<u>Address Type Codes</u>	<u>140</u>
<u>Change of Address Codes</u>	<u>141</u>
<u>Social Security Number Codes</u>	<u>142</u>
<u>Address Unit Mismatch Codes</u>	<u>143</u>
<u>Phone Unit Mismatch Codes</u>	<u>143</u>
<u>Driver's License Result Codes</u>	<u>144</u>
<u>Date of Birth Match Codes</u>	<u>144</u>
<u>High Risk Address Codes</u>	<u>145</u>
<u>High Risk Phone Codes</u>	<u>145</u>
<u>OFAC Validation Results Codes</u>	<u>146</u>
<u>Address Residential Match Codes</u>	<u>146</u>
<u>Address Business Match Codes</u>	<u>147</u>
<u>Phone Number Residential Match Codes</u>	<u>147</u>
<u>Phone Number Business Match Codes</u>	<u>147</u>
Appendix J	
<u>Fraud Shield Indicator Codes</u>	<u>149</u>
<u>Fraud Shield Indicator Codes</u>	<u>149</u>
<u>Index</u>	<u>151</u>

Documentation Changes and Enhancements

The following table lists changes made in recent releases of this document.

Month of Release	Changes
May 2010	Added the link-to-request fields: • Simple Order API—See linkToRequest in Table 17 on page 80. • SCMP API—See link_to_request in Table 19 on page 100.
February 2010	Updated the information about subscriptions and customer profiles: • Simple Order API—See “Payment Tokenization” on page 62 and “Recurring Billing” on page 63. • SCMP API—See “Payment Tokenization” on page 70 and “Recurring Billing” on page 71.
January 2010	Added information about a new processor: RBS WorldPay Atlanta. See “Payment Processors” on page 1. • For information about which services are supported: - Chapter 2, “Electronic Check Processing with the Simple Order API,” on page 7 - Chapter 3, “Electronic Check Processing with the SCMP API,” on page 33 • For information about which optional features are supported: - Chapter 4, “Optional Features for the Simple Order API,” on page 57 - Chapter 5, “Optional Features for the SCMP API,” on page 65 • For information about which other basic features and functionality are supported: - “Determining if a Check Has Cleared” on page 3 - Chapter 6, “Testing Electronic Check Services,” on page 73
September 2009	The SEC code is supported as an optional field for Chase Paymentech Solutions for PIN-less debits: • Simple Order API—See the check_secCode field in Table 7 on page 17. • SCMP API—See the ecp_sec_code field in Table 11 on page 43. Removed the “Basics About the API” chapters. Most of this information has been moved to Getting Started with CyberSource Advanced. Information specific to the electronic check services has been moved to the “Introduction” chapter: “Order Tracking” on page 4.

Chapter 1

Introduction to the Electronic Check Services

Topics:

[Payment Processors](#)

[Changes to Your Web Site for Electronic Checks](#)

[Determining if a Check Has Cleared](#)

[Order Tracking](#)

Payment Processors

To use the Electronic Check Services, you must sign up with one of these processors:

- Chase Paymentech Solutions
- CyberSource ACH Service
- RBS WorldPay Atlanta
- TeleCheck

These processors support transactions in U.S. dollars for U.S. bank accounts. Chase Paymentech Solutions also supports Canadian dollars and Canadian bank accounts.

Your processor will provide you with unique identification numbers for your account. You must provide these identification numbers to Customer Support.

Chase Paymentech Solutions acts as both a processor and a merchant bank, which is a bank that offers accounts for businesses that accept credit card or electronic check payments. If you choose Chase Paymentech Solutions as your processor, you must also open a check-enabled merchant bank account with them. However, you can set up the account to deposit the electronic check funds you receive directly into your primary account at another bank.

If CyberSource ACH Service is your processor, you need to have a treasury relationship with one of the following originating depository financial institutions (ODFIs):

- Bank of America
- Wells Fargo

If TeleCheck is your processor, you do not need to open a check-enabled merchant bank account. TeleCheck can deposit funds directly into your existing bank account.

Changes to Your Web Site for Electronic Checks

To process electronic checks, you need to add the following items to your Web site:

- 1 On your Web site, add a link to the table of current state returned check fees:
http://www.achex.com/html/NSF_pop.jsp
Because this table is updated regularly, CyberSource recommends that you link directly to it. You can display the state fees table in a pop-up window, a full browser window, or directly on the checkout page.
- 2 At the end of the checkout process on your Web site, display a consent statement for the check authorization that your customer *must accept* before submitting the order. The authorization consent statement must:
 - Be readily identifiable as an authorization.
 - Clearly and conspicuously state its terms including the transaction amount and the effective date of the transfer.
 - Include the routing number and bank account number to be debited.

Figure 1 Sample Language for a Payment Authorization

Bank ABA Routing Number: [ABA routing number for the customer's bank]
Bank Account Type: [checking, business checking, savings]
Bank Account Number: [customer's bank account number]
By clicking the button below, I authorize [merchant's name] to debit my [bank account type] account on [month, day, year] for the amount of [\$X.XX] for [description of the transaction]. If the entire order is not available at one time, I authorize partial debits from my account to take place upon each shipment of partial goods, not to exceed the total authorized amount. If any of my payments are returned unpaid, I understand that I may be charged a returned item fee.

Determining if a Check Has Cleared

You can use the Payment Events Report to keep track of your electronic check debits and identify problems that occur with funds transfers. The report is available daily and includes information from the past 24 hours that the processor has provided about your transactions, such as the clearing of a check or the denial of a check due to insufficient funds. The following table describes the event types that indicate that a check has likely cleared. For a list of all the event values, see the description of the Payment Events Report in the [Reporting Developer's Guide](#).

Important Due to the nature of electronic check processing, CyberSource does not guarantee that a check has truly cleared.

Table 1 Determining if a Check Has Cleared

Processor	Determining if a Check Has Cleared
Chase Paymentech Solutions	The Payment Events Report will not indicate that a check has cleared; it will show only problems that occur with funds transfers. Important If you use Chase Paymentech Solutions, you must contact them and request that they send their electronic check declines file to CyberSource. Then contact CyberSource Customer Support with your Chase Paymentech Solutions MA number so that your CyberSource account can be configured appropriately.
CyberSource ACH Service	The event type listed in the Payment Events Report is "Payment" when the ODFI receives a debit request. • <i>Bank of America ACH:</i> To see an event type of "Completed" when the check clears, contact Customer Support to have your account configured. CyberSource does not recommend using this event type because it is not a reliable indication that a check has cleared. • <i>Wells Fargo ACH:</i> After receiving the debit request, the ODFI waits for three days and if the bank does not inform them about any problems with the funds transfer, they consider the check cleared. The event type listed in the report is "Completed" when the check clears. CyberSource does not guarantee that the check has truly cleared.
TeleCheck	The event type listed in the Payment Events Report is "Payment" when a check clears.

Order Tracking

For general information about order tracking, see [Getting Started with CyberSource Advanced](#).

Reconciliation IDs and Transaction Reference Numbers

The following table lists the field names for the electronic check reconciliation IDs and transaction reference numbers in the reply messages.

Table 2 Reconciliation IDs and Transaction Reference Numbers in Replies

Service	Field Names
Electronic check debit	- Simple Order API: ecDebitReply_reconciliationID - SCMP API: ecp_debit_ref_no
Electronic check credit	- Simple Order API: ecCreditReply_reconciliationID - SCMP API: ecp_credit_ref_no

Request IDs

For all electronic check services, the request ID is returned in the reply message:

- **requestID** for the Simple Order API
- **request_id** for the SCMP API

The following table lists the field names for the electronic check request IDs in the request messages.

Table 3 Request IDs in Requests

Service	Field Names
Electronic check credit	- Simple Order API: ecCreditService_debitRequestID - SCMP API: ecp_debit_request_id
Void	- Simple Order API: voidService_voidRequestID - SCMP API: void_request_id

Check Reference Numbers

The check reference number is a value you can send in a request to track transactions through to the processor for reconciliation. If you do not include this field in your request, CyberSource generates a unique value for you and returns it in the reply message.

TeleCheck uses the check reference number to track all the transactions for a given order in their system. For example, you can use the check reference number to link a deferred or partial payment to the original debit. To accomplish this, set the check reference number request field for the debit service to the value of the check reference number reply field from the original debit's reply and include the request field in your request for the deferred or partial payment.

The following table lists the field names for the check reference numbers in the request and reply messages.

Table 4 Check Reference Numbers in Requests and Replies

Service	Field Names
Electronic check debit request	- Simple Order API: ecDebitService_referenceNumber - SCMP API: ecp_ref_no
Electronic check debit reply	- Simple Order API: ecDebitReply_reconciliationID - SCMP API: ecp_debit_ref_no
Electronic check credit request	- Simple Order API: ecCreditService_referenceNumber - SCMP API: ecp_ref_no
Electronic check credit reply	- Simple Order API: ecCreditReply_reconciliationID - SCMP API: ecp_credit_ref_no

Note The reply fields for the check reference numbers are the same as the reconciliation ID and transaction reference number fields.

Processor Transaction Identifiers

Note This value is not supported for Chase Paymentech Solutions.

The processor transaction identifier is a value assigned by the processor that you can use for reconciliation. The following table lists the field names for the processor transaction identifiers in the reply messages.

Table 5 Processor Transaction Identifiers in Replies

Service	Field Names
Electronic check debit	- Simple Order API: ecDebitReply_processorTransactionID - SCMP API: ecp_debit_processor_trans_id
Electronic check credit	- Simple Order API: ecCreditReply_processorTransactionID - SCMP API: ecp_credit_processor_trans_id

Chapter 2

Electronic Check Processing with the Simple Order API

Topics:

[Electronic Check Debits](#)
[Electronic Check Credits](#)
[Authenticate](#)
[Voids](#)

Electronic Check Debits

Topics:

[Requesting a Debit with the Simple Order API](#)
[Handling Customer Account Information](#)
[Verification and Validation](#)
[Optional Features for Debits](#)
[Requirements for the Debit Request Fields](#)

Requesting a Debit with the Simple Order API

To request an electronic check debit with the Simple Order API, set the **ecDebitService_run** field to **true**. When you request a debit, do not request any of the following services at the same time:

- Any credit card services: **ccAuthService**, **ccAuthReversalService**, **ccCaptureService**, **ccCreditService**
- Electronic check credit: **ecCreditService**
- Any Global Payment services: **bankTransferService**, **bankTransferRefundService**, **directDebitService**, **directDebitRefundService**, **bankTransferRealTimeService**
- PayPal® payment or credit: **paypalPaymentService**, **paypalCreditService**
- Advanced Fraud Screen: **afsService**
- Risk update: **riskUpdateService**

See the [Credit Card Services Implementation Guide](#), the [Global Payment Service Developer's Guide](#), the [PayPal Services Implementation Guide](#), and the [Decision Manager Developer's Guide](#) for information about these services.

Handling Customer Account Information

Topics:

[Merchant-Provided Data](#)

[Processor-Managed Data](#)

[Notifications of Change \(NOC\)](#)

Merchant-Provided Data

Applicable services:

- Debit

Supported processors:

- Chase Paymentech Solutions
- CyberSource ACH Service
- RBS WorldPay Atlanta
- TeleCheck

Merchant-provided data handling requires you to collect the customer's account information and provide it in your service request. The required fields are:

- **check_accountNumber**
- **check_accountType**
- **check_bankTransitNumber**

Using this method, you must modify your Web site to collect the pertinent account information. You must also retain the account information for future transactions, such as credits.

Important If you send the customer's account number in your request, then CyberSource assumes you are using the merchant-provided data method.

Customers might not know how to use their printed checks to find the bank routing number and the bank account number. Consider using a graphic like the following example on your Web site to help customers locate their information.

Figure 2 Sample Check Showing Routing Number and Account Number

NAME
ADDRESS
CITY, STATE ZIP

0123
01-2345/6789

DATE

PAY TO THE
ORDER OF

\$

BANK NAME
ADDRESS
CITY, STATE ZIP

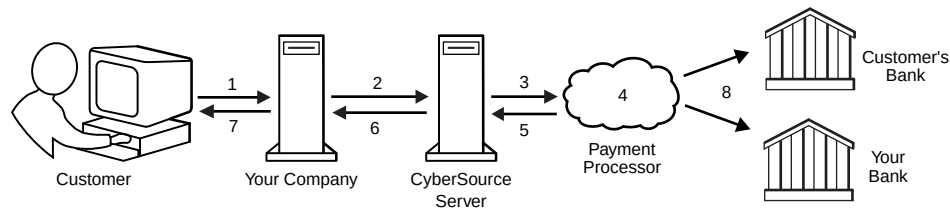
DOLLARS

FOR

⑆0123456789⑆ 01234567890123⑆ 0123

Routing Number Account Number

The following steps take place when you request a debit using merchant-provided data:

Figure 3 Processing an Electronic Check Debit with Merchant-Provided Data

- 1 Your customer places an order.
- 2 You request an electronic check debit.
In your request, you provide the customer's account information.
- 3 CyberSource sends the customer's account information and other information about the transaction to the check processor.
- 4 The payment processor validates the information and performs basic fraud screening.
The processor does not contact the customer's bank to verify the existence of the customer's account; it only makes sure that the information provided by the customer is reasonable and that the account is not a known source of fraud.
Depending on which processor you use, if there are problems with the account that prevent the transaction from being completed, the processor might charge you a returned check fee.
- 5 The payment processor sends a reply to CyberSource indicating whether the debit will be processed.
- 6 CyberSource sends a reply to you.
- 7 You display an appropriate message to your customer.
- 8 The processor sends the request for clearing.

Processor-Managed Data

Applicable services:

- Debit

Supported processors:

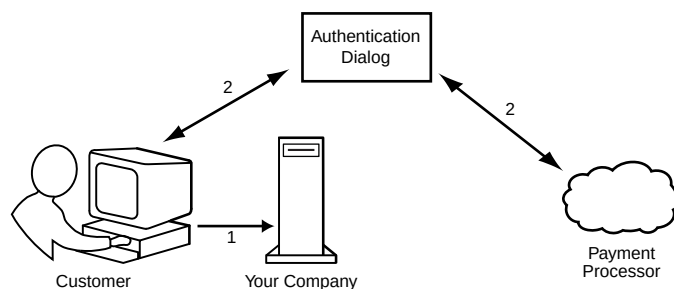
- TeleCheck

Processor-managed data handling does not require you to collect the customer's account information. Instead, the customer provides credentials directly to the processor and these credentials are authenticated before you request a debit.

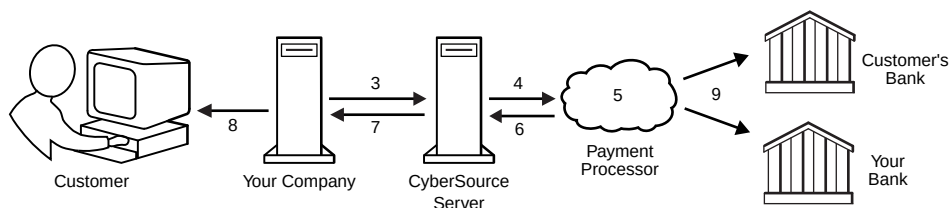
You must adjust your checkout process to route the customer's browser to the processor's authentication interface. You can do this through different methods, such as a popup window or simple response redirection. Contact your processor for information about using their SDK to integrate a dialog into your Web site.

The following steps take place when you request a debit using processor-managed data:

Figure 4 Processing an Electronic Check Debit with Processor-Managed Data



- 1 Your customer places an order.
- 2 The processor and customer hold a dialog.



- 3 You request an electronic check debit.
- 4 CyberSource sends information about the transaction to the processor.

- 5 The check processor validates the information and performs basic fraud screening. The check processor does not contact the customer's bank to verify the existence of the customer's account; it only makes sure that the information provided by the customer is reasonable and that the account is not a known source of fraud. Depending on which processor you use, if there are problems with the account that prevent the transaction from being completed, the processor might charge you a returned check fee.
- 6 The check processor sends a reply to CyberSource indicating whether the debit will be processed.
- 7 CyberSource sends a reply to you.
- 8 You display an appropriate message to your customer.
- 9 The processor sends the request for clearing.

[Notifications of Change \(NOC\)](#)

Applicable services:

- Credit
- Debit

Supported processors:

- CyberSource ACH Service
- RBS WorldPay Atlanta

A Notification of Change (NOC) is a notice from a customer's bank indicating that an electronic check transaction included incorrect customer or payment information. The customer's bank:

- Corrects the information.
- Posts the transaction to the customer's bank account.
- Notifies you that payment information needs to be updated.

Each NOC includes a code that specifies what needs to be changed. You are responsible for taking the appropriate action when you receive an NOC.

You must correct all applicable records before submitting additional electronic check transactions for the customer. If you are using Payment Tokenization or Recurring Billing, you must update the information in your subscriptions or customer profiles. CyberSource maintains a database of all NOC entries. Repeated attempts to resubmit an uncorrected transaction could result in a fine and possible sanctions from the National Automated Clearing House Association (NACHA).

To get information about the NOCs for your transactions, you need to talk to your bank about getting a report that includes NOCs.

For descriptions of the NOC codes, see Appendix H, "[NOC Codes](#)," on page 133.

Verification and Validation

Important Even if an account passes validation and verification tests, the transaction can be rejected at the time of settlement. The bank against which the check is drawn does not participate in the verification or validation process. Therefore, an account can pass the verification and validation tests and the transaction can still be rejected if there are not sufficient funds in the account or if the bank account number is invalid.

The following table indicates the types of verification and validation supported for each processor.

Table 6 Types of Verification and Validation

Verification and Validation Method	Chase Paymentech Solutions	CyberSource ACH Service	TeleCheck	RBS
Validation	Yes	No	Yes	No
ACH Verification	No	Yes	No	Yes
Guarantees	No	No	Yes	No
Paymentech Verification	Yes	No	No	No
eFunds ID Verification	Yes	No	No	No

Validation

Applicable services:

- Debit

Supported processors:

- Chase Paymentech Solutions
- TeleCheck

Validation consists of format tests, bank routing number tests, and a comparison with the check processing partner's internal negative file. Set **ecDebitService_verificationLevel** to 1 to request validation with your debit request.

Note For the CyberSource ACH Service, validation is included in the ACH verification functionality, which happens automatically when you call the debit or credit services.

ACH Verification

Applicable services:

- Credit
- Debit

Supported processors:

- CyberSource ACH Service
- RBS WorldPay Atlanta

ACH verification is performed automatically for all debit and credit requests for the CyberSource ACH Service and RBS WorldPay Atlanta. ACH verification:

- 1 Validates the format and structure of the customer's bank account number. If the account number needs to be corrected and if a corrected account number is available, CyberSource will return the corrected account number to you in one of these fields:
 - **ecDebitReply_correctedAccountNumber**
 - **ecCreditReply_correctedAccountNumber**
- 2 Verifies that the customer's routing number is a valid routing number and valid for electronic transactions. If the routing number needs to be corrected and if a corrected routing number is available, CyberSource will return the corrected routing number to you in one of these fields:
 - **ecDebitReply_correctedRoutingNumber**
 - **ecCreditReply_correctedRoutingNumber**

Note If a corrected account number or corrected routing number is returned to you, you can use the value to update the information in your system. You do not need to update the information for the current transaction because CyberSource already updated the information before sending the transaction request to your bank.

3 Returns verification codes to you regardless of whether or not the account number or routing number was corrected. These verification codes indicate the results of the ACH verification. One of these verification codes is a mapped value and is returned in one of these fields:

- **ecDebitReply_verificationCode**
- **ecCreditReply_verificationCode**

The other verification code is a raw value and is returned in one of these fields:

- **ecDebitReply_verificationCodeRaw**
- **ecCreditReply_verificationCodeRaw**

The verification codes have enumerated values that are described in [Table 18](#) on page 88.

Guarantees

Applicable services:

- Debit

Supported processors:

- TeleCheck

Contact your processor for information about check guarantees. Set **ecDebitService_verificationLevel** to 3 to request a check guarantee with your debit request.

Paymentech Verification

Applicable services:

- Debit

Supported processors:

- Chase Paymentech Solutions

Paymentech verification compares the transaction information with an external negative file to identify accounts that have a history of bad checks or that were closed for cause. Paymentech verification is available only for transactions in U.S. dollars. Set **ecDebitService_verificationLevel** to 2 to request Paymentech verification with your debit request.

eFunds ID Verification

Applicable services:

- Debit

Supported processors:

- Chase Paymentech Solutions

eFunds ID Verification uses various data sources to confirm that a customer's identity information has been previously associated with the bank account number and routing number. The customer information items that are verified are:

- Name
- Address
- Phone number

Before using eFunds ID Verification, you must contact your processor to sign up for this feature. For each customer information item, the result can be:

- A positive match—Matches the value in the database.
- A negative match—Does not match the value in the database.
- No match—No value appears in the database.

The reply includes the processor's raw match results for each of the customer information items. To interpret the raw results, see your processor's specification for eFunds ID Verification. The reply also includes a single value that summarizes the match result for all the customer information items:

- Y: At least one of the items has a positive match.
- N: At least one of the items has a negative match and the other items do not have positive matches.
- U: None of the items has either a positive or negative match.

You can use this feature in one of two ways:

- Request eFunds ID Verification before requesting a debit. Use the verification results to decide whether to accept the customer's order.
- Request eFunds ID Verification when you request a debit, letting CyberSource accept or decline your debit request based on the verification results. The default actions for the verification results are:
 - Y or U: CyberSource accepts your request and processes the debit.
 - N: CyberSource declines your debit request and does not settle the funds.

In the request, you can change these defaults by specifying which verification results should trigger a decline.

In either case, if the customer's information fails the eFunds ID Verification, you receive **decision=REJECT** and **ecDebitReply_reasonCode=200**. You can look at **ecDebitReply_avsCode** to determine the overall result of the verification and **ecDebitReply_avsCodeRaw** to see the processor's raw results.

To specify which **ecDebitReply_avsCode** values trigger a decline, use the **businessRules_declineAVSFlags** field to specify a list of those values. By default, CyberSource declines the request if you receive **ecDebitReply_avsCode=N**. If you want CyberSource to decline the request for a U result also, set **businessRules_declineAVSFlags=N U**. Include the N in the list even though it represents the default behavior.

To Make the Decision Yourself

- 1 In your debit request, set **ecDebitService_verificationLevel** to 4 and set **ecDebitService_paymentMode** to 1. Set **item_0_unitPrice** to the actual amount you plan to charge the customer for the order.
- 2 Evaluate the result.
- 3 To accept the order, send another debit request, but this time set **ecDebitService_verificationLevel** to 1 and **ecDebitService_paymentMode** to 0. For this request, the eFunds ID Verification is not performed and CyberSource processes the debit as usual.

To Let CyberSource Decide for You

- 1 In your debit request, set **ecDebitService_verificationLevel** to 4.
- 2 If the eFunds ID Verification fails, CyberSource declines the request with reason code 200 and does not process the settlement. You can request a different payment method from your customer.

If the eFunds ID Verification is successful, CyberSource processes the settlement as usual.

Optional Features for Debits

For information about optional features, such as subscriptions and deferred payments, see Chapter 4, “[Optional Features for the Simple Order API](#),” on page 57.

Requirements for the Debit Request Fields

The following table lists the requirements for the fields used to request an electronic check debit. For detailed descriptions of these fields, see “[Request Fields](#)” on page 80.

Table 7 Electronic Check Debit Request Fields for the Simple Order API

Field	Required /Optional for Merchant-Provided Data				Required / Optional for Processor-Managed Data	Notes
	ACH	Chase	Tele	RBS	TeleCheck	
billTo_city	R	R	R	R	R	
billTo_company	N/A	N/A	See notes	O	N/A	<i>TeleCheck:</i> For personal checks, you must provide billTo_driversLicenseNumber and billTo_driversLicenseState. For corporate checks, you must provide either: • billTo_driversLicenseNumber and billTo_driversLicenseState or • billTo_company and billTo_companyTaxID
billTo_companyTaxID	N/A	N/A	See notes	N/A	N/A	<i>TeleCheck:</i> See the notes for billTo_company.
billTo_country	R	R	R	R	R	
billTo_dateOfBirth	N/A	N/A	N/A	N/A	N/A	
billTo_driversLicenseNumber	N/A	N/A	See notes	N/A	N/A	<i>TeleCheck:</i> See billTo_company.
billTo_driversLicenseState	N/A	N/A	See notes	N/A	N/A	<i>TeleCheck:</i> See billTo_company.

Table 7 Electronic Check Debit Request Fields for the Simple Order API (Continued)

Field	Required /Optional for Merchant-Provided Data				Required / Optional for Processor-Managed Data	Notes
	ACH	Chase	Tele	RBS	TeleCheck	
billTo_email	R	R	R	R	R	
billTo_firstName	R	R	R	R	R	
billTo_ipAddress	N/A	O	O	N/A	O	
billTo_lastName	R	R	R	R	R	
billTo_phoneNumber	R	See notes	R	N/A	R	<i>Chase Paymentech Solutions:</i> Required for eFunds ID Verification.
billTo_postalCode	R	R	R	R	R	
billTo_state	R	R	R	R	R	
billTo_street1	R	R	R	R	R	
billTo_street2	O	O	O	N/A	O	
businessRules_declineAVSFlags	N/A	O	N/A	N/A	N/A	
check_accountEncoderID	N/A	See notes	N/A	N/A	N/A	<i>Chase Paymentech Solutions:</i> Required for encoded account numbers.
check_accountNumber	R	R	R	R	N/A	
check_accountType	R	R	R	R	N/A	
check_bankTransitNumber	R	R	R	R	N/A	
check_checkNumber	N/A	O	O	O	O	
check_secCode	O	O	N/A	R	N/A	
ecDebitService_commerceIndicator	O	N/A	O	O	O	
ecDebitService_partialPaymentID	N/A	N/A	See notes	N/A	See notes	<i>TeleCheck:</i> Required for partial payments.

Table 7 Electronic Check Debit Request Fields for the Simple Order API (Continued)

Field	Required /Optional for Merchant-Provided Data				Required / Optional for Processor-Managed Data	Notes
	ACH	Chase	Tele	RBS	TeleCheck	
ecDebitService_paymentInfo	N/A	N/A	N/A	R	N/A	
ecDebitService_paymentMode	N/A	See notes	See notes	N/A	See notes	TeleCheck: Use for deferred payments. Chase Paymentech Solutions: Use for deferred payments and/or eFunds ID Verification.
ecDebitService_referenceNumber	O	O	See notes	N/A	See notes	TeleCheck: Required for deferred payments.
ecDebitService_run	R	R	R	R	R	
ecDebitService_settlementMethod	N/A	O	N/A	N/A	N/A	
ecDebitService_verificationLevel	N/A	O	O	N/A	O	
invoiceHeader_merchantDescriptor	N/A	O	N/A	N/A	N/A	
item_#_productCode	O	O	O	N/A	O	
item_#_productName	O	O	O	N/A	O	
item_#_productSKU	O	O	O	N/A	O	
item_#_quantity	See notes	See notes	See notes	N/A	See notes	All Processors: Required if item_#_productCode is not default, stored_value, or one of the values related to shipping and/or handling.
item_#_taxAmount	O	O	O	N/A	O	
item_#_unitPrice	See notes	See notes	See notes	N/A	See notes	All Processors: You must include either this field or purchaseTotals_grandTotalAmount in your request.
linkToRequest	O	O	O	O	O	

Table 7 Electronic Check Debit Request Fields for the Simple Order API (Continued)

Field	Required /Optional for Merchant-Provided Data				Required / Optional for Processor-Managed Data	Notes
	ACH	Chase	Tele	RBS	TeleCheck	
merchantID	R	R	R	R	R	
merchantReferenceCode	R	R	R	R	R	
purchaseTotals_currency	R	R	R	R	R	
purchaseTotals_grandTotalAmount	See notes	See notes	See notes	R	See notes	<i>All Processors:</i> You must include either this field or item_#_unitPrice in your request.
recurringSubscriptionInfo_subscriptionID	O	O	O	O	O	<i>All Processors:</i> When you include this field, many other fields are no longer required. See “Payment Tokenization” on page 62 and “Recurring Billing” on page 63.

Electronic Check Credits

Topics:

[Requesting a Credit with the Simple Order API](#)
[Follow-on Credits and Stand-Alone Credits](#)
[Deciding What Kind of Credit to Request](#)
[ACH Verification](#)
[Notifications of Change \(NOC\)](#)
[Optional Features for Credits](#)
[Requirements for the Credit Request Fields](#)

Requesting a Credit with the Simple Order API

To request an electronic check credit with the Simple Order API, set the **ecCreditService_run** field to **true**. When you request a credit, do not request any of the following services at the same time:

- Any credit card services: **ccAuthService**, **ccAuthReversalService**, **ccCaptureService**, **ccCreditService**
- Electronic check debit: **ecDebitService**
- Any Global Payment services: **bankTransferService**, **bankTransferRefundService**, **directDebitService**, **directDebitRefundService**, **bankTransferRealTimeService**
- PayPal® payment or credit: **paypalPaymentService**, **paypalCreditService**
- Advanced Fraud Screen: **afsService**
- Risk update: **riskUpdateService**

For information about these services, see:

- [Credit Card Services with the Simple Order API](#) or [Credit Card Services with the SCMP API](#)
- [Global Payment Service Developer's Guide](#)
- [Decision Manager Developer's Guide](#)
- [PayPal Services Implementation Guide](#)

Follow-on Credits and Stand-Alone Credits

There are two kinds of credits:

- **Follow-on**—Processors that support this feature: All processors. Send the credit request with the request ID and request token from the debit reply. CyberSource uses these values to retrieve all the customer's billing and account information that you sent with the debit so you do not have to send it again with the credit.
- **Stand-alone**—Processors that support this feature: All processors except TeleCheck. You need to include all the customer's billing and account information because CyberSource does not retrieve anything from the database.

Important CyberSource stores the debit information for 60 days, so you must process follow-on credits within 60 days of requesting the debit. If the 60 days have passed or if you are not sure if the 60 days have passed, then use a stand-alone credit and provide all the customer's billing and account information.

Deciding What Kind of Credit to Request

- **All Processors Except TeleCheck**—If you are sending the credit request within 60 days of the debit request, send a follow-on credit so that you are not required to provide all the customer information. If you are sending the credit request more than 60 days after the debit request, send a stand-alone credit.
- **TeleCheck**—You must send the credit request within 60 days of the debit request. The credit request must be a follow-on credit, which means you do not need to provide all the customer information. CyberSource retrieves all the required information from the database, including the identifier that the processor uses to link the credit to the debit. By linking the credit to the debit, the processor can prohibit a credit amount that exceeds the debit amount.

Follow-On Credits

A follow-on credit uses the **requestID** and **requestToken** returned from a previous **ecDebitService** request to link the credit to the debit. Send the request ID value in the **ecCreditService_debitRequestID** field and send the request token value in the **orderRequestToken** field. CyberSource uses these values to look up the customer's billing and account information from the original debit, which means that you are not required to include those fields in the **ecCreditService** request.

Note A follow-on credit must be for a debit request that included a payment. In other words, **ecDebitService_paymentMode=0** or **2**. A follow-on credit cannot be for a debit request in which **ecDebitService_paymentMode=1**.

Important If you combine a request for a follow-on credit with a request for another service, you must provide the customer's billing and account information.

Stand-Alone Credits

A stand-alone credit does not link the credit to a previous debit request. Instead of sending the **ecCreditService_debitRequestID** and **orderRequestToken** fields in the credit request, the request must include the fields for the customer's billing and account information.

ACH Verification

Applicable services:

- Credit
- Debit

Supported processors:

- CyberSource ACH Service
- RBS WorldPay Atlanta

ACH verification is performed automatically for all debit and credit requests for the CyberSource ACH Service and RBS WorldPay Atlanta. ACH verification:

- 1 Validates the format and structure of the customer's bank account number. If the account number needs to be corrected and if a corrected account number is available, CyberSource will return the corrected account number to you in one of these fields:
 - **ecDebitReply_correctedAccountNumber**
 - **ecCreditReply_correctedAccountNumber**

- 2 Verifies that the customer's routing number is a valid routing number and valid for electronic transactions. If the routing number needs to be corrected and if a corrected routing number is available, CyberSource will return the corrected routing number to you in one of these fields:

- **ecDebitReply_correctedRoutingNumber**
- **ecCreditReply_correctedRoutingNumber**

Note If a corrected account number or corrected routing number is returned to you, you can use the value to update the information in your system. You do not need to update the information for the current transaction because CyberSource already updated the information before sending the transaction request to your bank.

- 3 Returns verification codes to you regardless of whether or not the account number or routing number was corrected. These verification codes indicate the results of the ACH verification. One of these verification codes is a mapped value and is returned in one of these fields:

- **ecDebitReply_verificationCode**
- **ecCreditReply_verificationCode**

The other verification code is a raw value and is returned in one of these fields:

- **ecDebitReply_verificationCodeRaw**
- **ecCreditReply_verificationCodeRaw**

The verification codes have enumerated values that are described in [Table 18](#) on page 88.

Notifications of Change (NOC)

Applicable services:

- Credit
- Debit

Supported processors:

- CyberSource ACH Service
- RBS WorldPay Atlanta

A Notification of Change (NOC) is a notice from a customer's bank indicating that an electronic check transaction included incorrect customer or payment information. The customer's bank:

- Corrects the information.
- Posts the transaction to the customer's bank account.
- Notifies you that payment information needs to be updated.

Each NOC includes a code that specifies what needs to be changed. You are responsible for taking the appropriate action when you receive an NOC.

You must correct all applicable records before submitting additional electronic check transactions for the customer. If you are using Payment Tokenization or Recurring Billing, you must update the information in your subscriptions or customer profiles. CyberSource maintains a database of all NOC entries. Repeated attempts to resubmit an uncorrected transaction could result in a fine and possible sanctions from the National Automated Clearing House Association (NACHA).

To get information about the NOCs for your transactions, you need to talk to your bank about getting a report that includes NOCs.

For descriptions of the NOC codes, see Appendix H, "[NOC Codes](#)," on page 133.

Optional Features for Credits

For information about optional features, such as merchant descriptors and multiple partial credits, see Chapter 4, "[Optional Features for the Simple Order API](#)," on page 57.

Requirements for the Credit Request Fields

The following table lists the requirements for the fields used to request an electronic check credit. For detailed descriptions of these fields, see “[Request Fields](#)” on page 80.

Table 8 Electronic Check Credit Request Fields for the Simple Order API

Field	Required /Optional for Merchant-Provided Data				Required / Optional for Processor-Managed Data	Notes
	ACH	Chase	Tele	RBS	TeleCheck	
billTo_city	See notes	See notes	See notes	R	See notes	<i>All Processors:</i> Required for stand-alone credits. Optional for follow-on credits, except for RBS Worldpay Atlanta.
billTo_country	See notes	See notes	See notes	R	N/A	<i>All Processors:</i> Required for stand-alone credits. Optional for follow-on credits.
billTo_dateOfBirth	N/A	N/A	N/A	O	N/A	
billTo_email	See notes	See notes	See notes	R	See notes	<i>All Processors:</i> Required for stand-alone credits. Optional for follow-on credits.
billTo_firstName	See notes	See notes	See notes	R	See notes	<i>All Processors:</i> Required for stand-alone credits. Optional for follow-on credits.
billTo_ipAddress	O	O	O	O	O	
billTo_lastName	See notes	See notes	See notes	R	See notes	<i>All Processors:</i> Required for stand-alone credits. Optional for follow-on credits.
billTo_phoneNumber	O	O	O	R	O	
billTo_postalCode	See notes	See notes	See notes	R	See notes	<i>All Processors:</i> Required for stand-alone credits. Optional for follow-on credits.
billTo_state	See notes	See notes	See notes	R	See notes	<i>All Processors:</i> Required for stand-alone credits. Optional for follow-on credits.

Table 8 Electronic Check Credit Request Fields for the Simple Order API (Continued)

Field	Required /Optional for Merchant-Provided Data				Required / Optional for Processor-Managed Data	Notes
	ACH	Chase	Tele	RBS	TeleCheck	
billTo_street1	See notes	See notes	See notes	R	See notes	<i>All Processors:</i> Required for stand-alone credits. Optional for follow-on credits.
billTo_street2	O	O	O	See notes	O	<i>RBS WorldPay Atlanta:</i> Required for stand-alone credits. Optional for follow-on credits.
check_accountEncoderID	N/A	See notes	N/A	R	N/A	<i>Chase Paymentech Solutions:</i> Required for encoded account numbers.
check_accountNumber	See notes	See notes	See notes	R	N/A	<i>All Processors:</i> Required for stand-alone credits. Optional for follow-on credits.
check_accountType	See notes	See notes	See notes	R	N/A	<i>All Processors:</i> Required for stand-alone credits. Optional for follow-on credits.
check_bankTransitNumber	See notes	See notes	See notes	R	N/A	<i>All Processors:</i> Required for stand-alone credits. Optional for follow-on credits.
check_checkNumber	N/A	O	O	See notes	O	<i>RBS WorldPay Atlanta:</i> Required for follow-on credits. Optional for stand-alone credits.
check_paymentInfo	N/A	N/A	N/A	R	N/A	
check_secCode	O	O	N/A	R	N/A	
ecCreditService_commerceIndicator	O	N/A	O	O	O	
ecCreditService_debitRequestID	See notes	See notes	See notes	See notes	See notes	<i>All Processors:</i> Required for follow-on credits.

Table 8 Electronic Check Credit Request Fields for the Simple Order API (Continued)

Field	Required /Optional for Merchant-Provided Data				Required / Optional for Processor-Managed Data	Notes
	ACH	Chase	Tele	RBS	TeleCheck	
ecCreditService_debitRequestToken	See notes	See notes	See notes	See notes	See notes	<i>All Processors:</i> For follow-on credits, you must include either this field or orderRequestToken in your request depending on when you implemented request tokens. See the information about request tokens in Getting Started with CyberSource Advanced .
ecCreditService_partialPaymentID	N/A	N/A	See notes	O	See notes	<i>TeleCheck:</i> Required for partial credits.
ecCreditService_referenceNumber	O	O	O	O	O	
ecCreditService_run	R	R	R	R	R	
ecCreditService_settlementMethod	N/A	O	N/A	O	N/A	
invoiceHeader_merchantDescriptor	N/A	O	N/A	O	N/A	
item_#_productCode	O	O	O	O	O	
item_#_productName	O	O	O	O	O	
item_#_productSKU	O	O	O	O	O	
item_#_quantity	O	O	O	O	O	
item_#_taxAmount	O	O	O	O	O	
item_#_unitPrice	See notes	See notes	See notes	O	See notes	<i>All Processors:</i> You must include either this field or purchaseTotals_grandTotalAmount in your request.
merchantID	R	R	R	R	R	
merchantReferenceCode	R	R	R	R	R	

Table 8 Electronic Check Credit Request Fields for the Simple Order API (Continued)

Field	Required /Optional for Merchant-Provided Data				Required / Optional for Processor-Managed Data	Notes
	ACH	Chase	Tele	RBS	TeleCheck	
orderRequestToken	See notes	See notes	See notes	O	See notes	<i>All Processors:</i> For follow-on credits, you must include either this field or ecCreditService_debitRequestToken in your request depending on when you implemented request tokens. See the information about request tokens in Getting Started with CyberSource Advanced .
purchaseTotals_currency	R	R	R	R	R	
purchaseTotals_grandTotalAmount	See notes	See notes	See notes	R	See notes	<i>All Processors:</i> You must include either this field or item_#_unitPrice in your request.
recurringSubscriptionInfo_subscriptionID	O	O	O	See notes	O	<i>All Processors:</i> When you include this field, many other fields are no longer required. See “Payment Tokenization” on page 62 and “Recurring Billing” on page 63.

Authenticate

Supported processors:

- RBS WorldPay Atlanta

The authentication service is an optional service you can request for RBS WorldPay Atlanta.

To request an electronic check authentication, set the **ecAuthenticateService_run** field to true. The following fields are required when requesting this service:

- **ecAuthenticateService_run**
- **merchantID**
- **merchantReferenceCode**
- **billTo_firstName**
- **billTo_lastName**

- **billTo_street1**
- **billTo_city**
- **billTo_state**
- **billTo_country**
- **billTo_postalCode**

The authentication service:

- Validates customer information such as name, address, and date of birth and returns information to you in the **ecAuthenticateReply_checkpointSummary** field.
- Provides you with consumer fraud information to help protect you against fraudulent transactions. Information is returned to you in the **ecAuthenticateReply_fraudShieldIndicators** field.

For more information, see:

- Appendix I, "[Check Point Summary Codes](#)," on page 135
- Appendix J, "[Fraud Shield Indicator Codes](#)," on page 149

Voids

Topics:

[Requesting a Void with the Simple Order API](#)
[Requirements for the Void Request Fields](#)

Requesting a Void with the Simple Order API

Supported processors:

- Chase Paymentech Solutions
- RBS WorldPay Atlanta
- TeleCheck

A void cancels an electronic check debit or credit request that you have submitted to CyberSource. A transaction can be voided only if CyberSource has not already submitted the debit or credit information to your processor. CyberSource usually submits transaction information to your processor once a day, so your window for successfully performing a void is relatively small. CyberSource will decline your void request if the debit or credit information has already been sent to the processor. You cannot undo a void and you cannot perform a follow-on credit for a debit that has been voided.

For RBS WorldPay Atlanta, voids are supported for credit requests only.

To request a void for an electronic check debit or credit with the Simple Order API, set the **voidService_run** field to `true`. When you request a void, do not request any other services at the same time.

A void is a follow-on transaction that uses the **requestID** and **requestToken** returned from a previous **ecDebitService** or **ecCreditService** request to link the void to the debit or credit. Send the request ID value in the **voidService_voidRequestID** field and send the request token value in the **orderRequestToken** field. CyberSource uses these values to look up the customer's billing and account information from the original debit or credit, which means that you are not required to include those fields in the **voidService** request.

Requirements for the Void Request Fields

The following table lists the requirements for the fields used to request a void. For detailed descriptions of these fields, see “[Request Fields](#)” on page 80.

Table 9 Void Request Fields for the Simple Order API

Field	Required /Optional for Merchant-Provided Data				Required / Optional for Processor-Managed Data	Notes
	ACH	Chase	RBS	Tele	TeleCheck	
merchantID	N/A	R	R	R	R	
merchantReferenceCode	N/A	R	R	R	R	
orderRequestToken	N/A	See notes	See notes	See notes	See notes	<i>All Processors:</i> You must include either this field or voidService_voidRequestToken in your request depending on when you implemented request tokens. See the information about request tokens in Getting Started with CyberSource Advanced .
voidService_run	N/A	R	R	R	R	
voidService_voidRequestID	N/A	R	R	R	R	
voidService_voidRequestToken	N/A	See notes	O	See notes	See notes	<i>All Processors:</i> You must include either this field or orderRequestToken in your request depending on when you implemented request tokens. See the information about request tokens in Getting Started with CyberSource Advanced .

Chapter 3

Electronic Check Processing with the SCMP API

Topics:

[Electronic Check Debits](#)
[Electronic Check Credits](#)
[Authenticate](#)
[Voids](#)

Electronic Check Debits

Topics:

[Requesting a Debit with the SCMP API](#)
[Handling Customer Account Information](#)
[Verification and Validation](#)
[Optional Features for Debits](#)
[Requirements for the Debit Request Fields](#)

Requesting a Debit with the SCMP API

To request an electronic check debit with the SCMP API, set the **ics_applications** field to **ics_ecp_debit**. When you request a debit, do not request any of the following services at the same time:

- Any credit card services: **ics_auth**, **ics_auth_reversal**, **ics_bill**, **ics_credit**
- Electronic check credit: **ics_ecp_credit**
- Any Global Payment services: **ics_bank_transfer**, **ics_bank_transfer_refund**, **ics_direct_debit**, **ics_direct_debit_refund**, **ics_bank_transfer_real_time**
- PayPal® payment or credit: **ics_paypal_payment**, **ics_paypal_credit**
- Advanced Fraud Screen: **ics_score**
- Risk update: **ics_risk_update**

See the [Credit Card Services Implementation Guide](#), the [Global Payment Service Developer's Guide](#), the [PayPal Services Implementation Guide](#), and the [Decision Manager Developer's Guide](#) for information about these services.

Handling Customer Account Information

Topics:

[Merchant-Provided Data](#)

[Processor-Managed Data](#)

[Notifications of Change \(NOC\)](#)

Merchant-Provided Data

Applicable services:

- Debit

Supported processors:

- Chase Paymentech Solutions
- CyberSource ACH Service
- RBS WorldPay Atlanta
- TeleCheck

Merchant-provided data handling requires you to collect the customer's account information and provide it in your service request. The required fields are:

- **eCP_account_no**
- **eCP_account_type**
- **eCP_rdfi**

Using this method, you must modify your Web site to collect the pertinent account information. You must also retain the account information for future transactions, such as credits.

Important If you send the customer's account number in your request, then CyberSource assumes you are using the merchant-provided data method.

Customers might not know how to use their printed checks to find the bank routing number and the bank account number. Consider using a graphic like the following example on your Web site to help customers locate their information.

Figure 5 Sample Check Showing Routing Number and Account Number

NAME
ADDRESS
CITY, STATE, ZIP

0123
01-2345/6789

DATE

PAY TO THE
ORDER OF

\$

BANK NAME
ADDRESS
CITY, STATE, ZIP

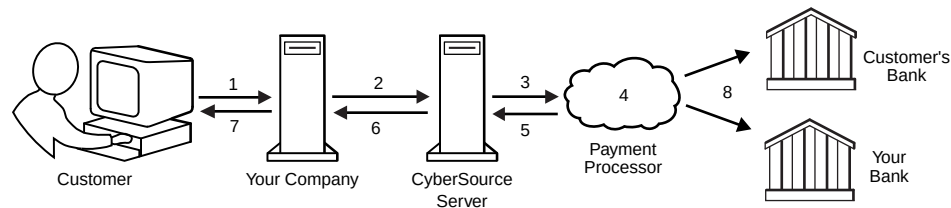
DOLLARS

FOR

⑆0123456789⑆ 01234567890123⑈ 0123

Routing Number Account Number

The following steps take place when you request a debit using merchant-provided data:

Figure 6 Processing an Electronic Check Debit with Merchant-Provided Data

- 1 Your customer places an order.
- 2 You request an electronic check debit.
In your request, you provide the customer's account information.
- 3 CyberSource sends the customer's account information and other information about the transaction to the check processor.
- 4 The payment processor validates the information and performs basic fraud screening.
The processor does not contact the customer's bank to verify the existence of the customer's account; it only makes sure that the information provided by the customer is reasonable and that the account is not a known source of fraud.
Depending on which processor you use, if there are problems with the account that prevent the transaction from being completed, the processor might charge you a returned check fee.
- 5 The payment processor sends a reply to CyberSource indicating whether the debit will be processed.
- 6 CyberSource sends a reply to you.
- 7 You display an appropriate message to your customer.
- 8 The processor sends the request for clearing.

Processor-Managed Data

Applicable services:

- Debit

Supported processors:

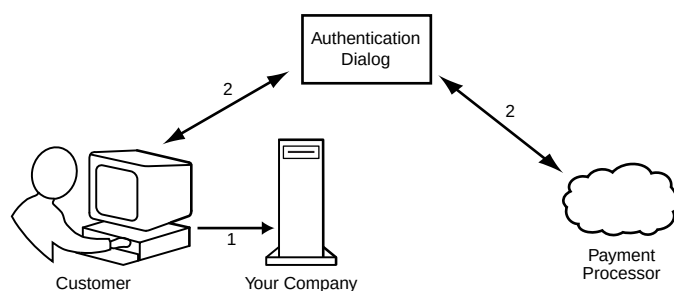
- TeleCheck

Processor-managed data handling does not require you to collect the customer's account information. Instead, the customer provides credentials directly to the processor and these credentials are authenticated before you request a debit.

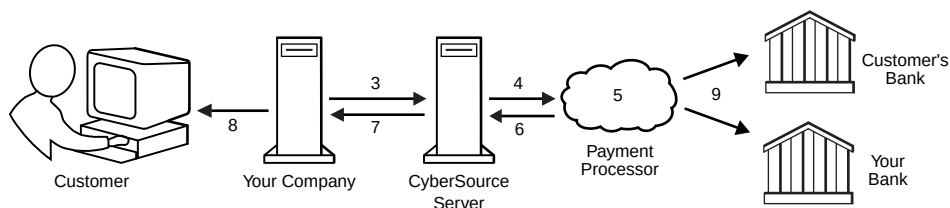
You must adjust your checkout process to route the customer's browser to the processor's authentication interface. You can do this through different methods, such as a popup window or simple response redirection. Contact your processor for information about using their SDK to integrate a dialog into your Web site.

The following steps take place when you request a debit using processor-managed data:

Figure 7 Processing an Electronic Check Debit with Processor-Managed Data



- 1 Your customer places an order.
- 2 The processor and customer hold a dialog.



- 3 You request an electronic check debit.
- 4 CyberSource sends information about the transaction to the processor.

- 5 The check processor validates the information and performs basic fraud screening. The check processor does not contact the customer's bank to verify the existence of the customer's account; it only makes sure that the information provided by the customer is reasonable and that the account is not a known source of fraud. Depending on which processor you use, if there are problems with the account that prevent the transaction from being completed, the processor might charge you a returned check fee.
- 6 The check processor sends a reply to CyberSource indicating whether the debit will be processed.
- 7 CyberSource sends a reply to you.
- 8 You display an appropriate message to your customer.
- 9 The processor sends the request for clearing.

Notifications of Change (NOC)

Applicable services:

- Credit
- Debit

Supported processors:

- CyberSource ACH Service
- RBS WorldPay Atlanta

A Notification of Change (NOC) is a notice from a customer's bank indicating that an electronic check transaction included incorrect customer or payment information. The customer's bank:

- Corrects the information.
- Posts the transaction to the customer's bank account.
- Notifies you that payment information needs to be updated.

Each NOC includes a code that specifies what needs to be changed. You are responsible for taking the appropriate action when you receive an NOC.

You must correct all applicable records before submitting additional electronic check transactions for the customer. If you are using Payment Tokenization or Recurring Billing, you must update the information in your subscriptions or customer profiles. CyberSource maintains a database of all NOC entries. Repeated attempts to resubmit an uncorrected transaction could result in a fine and possible sanctions from the National Automated Clearing House Association (NACHA).

To get information about the NOCs for your transactions, you need to talk to your bank about getting a report that includes NOCs.

For descriptions of the NOC codes, see Appendix H, "[NOC Codes](#)," on page 133.

Verification and Validation

Important Even if an account passes validation and verification tests, the transaction can be rejected at the time of settlement. The bank against which the check is drawn does not participate in the verification or validation process. Therefore, an account can pass the verification and validation tests and the transaction can still be rejected if there are not sufficient funds in the account or if the bank account number is invalid.

The following table indicates the types of verification and validation supported for each processor.

Table 10 Types of Verification and Validation

Verification and Validation Method	Chase Paymentech Solutions	CyberSource ACH Service	TeleCheck	RBS
Validation	Yes	No	Yes	No
ACH Verification	No	Yes	No	Yes
Guarantees	No	No	Yes	No
Paymentech Verification	Yes	No	No	No
eFunds ID Verification	Yes	No	No	No

Validation

Applicable services:

- Debit

Supported processors:

- Chase Paymentech Solutions
- TeleCheck

Validation consists of format tests, bank routing number tests, and a comparison with the check processing partner's internal negative file. Set **ecp_verification_level** to 1 to request validation with your debit request.

Note For the CyberSource ACH Service, validation is included in the ACH verification functionality, which happens automatically when you call the debit or credit services.

ACH Verification

Applicable services:

- Credit
- Debit

Supported processors:

- CyberSource ACH Service
- RBS WorldPay Atlanta

ACH verification is performed automatically for all debit and credit requests for the CyberSource ACH Service and RBS WorldPay Atlanta. ACH verification:

- 1 Validates the format and structure of the customer's bank account number. If the account number needs to be corrected and if a corrected account number is available, CyberSource will return the corrected account number to you in one of these fields:
 - **eCP_debit_corrected_account_number**
 - **eCP_credit_corrected_account_number**
- 2 Verifies that the customer's routing number is a valid routing number and valid for electronic transactions. If the routing number needs to be corrected and if a corrected routing number is available, CyberSource will return the corrected routing number to you in one of these fields:
 - **eCP_debit_corrected_routing_number**
 - **eCP_credit_corrected_routing_number**

Note If a corrected account number or corrected routing number is returned to you, you can use the value to update the information in your system. You do not need to update the information for the current transaction because CyberSource already updated the information before sending the transaction request to your bank.

3 Returns verification codes to you regardless of whether or not the account number or routing number was corrected. These verification codes indicate the results of the ACH verification. One of these verification codes is a mapped value and is returned in one of these fields:

- **ecp_debit_verification_code**
- **ecp_credit_verification_code**

The other verification code is a raw value and is returned in one of these fields:

- **ecp_debit_verification_code_raw**
- **ecp_credit_verification_code_raw**

The verification codes have enumerated values that are described in [Table 21](#) on page 108.

Guarantees

Applicable services:

- Debit

Supported processors:

- TeleCheck

Contact your processor for information about check guarantees. Set **ecp_verification_level** to 3 to request a check guarantee with your debit request.

Paymentech Verification

Applicable services:

- Debit

Supported processors:

- Chase Paymentech Solutions

Paymentech verification compares the transaction information with an external negative file to identify accounts that have a history of bad checks or that were closed for cause. Paymentech verification is available only for transactions in U.S. dollars. Set **ecp_verification_level** to 2 to request Paymentech verification with your debit request.

eFunds ID Verification

Applicable services:

- Debit

Supported processors:

- Chase Paymentech Solutions

eFunds ID Verification uses various data sources to confirm that a customer's identity information has been previously associated with the bank account number and routing number. The customer information items that are verified are:

- Name
- Address
- Phone number

Before using eFunds ID Verification, you must contact your processor to sign up for this feature. For each customer information item, the result can be:

- A positive match—Matches the value in the database.
- A negative match—Does not match the value in the database.
- No match—No value appears in the database.

The reply includes the processor's raw match results for each of the customer information items. To interpret the raw results, see your processor's specification for eFunds ID Verification. The reply also includes a single value that summarizes the match result for all the customer information items:

- Y: At least one of the items has a positive match.
- N: At least one of the items has a negative match and the other items do not have positive matches.
- U: None of the items has either a positive or negative match.

You can use this feature in one of two ways:

- Request eFunds ID Verification before requesting a debit. Use the verification results to decide whether to accept the customer's order.
- Request eFunds ID Verification when you request a debit, letting CyberSource accept or decline your debit request based on the verification results. The default actions for the verification results are:
 - Y or U: CyberSource accepts your request and processes the debit.
 - N: CyberSource declines your debit request and does not settle the funds.

In the request, you can change these defaults by specifying which verification results should trigger a decline.

In either case, if the customer's information fails the eFunds ID Verification, you receive **eCP_debit_rcode=0** and **eCP_debit_rflag=DAVSNO**. You can look at **eCP_debit_avs** to determine the overall result of the verification and **eCP_debit_avs_raw** to see the processor's raw results.

To specify which **eCP_debit_avs** values trigger a decline, use the **decline_avs_flags** field to specify a list of those values. By default, CyberSource declines the request if you receive **eCP_debit_avs=N**. If you want CyberSource to decline the request for a U result also, set **decline_avs_flags=N, U**. Include the N in the list even though it represents the default behavior.

To Make the Decision Yourself

- 1 In your debit request, set **eCP_verification_level** to 4 and set **eCP_payment_mode** to 1. Set the **amount** offer-level field for **offer0** to the actual amount you plan to charge the customer for the order.
- 2 Evaluate the result.
- 3 To accept the order, send another debit request, but this time set **eCP_verification_level** to 1 and **eCP_payment_mode** to 0. For this request, the eFunds ID Verification is not performed and CyberSource processes the debit as usual.

To Let CyberSource Decide for You

- 1 In your debit request, set **eCP_verification_level** to 4.
- 2 If the eFunds ID Verification fails, CyberSource declines the request with a DAVSNO reply flag and does not process the settlement. You can request a different payment method from your customer.

If the eFunds ID Verification is successful, CyberSource processes the settlement as usual.

Optional Features for Debits

For information about optional features, such as subscriptions and deferred payments, see Chapter 5, "[Optional Features for the SCMP API](#)," on page 65.

Requirements for the Debit Request Fields

The following table lists the requirements for the fields used to request an electronic check debit. For detailed descriptions of these fields, see [“Request-Level Fields”](#) on page 100 and [“Offer-Level Fields”](#) on page 107.

Table 11 Electronic Check Debit Request Fields for the SCMP API

Field	Required /Optional for Merchant-Provided Data				Required / Optional for Processor-Managed Data	Notes
	ACH	Chase	Tele	RBS	TeleCheck	
account_encoder_id	N/A	See notes	N/A	N/A	N/A	<i>Chase Paymentech Solutions:</i> Required for encoded account numbers.
bill_address1	R	R	R	R	R	
bill_address2	O	O	O	O	O	
bill_city	R	R	R	R	R	
bill_company_tax_id	N/A	N/A	See notes	N/A	N/A	<i>TeleCheck:</i> driver_license_no and driver_license_state . For corporate checks, you must provide either: • driver_license_no and driver_license_state <i>or</i> • company_name and bill_company_tax_id
bill_country	R	R	R	R	R	
bill_state	R	R	R	R	R	
bill_zip	R	R	R	R	R	
company_name	N/A	N/A	See notes	N/A	N/A	<i>TeleCheck:</i> See the notes for bill_company_tax_id .
currency	R	R	R	R	R	
customer_email	R	R	R	R	R	
customer_firstname	R	R	R	R	R	
customer_ipaddress	N/A	O	O	N/A	O	

Table 11 Electronic Check Debit Request Fields for the SCMP API (Continued)

Field	Required /Optional for Merchant-Provided Data				Required / Optional for Processor-Managed Data	Notes
	ACH	Chase	Tele	RBS	TeleCheck	
customer_lastname	R	R	R	R	R	
customer_phone	R	See notes	R	N/A	R	Chase Paymentech Solutions: Required for eFunds ID Verification.
date_of_birth	N/A	N/A	N/A	N/A	N/A	
decline_avs_flags	N/A	O	N/A	N/A	N/A	
driver_license_no	N/A	N/A	See notes	N/A	N/A	TeleCheck: See the notes for bill_company_tax_id .
driver_license_state	N/A	N/A	See notes	N/A	N/A	TeleCheck: See the notes for bill_company_tax_id .
e_commerce_indicator	O	N/A	O	O	O	
ecp_account_no	R	R	R	R	N/A	
ecp_account_type	R	R	R	R	N/A	
ecp_check_no	N/A	O	O	O	O	
ecp_payment_mode	N/A	See notes	See notes	N/A	See notes	TeleCheck: Use for deferred payments. Chase Paymentech Solutions: Use for deferred payments and/or eFunds ID Verification.
ecp_rdfi	R	R	R	R	N/A	
ecp_ref_no	O	O	See notes	O	See notes	TeleCheck: Required for deferred payments.
ecp_sec_code	O	O	N/A	R	N/A	
ecp_settlement_method	N/A	O	N/A	N/A	N/A	
ecp_verification_level	N/A	O	O	N/A	O	
grand_total_amount	See notes	See notes	See notes	R	See notes	All Processors: You must include either this field or offerN: amount in your request.

Table 11 Electronic Check Debit Request Fields for the SCMP API (Continued)

Field	Required /Optional for Merchant-Provided Data				Required / Optional for Processor-Managed Data	Notes
	ACH	Chase	Tele	RBS	TeleCheck	
ics_applications	R	R	R	R	R	
link_to_request	O	O	O	O	O	
merchant_descriptor	N/A	O	N/A	O	N/A	
merchant_id	R	R	R	R	R	
merchant_ref_number	R	R	R	R	R	
offerN: amount	See notes	See notes	See notes	See note	See notes	All Processors: You must include either this field or grand_total_amount in your request.
offerN: merchant_product_sku	O	O	O	N/A	O	
offerN: product_code	O	O	O	N/A	O	
offerN: product_name	O	O	O	N/A	O	
offerN: quantity	See notes	See notes	See notes	N/A	See notes	All Processors: Required if offerN: product_code is not default, stored_value, or one of the values related to shipping and/or handling.
offerN: tax_amount	O	O	O	N/A	O	
partial_payment_id	N/A	N/A	See notes	N/A	See notes	TeleCheck: Required for partial payments.
subscription_id	O	O	O	See notes	O	All Processors: When you include this field, many other fields are no longer required. See “Payment Tokenization” on page 62 and “Recurring Billing” on page 63.
timeout	O	O	O	O	O	

Electronic Check Credits

Topics:

- [Requesting a Credit with the SCMP API](#)
 - [Follow-on Credits and Stand-Alone Credits](#)
 - [Deciding What Kind of Credit to Request](#)
 - [ACH Verification](#)
 - [Notifications of Change \(NOC\)](#)
 - [Optional Features for Credits](#)
 - [Requirements for the Credit Request Fields](#)
-

Requesting a Credit with the SCMP API

To request an electronic check credit with the SCMP API, set the **ics_applications** field to **ics_ech_credit**. When you request a credit, do not request any of the following services at the same time:

- Any credit card services: **ics_auth**, **ics_auth_reversal**, **ics_bill**, **ics_credit**
- Electronic check debit: **ics_ech_debit**
- Any Global Payment services: **ics_bank_transfer**, **ics_bank_transfer_refund**, **ics_direct_debit**, **ics_direct_debit_refund**, **ics_bank_transfer_real_time**
- PayPal® payment or credit: **ics_paypal_payment**, **ics_paypal_credit**
- Advanced Fraud Screen: **ics_score**
- Risk update: **ics_risk_update**

For information about these services, see:

- [Credit Card Services with the Simple Order API](#) or [Credit Card Services with the SCMP API](#)
- [Global Payment Service Developer's Guide](#)
- [Decision Manager Developer's Guide](#)
- [PayPal Services Implementation Guide](#)

Follow-on Credits and Stand-Alone Credits

There are two kinds of credits:

- **Follow-on**—Processors that support this feature: All processors. Send the credit request with the request ID and request token from the debit reply. CyberSource uses these values to retrieve all the customer's billing and account information that you sent with the debit so you do not have to send it again with the credit.
- **Stand-alone**—Processors that support this feature: All processors except TeleCheck. You need to include all the customer's billing and account information because CyberSource does not retrieve anything from the database.

Important CyberSource stores the debit information for 60 days, so you must process follow-on credits within 60 days of requesting the debit. If the 60 days have passed or if you are not sure if the 60 days have passed, then use a stand-alone credit and provide all the customer's billing and account information.

Deciding What Kind of Credit to Request

- **All Processors Except TeleCheck**—If you are sending the credit request within 60 days of the debit request, send a follow-on credit so that you are not required to provide all the customer information. If you are sending the credit request more than 60 days after the debit request, send a stand-alone credit.
- **TeleCheck**—You must send the credit request within 60 days of the debit request. The credit request must be a follow-on credit, which means you do not need to provide all the customer information. CyberSource retrieves all the required information from the database, including the identifier that the processor uses to link the credit to the debit. By linking the credit to the debit, the processor can prohibit a credit amount that exceeds the debit amount.

Follow-On Credits

A follow-on credit uses the **request_id** and **request_token** returned from a previous **ics_ecp_debit** request to link the credit to the debit. Send the request ID value in the **ecp_debit_request_id** field and send the request token value in the **order_request_token** field. CyberSource uses these values to look up the customer's billing and account information from the original debit, which means that you are not required to include those fields in the **ics_ecp_credit** request.

Note A follow-on credit must be for a debit request that included a payment. In other words, **ecp_payment_mode**=0 or 2. A follow-on credit cannot be for a debit request in which **ecp_payment_mode**=1.

Important If you combine a request for a follow-on credit with a request for another service, you must provide the customer's billing and account information.

Stand-Alone Credits

A stand-alone credit does not link the credit to a previous debit request. Instead of sending the **ecp_debit_request_id** and **order_request_token** fields in the credit request, the request must include the fields for the customer's billing and account information.

ACH Verification

Applicable services:

- Credit
- Debit

Supported processors:

- CyberSource ACH Service
- RBS WorldPay Atlanta

ACH verification is performed automatically for all debit and credit requests for the CyberSource ACH Service and RBS WorldPay Atlanta. ACH verification:

- 1 Validates the format and structure of the customer's bank account number. If the account number needs to be corrected and if a corrected account number is available, CyberSource will return the corrected account number to you in one of these fields:
 - **ecp_debit_corrected_account_number**
 - **ecp_credit_corrected_account_number**

- 2 Verifies that the customer's routing number is a valid routing number and valid for electronic transactions. If the routing number needs to be corrected and if a corrected routing number is available, CyberSource will return the corrected routing number to you in one of these fields:
- **eCP_debit_corrected_routing_number**
 - **eCP_credit_corrected_routing_number**

Note If a corrected account number or corrected routing number is returned to you, you can use the value to update the information in your system. You do not need to update the information for the current transaction because CyberSource already updated the information before sending the transaction request to your bank.

- 3 Returns verification codes to you regardless of whether or not the account number or routing number was corrected. These verification codes indicate the results of the ACH verification. One of these verification codes is a mapped value and is returned in one of these fields:

- **eCP_debit_verification_code**
- **eCP_credit_verification_code**

The other verification code is a raw value and is returned in one of these fields:

- **eCP_debit_verification_code_raw**
- **eCP_credit_verification_code_raw**

The verification codes have enumerated values that are described in [Table 21](#) on page 108.

Notifications of Change (NOC)

Applicable services:

- Credit
- Debit

Supported processors:

- CyberSource ACH Service
- RBS WorldPay Atlanta

A Notification of Change (NOC) is a notice from a customer's bank indicating that an electronic check transaction included incorrect customer or payment information. The customer's bank:

- Corrects the information.
- Posts the transaction to the customer's bank account.
- Notifies you that payment information needs to be updated.

Each NOC includes a code that specifies what needs to be changed. You are responsible for taking the appropriate action when you receive an NOC.

You must correct all applicable records before submitting additional electronic check transactions for the customer. If you are using Payment Tokenization or Recurring Billing, you must update the information in your subscriptions or customer profiles. CyberSource maintains a database of all NOC entries. Repeated attempts to resubmit an uncorrected transaction could result in a fine and possible sanctions from the National Automated Clearing House Association (NACHA).

To get information about the NOCs for your transactions, you need to talk to your bank about getting a report that includes NOCs.

For descriptions of the NOC codes, see Appendix H, "[NOC Codes](#)," on page 133.

Optional Features for Credits

For information about optional features, such as merchant descriptors and multiple partial credits, see Chapter 5, "[Optional Features for the SCMP API](#)," on page 65.

Requirements for the Credit Request Fields

The following table lists the requirements for the fields used to request an electronic check credit. For detailed descriptions of these fields, see “[Request-Level Fields](#)” on page 100 and “[Offer-Level Fields](#)” on page 107.

Table 12 Electronic Check Credit Request Fields for the SCMP API

Field	Required /Optional for Merchant-Provided Data				Required / Optional for Processor-Managed Data	Notes
	ACH	Chase	Tele	RBS	TeleCheck	
account_encoder_id	N/A	See notes	N/A	R	N/A	<i>Chase Paymentech Solutions:</i> Required for encoded account numbers.
bill_address1	See notes	See notes	See notes	R	See notes	<i>All Processors:</i> Required for stand-alone credits. Optional for follow-on credits.
bill_address2	O	O	O	See notes	O	<i>RBS WorldPay Atlanta:</i> Required for stand-alone credits. Optional for follow-on credits.
bill_city	See notes	See notes	See notes	R	See notes	<i>All Processors:</i> Required for stand-alone credits. Optional for follow-on credits.
bill_country	See notes	See notes	See notes	R	N/A	Required for stand-alone credits. Optional for follow-on credits.
bill_state	See notes	See notes	See notes	R	See notes	<i>All Processors:</i> Required for stand-alone credits. Optional for follow-on credits.
bill_zip	See notes	See notes	See notes	See notes	See notes	<i>All Processors:</i> Required for stand-alone credits. Optional for follow-on credits.
currency	R	R	R	R	R	
customer_email	See notes	See notes	See notes	R	See notes	<i>All Processors:</i> Required for stand-alone credits. Optional for follow-on credits.
customer_firstname	See notes	See notes	See notes	R	See notes	<i>All Processors:</i> Required for stand-alone credits. Optional for follow-on credits.

Table 12 Electronic Check Credit Request Fields for the SCMP API (Continued)

Field	Required /Optional for Merchant-Provided Data				Required / Optional for Processor-Managed Data	Notes
	ACH	Chase	Tele	RBS	TeleCheck	
customer_ipaddress	O	O	O	O	O	
customer_lastname	See notes	See notes	See notes	R	See notes	<i>All Processors:</i> Required for stand-alone credits. Optional for follow-on credits.
customer_phone	O	O	O	R	O	
date_of_birth	N/A	N/A	N/A	O	N/A	
e_commerce_indicator	O	N/A	O	See notes	O	<i>All Processors:</i> Required for stand-alone credits. Optional for follow-on credits.
ecp_account_no	See notes	See notes	See notes	See notes	N/A	<i>All Processors:</i> Required for stand-alone credits. Optional for follow-on credits.
ecp_account_type	See notes	See notes	See notes	See notes	N/A	<i>All Processors:</i> Required for stand-alone credits. Optional for follow-on credits.
ecp_check_no	N/A	O	O	O	O	<i>All Processors:</i> Required for follow-on credits. Optional for stand-alone credits.
ecp_debit_request_id	See notes	See notes	See notes	See notes	See notes	<i>All Processors:</i> Required for follow-on credits.
ecp_debit_request_token	See notes	See notes	See notes	O	See notes	<i>All Processors:</i> For follow-on credits, you must include either this field or order_request_token in your request depending on when you implemented request tokens. See the information about request tokens in Getting Started with CyberSource Advanced .
ecp_payment_info	N/A	N/A	N/A	R	N/A	
ecp_rdfi	See notes	See notes	See notes	See notes	N/A	<i>All Processors:</i> Required for stand-alone credits. Optional for follow-on credits.

Table 12 Electronic Check Credit Request Fields for the SCMP API (Continued)

Field	Required /Optional for Merchant-Provided Data				Required / Optional for Processor-Managed Data	Notes
	ACH	Chase	Tele	RBS	TeleCheck	
ecp_ref_no	O	O	O	See notes	O	<i>RBS WorldPay Atlanta and TeleCheck: Required for stand-alone credits.</i>
ecp_settlement_method	N/A	O	N/A	O	N/A	
grand_total_amount	See notes	See notes	See notes	R	See notes	<i>All Processors: You must include either this field or offerN: amount in your request.</i>
ics_applications	R	R	R	R	R	
merchant_descriptor	N/A	O	N/A	O	N/A	
merchant_id	R	R	R	R	R	
merchant_ref_number	R	R	R	R	R	
offerN: amount	See notes	See notes	See notes	O	See notes	<i>All Processors: You must include either this field or grand_total_amount in your request.</i>
offerN: merchant_product_sku	O	O	O	O	O	
offerN: product_code	O	O	O	O	O	
offerN: product_name	O	O	O	O	O	
offerN: quantity	O	O	O	O	O	
offerN: tax_amount	O	O	O	O	O	
order_request_token	See notes	See notes	See notes	O	See notes	<i>All Processors: For follow-on credits, you must include either this field or ecp_debit_request_token in your request depending on when you implemented request tokens. See the information about request tokens in Getting Started with CyberSource Advanced.</i>

Table 12 Electronic Check Credit Request Fields for the SCMP API (Continued)

Field	Required /Optional for Merchant-Provided Data				Required / Optional for Processor-Managed Data	Notes
	ACH	Chase	Tele	RBS	TeleCheck	
partial_payment_id	N/A	N/A	See notes	O	See notes	<i>TeleCheck</i> : Required for partial credits.
subscription_id	O	O	O	O	O	<i>All Processors</i> : When you include this field, many other fields are no longer required. See “Payment Tokenization” on page 62 and “Recurring Billing” on page 63.
timeout	O	O	O	O	O	

Authenticate

Supported processors:

- RBS WorldPay Atlanta

The authentication service is an optional service you can request for RBS WorldPay Atlanta.

To request an electronic check authentication, set the **ics_applications** field to **ics_ecp_authenticate**. The following fields are required when requesting this service:

- **ics_applications**
- **merchant_id**
- **merchant_ref_number**
- **customer_firstname**
- **customer_lastname**
- **bill_address1**
- **bill_city**
- **bill_state**
- **bill_country**
- **bill_zip**

The authentication service:

- Validates customer information such as name, address, and date of birth and returns information to you in the `ecp_authenticate_checkpoint_summary` field.
- Provides you with consumer fraud information to help protect you against fraudulent transactions. Information is returned to you in the `ecp_authenticate_fraud_shield_indicators` field.

For more information, see:

- Appendix I, “[Check Point Summary Codes](#),” on page 135
- Appendix J, “[Fraud Shield Indicator Codes](#),” on page 149

Voids

Topics:

[Requesting a Void with the SCMP API](#)
[Requirements for the Void Request Fields](#)

Requesting a Void with the SCMP API

Supported processors:

- Chase Paymentech Solutions
- RBS WorldPay Atlanta
- TeleCheck

A void cancels an electronic check debit or credit request that you have submitted to CyberSource. A transaction can be voided only if CyberSource has not already submitted the debit or credit information to your processor. CyberSource usually submits transaction information to your processor once a day, so your window for successfully performing a void is relatively small. CyberSource will decline your void request if the debit or credit information has already been sent to the processor. You cannot undo a void and you cannot perform a follow-on credit for a debit that has been voided.

For RBS WorldPay Atlanta, voids are supported for credit requests only.

To request a void for an electronic check debit or credit with the SCMP API, set the `ics_applications` field to `ics_void`. When you request a void, do not request any other services at the same time.

A void is a follow-on transaction that uses the **request_id** and **request_token** returned from a previous **ics_ecp_debit** or **ics_ecp_credit** request to link the void to the debit or credit. Send the request ID value in the **void_request_id** field and send the request token value in the **order_request_token** field. CyberSource uses these values to look up the customer's billing and account information from the original debit or credit, which means that you are not required to include those fields in the **ics_void** request.

Requirements for the Void Request Fields

The following table lists the requirements for the fields used to request a void. For detailed descriptions of these fields, see "[Request-Level Fields](#)" on page 100.

Table 13 Void Request Fields for the Simple Order API

Field	Required /Optional for Merchant-Provided Data				Required / Optional for Processor-Managed Data	Notes
	ACH	Chase	RBS	Tele	TeleCheck	
merchant_id	N/A	R	R	R	R	
merchant_ref_number	N/A	R	R	R	R	
ics_applications	N/A	R	R	R	R	
order_request_token	N/A	See notes	See notes	See notes	See notes	<i>All Processors:</i> You must include either this field or void_request_token in your request depending on when you implemented request tokens. See the information about request tokens in Getting Started with CyberSource Advanced .
void_request_id	N/A	R	R	R	R	
void_request_token	N/A	See notes	O	See notes	See notes	<i>All Processors:</i> You must include either this field or order_request_token in your request depending on when you implemented request tokens. See the information about request tokens in Getting Started with CyberSource Advanced .

Chapter 4

Optional Features for the Simple Order API

Topics:

[Corporate Checks](#)
[Deferred and Partial Payments](#)
[Encoded Account Numbers](#)
[Merchant Descriptors](#)
[Multiple Partial Credits](#)
[Non-Sufficient Funds \(NSF\) Service](#)
[Payment Tokenization](#)
[Recurring Billing](#)
[Settlement Delivery Methods](#)

Corporate Checks

Applicable services:

- Debit

Supported processors:

- RBS WorldPay Atlanta
- TeleCheck

To process corporate checks with TeleCheck, include these fields in your debit request:

- **billTo_driversLicenseNumber** and **billTo_driversLicenseState**
or
- **billTo_company** and **billTo_companyTaxID**

Deferred and Partial Payments

Applicable services:

- Debit

Supported processors:

- Chase Paymentech Solutions
- TeleCheck

Definitions:

- *Deferred payment*—If there is a delay between the time you take the order and the time you ship the product, you need to defer your payment request.
- *Partial payment*—If a customer orders multiple products but you ship them separately on different dates, you need to perform multiple partial payments as you ship the products.

Chase Paymentech Solutions

- 1 For the first debit request, set **ecDebitService_paymentMode** to 1 to indicate that the debit uses deferred payment and full payment or partial payments will occur later. The default value of 0 indicates a normal debit with immediate payment.
- 2 When you are ready to process a payment, regardless of whether it's for the full amount or a partial amount, send another debit request and set **ecDebitService_paymentMode** to 2 to indicate that you are triggering a payment.
- 3 Repeat Step 2 for each partial payment for the order.

TeleCheck

- 1 For the first debit request, set **ecDebitService_paymentMode** to 1 to indicate that the debit uses deferred payment and the full payment or partial payments will occur later. The default value of 0 indicates a normal debit with immediate payment.
- 2 When you are ready to process a payment, regardless of whether it's for the full amount or a partial amount, send another debit request and do the following:
 - Set **ecDebitService_paymentMode** to 2 to indicate that you are triggering a payment.
 - Set **ecDebitService_referenceNumber** to the same value that you sent in the original debit request in Step 1. TeleCheck uses this value to track all of the transactions related to a given order.
 - If the payment is partial, set **ecDebitService_partialPaymentID** to a value of your choice that is unique within the scope of the order. For example, if this is the first partial payment, set the field to 1. If this is the second partial payment, set the field to 2, and so on. TeleCheck uses this value to identify the different payments related to the order.
- 3 Repeat Step 2 for each partial payment for the order.

Encoded Account Numbers

Applicable services:

- Debit
- Credit

Supported processors:

- Chase Paymentech Solutions

Depending on what type of business you are in, you might be eligible to acquire from a bank a list of customers who have accounts with that bank. The list does not include the customers' account numbers, but instead includes encoded account numbers. Some processors refer to this type of program as issuer encryption and to the numbers as encrypted account numbers. This type of program is designed to protect customer data according to the provisions of the Graham Leach Bliley Act.

When processing a payment or credit for one of these customers, you use the encoded account number instead of the customer's account number. The bank then matches the encoded account number to the customer's account number when processing the payment.

You must contact the processor to obtain information required for their account number encryption program and you must have a relationship with the bank to acquire their list of customers.

To process an electronic check debit or credit with an encoded account number:

- Set **check_accountNumber** to the encoded account number.
- Set **check_accountEncoderID** to the value assigned to the bank that supplied the customer information. Contact your processor to obtain the ID for the bank.

Merchant Descriptors

Applicable services:

- Debit
- Credit

Supported processors:

- Chase Paymentech Solutions

You can provide a merchant descriptor that will be displayed on the customer's bank account statement. The descriptor includes your company's name and a description of the product or service that was purchased.

Before sending a merchant descriptor with a debit or credit request, check with your processor to find out if you need to register your merchant descriptor information with them.

The **invoiceHeader_merchantDescriptor** field requires a particular format:

- **Characters 1–15:** Name of your company. If the name is less than 15 characters, use spaces to fill in the full 15 characters. If the name is greater than 15 characters, provide only the first 15 characters of the name.
- **Characters 16–25:** Description of the product or service.

Note If you use more than one consecutive space, extra spaces will be removed.

Multiple Partial Credits

Applicable services:

- Credit

Supported processors:

- RBS WorldPay Atlanta
- TeleCheck

When performing multiple partial credits:

- The amount of each individual credit cannot exceed the debit amount.
- The total amount of all the credits cannot exceed the debit amount.

In your follow-on credit request, use the **requestID** from the debit. Do not use the **requestID** from a previous partial credit. For each partial credit, set **ecCreditService_partialPaymentID** to a value of your choice that is unique within the scope of the order. The processor uses the payment IDs to identify the credits that are related to an order.

If you performed partial payments for this order, you specified a unique value for **ecDebitService_partialPaymentID** for each payment. You cannot reuse any of those values for the order's partial credits. For example, if you used 1 and 2 for the partial payments, you must use different values, such as 3 and 4, for the partial credits.

Non-Sufficient Funds (NSF) Service

Applicable services:

- Debit

Supported processors:

- CyberSource ACH Service

A non-sufficient funds (NSF) return occurs when the customer's bank account does not have sufficient funds to cover a specific electronic check transaction. CyberSource does not automatically resubmit charges returned from a customer's bank due to NSF. You can resubmit transactions returned as NSF up to two additional times for a total of three submissions: original submission plus two resubmissions. Continued attempts after this point may result in a fine and possible sanctions from the National Automated Clearing House Association (NACHA).

Contact your ODFI to enable the NSF service at your bank.

Payment Tokenization

Applicable services:

- Debit
- Credit

Supported processors:

- Chase Paymentech Solutions
- CyberSource ACH Service
- RBS WorldPay Atlanta
- TeleCheck

If you are using Payment Tokenization, you can process a debit or credit by using information that is stored in a customer profile. CyberSource uses the subscription ID to reference the customer profile information in the CyberSource database. Instead of providing all the information that is normally required for a transaction, you only need to provide the following values:

- Merchant ID
- Merchant reference code
- Amount of the payment or credit
- Subscription ID

You can override most of the information stored in the customer profile by including the relevant API fields in the debit or credit request. For example, you could provide a different billing or shipping address in the request. You cannot override the account number.

For complete information about Payment Tokenization, see [Payment Tokenization Using Secure Storage](#).

Recurring Billing

Applicable services:

- Debit
- Credit

Supported processors:

- Chase Paymentech Solutions
- CyberSource ACH Service
- RBS WorldPay Atlanta
- TeleCheck

If you are using Recurring Billing, you can process a debit or credit by using information that is stored in a subscription. CyberSource uses the subscription ID to reference the subscription information in the CyberSource database. Instead of providing all the information that is normally required for a transaction, you only need to provide the following values:

- Merchant ID
- Merchant reference code
- Amount of the payment or credit
- Subscription ID

You can override most of the information stored in the subscription by including the relevant API fields in the debit or credit request. For example, you could provide a different billing or shipping address in the request. You cannot override the account number.

For complete information about Recurring Billing, see the [Recurring Billing Implementation Guide](#).

Settlement Delivery Methods

Applicable services:

- Debit
- Credit

Supported processors:

- Chase Paymentech Solutions

You must specify a default method for delivering settlements to and from the customer's bank. For debits, you can use the **ecDebitService_settlementMethod** field to override the default method for a single transaction. For credits, you can use the **ecCreditService_settlementMethod** field to override the default method for a single transaction.

The following delivery methods are available:

- Automated Clearing House (ACH) for U.S. accounts or the Canadian Payment Association (CPA) for Canadian accounts
The transaction is deposited through the ACH or CPA. If the check fails the validation or verification process, the transaction will be rejected.
- Facsimile draft
The transaction is deposited as a facsimile draft. Only available for transactions in U.S. dollars. Use this method when the issuing bank is not an ACH member.
- Best possible
The transaction is deposited through the ACH system unless the customer's bank is not an ACH participant, in which case a facsimile draft is created and deposited on your behalf. Only available for transactions in U.S. dollars.

Chapter 5

Optional Features for the SCMP API

Topics:

[Corporate Checks](#)
[Deferred and Partial Payments](#)
[Encoded Account Numbers](#)
[Merchant Descriptors](#)
[Multiple Partial Credits](#)
[Non-Sufficient Funds \(NSF\) Service](#)
[Payment Tokenization](#)
[Recurring Billing](#)
[Settlement Delivery Methods](#)

Corporate Checks

Applicable services:

- Debit

Supported processors:

- RBS WorldPay Atlanta
- TeleCheck

To process corporate checks with TeleCheck, include these fields in your debit request:

- **driver_license_no** and **driver_license_state**
or
- **company_name** and **bill_company_tax_id**

Deferred and Partial Payments

Applicable services:

- Debit

Supported processors:

- Chase Paymentech Solutions
- TeleCheck

Definitions:

- *Deferred payment*—If there is a delay between the time you take the order and the time you ship the product, you need to defer your payment request.
- *Partial payment*—If a customer orders multiple products but you ship them separately on different dates, you need to perform multiple partial payments as you ship the products.

Chase Paymentech Solutions

- 1 For the first debit request, set **ecp_payment_mode** to 1 to indicate that the debit uses deferred payment and full payment or partial payments will occur later. The default value of 0 indicates a normal debit with immediate payment.
- 2 When you are ready to process a payment, regardless of whether it's for the full amount or a partial amount, send another debit request and set **ecp_payment_mode** to 2 to indicate that you are triggering a payment.
- 3 Repeat Step 2 for each partial payment for the order.

TeleCheck

- 1 For the first debit request, set **ecp_payment_mode** to 1 to indicate that the debit uses deferred payment and the full payment or partial payments will occur later. The default value of 0 indicates a normal debit with immediate payment.
- 2 When you are ready to process a payment, regardless of whether it's for the full amount or a partial amount, send another debit request and do the following:
 - Set **ecp_payment_mode** to 2 to indicate that you are triggering a payment.
 - Set **ecp_ref_no** to the same value that you sent in the original debit request in Step 1. TeleCheck uses this value to track all of the transactions related to a given order.
 - If the payment is partial, set **partial_payment_id** to a value of your choice that is unique within the scope of the order. For example, if this is the first partial payment, set the field to 1. If this is the second partial payment, set the field to 2, and so on. TeleCheck uses this value to identify the different payments related to the order.
- 3 Repeat Step 2 for each partial payment for the order.

Encoded Account Numbers

Applicable services:

- Debit
- Credit

Supported processors:

- Chase Paymentech Solutions

Depending on what type of business you are in, you might be eligible to acquire from a bank a list of customers who have accounts with that bank. The list does not include the customers' account numbers, but instead includes encoded account numbers. Some processors refer to this type of program as issuer encryption and to the numbers as encrypted account numbers. This type of program is designed to protect customer data according to the provisions of the Graham Leach Bliley Act.

When processing a payment or credit for one of these customers, you use the encoded account number instead of the customer's account number. The bank then matches the encoded account number to the customer's account number when processing the payment.

You must contact the processor to obtain information required for their account number encryption program and you must have a relationship with the bank to acquire their list of customers.

To process an electronic check debit or credit with an encoded account number:

- Set **check_accountNumber** to the encoded account number.
- Set **check_accountEncoderID** to the value assigned to the bank that supplied the customer information. Contact your processor to obtain the ID for the bank.

Merchant Descriptors

Applicable services:

- Debit
- Credit

Supported processors:

- Chase Paymentech Solutions

You can provide a merchant descriptor that will be displayed on the customer's bank account statement. The descriptor includes your company's name and a description of the product or service that was purchased.

Before sending a merchant descriptor with a debit or credit request, check with your processor to find out if you need to register your merchant descriptor information with them.

The **merchant_descriptor** field requires a particular format:

- **Characters 1–15:** Name of your company. If the name is less than 15 characters, use spaces to fill in the full 15 characters. If the name is greater than 15 characters, provide only the first 15 characters of the name.
- **Characters 16–25:** Description of the product or service.

Note If you use more than one consecutive space, extra spaces will be removed.

Multiple Partial Credits

Applicable services:

- Credit

Supported processors:

- RBS WorldPay Atlanta
- TeleCheck

When performing multiple partial credits:

- The amount of each individual credit cannot exceed the debit amount.
- The total amount of all the credits cannot exceed the debit amount.

In your follow-on credit request, use the **request_id** from the debit. Do not use the **request_id** from a previous partial credit. For each partial credit, set **partial_payment_id** to a value of your choice that is unique within the scope of the order. The processor uses the payment IDs to identify the credits that are related to an order.

If you performed partial payments for this order, you specified a unique value for **partial_payment_id** for each payment. You cannot reuse any of those values for the order's partial credits. For example, if you used 1 and 2 for the partial payments, you must use different values, such as 3 and 4, for the partial credits.

Non-Sufficient Funds (NSF) Service

Applicable services:

- Debit

Supported processors:

- CyberSource ACH Service

A non-sufficient funds (NSF) return occurs when the customer's bank account does not have sufficient funds to cover a specific electronic check transaction. CyberSource does not automatically resubmit charges returned from a customer's bank due to NSF. You can resubmit transactions returned as NSF up to two additional times for a total of three submissions: original submission plus two resubmissions. Continued attempts after this point may result in a fine and possible sanctions from the National Automated Clearing House Association (NACHA).

Contact your ODFI to enable the NSF service at your bank.

Payment Tokenization

Applicable services:

- Debit
- Credit

Supported processors:

- Chase Paymentech Solutions
- CyberSource ACH Service
- RBS WorldPay Atlanta
- TeleCheck

If you are using Payment Tokenization, you can process a debit or credit by using information that is stored in a customer profile. CyberSource uses the subscription ID to reference the customer profile information in the CyberSource database. Instead of providing all the information that is normally required for a transaction, you only need to provide the following values:

- Merchant ID
- Merchant reference number
- Amount of the payment or credit
- Subscription ID

You can override most of the information stored in the customer profile by including the relevant API fields in the debit or credit request. For example, you could provide a different billing or shipping address in the request. You cannot override the account number.

For complete information about Payment Tokenization, see [Payment Tokenization Using Secure Storage](#).

Recurring Billing

Applicable services:

- Debit
- Credit

Supported processors:

- Chase Paymentech Solutions
- CyberSource ACH Service
- RBS WorldPay Atlanta
- TeleCheck

If you are using Recurring Billing, you can process a debit or credit by using information that is stored in a subscription. CyberSource uses the subscription ID to reference the subscription information in the CyberSource database. Instead of providing all the information that is normally required for a transaction, you only need to provide the following values:

- Merchant ID
- Merchant reference number
- Amount of the payment or credit
- Subscription ID

You can override most of the information stored in the subscription by including the relevant API fields in the debit or credit request. For example, you could provide a different billing or shipping address in the request. You cannot override the account number.

For complete information about Recurring Billing, see the [Recurring Billing Implementation Guide](#).

Settlement Delivery Methods

Applicable services:

- Debit
- Credit

Supported processors:

- Chase Paymentech Solutions

You must specify a default method for delivering settlements to and from the customer's bank. You can use the **ecp_settlement_method** field to override the default method for a single transaction.

The following delivery methods are available:

- Automated Clearing House (ACH) for U.S. accounts or the Canadian Payment Association (CPA) for Canadian accounts
The transaction is deposited through the ACH or CPA. If the check fails the validation or verification process, the transaction will be rejected.
- Facsimile draft
The transaction is deposited as a facsimile draft. Only available for transactions in U.S. dollars. Use this method when the issuing bank is not an ACH member.
- Best possible
The transaction is deposited through the ACH system unless the customer's bank is not an ACH participant, in which case a facsimile draft is created and deposited on your behalf. Only available for transactions in U.S. dollars.

Chapter 6

Testing Electronic Check Services

Topics:

[Requirements for Testing](#)

[Testing Chase Paymentech Solutions Transactions](#)

[Testing CyberSource ACH Service Transactions](#)

[Going Live](#)

Requirements for Testing

Important Before you can test, you must contact Customer Support to activate Electronic Check Services and configure your account for electronic check testing. You must also contact your processor to set up your processor account.

- Use your regular merchant ID to perform testing.
- Use the test server `ics2wstest.ic3.com`.
- Use a real city and state, as well as the correct postal code for that city and state.
- Use a real combination for the area code and telephone number.
- Use a non-existent account and domain name for the customer's email address. For example: `random@example.com`.

Testing Chase Paymentech Solutions Transactions

Successful Transactions

Use the data in the following table to simulate successful debits and credits for Chase Paymentech Solutions.

Table 14 Test Data for Chase Paymentech Solutions Debits and Credits

Field	Test Values	Required / Optional
check_accountNumber for the Simple Order API	For transactions in U.S. or Canadian dollars:	Required
	• 4100	
ecp_account_no for the SCMP API	• 4101	
	• 4102	
	• 4103	
check_accountType for the Simple Order API	For transactions in U.S. or Canadian dollars:	Required
	• C	
ecp_account_type for the SCMP API	For transactions in U.S. dollars only:	
	• S	
	• X	
check_bankTransitNumber for the Simple Order API	For transactions in U.S. dollars only:	Required
	• 121042882	
ecp_rdfi for the SCMP API	• 121107882	
	• 071923284	
	• 122101191	
	For transactions in Canadian dollars, use any eight-digit number.	
ecDebitService_settlementMethod for the Simple Order API	For transactions in U.S. or Canadian dollars:	Optional
	• A	
ecp_settlement_method for the SCMP API	For transactions in U.S. dollars only:	
	• B	
	• F	
ecDebitService_verificationLevel for the Simple Order API	For transactions in U.S. or Canadian dollars:	Optional
	• 1	
ecp_verification_level for the SCMP API	For transactions in U.S. dollars only:	
	• 2	

Testing Chase Paymentech Solutions Declines

For Chase Paymentech Solutions, you can simulate electronic check declines using specific bank account numbers for debits. For a list of these values and the expected results, go to the [Testing Information page](#) on the Support Center.

Testing CyberSource ACH Service Transactions

Use the data in the following table to simulate ACH verification by requesting a debit for the CyberSource ACH Service. As an alternative, you can simulate ACH verification by requesting a credit: the reply fields will be for the credit service instead of the debit service.

Table 15 ACH Verification Test Data

	Triggers		Reply Fields			
Field:	Account Number	Routing Number	Mapped ACH Verification Code	Raw ACH Verification Code	Corrected Account Number	Corrected Routing Number
Simple Order API Field Name:	check_account Number	check_bankTransit Number	ecDebitReply_verification Code	ecDebitReply_verification CodeRaw	ecDebitReply_corrected Account Number	ecDebitReply_corrected Routing Number
SCMP API Field Name:	ecp_account_no	ecp_rdfi	ecp_debit_verification_code	ecp_debit_verification_code_raw	ecp_debit_corrected_account_number	ecp_debit_corrected_routing_number
	12345678	112200439	00	1	N/A	N/A
	001111111111	011000028	01	2	00111111	N/A
	1231231230	231385154	00	3	N/A	N/A
	123123123	231385154	00	4	N/A	N/A
	00111111	011201762	02	5	N/A	011201830
	001234567895	011400039	03	6	1234567895	011401533
	01111111	011301073	02	7	N/A	211070175
	1231231230	011001742	02	8	N/A	011000138
	1231231230	231382704	04	9 *	N/A	N/A
	12345678	115101438	04	10 *	N/A	N/A

* See the following table for the reply values for this error.

Table 16 ACH Verification Error Reply Values

Raw ACH Verification Code	Simple Order API	SCMP API
9	decision=REJECT reasonCode=388	ics_rcode=0 ics_rflag=DACHVERIFICATION ics_rmsg=ACH Verification Declined: Routing number is not ACH-able. No substitute available.
10	decision=REJECT reasonCode=388	ics_rcode=0 ics_rflag=DACHVERIFICATION ics_rmsg=Verification Declined: Routing number formatted correctly, but not found during lookup.

Testing RBS WorldPay Atlanta

See the test data on the Support Center:

http://www.cybersource.com/support_center/implementation/testing_info/

Going Live

You must go live with CyberSource before you begin to accept productions transactions. When you go live, your account is updated so that you can send transactions to the CyberSource production server. If you have not already done so, you will need to provide your banking information to CyberSource so that your processor can deposit funds to your merchant bank account.

To go live:

- 1 Go to the CyberSource Knowledgebase at <http://www.cybersource.com/esupport>.
- 2 Submit a question by clicking the **Submit a Question** link.
- 3 Log in with your merchant ID and password. These are the same values you use to log in to the Business Center.
- 4 On the **Submit a Question** page, fill in the contact information.
- 5 For the **General Topic**, select **Getting Set Up on CyberSource**.
- 6 For the **Specific Topic**, select **Go Live**.
- 7 For the **Subject**, enter "Go Live".
- 8 In the question field, indicate that you would like to go live.
- 9 If you have not already submitted your banking information to CyberSource, include the information in the question field and select the check box for **This question contains sensitive banking information**.

10 Click *Submit question*.

You will receive an email with your support ticket number, and a CyberSource representative will contact you to complete the process.

11 After CyberSource has confirmed that you are live, you need to update your system to send requests to the production server instead of the test server. See your client documentation for instructions.

After you go live, use real account numbers and other data to test every currency and CyberSource service that your integration supports. Because these are real transactions, use small offer amounts, such as one dollar, to do the tests. If you have more than one merchant ID, test each one separately. Process a debit and credit for each configuration, and use your bank statements to verify that money is deposited into and withdrawn from your merchant bank account.

Appendix A

Fields for the Simple Order API

Topics:

[Formatting Restrictions](#)

[Data Type Definitions](#)

[Request Fields](#)

[Reply Fields](#)

Formatting Restrictions

Unless otherwise noted, all field names are case sensitive and the fields accept special characters such as @, #, and %.

Note The values of the **item_#** fields must not contain carets (^) or colons (:), because these characters are reserved for use by the ICS services. The values of all request fields must not contain newlines or carriage returns. However, they can contain embedded spaces and any other printable characters. All leading and trailing spaces will be removed.

Data Type Definitions

For more information about these data types, see the [World Wide Web Consortium \(W3C\) XML Schema Part 2: Datatypes specification](#).

- **Integer** — A whole number. {..., -3, -2, -1, 0, 1, 2, 3, ...}
- **String** — Contains letters, numbers, spaces, and special characters, such as @ and #.

Request Fields

The following table describes the request fields for the Electronic Check Services. The requirements for these fields are specified separately:

- Debit service—See “[Requirements for the Debit Request Fields](#)” on page 17.
- Credit service—See “[Requirements for the Credit Request Fields](#)” on page 26.
- Void service—See “[Requirements for the Void Request Fields](#)” on page 32.

Note If you are using Payment Tokenization or Recurring Billing and you include a subscription ID in your request, many of the fields in the following table that are normally required for a debit or credit become optional. See “[Payment Tokenization](#)” on page 62 and “[Recurring Billing](#)” on page 63.

Table 17 Request Fields for the Simple Order API

Field Name	Description	Data Type & Length
billTo_city	City in the billing address.	String (50)
billTo_company	Name of the customer's company.	String (40)
billTo_companyTaxID	Company's tax identifier.	String with numbers only (9)
billTo_country	Country in the billing address. Use the two-character ISO country codes .	String (2)
billTo_dateOfBirth	Date of birth of the customer. Format: YYYY-MM-DD or YYYYMMDD	String (10)
billTo_driversLicenseNumber	Driver's license number of the customer.	String (30)
billTo_driversLicenseState	State or province where the customer's driver's license was issued. Use the two-character codes .	String (2)
billTo_email	Customer's email address, including the full domain name. Format: <i>name@host.domain</i>	String (255)
billTo_firstName	Customer's first name. If the first name is unavailable or inapplicable, such as for a corporate account, enter a dummy value such as NA.	String (60)
billTo_ipAddress	IP address for the customer. Example: 10.1.27.63.	String (15)

Table 17 Request Fields for the Simple Order API (Continued)

Field Name	Description	Data Type & Length
billTo_lastName	Customer's last name. If the transaction is for a corporate account, use this field for the company name.	String (60)
billTo_phoneNumber	Customer's phone number.	String (15)
billTo_postalCode	Postal code for the billing address. The postal code must consist of 5 to 9 digits. If the billing country is the U.S., the 9-digit postal code must follow this format: [5 digits][dash][4 digits] Example: 12345-6789 If the billing country is Canada, the 6-digit postal code must follow this format: [alpha][numeric][alpha][space] [numeric][alpha][numeric] Example: A1B 2C3	String (10)
billTo_state	State in the billing address. Use the two-character codes .	String (2)
billTo_street1	First line of the billing street address.	String (60)
billTo_street2	Second line of the billing street address. Used for additional address information. Example: Attention: Accounts Payable	String (60)
businessRules_declineAVSFlags	List of eFunds ID Verification results that should result in a rejected request. Use a space to separate each value in the list. See the eFunds section in "Verification and Validation" on page 12. Example: N U Important The value N must be included in the list even though it represents the default behavior.	String (255)
check_accountEncoderID	Identifier for the bank that provided the customer's encoded account number. To obtain the bank identifier, contact your processor. See " Encoded Account Numbers " on page 59.	String (3)
check_accountNumber	Account number. When processing encoded account numbers, use this field for the encoded account number.	String with numbers only (17)

Table 17 Request Fields for the Simple Order API (Continued)

Field Name	Description	Data Type & Length
check_accountType	Account type. Possible values: • C: Checking • S: Savings (U.S. dollars only) • X: Corporate checking (U.S. dollars only)	String (1)
check_bankTransitNumber	Bank routing number. This is also called the transit number.	String with numbers only (9)
check_checkNumber	Check number.	String with numbers only (8)
check_paymentInfo	Additional payment information on the customer's statement.	String (80)
check_secCode	Code that specifies the authorization method for the transaction. Possible values: • CCD : Corporate cash disbursement—A charge or credit against a business checking account. You can use one-time or recurring CCD transactions to transfer funds to or from a corporate entity. A standing authorization is required for recurring transactions. • PPD : Prearranged payment and deposit entry—A charge or credit against a personal checking or savings account. You can originate a PPD entry only when the payment and deposit terms between you and the customer are prearranged. A written authorization from the customer is required for one-time transactions and a written standing authorization is required for recurring transactions. • TEL : Telephone-initiated entry—A one-time charge against a personal checking or savings account. You can originate a TEL entry only when there is a business relationship between you and the customer or when the customer initiates a telephone call to you. For a TEL entry, you must obtain a payment authorization from the customer over the telephone. There is no recurring billing option for TEL. • WEB : Internet-initiated entry—A charge against a personal checking or savings account. You can originate a one-time or recurring WEB entry when the customer initiates the transaction over the Internet. For a WEB entry, you must obtain payment authorization from the customer over the Internet.	String (3)

Table 17 Request Fields for the Simple Order API (Continued)

Field Name	Description	Data Type & Length
customer_email	Customer's email address.	Alphanumeric (50)
customer_phone	Customer's phone number including area code. Format: NNNNNNNNNN or NNN - NNN - NNNN.	Alphanumeric (12)
customer_ssn	Customer's social security number. Format: NNN - NN - NNNN.	Alphanumeric (11)
ecCreditService_commerceIndicator	Type of transaction. Possible values: - internet (default): e-commerce order placed using a Web site. - moto: Mail order or telephone order.	String (13)
ecCreditService_debitRequestID	The requestID for the debit that is being credited. See "Follow-on Credits and Stand-Alone Credits" on page 22.	String (26)
ecCreditService_debitRequestToken	The requestToken for the debit that is being credited. The field is an encoded string that contains no confidential information, such as an account number or card verification number. The string can contain up to 256 characters. This field is available only for merchants who implemented request tokens before July 2008. For new implementations, use the orderRequestToken field. See the information about request tokens in Getting Started with CyberSource Advanced .	String (256)
ecCreditService_partialPaymentID	Identifier for a partial credit. The value for each credit request must be unique within the scope of the order. See "Multiple Partial Credits" on page 61.	String (25)
ecCreditService_referenceNumber	Check reference number. Identifier used for tracking a request through to the payment processor for reconciliation. If you do not provide this value, CyberSource will generate a unique value and return it to you in the ecCreditReply_reconciliationID field. See the information about tracking orders in Getting Started with CyberSource Advanced .	TeleCheck: String (25) All other processors: String (60)
ecCreditService_run	Set this field to <code>true</code> to include ecCreditService in your request.	String (5)

Table 17 Request Fields for the Simple Order API (Continued)

Field Name	Description	Data Type & Length
ecCreditService_settlementMethod	Method used for settlement. Possible values: • A: Automated Clearing House (default for credits and for transactions using Canadian dollars) • F: Facsimile draft (U.S. dollars only) • B: Best possible (U.S. dollars only) (default if the field has not already been configured for your merchant ID) See “Settlement Delivery Methods” on page 64.	String (1)
ecDebitService_authenticateID	Identification number returned when an Authenticate request is processed and returned in subsequent monetary transactions such as with the debit service.	Numeric (32)
ecDebitService_commerceIndicator	Type of transaction. Possible values: • internet (default): e-commerce order placed using a Web site. • moto: Mail order or telephone order.	String (13)
ecDebitService_partialPaymentID	Identifier for a partial payment. The value for each debit request must be unique within the scope of the order. See “Deferred and Partial Payments” on page 58.	String (25)
ecDebitService_paymentInfo	Payment related information. This information is included on the customer’s statement.	Alphanumeric (80)
ecDebitService_paymentMode	Flag that indicates whether to process the payment. Use with deferred payments or eFunds ID Verification. See “Deferred and Partial Payments” on page 58 and the section about eFunds in “Verification and Validation” on page 12. Possible values: • 0: Standard debit with immediate payment (default) • 1: For eFunds, notifies CyberSource to not process the payment now. For deferred payments, indicates that this is a deferred payment and that you will send a debit request with ecDebitService_paymentMode=2 in the future. • 2: For deferred payments, indicates notification to initiate payment	Integer (1)

Table 17 Request Fields for the Simple Order API (Continued)

Field Name	Description	Data Type & Length
ecDebitService_referenceNumber	Check reference number. Identifier used for tracking a request through to the payment processor for reconciliation. If you do not provide this value, CyberSource will generate a unique value and return it to you in the ecDebitReply_reconciliationID field. See the information about tracking orders in Getting Started with CyberSource Advanced . For deferred payments with TeleCheck, set this field to the value you received in the ecDebitReply_reconciliationID field in the associated debit's reply. See "Deferred and Partial Payments" on page 58.	TeleCheck: String (25) All other processors: String (60)
ecDebitService_run	Set this field to <code>true</code> to include ecDebitService in your request.	String (5)
ecDebitService_settlementMethod	Method used for settlement. Possible values: • A: Automated Clearing House (default for credits and for transactions using Canadian dollars) • F: Facsimile draft (U.S. dollars only) • B: Best possible (U.S. dollars only) (default if the field has not already been configured for your merchant ID) See "Settlement Delivery Methods" on page 64.	String (1)
ecDebitService_verificationLevel	Level of fraud screening. Possible values: • 1: Validation—Default if the field has not already been configured for your merchant ID • 2: Verification • 3: Guarantee—U.S. dollars only • 4: eFunds ID Verification For a description of this feature and a list of supported processors, see "Verification and Validation" on page 12.	Integer (1)
invoiceHeader_merchantDescriptor	Merchant description that appears on the customer's bank statement. For a description of this feature, a list of supported processors, and special formatting requirements, see "Merchant Descriptors" on page 60.	String (25)
item_#_productCode	Type of product. This value is used to determine the category that the product is in: electronic, handling, physical, service, or shipping. The default value is <code>default</code> . See Appendix G, "Product Codes" , on page 131 for a list of valid values.	String (30)
item_#_productName	Name of the product.	String (30)

Table 17 Request Fields for the Simple Order API (Continued)

Field Name	Description	Data Type & Length
item_#_productSKU	Product's identifier code.	String (30)
item_#_quantity	Quantity of the product being purchased. The default value is 1.	Integer (10)
item_#_taxAmount	Total tax to apply to the product. This value cannot be negative. The item_#_taxAmount field is additive. For example, if you send one item with unitPrice of \$10.00 and taxAmount of \$0.80, and you send another item with unitPrice of \$20.00 and taxAmount of \$1.60, the total amount authorized will be for \$32.40, not \$30.00 with \$2.40 of tax included. The item_#_taxAmount and the item_#_unitPrice must be in the same currency. If you want to include item_#_taxAmount and also request the taxService service, see the Tax Calculation Implementation Guide, which is available on the Support Center.	String (15)
item_#_unitPrice	Per-item price of the product. This value cannot be negative. You can include a decimal point (.) in this field, but you cannot include any other special characters. The amount will be truncated at the request level to the correct number of decimal places.	String (15)
linkToRequest	Value that links the current request to a previous authorization request for a debit card or prepaid card. This value is useful when using multiple payment methods to complete an order. For details, see the information about partial authorizations in Credit Card Services for the Simple Order API .	String (26)
merchantID	Your merchant ID. Use the same merchantID for evaluation, testing, and production.	String (30)
merchantReferenceCode	Merchant-generated order reference or tracking number. See the information about tracking orders in Getting Started with CyberSource Advanced .	String (50)

Table 17 Request Fields for the Simple Order API (Continued)

Field Name	Description	Data Type & Length
orderRequestToken	The requestToken value returned from a previous request. This value links the previous request to the current follow-on request. This field is an encoded string that does not contain any confidential information, such as account numbers or card verification numbers. The string can contain up to 256 characters. See the information about request tokens in Getting Started with CyberSource Advanced.	String (256)
purchaseTotals_currency	Currency used for the order. Possible values: • CAD : Canadian dollars • USD : U.S. dollars	String (5)
purchaseTotals_grandTotalAmount	Grand total for the order. See the information about using items or a grand total in Getting Started with CyberSource Advanced.	String (15)
recurringSubscriptionInfo_subscriptionID	If you are using Payment Tokenization or Recurring Billing and you include this value in your request, many of the fields that are normally required for a debit or credit become optional. See “Payment Tokenization” on page 62 and “Recurring Billing” on page 63.	String (26)
voidService_run	Set this field to <code>true</code> to include voidService in your request.	String (5)
voidService_voidRequestID	The requestID of the debit or credit you want to void.	String (26)
voidService_voidRequestToken	The requestToken value returned from a previous request for a service that you want to void. You can void debits and credits that have not been batched or voided before. For example, if you want to void a credit, set this field to the requestToken value returned in the reply for the credit request. The field is an encoded string that contains no confidential information, such as an account number or card verification number. The string can contain up to 256 characters. This field is available only for merchants who implemented request tokens before July 2008. For new implementations, use the orderRequestToken field. See the information about request tokens in Getting Started with CyberSource Advanced.	String (256)

Reply Fields

Table 18 Reply Fields

Field Name	Description	Returned By	Data Type & Length
decision	Summarizes the result of the overall request. The field can contain one of the following values: • ACCEPT • ERROR • REJECT	All Electronic Check Services	String (6)
ecAuthenticateReply_checkpointSummary	Contains information about the parameters sent in the authenticate message, which is described in “Check Point Summary Codes” on page 135.	Authenticate Reply	String (600)
ecAuthenticateReply_fraudShieldIndicators	Contains information about the fraud checks performed, which is described in “Fraud Shield Indicator Codes” on page 149.	Authenticate Reply	String (300)
ecAuthenticateReply_processorResponse	Result code returned by the payment processor.	Authenticate Reply	String (6)
ecAuthenticateReply_reasonCode	Numeric value corresponding to the result of the overall request. See “Reason Codes for the Simple Order API” on page 125 for a list of possible values.	Authenticate Reply	Integer (5)
ecAuthenticateReply_reconciliationID	Reference number you use to reconcile CyberSource reports with processor reports. See the information about tracking orders in Getting Started with CyberSource Advanced .	Authenticate Reply	String (60)
ecAuthenticateReply_requestDateTime	Time of the request. The format is YYYY-MM-DDThh:mm:ssZ. For example, 2003-08-11T22:47:57 is equal to August 11, 2003, at 10:47:57 P.M. The T separates the date and the time, and the Z indicates UTC.	Authenticate Reply	String (20)
ecCreditReply_amount	Total amount submitted to the payment processor.	ecCreditReply	String (15)
ecCreditReply_correctedAccountNumber	Corrected account number from the ACH verification service, which is described in “ACH Verification” on page 23.	ecCreditReply	String (17)

Table 18 Reply Fields (Continued)

Field Name	Description	Returned By	Data Type & Length
ecCreditReply_ correctedRouting Number	Corrected routing number from the ACH verification service, which is described in "ACH Verification" on page 23.	ecCreditReply	String (9)
ecCreditReply_ ownerMerchantID	Merchant ID that was used to create the subscription or customer profile for which the service was requested. If you are enabled for Recurring Billing, this field is returned only if you are using subscription sharing and if your merchant ID is in the same merchant ID pool as the owner merchant ID. See the subscription sharing information in the Recurring Billing Implementation Guide. If you are enabled for Payment Tokenization, this field is returned only if you are using profile sharing and if your merchant ID is in the same merchant ID pool as the owner merchant ID. See the profile sharing information in Payment Tokenization Using Secure Storage.	ecCreditReply	String (30)
ecCreditReply_ processorResponse	Result code returned by the payment processor.	ecCreditReply	String (6)
ecCreditReply_ processorTransactionID	Transaction identifier or tracking ID returned by the payment processor. See the information about tracking orders in Getting Started with CyberSource Advanced .	ecCreditReply	String (87)
ecCreditReply_ reasonCode	A numeric value corresponding to the result of the credit request. See "Reason Codes for the Simple Order API" on page 125 for a list of possible values.	ecCreditReply	Integer (5)
ecCreditReply_ reconciliationID	Reference number for the transaction. For some processors, you can use this value to reconcile your CyberSource reports with your processor reports. See the information about tracking orders in Getting Started with CyberSource Advanced.	ecCreditReply	String (60)

Table 18 Reply Fields (Continued)

Field Name	Description	Returned By	Data Type & Length
ecCreditReply_ requestDateTime	Time when credit is requested. The format is <i>YYYY-MM-DDThh:mm:ssZ</i> . For example, 2003-08-11T22:47:57 is equal to August 11, 2003, at 10:47:57 P.M. The T separates the date and the time, and the Z indicates UTC.	ecCreditReply	String (20)
ecCreditReply_ settlementMethod	Method used to settle the credit. Possible values: • A: Automated Clearing House • B: Best possible • F: Facsimile	ecCreditReply	String (1)
ecCreditReply_ verificationCode	Indicates the results from the ACH verification service, which is described in " ACH Verification " on page 23. Possible values: • 00: Success: Account number and routing number are OK. • 01: Success: Account number was corrected; routing number is OK. • 02: Success: Routing number was corrected; account number is OK. • 03: Success: Account number and routing number were corrected. • 04: Declined: Routing number did not pass verification. • 98: Unavailable: Unable to perform ACH verification. • 99: Invalid: Response from ACH verification is invalid.	ecCreditReply	String (2)

Table 18 Reply Fields (Continued)

Field Name	Description	Returned By	Data Type & Length
ecCreditReply_verificationCodeRaw	Raw results from the ACH verification service, which is described in “ACH Verification” on page 23. Possible values: • 1 : Accepted: Routing number is valid. Account number is valid. • 2 : Accepted: Routing number is valid. Account number is invalid; use corrected account number. • 3 : Accepted: Routing number is valid. Account number is valid. • 4 : Accepted: Routing number is valid. Account number structure not recognized; account may be valid. • 5 : Accepted: Routing number is not usable for ACH; use corrected routing number. Account number is valid. • 6 : Accepted: Routing number is not usable for ACH; use corrected routing number. Account number is invalid; use corrected account number. • 7 : Accepted: Routing number is not usable for ACH; use corrected routing number. Account number is valid. • 8 : Accepted: Routing number is not usable for ACH; use corrected routing number. Account number structure not recognized; account may be valid. • 9 : Declined: Routing number is not usable for ACH; no corrected routing number available. • 10 : Declined: Routing number not found. • 11 : Declined: Invalid routing number.	ecCreditReply	String (2)
ecDebitReply_amount	Total amount submitted to the payment processor.	ecDebitReply	String (15)

Table 18 Reply Fields (Continued)

Field Name	Description	Returned By	Data Type & Length
ecDebitReply_avsCode	Overall result of the eFunds ID Verification. Possible values: • N: At least one customer information item had a negative match, and the other two did not have positive matches. • Y: At least one item had a positive match. • U: None of the items had either positive or negative matches, or an error occurred during processing.	ecDebitReply	String (1)
ecDebitReply_avsCodeRaw	Processor's raw match results for each of the customer information fields. See the eFunds section in " Verification and Validation " on page 12. To interpret the raw results, see your processor's specification for eFunds ID Verification.	ecDebitReply	String (2)
ecDebitReply_correctedAccountNumber	Corrected account number from the ACH verification service, which is described in " ACH Verification " on page 13.	ecDebitReply	String (17)
ecDebitReply_correctedRoutingNumber	Corrected routing number from the ACH verification service, which is described in " ACH Verification " on page 13.	ecDebitReply	String (9)
ecDebitReply_ownerMerchantID	Merchant ID that was used to create the subscription or customer profile for which the service was requested. If you are enabled for Recurring Billing, this field is returned only if you are using subscription sharing and if your merchant ID is in the same merchant ID pool as the owner merchant ID. See the subscription sharing information in the Recurring Billing Implementation Guide . If you are enabled for Payment Tokenization, this field is returned only if you are using profile sharing and if your merchant ID is in the same merchant ID pool as the owner merchant ID. See the profile sharing information in Payment Tokenization Using Secure Storage .	ecDebitReply	String (30)
ecDebitReply_processorResponse	Result code returned by the payment processor.	ecDebitReply	String (6)

Table 18 Reply Fields (Continued)

Field Name	Description	Returned By	Data Type & Length
ecDebitReply_processorTransactionID	Transaction identifier or tracking ID returned by the payment processor. See the information about tracking orders in Getting Started with CyberSource Advanced .	ecDebitReply	String (87)
ecDebitReply_reasonCode	A numeric value corresponding to the result of the debit request. See “ Reason Codes for the Simple Order API ” on page 125 for a list of possible values.	ecDebitReply	Integer (5)
ecDebitReply_reconciliationID	Reference number for the transaction. For some processors, you can use this value to reconcile your CyberSource reports with your processor reports. See the information about tracking orders in Getting Started with CyberSource Advanced .	ecDebitReply	String (60)
ecDebitReply_requestDateTime	Time when debit is requested. The format is YYYY-MM-DDThh:mm:ssZ. For example, 2003-08-11T22:47:57 is equal to August 11, 2003, at 10:47:57 P.M. The T separates the date and the time, and the Z indicates UTC.	ecDebitReply	String (20)
ecDebitReply_settlementMethod	Method used to settle the debit. Possible values: • A: Automated Clearing House • B: Best possible • F: Facsimile	ecDebitReply	String (1)

Table 18 Reply Fields (Continued)

Field Name	Description	Returned By	Data Type & Length
ecDebitReply_verificationCode	Results from the ACH verification service, which is described in "ACH Verification" on page 13. Possible values: • 00 : Success: Account number and routing number are OK. • 01 : Success: Account number was corrected; routing number is OK. • 02 : Success: Routing number was corrected; account number is OK. • 03 : Success: Account number and routing number were corrected. • 04 : Declined: Routing number did not pass verification. • 98 : Unavailable: Unable to perform ACH verification. • 99 : Invalid: Response from ACH verification is invalid.	ecDebitReply	String (2)

Table 18 Reply Fields (Continued)

Field Name	Description	Returned By	Data Type & Length
ecDebitReply_verificationCodeRaw	Raw results from the ACH verification service, which is described in “ACH Verification” on page 13. Possible values: • 1 : Accepted: Routing number is valid. Account number is valid. • 2 : Accepted: Routing number is valid. Account number is invalid; use corrected account number. • 3 : Accepted: Routing number is valid. Account number is valid. • 4 : Accepted: Routing number is valid. Account number structure not recognized; account may be valid. • 5 : Accepted: Routing number is not usable for ACH; use corrected routing number. Account number is valid. • 6 : Accepted: Routing number is not usable for ACH; use corrected routing number. Account number is invalid; use corrected account number. • 7 : Accepted: Routing number is not usable for ACH; use corrected routing number. Account number is valid. • 8 : Accepted: Routing number is not usable for ACH; use corrected routing number. Account number structure not recognized; account may be valid. • 9 : Declined: Routing number is not usable for ACH; no corrected routing number available. • 10 : Declined: Routing number not found. • 11 : Declined: Invalid routing number.	ecDebitReply	String (2)
ecDebitReply_verificationLevel	Level of screening for the request. Possible values: • 1: Validation • 2: Verification • 3: Guarantee • 4: eFunds ID Verification	ecDebitReply	Integer (1)

Table 18 Reply Fields (Continued)

Field Name	Description	Returned By	Data Type & Length
invalidField_0...N	Fields in the request that contained invalid data. These reply fields are included as an aid to software developers only. No attempt should be made to use these fields for end user interaction. See the information about missing and invalid fields in Getting Started with CyberSource Advanced .	All Electronic Check Services	String (100)
merchantReferenceCode	Order reference or tracking number that you provided in the request. If you included multi-byte characters in this field in the request, the returned value might contain corrupted characters.	All Electronic Check Services	String (50)
missingField_0...N	Required fields that were missing from the request. These reply fields are included as an aid to software developers only. No attempt should be made to use these fields for end user interaction. See the information about missing and invalid fields in Getting Started with CyberSource Advanced .	All Electronic Check Services	String (100)
purchaseTotals_currency	Currency used for the order. Possible values: • USD: U.S. dollars • CAD: Canadian dollars	All Electronic Check Services	String (5)
reasonCode	Numeric value corresponding to the result of the overall request. See " Reason Codes for the Simple Order API " on page 125 for a list of possible values.	All Electronic Check Services	Integer (5)
requestID	Identifier for the request.	All Electronic Check Services	String (26)
requestToken	Request token data created by CyberSource for each reply. You need to store the contents of this field so that you can retrieve and send it in follow-on requests. The field is an encoded string that contains no confidential information, such as an account or card verification number. The string can contain up to 256 characters.	All Electronic Check Services	String (256)

See the information about request tokens in [Getting Started with CyberSource Advanced](#).

Table 18 Reply Fields (Continued)

Field Name	Description	Returned By	Data Type & Length
voidReply_amount	Total amount of the void.	voidReply	String (15)
voidReply_currency	Currency used for the order. See the ISO currency codes .	voidReply	String (5)
voidReply_reasonCode	Numeric value corresponding to the result of the void request. See " Reason Codes for the Simple Order API " on page 125 for a list of possible values.	voidReply	Integer (5)
voidReply_requestDateTime	Time when void was requested. The format is <code>YYYY-MM-DDThh:mm:ssZ</code> . For example, <code>2005-08-11T22:47:57Z</code> is equal to August 11, 2005, at 10:47:57 P.M. The T separates the date and the time. The Z indicates UTC.	voidReply	String (20)

Appendix B

Fields for the SCMP API

Topics:

[Formatting Restrictions](#)

[Data Type Definitions](#)

[Request-Level Fields](#)

[Offer-Level Fields](#)

[Reply Fields](#)

Formatting Restrictions

Unless otherwise noted, all of the fields listed are order and case insensitive, and the fields accept special characters. For example: @, #, and %.

Request-level and offer-level field names and values must not contain carets (^) or colons (:). However, they can contain embedded spaces and any other printable characters. If you use more than one consecutive space, the extra spaces will be removed.

Data Type Definitions

- **Date and time** — The format is `YYYY-MM-DDThhmmssZ`. For example, `2002-08-11T224757Z` is equal to August 11, 2002, at 10:47:57 P.M. The T separates the date and the time. The Z indicates Coordinated Universal Time (UTC), also known as Greenwich Mean Time.
- **Decimal** — A number that includes a decimal point. For example, `23.45`, `-0.1`, `4.0`, `90809.0468`.
- **Integer** — A whole number. {..., -3, -2, -1, 0, 1, 2, 3, ...}
- **Non-negative integer** — A whole number greater than or equal to zero. {0, 1, 2, 3, ...}
- **Positive integer** — A whole number greater than zero. {1, 2, 3, ...}
- **String** — Contains letters, numbers, spaces, and special characters, such as @ and #.

Request-Level Fields

The following table describes the request-level fields for the Electronic Check Services. The requirements for these fields are described separately:

- Debit service—See “[Requirements for the Debit Request Fields](#)” on page 43.
- Credit service—See “[Requirements for the Credit Request Fields](#)” on page 51.
- Void service—See “[Requirements for the Void Request Fields](#)” on page 56.

Note If you are using Payment Tokenization or Recurring Billing and you include a subscription ID in your request, many of the fields in the following table that are normally required for an authorization or credit become optional. See “[Payment Tokenization](#)” on page 70 and “[Recurring Billing](#)” on page 71.

Table 19 Request-Level Fields

Field Name	Description	Data Type & Length
account_encoder_id	Identifier for the bank that provided the customer's encoded account number. To obtain the bank identifier, contact your processor. “ Encoded Account Numbers ” on page 59.	String (3)
bill_address1	First line of the billing street address.	String (60)
bill_address2	Second line of the billing street address. Used for additional address information. Example: Attention: Accounts Payable	String (60)
bill_city	City in the billing address.	String (50)
bill_company_tax_id	Company's tax identifier.	String with numbers only (9)
bill_country	Country in the billing address. Use the two-character ISO country codes .	String (2)
bill_state	State in the billing address. Use the two-character codes .	String (2)

Table 19 Request-Level Fields (Continued)

Field Name	Description	Data Type & Length
bill_zip	Postal code for the billing address. The postal code must consist of 5 to 9 digits. If the billing country is the U.S., the 9-digit postal code must follow this format: [5 digits][dash][4 digits] Example: 12345-6789 If the billing country is Canada, the 6-digit postal code must follow this format: [alpha][numeric][alpha][space] [numeric][alpha][numeric] Example: A1B 2C3	String (10)
company_name	Name of the customer's company.	String (40)
currency	Currency used for the order. Possible values: • CAD: Canadian dollars • USD: U.S. dollars	String (5)
customer_email	Customer's email address, including the full domain name. Format: <i>name@host.domain</i>	String (255)
customer_firstname	Customer's first name. If the first name is unavailable or inapplicable, such as for a corporate account, enter a dummy value such as NA.	String (60)
customer_ipaddress	IP address for the customer. Example: 10.1.27.63.	String (15)
customer_lastname	Customer's last name. If the transaction is for a corporate account, use this field for the company name.	String (60)
customer_phone	Customer's phone number.	String (15)
customer_ssn	Customer's social security number. Format: NNN-NN-NNNN.	Alphanumeric (11)
date_of_birth	Date of birth of the customer. Format: YYYY-MM-DD or YYYYMMDD	String (10)

Table 19 Request-Level Fields (Continued)

Field Name	Description	Data Type & Length
decline_avs_flags	List of eFunds ID Verification results that should result in a rejected request. Use a comma to separate each value in the list. See the eFunds section in “Verification and Validation” on page 38. Example: N, U Important The value N must be included in the list even though it represents the default behavior.	String (255)
driver_license_no	Driver's license number of the customer.	String (30)
driver_license_state	State or province where the customer's driver's license was issued. Use the two-character codes .	String (2)
e_commerce_indicator	Type of transaction. Possible values: - internet (default): e-commerce order placed using a Web site. - moto: Mail order or telephone order.	String (13)
ecp_account_no	Account number. When processing encoded account numbers, use this field for the encoded account number.	Non-negative integer (17)
ecp_account_type	Account type. Possible values: - C: Checking - S: Savings (U.S. dollars only) - X: Corporate checking (U.S. dollars only)	String (1)
ecp_check_no	Check number.	Integer (8)
ecp_debit_authenticate_id	Identification number returned when an Authenticate request is processed and returned in subsequent monetary transactions such as with the debit service.	Numeric (32)
ecp_debit_payment_info	Payment related information. This field contains payment related information that is passed to the customer's statement.	Alphanumeric (80)
ecp_debit_request_id	The request_id for the debit that is being credited. See “Follow-on Credits and Stand-Alone Credits” on page 47.	String (26)

Table 19 Request-Level Fields (Continued)

Field Name	Description	Data Type & Length
eCP_debit_request_token	The request_token for the debit that is being credited. The field is an encoded string that contains no confidential information, such as an account number or card verification number. The string can contain up to 256 characters. This field is available only for merchants who implemented request tokens before July 2008. For new implementations, use the order_request_token field. See the information about request tokens in Getting Started with CyberSource Advanced.	String (256)
eCP_payment_info	Payment related information. This information is included on the customer's statement.	String (80)
eCP_payment_mode	Flag that indicates whether to process the payment. Use with deferred payments or eFunds ID Verification. See "Deferred and Partial Payments" on page 66 and the eFunds section in "Verification and Validation" on page 38. Possible values: • 0: Standard debit with immediate payment (default). • 1: For eFunds, notifies CyberSource to not process the payment now. For deferred payments, indicates that this is a deferred payment and that you will send a debit request with eCP_payment_mode=2 in the future. • 2: For deferred payments, indicates notification to initiate payment.	Integer (1)
eCP_rdfi	Bank routing number. This is also called the transit number.	Non-negative integer (9)
eCP_ref_no	Check reference number. Identifier used for tracking a request through to the payment processor for reconciliation. If you do not provide this value, CyberSource will generate a unique value and return it to you in the following field: • eCP_debit_ref_no if you are requesting a debit • eCP_credit_ref_no if you are requesting a credit See the information about tracking orders in Getting Started with CyberSource Advanced. For deferred payments with TeleCheck, set this field to the value you received in the eCP_debit_ref_no field in the associated debit's reply. See "Deferred and Partial Payments" on page 66.	TeleCheck: String (25) All other processors: String (60)

Table 19 Request-Level Fields (Continued)

Field Name	Description	Data Type & Length
eCP_sec_code	Code that specifies the authorization method for the transaction. Possible values: • CCD : Corporate cash disbursement—A charge or credit against a business checking account. You can use one-time or recurring CCD transactions to transfer funds to or from a corporate entity. A standing authorization is required for recurring transactions. • PPD : Prearranged payment and deposit entry—A charge or credit against a personal checking or savings account. You can originate a PPD entry only when the payment and deposit terms between you and the customer are prearranged. A written authorization from the customer is required for one-time transactions and a written standing authorization is required for recurring transactions. • TEL : Telephone-initiated entry—A one-time charge against a personal checking or savings account. You can originate a TEL entry only when there is a business relationship between you and the customer or when the customer initiates a telephone call to you. For a TEL entry, you must obtain a payment authorization from the customer over the telephone. There is no recurring billing option for TEL. • WEB : Internet-initiated entry—A charge against a personal checking or savings account. You can originate a one-time or recurring WEB entry when the customer initiates the transaction over the Internet. For a WEB entry, you must obtain payment authorization from the customer over the Internet.	String (3)
eCP_settlement_method	Method used for settlement. Possible values: • A: Automated Clearing House (default for credits and for transactions using Canadian dollars) • F: Facsimile draft (U.S. dollars only) • B: Best possible (U.S. dollars only) (default if the field has not already been configured for your merchant ID) See “Settlement Delivery Methods” on page 72.	String (1)

Table 19 Request-Level Fields (Continued)

Field Name	Description	Data Type & Length
ecp_verification_level	Level of fraud screening. Possible values: • 1 : Validation—Default if the field has not already been configured for your merchant ID • 2 : Verification • 3 : Guarantee—U.S. dollars only • 4 : eFunds ID Verification For a description of this feature and a list of supported processors, see "Verification and Validation" on page 38.	Non-negative integer (1)
grand_total_amount	Grand total for the order. See the information about using offers or a grand total in Getting Started with CyberSource Advanced .	Decimal (15)
ics_applications	ICS services to process for the request. At least one service must be specified.	String (255)
link_to_request	Value that links the current request to a previous authorization request for a debit card or prepaid card. This value is useful when using multiple payment methods to complete an order. For details, see the information about partial authorizations in Credit Card Services for the SCMP API .	String (26)
merchant_descriptor	Merchant description that appears on the customer's bank statement. For a description of this feature, a list of supported processors, and special formatting requirements, see "Merchant Descriptors" on page 68.	String (25)
merchant_id	Your merchant ID. Use the same merchant_id for evaluation, testing, and production.	String (30)
merchant_ref_number	Merchant-generated order reference or tracking number. See the information about tracking orders in Getting Started with CyberSource Advanced .	String (50)
offer0...N	Offers for the request. At a minimum, offer0 must be included in the request. The offer-level fields are described in Table 20 on page 107.	String (50)

Table 19 Request-Level Fields (Continued)

Field Name	Description	Data Type & Length
order_request_token	The request_token value returned from a previous request. This value links the previous request to the current follow-on request. This field is an encoded string that does not contain any confidential information, such as account numbers or card verification numbers. The string can contain up to 256 characters. See the information about request tokens in Getting Started with CyberSource Advanced.	String (256)
partial_payment_id	Identifier for a partial payment or partial credit. The value for each debit request or credit request must be unique within the scope of the order. See “ Deferred and Partial Payments ” on page 66 and “ Multiple Partial Credits ” on page 69.	String (60)
subscription_id	If you are using Payment Tokenization or Recurring Billing and you include this value in your request, many of the fields that are normally required for a debit or credit become optional. See “ Payment Tokenization ” on page 70 and “ Recurring Billing ” on page 71.	String (26)
timeout	Number of seconds the system waits before returning a timeout error. The default is 110 seconds.	Positive integer (3)
void_request_id	The request_id of the debit or credit you want to void.	String (26)
void_request_token	The request_token value returned from a previous request for a service that you want to void. You can void debits and credits that have not been batched or voided before. For example, if you want to void a credit, set this field to the request_token value returned in the reply for the credit request. The field is an encoded string that contains no confidential information, such as an account number or card verification number. The string can contain up to 256 characters. This field is available only for merchants who implemented request tokens before July 2008. For new implementations, use the order_request_token field. See the information about request tokens in Getting Started with CyberSource Advanced.	String (256)

Offer-Level Fields

The following table describes the offer-level fields for the Electronic Check Services. The requirements for these fields are described separately:

- Debit service—See “[Requirements for the Debit Request Fields](#)” on page 43.
- Credit service—See “[Requirements for the Credit Request Fields](#)” on page 51.
- Void service—See “[Requirements for the Void Request Fields](#)” on page 56.

Table 20 Offer-Level Fields

Field Name	Description	Data Type & Length
amount	Per-item price of the product. This value cannot be negative. You can include a decimal point (.) in this field, but you cannot include any other special characters. The amount will be truncated at the request level to the correct number of decimal places.	Decimal (15)
merchant_product_sku	Product's identifier code.	String (30)
product_code	Type of product. This value is used to determine the category that the product is in: electronic, handling, physical, service, or shipping. The default value is default. See Appendix G, “ Product Codes ,” on page 131 for a list of valid values.	String (30)
product_name	Name of the product.	String (30)
quantity	Quantity of the product being purchased. The default value is 1.	Non-negative integer (10)
tax_amount	Total tax to apply to the product. This value cannot be negative. The tax_amount field is additive. For example, if you send the following offer lines: <pre>offer0=amount:10.00^quantity:1^tax_ amount:0.80 offer1=amount:20.00^quantity:1^tax_ amount:1.60</pre> the total amount authorized will be for \$32.40, not \$30.00 with \$2.40 of tax included. The tax_amount and the amount must be in the same currency. If you want to include tax_amount and also request the ics_tax service, see the Tax Calculation Implementation Guide, which is available on the Support Center.	Decimal (15)

Reply Fields

Table 21 Reply Fields

Field Name	Description	Returned By	Data Type & Length
client_lib_version	Information about the client library used to request the transaction.	All Electronic Check Services	String (50)
currency	Currency used for the order. The value will be USD for U.S. dollars or CAD for Canadian dollars.	All Electronic Check Services	String (5)
ecp_authenticate_checkpoint_summary	Contains information about the parameters sent in the authenticate message, which is described in “Check Point Summary Codes” on page 135.	ics_ecp_authenticate	String (600)
ecp_authenticate_fraud_shield_indicators	Contains information about the fraud checks performed, which is described in “Fraud Shield Indicator Codes” on page 149.	ics_ecp_authenticate	String (300)
ecp_authenticate_rcode	One-digit code that indicates whether the ics_ecp_credit request was successful. The field will contain one of the following values: • -1: An error occurred • 0: The request was declined • 1: The request was successful	ics_ecp_authenticate	Integer (1)
ecp_authenticate_ref_no	Reference number you use to reconcile CyberSource reports with processor reports. See the information about tracking orders in Getting Started with CyberSource Advanced .	ics_ecp_authenticate	String (60)
ecp_authenticate_rflag	Numeric value corresponding to the result of the overall request.	ics_ecp_authenticate	String (50)
ecp_authenticate_rmsg	Message that explains the reply flag ecp_credit_rflag . Do not display this message to your customer, and do not use this field to write an error handler.	ics_ecp_authenticate	String (255)
ecp_authenticate_request_time	Time of the request. The format is YYYY-MM-DDThh:mm:ssZ. For example, 2003-08-11T22:47:57 is equal to August 11, 2003, at 10:47:57 P.M. The T separates the date and the time, and the Z indicates UTC.	ics_ecp_authenticate	String (20)

Table 21 Reply Fields (Continued)

Field Name	Description	Returned By	Data Type & Length
eep_authenticate_response	Result code returned by the payment processor.	ics_eep_authenticate	String (6)
eep_credit_corrected_account_number	Corrected account number from the ACH verification service, which is described in “ACH Verification” on page 48.	ics_eep_credit	String (17)
eep_credit_corrected_routing_number	Corrected account number from the ACH verification service, which is described in “ACH Verification” on page 48.	ics_eep_credit	String (9)
eep_credit_owner_merchant_id	Merchant ID that was used to create the subscription or customer profile for which the service was requested. If you are enabled for Recurring Billing, this field is returned only if you are using subscription sharing and if your merchant ID is in the same merchant ID pool as the owner merchant ID. See the subscription sharing information in the Recurring Billing Implementation Guide. If you are enabled for Payment Tokenization, this field is returned only if you are using profile sharing and if your merchant ID is in the same merchant ID pool as the owner merchant ID. See the profile sharing information in Payment Tokenization Using Secure Storage.	ics_eep_credit	String (30)
eep_credit_processor_trans_id	Transaction identifier or tracking ID returned by the payment processor. See the information about tracking orders in Getting Started with CyberSource Advanced .	ics_eep_credit	String (87)
eep_credit_rcode	One-digit code that indicates whether the ics_eep_credit request was successful. The field will contain one of the following values: • -1: An error occurred • 0: The request was declined • 1: The request was successful	ics_eep_credit	Integer (1)

Table 21 Reply Fields (Continued)

Field Name	Description	Returned By	Data Type & Length
eCP_credit_ref_no	Reference number for the transaction. For some processors, you can use this value to reconcile your CyberSource reports with your processor reports. See the information about tracking orders in Getting Started with CyberSource Advanced .	ics_eCP_credit	String (60)
eCP_credit_result_code	Result code returned by the payment processor.	ics_eCP_credit	String (6)
eCP_credit_rflag	One-word description of the result of the ics_eCP_credit request.	ics_eCP_credit	String (50)
eCP_credit_rmsg	Message that explains the reply flag eCP_credit_rflag . Do not display this message to your customer, and do not use this field to write an error handler.	ics_eCP_credit	String (255)
eCP_credit_settlement_method	Method used to settle the credit. Possible values: • A: Automated Clearing House • B: Best possible • F: Facsimile	ics_eCP_credit	String (1)
eCP_credit_submit_time	Time when credit was requested in UTC. See “ Data Type Definitions ” on page 99 for the field's format.	ics_eCP_credit	Date and time (20)
eCP_credit_total_amount	Total amount submitted to the payment processor.	ics_eCP_credit	Decimal (15)

Table 21 Reply Fields (Continued)

Field Name	Description	Returned By	Data Type & Length
eep_credit_verification_code	Indicates the results from the ACH verification service, which is described in “ACH Verification” on page 48. Possible values: • 00 : Success: Account number and routing number are OK. • 01 : Success: Account number was corrected; routing number is OK. • 02 : Success: Routing number was corrected; account number is OK. • 03 : Success: Account number and routing number were corrected. • 04 : Declined: Routing number did not pass verification. • 98 : Unavailable: Unable to perform ACH verification. • 99 : Invalid: Response from ACH verification is invalid.	ics_eep_credit	String (2)

Table 21 Reply Fields (Continued)

Field Name	Description	Returned By	Data Type & Length
eep_credit_verification_code_raw	Raw results from the ACH verification service, which is described in "ACH Verification" on page 48. Possible values: • 1 : Accepted: Routing number is valid. Account number is valid. • 2 : Accepted: Routing number is valid. Account number is invalid; use corrected account number. • 3 : Accepted: Routing number is valid. Account number is valid. • 4 : Accepted: Routing number is valid. Account number structure not recognized; account may be valid. • 5 : Accepted: Routing number is not usable for ACH; use corrected routing number. Account number is valid. • 6 : Accepted: Routing number is not usable for ACH; use corrected routing number. Account number is invalid; use corrected account number. • 7 : Accepted: Routing number is not usable for ACH; use corrected routing number. Account number is valid. • 8 : Accepted: Routing number is not usable for ACH; use corrected routing number. Account number structure not recognized; account may be valid. • 9 : Declined: Routing number is not usable for ACH; no corrected routing number available. • 10 : Declined: Routing number not found. • 11 : Declined: Invalid routing number.	ics_eep_credit	String (2)
eep_debit_avs	Overall result of the eFunds ID Verification. Possible values: • N: At least one customer information item had a negative match, and the other two did not have positive matches. • Y: At least one item had a positive match. • U: None of the items had either positive or negative matches, or an error occurred during processing.	ics_eep_debit	String (1)

Table 21 Reply Fields (Continued)

Field Name	Description	Returned By	Data Type & Length
eCP_debit_avs_raw	Processor's raw match results for each of the customer information fields. See the eFunds section in " Verification and Validation " on page 38. To interpret the raw results, see your processor's specification for eFunds ID Verification.	ics_eCP_debit	String (2)
eCP_debit_corrected_account_number	Corrected account number from the ACH verification service, which is described in " ACH Verification " on page 39.	ics_eCP_debit	String (17)
eCP_debit_corrected_routing_number	Corrected account number from the ACH verification service, which is described in " ACH Verification " on page 39.	ics_eCP_debit	String (9)
eCP_debit_owner_merchant_id	Merchant ID that was used to create the subscription or customer profile for which the service was requested. If you are enabled for Recurring Billing, this field is returned only if you are using subscription sharing and if your merchant ID is in the same merchant ID pool as the owner merchant ID. See the subscription sharing information in the Recurring Billing Implementation Guide. If you are enabled for Payment Tokenization, this field is returned only if you are using profile sharing and if your merchant ID is in the same merchant ID pool as the owner merchant ID. See the profile sharing information in Payment Tokenization Using Secure Storage.	ics_eCP_debit	String (30)
eCP_debit_processor_trans_id	Transaction identifier returned by the payment processor. See the information about tracking orders in Getting Started with CyberSource Advanced .	ics_eCP_debit	String (87)
eCP_debit_rcode	One-digit code that indicates whether the ics_eCP_debit request was successful. The field will contain one of the following values: • -1: An error occurred • 0: The request was declined • 1: The request was successful	ics_eCP_debit	Integer (1)

Table 21 Reply Fields (Continued)

Field Name	Description	Returned By	Data Type & Length
eCP_debit_ref_no	Reference number for the transaction. For some processors, you can use this value to reconcile your CyberSource reports with your processor reports. See the information about tracking orders in Getting Started with CyberSource Advanced .	ics_eCP_debit	String (60)
eCP_debit_result_code	Result code returned by the payment processor.	ics_eCP_debit	String (6)
eCP_debit_rflag	One-word description of the result of the ics_eCP_debit request.	ics_eCP_debit	String (50)
eCP_debit_rmsg	Message that explains the reply flag eCP_debit_rflag . Do not display this message to your customer, and do not use this field to write an error handler.	ics_eCP_debit	String (255)
eCP_debit_settlement_method	Method used to settle the debit. Possible values: • A: Automated Clearing House • B: Best possible • F: Facsimile	ics_eCP_debit	String (1)
eCP_debit_submit_time	Time when debit was requested in UTC. See “ Data Type Definitions ” on page 99 for the field's format.	ics_eCP_debit	Date and time (20)
eCP_debit_total_amount	Total amount submitted to the payment processor.	ics_eCP_debit	Decimal (15)

Table 21 Reply Fields (Continued)

Field Name	Description	Returned By	Data Type & Length
eep_debit_verification_code	Indicates the results from the ACH verification service, which is described in “ACH Verification” on page 39. Possible values: • 00 : Success: Account number and routing number are OK. • 01 : Success: Account number was corrected; routing number is OK. • 02 : Success: Routing number was corrected; account number is OK. • 03 : Success: Account number and routing number were corrected. • 04 : Declined: Routing number did not pass verification. • 98 : Unavailable: Unable to perform ACH verification. • 99 : Invalid: Response from ACH verification is invalid.	ics_eep_debit	String (2)

Table 21 Reply Fields (Continued)

Field Name	Description	Returned By	Data Type & Length
eep_debit_verification_code_raw	Raw results from the ACH verification service, which is described in "ACH Verification" on page 39. Possible values: • 1 : Accepted: Routing number is valid. Account number is valid. • 2 : Accepted: Routing number is valid. Account number is invalid; use corrected account number. • 3 : Accepted: Routing number is valid. Account number is valid. • 4 : Accepted: Routing number is valid. Account number structure not recognized; account may be valid. • 5 : Accepted: Routing number is not usable for ACH; use corrected routing number. Account number is valid. • 6 : Accepted: Routing number is not usable for ACH; use corrected routing number. Account number is invalid; use corrected account number. • 7 : Accepted: Routing number is not usable for ACH; use corrected routing number. Account number is valid. • 8 : Accepted: Routing number is not usable for ACH; use corrected routing number. Account number structure not recognized; account may be valid. • 9 : Declined: Routing number is not usable for ACH; no corrected routing number available. • 10 : Declined: Routing number not found. • 11 : Declined: Invalid routing number.	ics_eep_debit	String (2)
eep_debit_verification_level	Level of screening for the request. Possible values: • 1: Validation • 2: Verification • 3: Guarantee • 4: eFunds ID Verification	ics_eep_debit	Non-negative integer (1)

Table 21 Reply Fields (Continued)

Field Name	Description	Returned By	Data Type & Length
ics_rcode	One-digit code that indicates whether the entire request was successful. The field will contain one of the following values: • -1: An error occurred • 0: The request was declined • 1: The request was successful	All Electronic Check Services	Integer (1)
ics_rflag	One-word description of the result of the entire request.	All Electronic Check Services	String (50)
ics_rmsg	Message that explains the reply flag ics_rflag . Do not display this message to your customer, and do not use this field to write an error handler	All Electronic Check Services	String (255)
merchant_ref_number	Order reference or tracking number that you provided in the request. If you included multi-byte characters in this field in the request, the returned value might contain corrupted characters.	All Electronic Check Services	String (50)
request_id	Identifier for the request generated by the client.	All Electronic Check Services	String (26)
request_token	Request token data created by CyberSource for each reply. You need to store the contents of this field so that you can retrieve and send it in follow-on requests. The field is an encoded string that contains no confidential information, such as an account or card verification number. The string can contain up to 256 characters. See the information about request tokens in Getting Started with CyberSource Advanced .	All Services	String (256)
void_rcode	One-digit code that indicates whether the ics_void request was successful. The field will contain one of the following values: • -1: An error occurred • 0: The request was declined • 1: The request was successful	ics_void	Integer (1)
void_rflag	One-word description of the result of the ics_void request.	ics_void	String (50)

Table 21 Reply Fields (Continued)

Field Name	Description	Returned By	Data Type & Length
void_rmsg	Message that explains the reply flag void_rflag . Do not display this message to your customer, and do not use this field to write an error handler.	ics_void	String (255)
void_void_amount	Total amount of the void.	ics_void	Decimal (15)
void_void_currency	Currency used for the order. Formatted using the ISO currency codes .	ics_void	String (5)
void_void_request_time	Time when void was requested in UTC. See " Data Type Definitions " on page 99 for the field's format.	ics_void	Date and time (20)

Appendix C

Examples for the Simple Order API

Examples:

[Name-Value Pair Examples](#)

[XML Examples](#)

Name-Value Pair Examples

Example Electronic Check Debit Request

This example uses merchant-provided data.

```
ecDebitService_run=true
merchantID=infodev
merchantReferenceCode=15363553D21528F
billTo_firstName=John
billTo_lastName=Doe
billTo_street1=1295 Charleston Rd.
billTo_city=Mountain View
billTo_state=CA
billTo_postalCode=94043
billTo_country=US
billTo_phoneNumber=650-965-6000
billTo_email=jdoe@example.com
item_0_unitPrice=100.00
purchaseTotals_currency=USD
check_accountNumber=4100
check_accountType=c
check_bankTransitNumber=071923284
```

Example Electronic Check Debit Reply

```
requestID=9980055975450167905139
requestToken=kL34fkL0J0Q498xoiYcyyVr1wpbWEL2RKMYv
merchantReferenceCode=15363553D21528F
decision=ACCEPT
reasonCode=100
ecDebitReply_reasonCode=100
ecDebitReply_settlementMethod=A
ecDebitReply_requestDateTime=2003-05-16T23:48:09Z
ecDebitReply_amount=100.00
ecDebitReply_verificationLevel=1
ecDebitReply_reconciliationID=02RYXSPGCQH60NWA
ecDebitReply_processorResponse=123456
```

XML Examples

Example Electronic Check Debit Request

This example uses merchant-provided data.

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data 1.23">
  <merchantID>infodev</merchantID>
  <merchantReferenceCode>15363553D21528F</merchantReferenceCode>
  <billTo>
    <firstName>John</firstName>
    <lastName>Doe</lastName>
    <street1>1295 Charleston Rd.</street1>
    <city>Mountain View</city>
    <state>CA</state>
    <postalCode>94043</postalCode>
    <country>US</country>
    <phoneNumber>650-965-6000</phoneNumber>
    <email>jdoe@example.com</email>
  </billTo>
  <item id="0">
    <unitPrice>100.00</unitPrice>
  </item>
  <purchaseTotals>
    <currency>USD</currency>
  </purchaseTotals>
  <check>
    <accountNumber>4100</accountNumber>
    <accountType>c</accountType>
    <bankTransitNumber>071923284</bankTransitNumber>
  </check>
  <ecDebitService run="true"/>
</requestMessage>
```

Example Electronic Check Debit Reply

```
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.23">
  <c:merchantReferenceCode>15363553D21528F</c:merchantReferenceCode>
  <c:requestID>9980055975450167905139</c:requestID>
  <c:requestToken>kL34fkL0J0Q498xoiYcyyVr1wpbWEL2RKMYv</c:requestToken>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:purchaseTotals>
    <c:currency>USD</c:currency>
  </c:purchaseTotals>
  <c:ecDebitReply>
    <c:reasonCode>100</c:reasonCode>
    <c:settlementMethod>A</c:settlementMethod>
    <c:requestDateTime>2003-05-16T23:48:09Z</c:requestDateTime>
    <c:amount>100.00</c:amount>
    <c:verificationLevel>1</c:verificationLevel>
    <c:reconciliationID>02RYXSPGCQH60NWA</c:reconciliationID>
    <c:processorResponse>123456</c:processorResponse>
  </c:ecDebitReply>
</c:replyMessage>
```

Appendix D

Examples for the SCMP API

Example Electronic Check Debit Request

This example uses merchant-provided data.

```
bill_address1=1295 Charleston Rd.  
bill_city=Mountain View  
bill_country=US  
bill_state=CA  
bill_zip=94043  
currency=USD  
customer_email=jdoe@example.com  
customer_firstname=John  
customer_lastname=Doe  
customer_phone=650-965-6000  
ecp_account_no=4100  
ecp_account_type=c  
ecp_rdfi=071923284  
ics_applications=ics_ecp_debit  
merchant_id=infodev  
merchant_ref_number=15363553D21528F23162D3E3A  
offer0=amount:100.00  
server_host=ics2test.ic3.com  
server_port=80
```

Example Electronic Check Debit Reply

```
merchant_ref_number=15363553D21528F23162D3E3A
currency=USD
ecp_debit_rcode=1
ecp_debit_ref_no=02RYXSPGCQH60NWA
ecp_debit_result_code=123456
ecp_debit_rflag=SOK
ecp_debit_rmsg=Request was processed successfully.
ecp_debit_settlement_method=A
ecp_debit_submit_time=2003-03-16T234809Z
ecp_debit_total_amount=100.00
ecp_debit_verification_level=1
ics_rcode=1
ics_rflag=SOK
ics_rmsg=Request was processed successfully.
request_id=9980055975450167905139
request_token=kL34fkL0J0Q498xoiYcyyVr1wpbWEL2RKMYv
```

Appendix E

Reason Codes for the Simple Order API

The following table lists the Electronic Check Services reason codes returned by the Simple Order API. See the information about handling replies in [Getting Started with CyberSource Advanced](#) for a discussion of replies, decisions, and reason codes.

Note CyberSource reserves the right to add new reason codes at any time. If your error handler receives a reason code that it does not recognize, it should use the **decision** field to determine the result.

Table 22 Reason Codes

Reason Code	Description
100	Successful transaction.
101	The request is missing one or more required fields. Possible action: See the reply fields missingField_0...N for which fields are missing. Resend the request with the complete information. See the information about missing and invalid fields in Getting Started with CyberSource Advanced .
102	One or more fields in the request contains invalid data. Possible action: See the reply fields invalidField_0...N for which fields are invalid. Resend the request with the correct information. See the information about missing and invalid fields in Getting Started with CyberSource Advanced .
150	Error: General system failure. See the documentation for your client for information about handling retries in the case of system errors.
151	Error: The request was received, but there was a server timeout. This error does not include timeouts between the client and the server. Possible Action: To avoid duplicating the transaction, do not resend the request until you have reviewed the transaction status in the Business Center. See the documentation for your client for information about handling retries in the case of system errors.

Table 22 Reason Codes (Continued)

Reason Code	Description
152	Error: The request was received but there was a service timeout. Possible action: To avoid duplicating the transaction, do not resend the request until you have reviewed the transaction status in the Business Center. See the documentation for your client for information about handling retries in the case of system errors.
200	The request was declined by CyberSource because it did not pass the eFunds ID Verification check. Possible action: You can accept the order, but consider reviewing the order for the possibility of fraud.
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.
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 merchant configuration. Possible action: Do not resend the request. Contact Customer Support to correct the configuration problem.
236	Processor failure. Possible action: Wait a few minutes and resend the request.
241	The request ID is invalid for the follow-on request. Possible action: Verify the request ID is valid and resend the request.
246	The debit or credit is not voidable because the debit or credit information has already been submitted to your processor. Or, you requested a void for a type of transaction that cannot be voided. Possible action: No action required.
247	You requested a credit for a debit that was previously voided. Possible action: No action required.

Table 22 Reason Codes (Continued)

Reason Code	Description
250	Error: The request was received, but there was a timeout at 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.
388	Error: The routing number did not pass verification as described in "ACH Verification" on page 13. Possible action: (1) Ask your customer to contact their bank to get an ACH routing number. (2) Ask your customer to provide the routing number and account number for a different bank account if they have one. (3) Request a different form of payment.

Appendix F

Reply Flags for the SCMP API

A reply flag is associated with an entire request or a specific service:

- *Entire request*—The flag is in the **ics_rflag** field with a message in the **ics_rmsg** field.
- *Individual service*—The flag is in the **<service>_rflag** with a message in the **<service>_rmsg** field.

Important CyberSource reserves the right to add new reply flags at any time. Your system must be able to process these new reply flags.

Table 23 Reply Flags

Reply Flag	Description	Services That Can Return This Flag
DAVSNO	The request failed the eFunds ID Verification.	ics_ecp_debit
DACHVERIFICATION	The routing number did not pass verification as described in “ ACH Verification ” on page 39. Possible action: (1) Ask your customer to contact their bank to get an ACH routing number. (2) Ask your customer to provide the routing number and account number for a different bank account if they have one. (3) Request a different form of payment.	ics_ecp_credit ics_ecp_debit
DCHECKREFUSED	The processor declined the transaction.	All Electronic Check Services
DINVALIDDATA	Data provided is not consistent with the request. For example, you requested a product with negative cost, or you tried to credit a debit that was previously voided.	All Electronic Check Services
DMISSINGFIELD	The request is missing a required field.	All Electronic Check Services

Table 23 Reply Flags (Continued)

Reply Flag	Description	Services That Can Return This Flag
DNOTVOIDABLE	The debit or credit is not voidable because the debit or credit information has already been submitted to your processor. Or, you requested a void for a type of transaction that cannot be voided.	ics_void
ESYSTEM	System error. You must design your transaction management system to include a way to correctly handle CyberSource system errors. Depending on which payment processor is handling the transaction, the error might indicate a valid CyberSource system error, or it might indicate a processor rejection because of some type of invalid data. In either case, CyberSource recommends that you do not design your system to endlessly retry sending a transaction in the case of a system error. See the documentation for your client for information about handling retries in the case of system errors.	All Electronic Check Services
ETIMEOUT	The request was received but there was a service timeout. To avoid duplicating the transaction, do not resend the request until you have reviewed the transaction status in the Business Center. See the documentation for your client for information about handling retries in the case of system errors.	All Electronic Check Services
SOK	Transaction was successful.	All Electronic Check Services

Appendix G

Product Codes

The following table lists the values you can use for the product code. For the Simple Order API, use the **item_#_productCode** request field to specify the product code. For the SCMP API, use the **product_code** offer-level field.

Table 24 Product Code Values

Product Code	Definition
adult_content	Adult content.
coupon	Coupon applied to the entire order.
default	Default value for the product code. CyberSource uses default when a request provides no value for the product code.
electronic_good	Electronic product other than software.
electronic_software	Software distributed electronically rather than on tapes, disks, or other media.
gift_certificate	Gift certificate not issued with CyberSource Stored Value Services.
handling_only	Separate charge that is generally a fee imposed by the seller on the customer. The fee pays for the seller's administrative selling costs.
service	Service that you perform for the customer.
shipping_and_handling	Shipping is a separate charge for shipping the product to the purchaser. Handling is generally a fee imposed by the seller to pay for administrative selling costs.
shipping_only	Charge for transporting tangible personal property from the seller to the purchaser. Documentation must be maintained that clearly establishes where title to the tangible personal property passed from the seller to the purchaser.
stored_value	Stored Value certificate.
subscription	Subscription to a Web site or other content.

Appendix H

NOC Codes

The following table describes the NOC codes. For more information, see:

- *Simple Order API*—“[Notifications of Change \(NOC\)](#)” on page 11
- *SCMP API*—“[Notifications of Change \(NOC\)](#)” on page 37

Table 25 NOC Codes

Code	NOC Reason	Description	Required Action
C01	Incorrect account number	The customer's bank account number is incorrect.	Correct all applicable records before submitting additional electronic check transactions for the customer.
C02	Incorrect routing number	The bank's routing number is incorrect.	Correct all applicable records before submitting additional electronic check transactions for the customer.
C03	Incorrect routing number and incorrect account number	The bank's routing number and the customer's bank account number are incorrect.	Correct all applicable records before submitting additional electronic check transactions for the customer.
C04	Incorrect customer name	The customer name associated with the bank account is incorrect.	Correct all applicable records before submitting additional electronic check transactions for the customer.
C05	Incorrect transaction code	The transaction was submitted to a specific type of account but includes a conflicting account type code (checking / savings).	Correct all applicable records before submitting additional electronic check transactions for the customer.

Table 25 NOC Codes (Continued)

Code	NOC Reason	Description	Required Action
C06	Incorrect account number and incorrect transaction code	The customer's bank account number is incorrect and the transaction was submitted to a specific type of account but includes a conflicting account type code (checking / savings).	Correct all applicable records before submitting additional electronic check transactions for the customer.
C07	Incorrect routing number, incorrect account number, and incorrect transaction code	The bank's routing number and the customer's bank account number are incorrect. Additionally, the transaction was submitted to a specific type of account but includes a conflicting account type code (checking / savings).	Correct all applicable records before submitting additional electronic check transactions for the customer.

Appendix I

Check Point Summary Codes

The check point summary returns information about the parameters sent in the authenticate message. If no parameters are sent, no information will be returned. The following table describes the codes returned in the check point summary. For more information, see:

- *Simple Order API*—“[Authenticate](#)” on page 29
- *SCMP API*—“[Authenticate](#)” on page 54

Topics:

[Primary Result Codes](#)
[Address Result Codes](#)
[Phone Codes](#)
[Address Type Codes](#)
[Change of Address Codes](#)
[Social Security Number Codes](#)
[Address Unit Mismatch Codes](#)
[Phone Unit Mismatch Codes](#)
[Driver's License Result Codes](#)
[Date of Birth Match Codes](#)
[High Risk Address Codes](#)
[High Risk Phone Codes](#)
[OFAC Validation Results Codes](#)
[Address Residential Match Codes](#)
[Address Business Match Codes](#)
[Phone Number Residential Match Codes](#)
[Phone Number Business Match Codes](#)

Primary Result Codes

Table 26 Primary Result Codes

Parameter	Description
00	Data found for search request
02	Receive error
03	Link to load balancing failed
04	Converse allocation error
05	System busy
06	SYSID error
08	Term error
12	No search criteria specified, request ignored
13	Response level must be "F", "B" or blank
15	Match to Full Name Only - PLC NS List
16	Match to Surname and First Initial - PLC NS List
17	Internal error - contact Help Desk
18	Internal error - contact Help Desk
25	System busy - try again later
50	Transaction timed out - contact Help Desk if problem persists
51	Poll error - contact Help Desk
52	Receive error - contact Help Desk
53	Create error - contact Help Desk
54	Connect error - contact Help Desk
55	Send error - contact Help Desk
56	Link error - contact Help Desk
57	EBCDIC to ASCII convert failed - contact Help Desk
58	ASCII to EBCDIC convert failed - contact Help Desk
59	Response exceeds storage - contact Help Desk

Table 26 Primary Result Codes (Continued)

Parameter	Description
63	Storage error during data parsing - contact Help Desk
64	Response data format invalid - contact Help Desk
65	Unable to allocate connection
66	System ID error
67	Connect/converse error
68	Checkpoint server dropped socket
69	Profile not on database
70	System change transaction fail (internal)
71	Name required but missing
72	Address required but missing
75	Audit number required but missing
77	Invalid audit number
78	Audit entry not a Checkpoint 2.0 request
79	Audit file not loaded
80	Server not available to process request
81	Error connecting to database

Address Result Codes

The address result code (AddrCode) identifies the results found based on the address submitted.

Table 27 Address Result Codes

Parameter	Description
A	Address ambiguous
B	Match to business name - residential address
BB	Match to business name - business address

Table 27 Address Result Codes

Parameter	Description
BM	Match to business name - mixed use address
E	Matching records exceed maximum defined on profile
H	House number not found on street
I	Incomplete or blank address
IV	Invalid address
N	No match to name - residential address
NA	Data not available
NB	No match to name - business address
NM	No match to name - mixed use address
NP	Test not in profile
NS	Standardization database has expired - contact Help
R	Road name - City/Zip mismatch
S	Match to surname - residential address
SB	Match to surname - business address
SM	Match to surname - mixed use address
SX	Standardization database has expired - contact Help
T	City - State mismatch
U	Address unverifiable - not in database
UR	Address residential - name match unavailable
Y	Match to full name - residential address
YB	Match to full name - business address
YM	Match to full name - mixed use address
Z	City/State - Zip mismatch
00	Unknown message code - contact help desk

Phone Codes

The phone code (PhnCode) identifies the results found based on the phone number submitted in the authenticate message.

Table 28 Phone Codes

Parameter	Value
A	Match to address only - residential phone
AB	Match to address only - business phone
AM	Match to address only - mixed use phone
B	Match to business name and address - residential phone
BB	Match to business name and address - business phone
BM	Match to business name and address - mixed use phone
C	Probable cellular phone
D	Match to business name - residential address
DB	Match to business name - business address
DM	Match to business name - mixed use phone
E	Matching records exceed maximum defined on profile
F	Match to full name only - residential phone
FB	Match to full name only - business phone
FM	Match to full name only - mixed use phone
H	Match to surname and address - residential phone
HB	Match to surname and address - business phone
HM	Match to surname and address - mixed use phone
I	Phone is incorrect length
IA	Invalid area code
M	Phone missing (search information not received)
MA	Match to header data
N	No match to name or address - residential phone

Table 28 Phone Codes

Parameter	Value
NA	Data not available
NB	No match to name or address - business phone
NM	No match to name or address - mixed use phone
NP	Test not in profile
P	Probable pager
S	Match to surname only - residential phone
SB	Match to surname only - business phone
SM	Match to surname only - mixed use phone
U	Phone unverifiable - not in database
X	Prefix - Zip mismatch
Y	Match to full name and address - residential phone
YB	Match to full name and address - business phone
YM	Match to full name and address - mixed use phone
00	Unknown message code - contact help desk

Address Type Codes

The address type code (AddrTypeCode) identifies the results found based on the address submitted.

Table 29 Address Type Codes

Parameter	Value
C	Single company
E	Test error
EB	Seasonal - business
EM	Seasonal - multi-family dwelling
EX	Seasonal - mixed use

Table 29 Address Type Codes

Parameter	Value
M	Multi-family dwelling
N	No information available
NA	Data not available
NP	Test not in profile
O	Office building
P	Post office box

Change of Address Codes

The change of address code (COACode) identifies the results found for a change of address check.

Table 30 Change of Address Codes

Parameter	Value
C	Change of address information found
N	No change of address information found
NA	Data not available
NP	Test not in profile
U	Test not available
YA	A high risk business was identified at this address
00	Unknown message code - contact help desk

Social Security Number Codes

The Social Security Number code (SSNCode) identifies the results found to verify the Social Security Number submitted.

Table 31 Social Security Number Codes

Parameter	Value
A	Match to address only
D	Deceased - unable to verify name
DN	Deceased - no match to name
DS	Deceased - match to surname
DY	Deceased - match to full name
E	Matching records exceed maximum defined on profile
F	SSN format is invalid
FF	Match to first name and address - match performed using SSN finder
FY	Match to full name and address - match performed using SSN finder
I	SSN is incorrect length
M	SSN is missing
N	No match to name or address
NA	Data not available
NI	SSN not issued
NP	Test not in profile
NV	Header search not available in NV due to state law
P	Match to previous address only
S	Match to surname only
SA	Match to surname and address
V	Valid SSN - SSN not found
Y	Match to full name only
YA	Match to full name and address - match performed using SSN

Table 31 Social Security Number Codes

Parameter	Value
YB	Match to full name and address - match performed using name and address
Z	SSN found - no surname entered
00	Unknown message code - contact help desk

Address Unit Mismatch Codes

The address unit mismatch code (AddrUnitMismatchCode) identifies the expected unit for the address.

Table 32 Address Unit Mismatch Codes

Parameter	Value
EU	Unit number is extra - not expected at this address
MU	Unit number is missing - expected at this address
WU	Unit number wrong - unit number does not match unit number at this address

Phone Unit Mismatch Codes

The phone unit mismatch code (PhnUnitMismatchCode) identifies the expected unit for the address associated with the phone number.

Table 33 Phone Unit Mismatch Codes

Parameter	Value
EU	Unit number is extra - not expected at this address
MU	Unit number is missing - expected at this address
WU	Unit number wrong - unit number does not match unit number at this address

Driver's License Result Codes

The driver's license result code (DLResultCode) identifies the results found based on the driver's licences information submitted.

Table 34 Driver's License Result Codes

Parameter	Value
A	Match to address only
I	DL state and number format invalid
M	Driver License number not submitted on inquiry
N	No match to name or address
NA	Data not available for this state
NI	Input DL state and number not on file
NP	Test not in profile
S	Match to surname only
SA	Match to surname and address
V	Valid DL state and number - Name match not available
Y	Match to full name only

Date of Birth Match Codes

The date of birth match code (DateOfBirthMatch) identifies the results found based on the date of birth information submitted.

Table 35 Date of Birth Match Codes

Parameter	Value
1	Match
2	Partial match
3	No Match
4	Not on file

Table 35 Date of Birth Match Codes

Parameter	Value
5	SSN not on file; search cannot be done
6	DOB not provided on search request
7	Invalid DOB format

High Risk Address Codes

The high risk address code (HighRiskAddrCode) identifies any high risk address information that is associated with the address information submitted.

Table 36 High Risk Address Codes

Parameters	Values
N	No address high risk information found
NP	Test not in profile
YA	A high risk business was identified at this address

High Risk Phone Codes

The high risk phone code (HighRiskPhnCode) identifies any high risk phone number information that is associated with the phone number information submitted.

Table 37 High Risk Phone Codes

Parameters	Values
N	No address high risk information found
NP	Test not in profile
YA	A high risk business was identified at this address

OFAC Validation Results Codes

The OFAC validation results code (OFACValidationResult) identifies if the information submitted in the request is on the OFAC list and specifies what pieces of information are present on the OFAC list if there is a match.

Table 38 OFAC Validation Results Codes

Parameter	Value
1	No match
2	Match to full name only
3	Match to SSN only
4	Match to name and SSN
5	Match to name and DOB
6	Match to name and YOB
7	Match to SSN and DOB
8	Match to SSN and YOB
9	Match to name, SSN, and DOB
10	Match to name, SSN, and YOB
11	Match to company name only
12	Match to company address only
13	Match to company name and address
14	Match to surname and first name
15	Match to full name only - PLC NS list
16	Match to surname and first initial - PLC NS list

Address Residential Match Codes

The address residential match code is a number (0000 through 9999) that identifies the number of residential records that matched the address given during address verification.

Address Business Match Codes

The address business match code is a number (0000 through 9999) that identifies the number of business records that matched the address given during address verification.

Phone Number Residential Match Codes

The phone number residential match code is a number (0000 through 9999) that identifies the number of residential records that matched the phone number given during address verification.

Phone Number Business Match Codes

The phone number business match code is a number (0000 through 9999) that identifies the number of business records that matched the phone number given during address verification.

Appendix J

Fraud Shield Indicator Codes

The fraud shield indicators provides you with consumer fraud information. This information is used to protect you against fraudulent transactions. However, no fraud check is perfect and a chance for fraudulent activity is always present. The following table describes the codes returned in the fraud shield indicators value. For more information, see:

- *Simple Order API*—“[Authenticate](#)” on page 29
- *SCMP API*—“[Authenticate](#)” on page 54

Fraud Shield Indicator Codes

Table 39 Fraud Shield Indicator Codes

Parameter	Description
FS01	Inquiry/Online current address conflict
FS02	Inquiry address first reported < 90 days
FS03	Inquiry current address not on-file
FS04	Input SSN not issued as of MM/YY
FS05	Input SSN recorded as deceased
FS06	Inquiry age younger than SSN issue date
FS10	INQ: Type of high risk address
FS11	INQ: Nonresidential address
FS13	High probability SSN belongs to another
FS14	Input SSN invalid
FS15	INQ: Address reported cautious

Table 39 Fraud Shield Indicator Codes (Continued)

Parameter	Description
FS16	FILE: Type of high risk address
FS17	FILE: Non Residential address
FS18	FILE: reported cautious
FS21	Telephone number inconsistent with address
FS25	Best on file SSN recorded as deceased
FS26	Best on file SSN not issued as of MM/YY
FS27	SSN Reported more frequently for another

Index

A

account information [8](#), [34](#)
account numbers
 encoded [59](#), [67](#)
 on checks [8](#), [34](#)
ACH verification
 for credits [23](#), [48](#)
 for debits [13](#), [39](#)
authentication [29](#), [54](#)
authorization consents [2](#)

B

banking requirements [1](#)

C

Chase Paymentech Solutions
 banking requirements [1](#)
 credits [21](#), [46](#)
 debits [7](#), [33](#)
 deferred payments [58](#), [66](#)
 eFunds ID Verification [15](#), [41](#)
 encoded account numbers [59](#), [67](#)
 merchant descriptors [60](#), [68](#)
 merchant-provided data [8](#), [34](#)
 partial payments [58](#), [66](#)
 payment tokenization [62](#), [70](#)
 recurring billing [63](#), [71](#)
 settlement delivery methods [64](#), [72](#)
 testing [74](#)
 validation [12](#), [38](#)
 verification [14](#), [40](#)
 voids [30](#), [55](#)

check point summary codes [135](#)
check reference numbers [4](#)
consent statements [2](#)
corporate checks [57](#), [65](#)
credits
 follow-on [22](#), [47](#)
 multiple partial [61](#), [69](#)
 stand-alone [22](#), [47](#)
customer profiles [62](#), [70](#)
CyberSource ACH Service
 banking requirements [1](#)
 credits [21](#), [46](#)
 debits [7](#), [33](#)
 merchant-provided data [8](#), [34](#)
 NOCs
 for credits [25](#), [50](#)
 for debits [11](#), [37](#)
 NSF service [61](#), [69](#)
 payment tokenization [62](#), [70](#)
 recurring billing [63](#), [71](#)
 testing [75](#)
 verification
 for credits [23](#), [48](#)
 for debits [13](#), [39](#)

D

data types [79](#), [99](#)
debits [7](#), [33](#)
deferred payments [58](#), [66](#)
delivery methods [64](#), [72](#)
descriptors [60](#), [68](#)

E

ecCreditService [21](#)
ecDebitService [7](#)
eFunds ID Verification [15](#), [41](#)
encoded account numbers [59](#), [67](#)
encrypted account numbers [59](#), [67](#)
examples [119](#), [123](#)

F

follow-on credits [22](#), [47](#)
fraud screening [12](#), [38](#)
fraud shield indicator codes [149](#)

G

GMT [99](#)
guaranteed checks [14](#), [40](#)

I

ics_ecp_credit [46](#)
ics_ecp_debit [33](#)
ics_void [55](#)
issuer encryption [59](#), [67](#)

L

live transactions [76](#)

M

merchant descriptors [60](#), [68](#)
merchant-provided data [8](#), [34](#)
multiple partial credits [61](#), [69](#)

N

NOC codes [133](#)
NOCs
 for credits [25](#), [50](#)
 for debits [11](#), [37](#)
non-sufficient funds. *See* NSF service
NSF service [61](#), [69](#)

O

order tracking [4](#)

P

partial credits [61](#), [69](#)
partial payments [58](#), [66](#)
Payment Events Report [3](#)
payment tokenization [62](#), [70](#)
processor transaction identifiers [4](#)
processor-managed data [10](#), [36](#)
product codes [131](#)
profiles [62](#), [70](#)

R

RBS WorldPay Atlanta
 authentication [29](#), [54](#)
 banking requirements [1](#)
 corporate checks [57](#), [65](#)
 merchant-provided data [8](#), [34](#)
 multiple partial credits [61](#), [69](#)
 NOCs
 for credits [25](#), [50](#)
 for debits [11](#), [37](#)
 payment tokenization [62](#), [70](#)
 recurring billing [63](#), [71](#)
 verification
 for credits [23](#), [48](#)
 for debits [13](#), [39](#)
 voids [30](#), [55](#)
reason codes [125](#)
reconciliation IDs [4](#)
recurring billing [63](#), [71](#)
reply flags [129](#)
request IDs [4](#)
returned check fees table [2](#)
routing numbers on checks [8](#), [34](#)

S

samples [119](#), [123](#)
SEC codes [82](#), [104](#)
secure data [62](#), [70](#)
secure storage [62](#), [70](#)

settlement delivery methods [64](#), [72](#)
soft descriptors [60](#), [68](#)
special characters [79](#), [99](#)
stand-alone credits [22](#), [47](#)
state returned check fees table [2](#)
subscriptions [63](#), [71](#)

T

TeleCheck

banking requirements [1](#)
corporate checks [57](#), [65](#)
credits [21](#), [46](#)
debits [7](#), [33](#)
deferred payments [58](#), [66](#)
guaranteed checks [14](#), [40](#)
merchant-provided data [8](#), [34](#)
multiple partial credits [61](#), [69](#)
partial payments [58](#), [66](#)
payment tokenization [62](#), [70](#)
processor-managed data [10](#), [36](#)
recurring billing [63](#), [71](#)
validation [12](#), [38](#)
voids [30](#), [55](#)
testing [73](#)
transaction reference numbers [4](#)

U

UTC [99](#)

V

validation [12](#), [38](#)
verification [12](#), [38](#)
voids [30](#), [55](#)
voidService [30](#)

