Chase Pay POS Transactions

Using the Simple Order API

December 2016
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 650-432-7350 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.

Copyright

© 2016 CyberSource Corporation. All rights reserved. CyberSource Corporation ("CyberSource") furnishes this document and the software described in this document under the applicable agreement between the reader of this document ("You") and CyberSource ("Agreement"). You may use this document and/or software only in accordance with the terms of the Agreement. Except as expressly set forth in the Agreement, the information contained in this document is subject to change without notice and therefore should not be interpreted in any way as a guarantee or warranty by CyberSource. CyberSource assumes no responsibility or liability for any errors that may appear in this document. The copyrighted software that accompanies this document is licensed to You for use only in strict accordance with the Agreement. You should read the Agreement carefully before using the software. Except as permitted by the Agreement, You may not reproduce any part of this document, store this document in a retrieval system, or transmit this document, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written consent of CyberSource.

Restricted Rights Legends

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

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 Unpublished rights reserved under the copyright laws of the United States.

Trademarks

CyberSource, The Power of Payment, CyberSource Payment Manager, CyberSource Risk Manager, CyberSource Decision Manager, CyberSource Connect, Authorize.Net, and eCheck.net 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

Recent Revisions to This Document 5

About This Guide 6
  Audience and Purpose 6
  Conventions 6
  Related Documentation 7
  Customer Support 7

Chapter 1 Introduction to Chase Pay POS Transactions 8
  Requirements 8
  Supported Card Type 8
  How Chase Pay POS Works 8
  Testing 9

Appendix A Examples 10
  Name-Value Pair Examples 10
    Chase Pay POS Authorization 10
    Chase Pay POS Capture 11
    Chase Pay POS Sale 12
    Chase Pay POS Authorization Reversal 13
    Chase Pay POS Follow-On Credit 13
  XML Examples 14
    Chase Pay POS Authorization 14
    Chase Pay POS Capture 15
    Chase Pay POS Sale 17
    Chase Pay POS Authorization Reversal 18
    Chase Pay POS Follow-On Credit 19
Recent Revisions to This Document

<table>
<thead>
<tr>
<th>Release</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>December 2016</td>
<td>Initial release.</td>
</tr>
</tbody>
</table>
About This Guide

Audience and Purpose

This guide is written for merchants who want to enable customers to use Chase Pay point of sale (POS) with a mobile app. This document provides an overview of integrating Chase Pay POS and CyberSource services into an order management system and describes how to use the CyberSource API to request authorizations.

Implementing the CyberSource services requires software development skills. You must write code that uses the API request and reply fields to integrate the services into your existing order management system.

Conventions

The following special statements are used in this document:

- **Note**: A *Note* contains helpful suggestions or references to material not contained in this document.

- **Important**: An *Important* statement contains information essential to successfully completing a task or learning a concept.
The following text conventions are used in this document:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>bold</td>
<td>Field and service names in text; for example: Include the <strong>ccAuthService_run</strong> field.</td>
</tr>
</tbody>
</table>
| monospace  | XML elements  
|            | Code examples  
|            | Values for API fields; for example:  
|            | Set the **ccAuthService_run** field to **true**. |

**Related Documentation**

- *Getting Started with CyberSource Advanced for the Simple Order API* (PDF | HTML)
- *Card-Present Processing Using the Simple Order API* (PDF | HTML)

Refer to the Support Center for complete CyberSource technical documentation:

http://www.cybersource.com/support_center/support_documentation

**Customer Support**

For support information about any CyberSource service, visit the Support Center:

http://www.cybersource.com/support
Introduction to Chase Pay POS Transactions

Requirements

You must:

- Create a CyberSource merchant evaluation account, if you do not have one already: https://www.cybersource.com/register/

- Obtain a merchant category code from your Chase account manager, and then contact CyberSource Customer Support to add the merchant category code to your CyberSource account.

- Have a merchant account with Chase Paymentech Solutions.

- Install a CyberSource Simple Order API client.

Supported Card Type

CyberSource supports the Visa card type for Chase Pay POS transactions.

How Chase Pay POS Works

1. The customer uses your app and chooses Chase Pay.

2. Your app uses the Chase Pay SDK to authenticate the customer credentials for the Chase Pay wallet with the Chase Pay host, obtains the session ID, and retrieves a list of the credit cards that are in the customer’s Chase Pay wallet.

3. Your app displays the list of credit cards, and the customer selects a card.

4. Your app sends the session ID, account index, and order details to your order management system.
Your order management system uses the session ID and account index to retrieve the payment details from the Chase Pay host.

Your order management system matches the customer's shopping cart to the customer and the payment details.

To process the payment, your order management system sends a request through CyberSource to Chase Paymentech Solutions. In addition to the basic fields required for retail chip card processing, you must also include the following fields in your request:

- **paymentNetworkToken_requestorID**—Set the value for this field to the value assigned by the token service provider.
- **paymentNetworkToken_transactionType**—Set the value for this field to 1, which indicates that tokenized track data is included in the request.
- **pos_entryMode**—Set the value for this field to **QRCode**.
- **pos_terminalCapability**—Set the value for this field to 4.
- **payment_solution**—Set the value for this field to 007, which indicates that this is a Chase Pay transaction.

The preceding fields are described in Appendix B, "API Fields," on page 20.

In the reply message that you receive from CyberSource, the `ccAuthReply_paymentNetworkTransactionID` value is the Visa transaction ID. This value is returned only for approved transactions.

### Testing

For the Chase Paymentech Solutions test data, see:

http://www.cybersource.com/developers/getting_started/test_and_manage/simple_order_api/paymentech/
Name-Value Pair Examples

Chase Pay POS Authorization

Example 1  Request Message: Chase Pay POS Authorization

ccAuthService_run=true
merchantID=JanesPlants
merchantReferenceCode=ABC123
purchaseTotals_currency=usd
purchaseTotals_grandTotalAmount=75.00
pos_entryMode=QRCode
pos_cardPresent=Y
pos_terminalCapability=4
pos_trackData=;4111111111111111=25122010123499999991?
pos_terminalID=87654321
card_cardType=001
ccAuthService_commerceIndicator=retail
evrRequest_combinedTags=9F33032040009505000000000009F3704238561349F10070
6011113A000009F26088717A1A173EA04D9F36020065820200009C01009F1A02084
09A03006209F02060000000020005F2A0208409F0306000000000000
paymentNetworkToken_requestorID=12323453456
paymentNetworkToken_transactionType=1
paymentSolution=007
Appendix A  Examples

Example 2  Reply Message: Chase Pay POS Authorization

merchantReferenceCode=ABC123
requestID=0305782650000167905080
decision=ACCEPT
reasonCode=100
purchaseTotals_currency=usd
ccAuthReply_reasonCode=100
ccAuthReply_amount=75.00
ccAuthReply_authorizationCode=831000
ccAuthReply_avsCode=2
ccAuthReply_authorizedDateTime=2016-08-11T22:47:57Z
ccAuthReply_processorResponse=100
ccAuthReply_reconciliationID=1094820975023470
ccAuthReply_paymentNetworkTransactionID=0412MCCNYJPWY
emvReply_combinedTags=9F330332040000950500000000009F3704518823719F100706011103A0000009F26081E1756ED02134E29F36020015820200009C01009F1A0208409A030006219F0206000000020005F2A028409F03060000000000

Chase Pay POS Capture

Example 3  Request Message: Chase Pay POS Capture

ccCaptureService_authRequestID=0305782650000167905080
merchantID=JanesPlants
merchantReferenceCode=ABC123
ccCaptureService_run=true
purchaseTotals_currency=usd
purchaseTotals_grandTotalAmount=75.00

Example 4  Reply Message: Chase Pay POS Capture

requestID=101982752034829057029
merchantReferenceCode=ABC123
decision=ACCEPT
reasonCode=100
purchaseTotals_currency=usd
ccCaptureReply_reasonCode=100
ccCaptureReply_amount=75.00
ccCaptureReply_reconciliationID=482046C3A7B94F5BD1FE3C66C
Chase Pay POS Sale

Example 5  Request Message: Chase Pay POS Sale

merchantID=JanesPlants
merchantReferenceCode=ABC123
purchaseTotals_currency=usd
purchaseTotals_grandTotalAmount=75.00
pos_entryMode=QRCode
pos_cardPresent=Y
pos_terminalCapability=4
pos_trackData=;4111111111111111=25122010123499999991?
pos_terminalID=87654321
card_cardType=001
ccAuthService_run=true
ccAuthService_commerceIndicator=retail
ccCaptureService_run=true
mvRequest_combinedTags=9F3303204000950500000000009F3704238561349F100706
  011103A000009F26088717A1A173EAA04D9F36020065820200009C01009F1A020840
  9A03006209F02060000000020005F2A020849F0306000000000000
paymentNetworkToken_requestorID=12323453456
paymentNetworkToken_transactionType=1
paymentSolution=007

Example 6  Reply Message: Chase Pay POS Sale

merchantReferenceCode=ABC123
requestID=030578265000167905080
decision=ACCEPT
reasonCode=100
purchaseTotals_currency=usd
ccAuthReply_reasonCode=100
ccAuthReply_amount=75.00
ccAuthReply_authorizationCode=831000
ccAuthReply_avsCode=2
ccAuthReply_authorizationDateTime=2016-08-11T22:47:57Z
ccAuthReply_processorResponse=100
ccAuthReply_reconciliationID=1094820975023470
ccAuthReply_paymentNetworkTransactionID=0412MCCNYJFY
ccAuthReply_cardCategory=J1
ccAuthReply_cardGroup=0
ccCaptureReply_reasonCode=100
ccCaptureReply_amount=75.00
ccCaptureReply_reconciliationID=1094820975023470
emvReply_combinedTags=9F3303204000950500000000009F3704518823719F100706
  011103A000009F26081E1756ED0E2133E29F36020015820200009C01009F1A020840
  9A03006219F020600000000020005F2A020849F0306000000000000
Chase Pay POS Authorization Reversal

Example 7  Request Message: Chase Pay POS Authorization Reversal

<table>
<thead>
<tr>
<th>merchantID=JanesPlants</th>
</tr>
</thead>
<tbody>
<tr>
<td>merchantReferenceCode=ABC123</td>
</tr>
<tr>
<td>purchaseTotals_currency=usd</td>
</tr>
<tr>
<td>purchaseTotals_grandTotalAmount=75.00</td>
</tr>
<tr>
<td>ccAuthReversalService_run=true</td>
</tr>
<tr>
<td>ccAuthReversalService_authRequestID=0305782650000167905080</td>
</tr>
</tbody>
</table>

Example 8  Reply Message: Chase Pay POS Authorization Reversal

<table>
<thead>
<tr>
<th>merchantReferenceCode=ABC123</th>
</tr>
</thead>
<tbody>
<tr>
<td>requestID=9827520348290570291013</td>
</tr>
<tr>
<td>decision=ACCEPT</td>
</tr>
<tr>
<td>reasonCode=100</td>
</tr>
<tr>
<td>purchaseTotals_currency=usd</td>
</tr>
<tr>
<td>ccAuthReversalReply_reasonCode=100</td>
</tr>
<tr>
<td>ccAuthReversalReply_amount=75.00</td>
</tr>
<tr>
<td>ccAuthReversalReply_processorResponse=100</td>
</tr>
</tbody>
</table>

Chase Pay POS Follow-On Credit

Example 9  Request Message: Chase Pay POS Follow-On Credit

<table>
<thead>
<tr>
<th>merchantID=JanesPlants</th>
</tr>
</thead>
<tbody>
<tr>
<td>merchantReferenceCode=ABC123</td>
</tr>
<tr>
<td>purchaseTotals_currency=usd</td>
</tr>
<tr>
<td>purchaseTotals_grandTotalAmount=75.00</td>
</tr>
<tr>
<td>ccCreditService_run=true</td>
</tr>
<tr>
<td>ccCreditService_captureRequestID=1019827520348290570293</td>
</tr>
</tbody>
</table>

Example 10  Reply Message: Chase Pay POS Follow-On Credit

<table>
<thead>
<tr>
<th>merchantReferenceCode=ABC123</th>
</tr>
</thead>
<tbody>
<tr>
<td>requestID=9057101982752034820293</td>
</tr>
<tr>
<td>decision=ACCEPT</td>
</tr>
<tr>
<td>reasonCode=100</td>
</tr>
<tr>
<td>purchaseTotals_currency=usd</td>
</tr>
<tr>
<td>ccCreditReply_reasonCode=100</td>
</tr>
<tr>
<td>ccCreditReply_amount=75.00</td>
</tr>
<tr>
<td>ccCreditReply_reconciliationID=C3A794F5BD1FE3C64820466C</td>
</tr>
</tbody>
</table>
XML Examples

Chase Pay POS Authorization

Example 11 Request Message: Chase Pay POS Authorization

```xml
:requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.128">
    <merchantID>JanesPlants</merchantID>
    <merchantReferenceCode>ABC123</merchantReferenceCode>
    <purchaseTotals>
        <currency>usd</currency>
        <grandTotalAmount>75.00</grandTotalAmount>
    </purchaseTotals>
    <pos>
        <entryMode>QRCode</entryMode>
        <cardPresent>Y</cardPresent>
        <terminalCapability>4</terminalCapability>
        <trackData>;4111111111111111=25122010123499999991</trackData>
        <terminalID>87654321</terminalID>
    </pos>
    <card>
        <cardType>001</cardType>
        <ccAuthService run="true">
            <commerceIndicator>retail</commerceIndicator>
        </ccAuthService>
        <emvRequest>
            <combinedTags>9F330320400095050000000009F3704238561349F100706011103A000009F260
            88717A173EAA04D9F36020065820200009C0109F1A0208409A00000209F020600000002
            0005F2A0208409F03060000000000</combinedTags>
        </emvRequest>
        <paymentNetworkToken>
            <requestorID>12323453456</requestorID>
            <transactionType>1</transactionType>
        </paymentNetworkToken>
        <paymentSolution>007</paymentSolution>
    </card>
</requestMessage>
```
Appendix A  
Examples

Example 12  
Reply Message: Chase Pay POS Authorization

```
<replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.128">
  <c:requestID>0305782650000167905080</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:purchaseTotals>
    <c:currency>usd</c:currency>
  </c:purchaseTotals>
  <c:ccAuthReply>
    <c:reasonCode>100</c:reasonCode>
    <c:amount>75.00</c:amount>
    <c:authorizationCode>831000</c:authorizationCode>
    <c:avsCode>2</c:avsCode>
    <c:authorizedDateTime>2016-08-11T22:47:57Z</c:authorizedDateTime>
    <c:processorResponse>100</c:processorResponse>
    <c:reconciliationID>1094820975023470</c:reconciliationID>
    <c:paymentNetworkTransactionID>0412MCCNYJPWY</c:paymentNetworkTransactionID>
  </c:ccAuthReply>
  <c:emvReply>
    <c:combinedTags>9F3303204000950500000000009F3704518823719F100706011103A000009F26081E17C6ED6A114E29F36020015820200009C0109F1A0208409A030006219F02060000000200000F2A0208409F0306000000000000</c:combinedTags>
  </c:emvReply>
</replyMessage>
```

---

**Chase Pay POS Capture**

Example 13  
Request Message: Chase Pay POS Capture

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.128">
  <merchantID>JanesPlants</merchantID>
  <merchantReferenceCode>ABC123</merchantReferenceCode>
  <purchaseTotals>
    <currency>usd</currency>
    <grandTotalAmount>75.00</grandTotalAmount>
  </purchaseTotals>
  <ccCaptureService run="true">
    <authRequestID>0305782650000167905080</authRequestID>
  </ccCaptureService>
</requestMessage>
```
Example 14  Reply Message: Chase Pay POS Capture

```
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.128">
  <c:requestID>1019827520348290570293</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:purchaseTotals>
    <c:currency>usd</c:currency>
  </c:purchaseTotals>
  <c:ccCaptureReply>
    <c:reasonCode>100</c:reasonCode>
    <c:amount>75.00</c:amount>
    <c:reconciliationID>482046C3A7E94F5BD1FE3C66C</c:reconciliationID>
  </c:ccCaptureReply>
</c:replyMessage>
```
Chase Pay POS Sale

Example 15  Request Message: Chase Pay POS Sale

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.128">
  <merchantID>JanesPlants</merchantID>
  <merchantReferenceCode>ABC123</merchantReferenceCode>
  <purchaseTotals>
    <currency>usd</currency>
    <grandTotalAmount>75.00</grandTotalAmount>
  </purchaseTotals>
  <pos>
    <entryMode>QRCode</entryMode>
    <cardPresent>Y</cardPresent>
    <terminalCapability>4</terminalCapability>
    <trackData>;4111111111111111=25122010123499999991?2</trackData>
    <terminalID>87654321</terminalID>
  </pos>
  <card>
    <cardType>001</cardType>
    <ccAuthService run="true">
      <commerceIndicator>retail</commerceIndicator>
    </ccAuthService>
    <ccCaptureService run="true"/>
    <emvRequest>
      <combinedTags>9F3303204000950500000000009F3704238561349F10076011103A00009F260
                    88717A1A173EAA04D9F36020065820200009C01009F1A0208409A030006209F020600000002
                    0005F2A0208409F03060000000000</combinedTags>
    </emvRequest>
  </card>
  <paymentNetworkToken>
    <requestorID>12323453456</requestorID>
    <transactionType>1</transactionType>
    <paymentNetworkToken>
    <paymentSolution>007</paymentSolution>
  </paymentSolution>
</requestMessage>
```
Example 16  Reply Message: Chase Pay POS Sale

```
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.128">
  <c:requestID>03057826500000167905080</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:purchaseTotals>
    <c:currency>usd</c:currency>
  </c:purchaseTotals>
  <c:ccAuthReply>
    <c:reasonCode>100</c:reasonCode>
    <c:amount>75.00</c:amount>
    <c:avsCode>2</c:avsCode>
    <c:authorizedDateTime>2016-08-11T22:47:57Z</c:authorizedDateTime>
    <c:processorResponse>100</c:processorResponse>
    <c:reconciliationID>1094820975023470</c:reconciliationID>
    <c:paymentNetworkTransactionID>0412MCNYPWY</c:paymentNetworkTransactionID>
    <c:cardCategory>J1</c:cardCategory>
    <c:cardGroup>0</c:cardGroup>
  </c:ccAuthReply>
  <c:emvReply>
    <c:combinedTags>
      9F330320400095050000000009F3704518823719F1007060111103A000009F2
      6081E1756ED0E2134E29F36020015820200009C0109F1A0208409A03006D219F0206000000
      020005F2A2028409F03060000000000</c:combinedTags>
  </c:emvReply>
</c:replyMessage>
```

Example 17  Request Message: Chase Pay POS Authorization Reversal

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.128">
  <merchantID>JanesPlants</merchantID>
  <merchantReferenceCode>ABC123</merchantReferenceCode>
  <purchaseTotals>
    <currency>usd</currency>
    <grandTotalAmount>75.00</grandTotalAmount>
  </purchaseTotals>
  <ccAuthReversalService run="true">
    <authRequestID>03057826500000167905080</authRequestID>
  </ccAuthReversalService>
</requestMessage>
```
Example 18  
Reply Message: Chase Pay POS Authorization Reversal

```xml
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.128">
  <c:requestID>9827520348290570291013</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:purchaseTotals>
    <c:currency>usd</c:currency>
  </c:purchaseTotals>
  <c:ccAuthReversalReply>
    <c:reasonCode>100</c:reasonCode>
    <c:amount>75.00</c:amount>
    <c:processorResponse>100</c:processorResponse>
  </c:ccAuthReversalReply>
</c:replyMessage>
```

---

Chase Pay POS Follow-On Credit

Example 19  
Request Message: Chase Pay POS Follow-On Credit

```xml
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.128">
  <merchantID>JanesPlants</merchantID>
  <merchantReferenceCode>ABC123</merchantReferenceCode>
  <purchaseTotals>
    <c:currency>usd</c:currency>
    <grandTotalAmount>75.00</grandTotalAmount>
  </purchaseTotals>
  <ccCreditService run="true">
    <captureRequestID>1019827520348290570293</captureRequestID>
  </ccCreditService>
</requestMessage>
```

---

Example 20  
Reply Message: Chase Pay POS Follow-On Credit

```xml
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.128">
  <c:requestID>9057101982752034820293</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:purchaseTotals>
    <c:currency>usd</c:currency>
  </c:purchaseTotals>
  <c:ccCreditReply>
    <c:reasonCode>100</c:reasonCode>
    <c:amount>75.00</c:amount>
    <c:reconciliationID>C3A7E94F5BD1FE3C64820466C</c:reconciliationID>
  </c:ccCreditReply>
</c:replyMessage>
```
API Fields

XML Schema Version

Use XML schema 1.128 or later.

Formatting Restrictions

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

The values of the item_#_ fields must not contain carets (^) or colons (;) because these characters are reserved for use by CyberSource services.

Values for request-level and item-level fields must not contain new lines or carriage returns. However, they can contain embedded spaces and any other printable characters. CyberSource removes all leading and trailing spaces.

Data Type Definitions

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

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integer</td>
<td>Whole number {..., -3, -2, -1, 0, 1, 2, 3, ...}</td>
</tr>
<tr>
<td>String</td>
<td>Sequence of letters, numbers, spaces, and special characters</td>
</tr>
</tbody>
</table>
# Request Fields

## Table 2  Request Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Used By: Required (R) or Optional (O)</th>
<th>Data Type &amp; Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>billTo_city</td>
<td>City of the billing address.</td>
<td>ccAuthService (O)</td>
<td>String (50)</td>
</tr>
<tr>
<td>billTo_country</td>
<td>Country of the billing address. Use the two-character ISO Standard Country Codes.</td>
<td>ccAuthService (O)</td>
<td>String (2)</td>
</tr>
<tr>
<td>billTo_email</td>
<td>Customer’s email address.</td>
<td>ccAuthService (O)</td>
<td>String (255)</td>
</tr>
<tr>
<td>billTo_firstName</td>
<td>Customer’s first name. For a credit card transaction, this name must match the name on the card.</td>
<td>ccAuthService (O)</td>
<td>String (60)</td>
</tr>
<tr>
<td>billTo_lastName</td>
<td>Customer’s last name. For a credit card transaction, this name must match the name on the card.</td>
<td>ccAuthService (O)</td>
<td>String (60)</td>
</tr>
<tr>
<td>billTo_phoneNumber</td>
<td>Customer’s phone number. CyberSource recommends that you include the country code when the order is from outside the U.S.</td>
<td>cc AuthService (O)</td>
<td>String (15)</td>
</tr>
</tbody>
</table>
| billTo_postalCode | Postal code for the billing address. The postal code must consist of 5 to 9 digits.  
|                 | When the billing country is the U.S., the 9-digit postal code must follow this format:  
|                 | [5 digits][dash][4 digits]  
|                 | Example: 12345-6789  
|                 | When the billing country is Canada, the 6-digit postal code must follow this format:  
|                 | [alpha][numeric][alpha][space][numeric][alpha][numeric]  
|                 | Example: A1B 2C3  
| billTo_state    | State or province of the billing address. Use the State, Province, and Territory Codes for the United States and Canada. | ccAuthService (O)                     | String (2)         |
| billTo_street1  | First line of the billing street address.                                  | ccAuthService (O)                     | String (60)        |
| billTo_street2  | Additional address information. Example: Attention: Accounts Payable.       | ccAuthService (O)                     | String (60)        |
| card_cardType   | Type of card to authorize. Possible value:  
|                 | • 001: Visa                                                               | ccAuthService (R)                     | String (3)         |
### Table 2  Request Fields (Continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Used By: Required (R) or Optional (O)</th>
<th>Data Type &amp; Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>ccAuthService_commerceIndicator</td>
<td>Type of transaction. For a Chase Pay POS transaction, you must set this field to retail.</td>
<td>ccAuthService (R)</td>
<td>String (13)</td>
</tr>
<tr>
<td>ccAuthService_run</td>
<td>Set to true to request credit card authorization.</td>
<td>ccAuthService (R)</td>
<td>String (255)</td>
</tr>
<tr>
<td>emvRequest_combinedTags</td>
<td>The EMV data is in the tag-length-value format and includes chip card tags, terminal tags, and transaction detail tags. Include all tags required by Chase Pay POS as described in the Chase Pay API Specification, Version 0.94. For information about the individual tags, see the “Application Specification” section in the EMV 4.3 Specifications: <a href="http://emvco.com">http://emvco.com</a>&lt;br&gt;&lt;br&gt;<strong>Important</strong> The following tags contain sensitive information and must not be included in this field:&lt;br&gt;« 56: Track 1 equivalent data&lt;br&gt;« 57: Track 2 equivalent data&lt;br&gt;« 5A: Application PAN&lt;br&gt;« 5F20: Cardholder name&lt;br&gt;« 5F24: Application expiration date&lt;br&gt;« 99: Transaction PIN&lt;br&gt;« 9F0B: Cardholder name (extended)&lt;br&gt;« 9F1F: Track 1 discretionary data&lt;br&gt;« 9F20: Track 2 discretionary data&lt;br&gt;</td>
<td>ccAuthService (R)</td>
<td>String (999)</td>
</tr>
<tr>
<td>item_#_productCode</td>
<td>Type of product. This value is used to identify the product category (electronic, handling, physical, service, or shipping). The default value is default. For a list of valid values, see the information about product codes in Credit Card Services Using the Simple Order API.&lt;br&gt;&lt;br&gt;When this field is not set to default or one of the values related to shipping and/or handling, the item_#<em>quantity, item</em>#<em>productName, and item</em>#_productSKU fields are required. For information about items and grand totals, see Getting Started with CyberSource Advanced for the Simple Order API.</td>
<td>ccAuthService (O)</td>
<td>String (30)</td>
</tr>
<tr>
<td>item_#_productName</td>
<td>Required when item_#_productCode is not default or one of the values related to shipping and/or handling.</td>
<td>ccAuthService (See description)</td>
<td>String (30)</td>
</tr>
</tbody>
</table>
### Table 2  Request Fields (Continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Used By: Required (R) or Optional (O)</th>
<th>Data Type &amp; Length</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>item_#_productSKU</code></td>
<td>Product identifier code. Required when <code>item_#_productCode</code> is not default or one of the values related to shipping and/or handling.</td>
<td>ccAuthService (See description)</td>
<td>String (15)</td>
</tr>
<tr>
<td><code>item_#_quantity</code></td>
<td>Default is 1. Required when <code>item_#_productCode</code> is not default or one of the values related to shipping and/or handling.</td>
<td>ccAuthService (See description)</td>
<td>Integer (10)</td>
</tr>
</tbody>
</table>
| `item_#_taxAmount`  | Total tax to apply to the product. This value cannot be negative. The tax amount and the unit price must be in the same currency. The tax amount field is additive. The following example uses a two-exponent currency such as USD:  

1. You include the following items in your request:  
   - `item_0_unitPrice=10.00`
   - `item_0_quantity=1`
   - `item_0_taxAmount=0.80`
   - `item_1_unitPrice=20.00`
   - `item_1_quantity=1`
   - `item_1_taxAmount=1.60`

2. The total amount authorized will be 32.40, not 30.00 with 2.40 of tax included. | ccAuthService (O) | String (15) |
<p>| <code>item_#_unitPrice</code>  | Per-item price of the product. You must include either this field or <code>purchaseTotals_grandTotalAmount</code> in your request. This value cannot be negative. For information about items and grand totals, see Getting Started with CyberSource Advanced for the Simple Order API. You can include a decimal point (.) in the value for this field, but you cannot include any other special characters. CyberSource truncates the amount to the correct number of decimal places. | ccAuthService (See description)       | String (15)        |
| <code>merchantID</code>        | Your CyberSource merchant ID. Use the same merchant ID for evaluation, testing, and production. | ccAuthService (R)                     | String (30)        |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Used By: Required (R) or Optional (O)</th>
<th>Data Type &amp; Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>merchantReferenceCode</td>
<td>Merchant-generated order reference or tracking number. CyberSource recommends that you send a unique value for each transaction so that you can perform meaningful searches for the transaction. For information about tracking orders, see <a href="#">Getting Started with CyberSource Advanced for the Simple Order API</a>.</td>
<td>ccAuthService (R)</td>
<td>String (50)</td>
</tr>
<tr>
<td>paymentNetworkToken</td>
<td>Value that identifies your business and indicates that the cardholder’s account number is tokenized. This value is assigned by the token service provider and is unique within the token service provider’s database.</td>
<td>ccAuthService (O)</td>
<td>String (11)</td>
</tr>
</tbody>
</table>
| _requestorID              | Type of transaction that provided the token data. This value does not specify the token service provider; it specifies the entity that provided you with information about the token. Possible value:  
  ■ 1: In-app transaction. An application on the customer’s mobile device provided the token data for an e-commerce transaction. | ccAuthService (R)                     | String (1)         |
| paymentSolution           | Digital solution identifier. Possible value:  
  ■ 007: Chase Pay | ccAuthService (R)                     | String (3)         |
| pos_cardPresent           | Indicates whether the card is present at the time of the transaction. Possible value:  
  ■ N: Card is not present. | ccAuthService (R)                     | String (1)         |
| pos_entryMode             | Method of entering credit card information into the POS terminal. Possible value:  
  ■ QRCode | ccAuthService (R)                     | String (11)        |
| pos_terminalCapability    | POS terminal’s capability. Possible value:  
  ■ 4: Terminal can read chip cards. | ccAuthService (R)                     | Integer (1)        |
### Table 2  Request Fields (Continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Used By: Required (R) or Optional (O)</th>
<th>Data Type &amp; Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>pos_terminalID</td>
<td>Identifier for the terminal at your retail location. You can define this value yourself, but consult the processor for requirements.</td>
<td>ccAuthService (R)</td>
<td>String (8)</td>
</tr>
<tr>
<td>pos_trackData</td>
<td>Track 2 data. This value must include:</td>
<td>ccAuthService (R)</td>
<td>String (119)</td>
</tr>
<tr>
<td></td>
<td>- Semicolon (;) for the start sentinel</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Equals sign (=) after the card account number</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Question mark (?) for the end sentinel</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example: ;4111111111111111=25122010123499999991?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>purchaseTotals_currency</td>
<td>Currency used for the order. Use the three-character ISO Standard Currency Codes.</td>
<td>ccAuthService (R)</td>
<td>String (5)</td>
</tr>
<tr>
<td>purchaseTotals_grandTotalAmount</td>
<td>Grand total for the order. This value cannot be negative. You can include a decimal point (.), but you cannot include any other special characters. CyberSource truncates the amount to the correct number of decimal places. You must include either this field or <strong>item_#_unitPrice</strong> in your request. For information about items and grand totals, Getting Started with CyberSource Advanced for the Simple Order API.</td>
<td>ccAuthService (See description)</td>
<td>String (15)</td>
</tr>
<tr>
<td>shipTo_city</td>
<td>City of shipping address.</td>
<td>ccAuthService (Required if any shipping address information is included in the request and shipping to the U.S. or Canada; otherwise, optional.)</td>
<td>String (50)</td>
</tr>
<tr>
<td>shipTo_country</td>
<td>Country of shipping address. Use the two-character ISO Standard Country Codes.</td>
<td>ccAuthService (Required if any shipping address information is included in the request; otherwise, optional.)</td>
<td>String (2)</td>
</tr>
<tr>
<td>shipTo_firstName</td>
<td>First name of the person receiving the shipment.</td>
<td>ccAuthService (O)</td>
<td>String (60)</td>
</tr>
<tr>
<td>shipTo_lastName</td>
<td>Last name of the person receiving the shipment.</td>
<td>ccAuthService (O)</td>
<td>String (60)</td>
</tr>
</tbody>
</table>
### Table 2  Request Fields (Continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Used By: Required (R) or Optional (O)</th>
<th>Data Type &amp; Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>shipTo_postalCode</td>
<td>Postal code for the shipping address. The postal code must consist of 5 to 9 digits. When the shipping country is the U.S., the 9-digit postal code must follow this format: [5 digits][dash][4 digits] Example: 12345-6789 When the shipping country is Canada, the 6-digit postal code must follow this format: [alpha][numeric][alpha][space][numeric][alpha] [numeric] Example: A1B 2C3</td>
<td>ccAuthService (Required if any shipping address information is included in the request and shipping to the U.S. or Canada; otherwise, optional.)</td>
<td>String (10)</td>
</tr>
<tr>
<td>shipTo_state</td>
<td>State or province of shipping address. Use the State, Province, and Territory Codes for the United States and Canada.</td>
<td>ccAuthService (Required if any shipping address information is included in the request and shipping to the U.S. or Canada; otherwise, optional.)</td>
<td>String (2)</td>
</tr>
<tr>
<td>shipTo_street1</td>
<td>First line of shipping address.</td>
<td>ccAuthService (Required if any shipping address information is included in the request; otherwise, optional.)</td>
<td>String (60)</td>
</tr>
<tr>
<td>shipTo_street2</td>
<td>Second line of shipping address.</td>
<td>ccAuthService (O)</td>
<td>String (60)</td>
</tr>
</tbody>
</table>
Because CyberSource can add reply fields and reason codes at any time:

- You must parse the reply data according to the names of the fields instead of the field order in the reply. For more information about parsing reply fields, see the documentation for your client.
- Your error handler should be able to process new reason codes without problems.
- Your error handler should use the decision field to determine the result if it receives a reason code that it does not recognize.

Your payment processor can include additional API reply fields that are not documented in this guide.

### Table 3  Reply Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Returned By</th>
<th>Data Type &amp; Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>ccAuthReply_amount</td>
<td>Amount that was authorized.</td>
<td>ccAuthReply</td>
<td>String (15)</td>
</tr>
<tr>
<td>ccAuthReply_authorizationCode</td>
<td>Authorization code. Returned only when the processor returns this value.</td>
<td>ccAuthReply</td>
<td>String (7)</td>
</tr>
<tr>
<td>ccAuthReply_avsCode</td>
<td>AVS results. See Credit Card Services Using the Simple Order API for a detailed list of AVS codes.</td>
<td>ccAuthReply</td>
<td>String (1)</td>
</tr>
<tr>
<td>ccAuthReply_avsCodeRaw</td>
<td>AVS result code sent directly from the processor. Returned only when the processor returns this value.</td>
<td>ccAuthReply</td>
<td>String (10)</td>
</tr>
<tr>
<td>ccAuthReply_paymentNetworkTransactionID</td>
<td>Network transaction identifier (TID). You can use this value to identify a specific transaction when you are discussing the transaction with your processor.</td>
<td>ccAuthReply</td>
<td>String (15)</td>
</tr>
</tbody>
</table>
### Table 3  Reply Fields (Continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Returned By</th>
<th>Data Type &amp; Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>ccAuthReply_processorResponse</td>
<td>For most processors, this is the error message sent directly from the bank. Returned only when the processor returns this value.</td>
<td>ccAuthReply</td>
<td>String (10)</td>
</tr>
<tr>
<td>ccAuthReply_reasonCode</td>
<td>Numeric value corresponding to the result of the credit card authorization request. See <em>Credit Card Services Using the Simple Order API</em> for a detailed list of reason codes.</td>
<td>ccAuthReply</td>
<td>Integer (5)</td>
</tr>
<tr>
<td>ccAuthReply_reconciliationID</td>
<td>Reference number for the transaction.</td>
<td>ccAuthReply</td>
<td>String (60)</td>
</tr>
<tr>
<td>ccAuthReversalReply_processorResponse</td>
<td>For most processors, this is the error message sent directly from the bank. Returned only when the processor returns this value.</td>
<td>ccAuthReversalReply</td>
<td>String (10)</td>
</tr>
<tr>
<td>decision</td>
<td>Summarizes the result of the overall request. Possible values: ☐ ACCEPT ☐ ERROR ☐ REJECT ☐ REVIEW: Returned only when you use CyberSource Decision Manager. For details about these values, see the information about handling replies in <em>Getting Started with CyberSource Advanced for the Simple Order API</em>.</td>
<td>ccAuthReply</td>
<td>String (6)</td>
</tr>
<tr>
<td>emvReply_combinedTags</td>
<td>The EMV data is in the tag-length-value format and includes chip card tags, terminal tags, and transaction detail tags.</td>
<td>ccAuthReply</td>
<td>String (999)</td>
</tr>
<tr>
<td>invalidField_0...N</td>
<td>Fields in the request that have invalid data. For information about missing or invalid fields, see <em>Getting Started with CyberSource Advanced for the Simple Order API</em>. Note These fields are included as an aid to software developers only. Do not use these fields to interact with your customers.</td>
<td>ccAuthReply</td>
<td>String (100)</td>
</tr>
</tbody>
</table>
### Table 3  Reply Fields (Continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Returned By</th>
<th>Data Type &amp; Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>merchantReference Code</td>
<td>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 include corrupted characters.</td>
<td>ccAuthReply</td>
<td>String (50)</td>
</tr>
<tr>
<td>missingField_0...N</td>
<td>Required fields that were missing from the request. For information about missing or invalid fields, see <a href="#">Getting Started with CyberSource Advanced for the Simple Order API</a>. These fields are included as an aid to software developers only. Do not use these fields to interact with your customers.</td>
<td>ccAuthReply</td>
<td>String (100)</td>
</tr>
<tr>
<td>purchaseTotals_currency</td>
<td>Currency used for the order. For the possible values, see <a href="#">ISO Standard Currency Codes</a>.</td>
<td>ccAuthReply</td>
<td>String (5)</td>
</tr>
<tr>
<td>reasonCode</td>
<td>Numeric value corresponding to the result of the overall request. See <a href="#">Credit Card Services Using the Simple Order API</a> for a detailed list of reason codes.</td>
<td>ccAuthReply</td>
<td>Integer (5)</td>
</tr>
<tr>
<td>requestID</td>
<td>Identifier for the request generated by the client.</td>
<td>ccAuthReply</td>
<td>String (26)</td>
</tr>
</tbody>
</table>