Apple Pay

Using the Simple Order API

November 2019
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: http://www.cybersource.com/support

Copyright

© 2019 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 Unpublished rights reserved under the copyright laws of the United States.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

Authorize.Net, eCheck.Net, and The Power of Payment are registered trademarks of CyberSource Corporation. CyberSource, 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

### Recent Revisions to This Document

5

### About This Guide

7

- Audience and Purpose 7
- Conventions 7
  - Note and Important Statements 7
  - Text and Command Conventions 8

### Related Documents

8

### Customer Support

8

---

### Chapter 1 Apple Pay Integrations

9

- In-App Transactions Using the CyberSource API 9
  - Merchant Decryption 9
  - CyberSource Decryption 10

- Web Transactions 11
  - Merchant Decryption 11
  - CyberSource Decryption 11

- Requirements 12

- Apple Pay JavaScript 13
  - Apple Pay Button 13
  - ApplePaySession Class 13
  - Create ApplePaySession Object 14

- Merchant Validation 14

- Payment Confirmation 14

- Merchant Decryption 14

- CyberSource Decryption 15
Chapter 2  Getting Started  16
  Requirements  16
  Supported Processors, Card Types, and Optional Features  17
  Enrolling in Apple Pay  18
    Generating a New CSR  19
  Transaction Reports  20

Chapter 3  Requesting the Authorization Service  21
  Option 1: Merchant Decryption  21
    Visa Transaction  21
    Mastercard Transaction  23
    American Express Transaction  25
    Discover Transaction  27
    JCB Transaction  29
  Option 2: CyberSource Decryption  32
    Visa Transaction  32
    Mastercard Transaction  35
    American Express Transaction  37
    Discover Transaction  39
    JCB Transaction  41
  Additional CyberSource Services  43

Appendix A  API Fields  44
  Data Type Definitions  44
  Numbered Elements  44
  Relaxed Requirements for Address Data and Expiration Date  45
  Request Fields  46
  Reply Fields  53
Recent Revisions to This Document

<table>
<thead>
<tr>
<th>Release</th>
<th>Changes</th>
</tr>
</thead>
</table>
| November 2019 | Changed payment network tokenization to authorizations with payment network tokens throughout.  
                   SIX: added an important note. See SIX, page 18. |
| September 2019| This revision contains only editorial changes and no technical updates.  |
| May 2019      | Removed the following request fields that were erroneously added in the April 2019 release:  
                   ● ccSaleService_directoryServerTransactionID  
                   ● ccSaleService_networkTokenCryptogram  
                   ● ccSaleService_paSpecificationVersion  
                   Removed the following reply fields that were erroneously added in the April 2019 release:  
                   ● payerAuthEnrollReply_directoryServerTransactionID  
                   ● payerAuthValidateReply_directoryServerTransactionID |
<table>
<thead>
<tr>
<th>Release</th>
<th>Changes</th>
</tr>
</thead>
</table>
| April 2019  | Added support for tokenized transactions using a network token with 3D Secure or SecureCode. See "Option 1: Merchant Decryption," page 21. Added the following request fields that support tokenized transactions using a network token with 3D Secure or SecureCode (see "Request Fields," page 46):  
  - ccAuthService_directoryServerTransactionID  
  - ccAuthService_networkTokenCryptogram  
  - ccAuthService_paSpecificationVersion  
  - ccSaleService_directoryServerTransactionID  
  - ccSaleService_networkTokenCryptogram  
  - ccSaleService_paSpecificationVersion  
  Added the following reply fields that support tokenized transactions using a network token with 3D Secure or SecureCode (see "Reply Fields," page 53):  
  - payerAuthEnrollReply_directoryServerTransactionID  
  - payerAuthValidateReply_directoryServerTransactionID  
  Added support for the processor Elavon Americas. See "Supported Processors, Card Types, and Optional Features," page 17. Added support for merchant-initiated transactions as an optional feature for the following processors (see "Supported Processors, Card Types, and Optional Features," page 17):  
  - Chase Paymentech Solutions  
  - CyberSource through VisaNet  
  - Elavon Americas  
  Added support for subsequent authorizations as an optional feature for the following processors (see "Supported Processors, Card Types, and Optional Features," page 17):  
  - FDC Nashville Global  
  - JCN Gateway  
  Added support for the following optional features by Elavon Americas (see "Supported Processors, Card Types, and Optional Features," page 17):  
  - Multiple partial captures  
  - Recurring payments                                                                                                                                 |
| March 2019  | Added support for the processor Credit Mutuel-CIC. See "Supported Processors, Card Types, and Optional Features," page 17. Added support for recurring payments as an optional feature for the processors Credit Mutuel-CIC and SIX. See "Supported Processors, Card Types, and Optional Features," page 17.
About This Guide

Audience and Purpose

This document is written for merchants who want to use Apple Pay in an iOS application and use information from Apple to process payments through CyberSource. This document provides an overview for integrating Apple and CyberSource services into an order management system.

Conventions

Note and Important Statements

A Note contains helpful suggestions or references to material not contained in the document.

An Important statement contains information essential to successfully completing a task or learning a concept.
Text and Command Conventions

<table>
<thead>
<tr>
<th>Convention</th>
<th>Usage</th>
</tr>
</thead>
</table>
| **Bold**   | ■ Field and service names in text; for example:  
  Include the **card_accountNumber** field.  
  ■ Items that you are instructed to act upon; for example:  
  Click **Save**. |

**Screen text**  
■ XML elements.  
■ Code examples and samples.  
■ Text that you enter in an API environment; for example:  
  Set the **ccAuthService_run** field to **true**.

Related Documents

CyberSource Documents:  
■ *Business Center User Guide* (PDF | HTML)  
■ *Business Center Reporting User Guide* (PDF | HTML)  
■ *Credit Card Services Using the Simple Order API* (PDF | HTML)  
■ *Authorizations with Payment Network Tokens Using the Simple Order API* (PDF | HTML)

Apple Documents:  
■ *PassKit Framework Reference*

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
In-App Transactions Using the CyberSource API

Merchant Decryption

1. When the customer chooses to pay with Apple Pay, you use the Apple PassKit Framework to request the encrypted payment data from Apple.

2. Apple uses the Secure Element to create a payment token (the `PKPaymentToken` structure) and encrypt the token’s payment data (the `paymentData` field of the `PKPaymentToken` structure) before it sends it to your application.

3. You forward the encrypted payment data to your e-commerce back-end system to decrypt. For information on decryption, see:

   https://developer.apple.com/library/ios/documentation/PassKit/Reference/PaymentTokenJSON/PaymentTokenJSON.html#//apple_ref/doc/uid/TP40014929-CH8-SW1

4. Using the CyberSource API, you submit the authorization request and include the decrypted payment data. See "Option 1: Merchant Decryption," page 21.
5 CyberSource forwards the information to the payment network, including your processor and the relevant payment card company.

---

**Important**

You must use the Business Center or one of the CyberSource API services to capture, credit, or void the authorization. See *Credit Card Services Using the Simple Order API.*

---

## CyberSource Decryption

1 When the customer chooses to pay with Apple Pay, you use the Apple PassKit Framework to request the encrypted payment data from Apple.

2 Apple uses the Secure Element to create a payment token (the `PKPaymentToken` structure) and encrypt the token’s payment data (the `paymentData` field of the `PKPaymentToken` structure) before it sends it to your application.

3 You forward the encrypted payment data to your e-commerce back-end system.

4 Using the CyberSource API, you submit the authorization request. In the `encryptedPayment_data` field include the Base64 encoded value obtained from the `paymentData` field of the `PKPaymentToken` structure. See "Option 2: CyberSource Decryption," page 32.

5 CyberSource decrypts the payment data and forwards the information to the payment network, including your processor and the relevant payment card company.

---

**Important**

You must use the Business Center or one of the CyberSource API services to capture, credit, or void the authorization. See *Credit Card Services Using the Simple Order API.*
Web Transactions

Merchant Decryption

1 When the customer chooses to pay with Apple Pay, you use the Apple Pay JavaScript to request the encrypted payment data from Apple.

2 Apple uses the Secure Element to create a payment token (the `PKPaymentToken` structure) and encrypt the token’s payment data (the `paymentData` field of the `PKPaymentToken` structure) before it sends it your application via the `onpaymentauthorized` callback function.

3 You forward the encrypted payment data to your e-commerce back-end system. For information on decryption, see:

```https://developer.apple.com/library/ios/documentation/PassKit/Reference/PaymentTokenJSON/PaymentTokenJSON.html#//apple_ref/doc/uid/TP40014929-CH8-SW1```

4 Using the CyberSource API, you submit the authorization request and include the decrypted payment data. See "Option 2: CyberSource Decryption," page 32.

5 CyberSource forwards the information to the payment network, including your processor and the relevant payment card company.

Important You must use the Business Center or one of the CyberSource API services to capture, credit, or void the authorization. See Credit Card Services Using the Simple Order API.

CyberSource Decryption

1 When the customer chooses to pay with Apple Pay, you use the Apple Pay JavaScript to request the encrypted payment data from Apple.

2 Apple uses the Secure Element to create a payment token (the `PKPaymentToken` structure) and encrypt the token’s payment data (the `paymentData` field of the `PKPaymentToken` structure) before it sends it your application via the `onpaymentauthorized` callback function.

3 You forward the encrypted payment data to your e-commerce back-end system.

4 Using the CyberSource API, you submit the authorization request. In the `encryptedPayment_data` field include the Base64 encoded value obtained from the `paymentData` field of the `PKPaymentToken` structure. See "Option 2: CyberSource Decryption," page 32.
CyberSource decrypts the payment data and forwards the information to the payment network, including your processor and the relevant payment card company.

**Important**
You must use the Business Center or one of the CyberSource API services to capture, credit, or void the authorization. See *Credit Card Services Using the Simple Order API*.

## Requirements

You must be an Admin or Team Agent user of your Apple Developer Program account.

For details on each requirement below, see:


### To configure your requirements:

**Step 1** Register your merchant ID.

- If you are currently processing In-App transactions, you can use the same merchant ID for processing Web transactions.

**Step 2** Create or upload a Certificate Signing Request (CSR), which is used to encrypt the payment information during the payment process.

- If you are using the merchant decryption method (see "Option 1: Merchant Decryption," page 21), create a CSR.

- If you are using the CyberSource decryption method (see "Option 2: CyberSource Decryption," page 32), upload the CSR that you created in the Business Center (see "Enrolling in Apple Pay," page 18).

- If you are currently processing In-App transactions, you can use the same CSR for processing Web transactions.

**Step 3** Register your domain. Registration is required in order to use Apple Pay on your web site.

**Step 4** Create a Merchant Identity Certificate. This certificate is required in order to connect to the Apple servers.
Apple Pay Integrations

Chapter 1

Apple Pay Using the Simple Order API

November 2019

13

Chapter 1  Apple Pay Integrations

Apple Pay JavaScript

Use the Apple Pay JavaScript to accept Apple Pay payments on your web site. The Apple Pay JavaScript tests that Apple Pay exists on your web site, displays the Apple Pay sheet, and receives the payment token.

Apple Pay Button

When a customer clicks or taps an Apple Pay button, it must invoke the Apple Pay payment sheet.

Important

For information on how to use Apple Pay buttons and the button styles, see:


You can use CSS templates provided by Apple to display the Apple Pay button on your web site. There are two templates: logo only button and buy with button. For more information, see Displaying the Apple Pay Button.

ApplePaySession Class

The ApplePaySession class manages the payment process on your web site. The ApplePaySession object is the entry point for Apple Pay on your web site.

Before displaying the Apple Pay button (see ”Apple Pay Button,” page 13) or creating an Apple Pay session (see ”Create ApplePaySession Object,” page 14), ensure that the Apple Pay JavaScript API is available and enabled on the device.

To enable the Apple Pay JavaScript API:

Step 1  Verify that the window.ApplePaySession class exists.

Step 2  Call its canMakePayments or canMakePaymentsWithActiveCard method:

- canMakePayments—verifies that the device is enabled for Apple Pay.

- canMakePaymentsWithActiveCard—verifies that the device is enabled for Apple Pay and the customer has a card stored on the device. You can call this method only if Apple Pay is the default payment method during your checkout flow, or if you want to add the Apple Pay button to your product detail page.

Note  All optional features are described in Authorizations with Payment Network Tokens Using the Simple Order API (PDF | HTML).
Create ApplePaySession Object

There are two required arguments when creating an ApplePaySession object:

- Version number—the API version is 1.
- Payment request—the PaymentRequest dictionary contains the information required in order to display the payment form.

When the session is created, call its begin method to display the payment form. This method can be called only when invoked by a user’s request.

Merchant Validation

When the payment form is displayed, the onvalidatemerchant callback function is called and provides a URL to pass to your server for validating the merchant session. Refer to the Merchant Validation section.

Payment Confirmation

When the customer confirms the payment by clicking or tapping the Apple Pay button, the onpaymentauthorized callback function is called and provides the payment token.

Merchant Decryption

Forward the encrypted payment data to your e-commerce back-end system to decrypt. For information on decryption, see:

https://developer.apple.com/library/ios/documentation/PassKit/Reference/PaymentTokenJSON/PaymentTokenJSON.html#//apple_ref/doc/uid/TP40014929-CH8-SW1

Using the CyberSource API, submit the authorization request and include the decrypted payment data. See "Option 1: Merchant Decryption," page 21.
CyberSource Decryption

Forward the encrypted payment data to your e-commerce back-end system.

Using the CyberSource API, submit the authorization request. In the `encryptedPayment_data` field include the Base64 encoded value obtained from the `paymentData` object. Example 1 shows the JavaScript for obtaining this value. See “Option 2: CyberSource Decryption,” page 32.

Example 1    JavaScript for Obtaining a Base64-Encoded Value

```javascript
session.onpaymentauthorized = function (event) {
    var paymentDataString = JSON.stringify(event.payment.token.paymentData);
    var paymentDataBase64 = btoa(paymentDataString);
    // Use paymentDataBase64 in your authorization request
}
```
Requirements

- CyberSource account. If you do not already have a CyberSource account, contact your local CyberSource sales representative. You can find your local Sales office here: http://www.cybersource.com/locations/

- Merchant account with a supported processor (see Table 1, “Processors, Card Types, and Optional Features,” on page 17).

- You must have an Admin or Team Agent user of the Apple Pay Developer account.

---

**Important**

Apple Pay relies on authorizations with payment network tokens. You can sign up for Apple Pay only when both of the following statements are true:

- Your processor supports payment network tokens.
- CyberSource supports payment network tokens with your processor.

If one or both of the preceding statements are not true, you must take one of the following actions before you can sign up for Apple Pay:

- Obtain a new merchant account with a processor that supports payment network tokens.
- Wait until your processor supports payment network tokens.
## Supported Processors, Card Types, and Optional Features

All optional features, except split shipments, are described in *Authorizations with Payment Network Tokens Using the Simple Order API* (PDF | HTML). Split shipments are described in *Credit Card Services Using the Simple Order API* (PDF | HTML).

### Table 1  Processors, Card Types, and Optional Features

<table>
<thead>
<tr>
<th>Processor</th>
<th>Card Types</th>
<th>Optional Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>American Express Direct</td>
<td>American Express</td>
<td>■ Multiple partial captures&lt;br&gt;■ Recurring payments</td>
</tr>
<tr>
<td>Barclays</td>
<td>Visa, Mastercard, Maestro (International), Maestro (UK Domestic)</td>
<td>■ Multiple partial captures&lt;br&gt;■ Recurring payments</td>
</tr>
<tr>
<td>Chase Paymentech Solutions</td>
<td>Visa, Mastercard, American Express, Discover, Maestro (International)</td>
<td>■ Merchant-Initiated transactions&lt;br&gt;■ Multiple partial captures&lt;br&gt;■ Recurring payments</td>
</tr>
<tr>
<td>Credit Mutuel-CIC</td>
<td>Visa, Mastercard, Cartes Bancaires</td>
<td>Recurring Payments</td>
</tr>
<tr>
<td>CyberSource through VisaNet.</td>
<td>Visa, Mastercard</td>
<td>■ Merchant-Initiated transactions&lt;br&gt;■ Recurring payments&lt;br&gt;■ Split shipments</td>
</tr>
<tr>
<td>The supported acquirers are:</td>
<td></td>
<td>■ Australia and New Zealand Banking Group Ltd. (ANZ)&lt;br&gt;■ CitiBank Singapore Ltd.&lt;br&gt;■ Global Payments Asia Pacific&lt;br&gt;■ Vantiv&lt;br&gt;■ Westpac</td>
</tr>
<tr>
<td>Elavon Americas</td>
<td>Visa, Mastercard, American Express, JCB, Discover</td>
<td>■ Merchant-Initiated transactions&lt;br&gt;■ Multiple partial captures&lt;br&gt;■ Recurring payments</td>
</tr>
<tr>
<td>FDC Compass</td>
<td>Visa, Mastercard, American Express</td>
<td>■ Multiple partial captures&lt;br&gt;■ Recurring payments</td>
</tr>
<tr>
<td>FDC Nashville Global</td>
<td>Visa, Mastercard, American Express, Discover</td>
<td>■ Multiple partial captures&lt;br&gt;■ Recurring payments&lt;br&gt;■ Subsequent authorizations</td>
</tr>
<tr>
<td>GPN</td>
<td>Visa, Mastercard, American Express</td>
<td>■ Recurring payments&lt;br&gt;■ Split shipments</td>
</tr>
</tbody>
</table>
**Enrolling in Apple Pay**

**To enroll in Apple Pay:**

**Step 1** Log in to the Business Center:
- Test transactions: https://ebctest.cybersource.com/ebc2/
- Live transactions: https://ebc2.cybersource.com/ebc2/

**Step 2** On the left navigation pane, click the **Payment Configuration** icon.

**Step 3** Click **Digital Payment Solutions**. The Digital Payments page appears.
Step 4  Click **Configure**. The Apple Pay Registration panel opens.

Step 5  Enter your Apple Merchant ID.

Step 6  Click **Generate New CSR**.

Step 7  To download your CSR, click the **Download** icon next to the key.

Step 8  Follow your browser’s instructions to save and open the file.

Step 9  Complete the enrollment process by submitting your CSR to Apple.

Step 10  For information about adding certificates to your Apple Merchant ID, refer to the Apple Pay PassKit:


---

**Note**

If you are using a CyberSource test account, you must connect to the Apple developer system and not to the Apple production system.

---

**Important**

After you complete your testing, you must create a new CSR for the CyberSource production system, and you must use that CSR for the Apple production system. Until you perform these steps, you cannot enable payments in your iOS application.

---

Step 12  Repeat Steps 1 through 10 with your CyberSource production account and the Apple production account.

---

**Generating a New CSR**

**To generate a new CSR:**

---

Step 1  Log in to the Business Center:

- Test transactions: [https://ebctest.cybersource.com/ebc2/](https://ebctest.cybersource.com/ebc2/)
- Live transactions: [https://ebc2.cybersource.com/ebc2/](https://ebc2.cybersource.com/ebc2/)

Step 2  On the left navigation pane, click the **Payment Configuration** icon.

Step 3  Click **Digital Payment Solutions**. The Digital Payments page appears.
Step 4  Click **Configure**. The Apple Pay Registration panel opens.

Step 5  To download your CSR, click the **Download** icon next to the key.

Step 6  Follow your browser's instructions to save and open the file.

Step 7  To edit your Apple Merchant ID, click the **Edit** icon. The Edit CSR panel opens.

Step 8  Modify your merchant ID as necessary and click **Update**.

---

**Transaction Reports**

Go to the Business Center and use the Transaction Request Report to obtain information about your transactions:

- In the Business Center, use the Transaction Search page to identify Apple transactions. You can search for transactions by date, application type, customer name, and other transaction identifiers.

- For information about the Transaction Request Report, see the *Business Center Reporting User Guide* (PDF | HTML)
Requesting the Authorization Service

**Option 1: Merchant Decryption**

**Visa Transaction**

To request an authorization for a Visa transaction:

- **Step 1** Set the `card_accountNumber` field to the payment network token value.
- **Step 2** Set the `card_expirationMonth` and `card_expirationYear` fields to the values from the payment network token expiration date field.
- **Step 3** Set the `ccAuthService_cavv` field to the 3D Secure cryptogram of the payment network token.
- **Step 4** Set the `ccAuthService_networkTokenCryptogram` field to the network token cryptogram.
- **Step 5** Set the `paymentNetworkToken_transactionType` field to 1.
- **Step 6** Set the `ccAuthService_commerceIndicator` field to the ECI value contained in the Apple Pay response payload (5=vvb and 7=internet).
- **Step 7** Set the `paymentSolution` field to 001.

- **Note** See the Relaxed Requirements for Address Data and Expiration Date page and "Request Fields," page 46, for details and field descriptions.
Example 2  Authorization Request (Visa)

```xml
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.121">
  <merchantID>demomerchant</merchantID>
  <merchantReferenceCode>demorefnum</merchantReferenceCode>
  <billTo>
    <firstName>Jane</firstName>
    <lastName>Smith</lastName>
    <street1>123 Main Street</street1>
    <city>Small Town</city>
    <state>CA</state>
    <postalCode>98765</postalCode>
    <country>US</country>
    <email>jsmith@example.com</email>
  </billTo>
  <purchaseTotals>
    <currency>USD</currency>
    <grandTotalAmount>5.00</grandTotalAmount>
  </purchaseTotals>
  <card>
    <accountNumber>4111111111111111</accountNumber>
    <expirationMonth>12</expirationMonth>
    <expirationYear>2020</expirationYear>
    <cvNumber>123</cvNumber>
    <cardType>001</cardType>
  </card>
  <ccAuthService run="true">
    <cavv>ABCDEFabcdefABCDEFabcdef0987654321234567</cavv>
    <commerceIndicator>internet</commerceIndicator>
  </ccAuthService>
  <paymentNetworkToken>
    <transactionType>1</transactionType>
  </paymentNetworkToken>
  <paymentSolution>001</paymentSolution>
</requestMessage>
```
Example 3  Authorization Response (Visa)

```xml
<replyMessage>
  <merchantReferenceCode>demorefnum</merchantReferenceCode>
  <requestID>4465840340765000001541</requestID>
  <decision>ACCEPT</decision>
  <reasonCode>100</reasonCode>
  <requestToken>Abj/7wSR5C/4Icd2fdAKakGLadfg5535r/ghx3Z90AoBj3u</requestToken>
  <purchaseTotals>
    <currency>USD</currency>
  </purchaseTotals>
  <ccAuthReply>
    <reasonCode>100</reasonCode>
    <amount>5.00</amount>
    <authorizationCode>888888</authorizationCode>
    <avsCode>X</avsCode>
    <avsCodeRaw>I1</avsCodeRaw>
    <authorizedDateTime>2015-11-03T20:53:54Z</authorizedDateTime>
    <processorResponse>100</processorResponse>
    <reconciliationID>11267051CGJSMQDC</reconciliationID>
  </ccAuthReply>
</replyMessage>
```

### Mastercard Transaction

To request an authorization for a Mastercard transaction:

- **Step 1** Set the `card_accountNumber` field to the payment network token value.
- **Step 2** Set the `card_expirationMonth` and `card_expirationYear` fields to the values from the payment network token expiration date field.
- **Step 3** Set the `ucaf_authenticationData` field to the 3D Secure cryptogram of the payment network token.
- **Step 4** Set the `ccAuthService_networkTokenCryptogram` field to the network token cryptogram.
- **Step 5** Set the `ucaf_collectionIndicator` field to 2.
- **Step 6** Set the `paymentNetworkToken_transactionType` field to 1.

**Note**

See the [Relaxed Requirements for Address Data and Expiration Date page](#) and "Request Fields," page 46, for details and field descriptions.
Step 7  Set the `ccAuthService_commerceIndicator` field to ECI value contained in the Apple Pay response payload.

Step 8  Set the `paymentSolution` field to 001.

Example 4  Authorization Request (Mastercard)

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.121">
  <merchantID>demomerchant</merchantID>
  <merchantReferenceCode>demorefnum</merchantReferenceCode>
  <billTo>
    <firstName>Jane</firstName>
    <lastName>Smith</lastName>
    <street1>123 Main Street</street1>
    <city>Small Town</city>
    <state>CA</state>
    <postalCode>98765</postalCode>
    <country>US</country>
    <email>jsmith@example.com</email>
  </billTo>
  <purchaseTotals>
    <currency>USD</currency>
    <grandTotalAmount>5.00</grandTotalAmount>
  </purchaseTotals>
  <card>
    <accountNumber>555555555555xxxx</accountNumber>
    <expirationMonth>12</expirationMonth>
    <expirationYear>2020</expirationYear>
    <cvNumber>123</cvNumber>
    <cardType>002</cardType>
  </card>
  <ucaf>
    <authenticationData>ABCDEFabcdefABCDscdef0987654321234567</authenticationData>
    <collectionIndicator>2</collectionIndicator>
  </ucaf>
  <ccAuthService run="true">
    <commerceIndicator>spa</commerceIndicator>
  </ccAuthService>
  <paymentNetworkToken>
    <transactionType>1</transactionType>
  </paymentNetworkToken>
  <paymentSolution>001</paymentSolution>
</requestMessage>
```
Example 5  Authorization Response (Mastercard)

<replyMessage>
  <merchantReferenceCode>demorefnum</merchantReferenceCode>
  <requestID>446584034076500001541</requestID>
  <decision>ACCEPT</decision>
  <reasonCode>100</reasonCode>
  <requestToken>Ahj/7wSR5C/41cd2fdAKakGLadfg5535r/ghx3Z90AoBj3u</requestToken>
  <purchaseTotals>
    <currency>USD</currency>
  </purchaseTotals>
  <ccAuthReply>
    <reasonCode>100</reasonCode>
    <amount>5.00</amount>
    <authorizationCode>888888</authorizationCode>
    <avsCode>X</avsCode>
    <avsCodeRaw>I1</avsCodeRaw>
    <authorizedDateTime>2015-11-03T20:53:54Z</authorizedDateTime>
    <processorResponse>100</processorResponse>
    <reconciliationID>11267051CGJSMQDC</reconciliationID>
  </ccAuthReply>
</replyMessage>

American Express Transaction

To request an authorization for an American Express transaction:

See the Relaxed Requirements for Address Data and Expiration Date page and "Request Fields," page 46, for details and field descriptions.

Note

Set the card_accountNumber field to the payment network token value.

Step 2  Set the card_expirationMonth and card_expirationYear fields to the values from the payment network token expiration date field.

Step 3  Set the ccAuthService_cavv field to the 3D Secure cryptogram of the payment network token.

Important

Include the whole 20-byte cryptogram in the ccAuthService_cavv field. For a 40-byte cryptogram, split the cryptogram into two 20-byte binary values (block A and block B). Set the ccAuthService_cavv field to the block A value and set the ccAuthService_xid field to the block B value.
Step 4  Set the `ccAuthService_networkTokenCryptogram` field to the network token cryptogram.

Step 5  Set the `paymentNetworkToken_transactionType` field to 1.

Step 6  Set the `ccAuthService_commerceIndicator` field to ECI value contained in the Apple Pay response payload.

Step 7  Set the `paymentSolution` field to 001.

Example 6  Authorization Request (American Express)

```xml
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.121">
  <merchantID>demomerchant</merchantID>
  <merchantReferenceCode>demorefnum</merchantReferenceCode>
  <billTo>
    <firstName>Jane</firstName>
    <lastName>Smith</lastName>
    <street1>123 Main Street</street1>
    <city>Small Town</city>
    <state>CA</state>
    <postalCode>98765</postalCode>
    <country>US</country>
    <email>jsmith@example.com</email>
  </billTo>
  <purchaseTotals>
    <currency>USD</currency>
    <grandTotalAmount>5.00</grandTotalAmount>
  </purchaseTotals>
  <card>
    <accountNumber>37828224631xxxx</accountNumber>
    <expirationMonth>12</expirationMonth>
    <expirationYear>2020</expirationYear>
    <cvNumber>123</cvNumber>
    <cardType>003</cardType>
  </card>
  <ccAuthService run="true">
    <cavv>ABCDEFabcdefABCDEFabcdef0987654321234567</cavv>
    <commerceIndicator>aesk</commerceIndicator>
  </ccAuthService>
  <paymentNetworkToken>
    <transactionType>1</transactionType>
  </paymentNetworkToken>
  <paymentSolution>001</paymentSolution>
</requestMessage>
```
Example 7  Authorization Response (American Express)

```xml
<c:replyMessage>
  <c:requestID>4465840340765000001541</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:requestToken>Ahj/7wSR5C/4Icd2fdAKakGLadfg5535r/ghx3Z90AoBj3u</c:requestToken>
  <c:purchaseTotals>
    <c:currency>USD</c:currency>
  </c:purchaseTotals>
  <c:ccAuthService_cavv>
    <c:reasonCode>100</c:reasonCode>
    <c:amount>5.00</c:amount>
    <c:avsCode>X</c:avsCode>
    <c:avsCodeRaw>I1</c:avsCodeRaw>
    <c:authorizedDateTime>2015-11-03T20:53:54Z</c:authorizedDateTime>
    <c:processorResponse>100</c:processorResponse>
    <c:reconciliationID>11267051CGJSMQDC</c:reconciliationID>
  </c:ccAuthService_cavv>
</c:replyMessage>
```

Discover Transaction

To request an authorization for a Discover transaction:

Step 1  Set the `card_accountNumber` field to the payment network token value.

Step 2  Set the `card_expirationMonth` and `card_expirationYear` fields to the values from the payment network token expiration date field.

Step 3  Set the `ccAuthService_cavv` field to the 3D Secure cryptogram of the payment network token.

Step 4  Set the `ccAuthService_networkTokenCryptogram` field to the network token cryptogram.

Step 5  Set the `paymentNetworkToken_transactionType` field to 1.

Note  See the Relaxed Requirements for Address Data and Expiration Date page and "Request Fields," page 46, for details and field descriptions.
**Step 6** Set the `ccAuthService_commerceIndicator` field to ECI value contained in the Apple Pay response payload.

**Step 7** Set the `paymentSolution` field to `001`.

### Example 8  Authorization Request (Discover)

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.121">
  <merchantID>demomerchant</merchantID>
  <merchantReferenceCode>demorefnum</merchantReferenceCode>
  <billTo>
    <firstName>Jane</firstName>
    <lastName>Smith</lastName>
    <street1>123 Main Street</street1>
    <city>Small Town</city>
    <state>CA</state>
    <postalCode>98765</postalCode>
    <country>US</country>
    <email>jsmith@example.com</email>
  </billTo>
  <purchaseTotals>
    <currency>USD</currency>
    <grandTotalAmount>5.00</grandTotalAmount>
  </purchaseTotals>
  <card>
    <accountNumber>601111111111xxxx</accountNumber>
    <expirationMonth>12</expirationMonth>
    <expirationYear>2020</expirationYear>
    <cvNumber>123</cvNumber>
    <cardType>004</cardType>
  </card>
  <ccAuthService run="true">  
    <cavv>ABCDEFabcdefABCDEFabcdef0987654321234567</cavv>
    <commerceIndicator>dipb</commerceIndicator>
  </ccAuthService>
  <paymentNetworkToken>
    <transactionType>1</transactionType>
  </paymentNetworkToken>
  <paymentSolution>001</paymentSolution>
</requestMessage>
```
Chapter 3 Requesting the Authorization Service

Example 9 Authorization Response (Discover)

```xml
<c:replyMessage>
  <c:requestID>4465840340765000001541</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:requestToken>AhJ/7wSR5C/4Icd2fdAkaKGladg5535r/ghx3Z90AoBj3u</c:requestToken>
  <c:purchaseTotals>
    <c:currency>USD</c:currency>
  </c:purchaseTotals>
  <c:ccAuthReply>
    <c:reasonCode>100</c:reasonCode>
    <c:amount>5.00</c:amount>
    <c:authorizationCode>888888</c:authorizationCode>
    <c:avsCode>X</c:avsCode>
    <c:avsCodeRaw>I1</c:avsCodeRaw>
    <c:authorizedDateTime>2015-11-03T20:53:54Z</c:authorizedDateTime>
    <c:processorResponse>100</c:processorResponse>
    <c:reconciliationID>11267051CGJSMQDC</c:reconciliationID>
  </c:ccAuthReply>
</c:replyMessage>
```

JCB Transaction

To request an authorization for a JCB transaction:

1. Set the `card_accountNumber` field to the payment network token value.
2. Set the `card_expiration_Month` and `card_expirationYear` fields to the values from the payment network token expiration date field.
3. Set the `ccAuthService_cavv` field to the 3D Secure cryptogram of the payment network token.
4. Set the `paymentNetworkToken_transactionType` field to 1.
5. Set the `ccAuthService_eciRaw` field to the ECI value contained in the Apple Pay response payload.
6. Set the `PaymentSolution` field to 001.

Note

See the Relaxed Requirements for Address Data and Expiration Date page and "Request Fields," page 46, for details and field descriptions.
Example 10  Authorization Request (JCB)

```xml
:requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.121">
  <merchantID>demomerchant</merchantID>
  <merchantReferenceCode>demorefnum</merchantReferenceCode>
  <billTo>
    <firstName>Jane</firstName>
    <lastName>Smith</lastName>
    <street1>123 Main Street</street1>
    <city>Small Town</city>
    <state>CA</state>
    <postalCode>98765</postalCode>
    <country>US</country>
    <email>jsmith@example.com</email>
  </billTo>
  <purchaseTotals>
    <currency>USD</currency>
    <grandTotalAmount>5.00</grandTotalAmount>
  </purchaseTotals>
  <card>
    <accountNumber>356611111111xxxx</accountNumber>
    <expirationMonth>12</expirationMonth>
    <expirationYear>2020</expirationYear>
    <cvNumber>123</cvNumber>
    <cardType>001</cardType>
  </card>
  <ccAuthService run="true">
    <cavv>ABCDEFabcdefABCDEFabcdef0987654321234567</cavv>
    <eciRaw>5</eciRaw>
  </ccAuthService>
  <paymentNetworkToken>
    <transactionType>1</transactionType>
  </paymentNetworkToken>
  <paymentSolution>001</paymentSolution>
</requestMessage>
```
Example 11  Authorization Reply (JCB)

```
<c:replyMessage>
  <c:requestID>4465840340765000001541</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:requestToken>
    Ahj/7wSR5C/41cd2fdAKakGLadfg5535r/ghx3290AcBj3u
  </c:requestToken>
  <c:purchaseTotals>
    <c:currency>USD</c:currency>
  </c:purchaseTotals>
  <c:ccAuthReply>
    <c:reasonCode>100</c:reasonCode>
    <c:amount>5.00</c:amount>
    <c:authorizationCode>888888</c:authorizationCode>
    <c:avsCode>X</c:avsCode>
    <c:avsCodeRaw>I1</c:avsCodeRaw>
    <c:authorizedDateTime>2015-11-03T20:53:54Z</c:authorizedDateTime>
    <c:processorResponse>100</c:processorResponse>
    <c:reconciliationID>11267051CGJSMQDC</c:reconciliationID>
  </c:ccAuthReply>
</c:replyMessage>
```

Example 12  NVP Request (JCB)

```
merchantID=demomerchant
merchantReferenceCode=demorefnum
billTo_firstName=Jane
billTo_lastName=Smith
billTo_street1=123 Main Street
billTo_city=Small Town
billTo_state=CA
billTo_postalCode=98765
billTo_country=US
billTo_email=jsmith@example.com
purchaseTotals_currency=USD
purchaseTotals_grandTotalAmount=5.00
card_accountNumber=356611111111xxxx
card_expirationYear=2020
card_cvnNumber=123
cardType=001
ccAuthService_cavv=ABCDEFabcdefABCDEFabcdef0987654321234567
ccAuthService_cavv=5
paymentNetworkToken_transactionType=1
paymentSolution=001
```
Example 13  NVP Reply (JCB)

| merchantReferenceCode=demorefnum |
| requestID=446584034076500000001541 |
| decision=accept |
| reasonCode=100 |
| requestToken=Ahj/7wSR5C/4Icd2fdAKakGLadfg5535r/ghx3290AoBj3u |
| purchaseTotals_currency=USD |
| ccAuthReply_reasonCode=100 |
| ccAuthReply_amount=5.00 |
| ccAuthReply_authorizationCode=888888 |
| ccAuthReply_avsCode=X |
| ccAuthReply_avsCodeRaw=I1 |
| ccAuthReply_authorizedDateTime=2015-11-03T20:53:54Z |
| ccAuthReply_processorResponse=100 |
| ccAuthReply_reconciliationID=11267051CGJSMQDC |

Option 2: CyberSource Decryption

Visa Transaction

To request an authorization for a Visa transaction:

See the Relaxed Requirements for Address Data and Expiration Date page and "Request Fields," page 46, for details and field descriptions.

**Step 1** Set the encryptedPayment_data field to the Base64-encoded value obtained from the paymentData property of the PKPaymentToken object. See "CyberSource Decryption," page 10.

**Step 2** Set the encryptedPayment_descriptor field to: RK1EPUNPTU1PT16BFBR55JTkFQUC5QQV1NRU5U

**Step 3** Set the paymentSolution field to 001.
Example 14  Authorization Request (Visa)

```xml
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.121">
  <merchantID>demomerchant</merchantID>
  <merchantReferenceCode>demorefnum</merchantReferenceCode>
  <billTo>
    <firstName>Jane</firstName>
    <lastName>Smith</lastName>
    <street1>123 Main Street</street1>
    <city>Small Town</city>
    <state>CA</state>
    <postalCode>98765</postalCode>
    <country>US</country>
    <email>jsmith@example.com</email>
  </billTo>
  <purchaseTotals>
    <currency>USD</currency>
    <grandTotalAmount>5.00</grandTotalAmount>
  </purchaseTotals>
  <encryptedPayment>
    <descriptor>Rk1EPUNPTU1PTi5BUFBMRS5JTkFQC5QQV1NRU5U</descriptor>
    <data>ABCDEFabcdefABCDEFabcdef0987654321234567</data>
    <encoding>Base64</encoding>
  </encryptedPayment>
  <card>
    <cardType>001</cardType>
  </card>
  <ccAuthService run="true"/>
  <paymentSolution>001</paymentSolution>
</requestMessage>
```
Example 15  Authorization Response (Visa)

```xml
<c:replyMessage>
  <c:requestID>4465840340765000001541</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:requestToken>Ahj/7wSR5C/41cd2fdAKakGLadfg5535r/ghx3Z90AoBj3u</c:requestToken>
  <c:token>
    <c:expirationMonth>07</c:expirationMonth>
    <c:expirationYear>2025</c:expirationYear>
    <c:prefix>239845</c:prefix>
    <c:suffix>2947</c:suffix>
  </c:token>
  <c:purchaseTotals>
    <c:currency>USD</c:currency>
  </c:purchaseTotals>
  <c:ccAuthReply>
    <c:reasonCode>100</c:reasonCode>
    <c:amount>5.00</c:amount>
    <c:authorizationCode>888888</c:authorizationCode>
    <c:avsCode>X</c:avsCode>
    <c:avsCodeRaw>I1</c:avsCodeRaw>
    <c:authorizedDateTime>2015-11-03T20:53:54Z</c:authorizedDateTime>
    <c:processorResponse>100</c:processorResponse>
    <c:reconciliationID>11267051CGJSMQDC</c:reconciliationID>
  </c:ccAuthReply>
</c:replyMessage>
```
Mastercard Transaction

To request an authorization for a Mastercard transaction:

Step 1 Set the encryptedPayment_data field to the Base64-encoded value obtained from the paymentData property of the PKPaymentToken object. See "CyberSource Decryption," page 10.

Step 2 Set the encryptedPayment_descriptor field to:
RklEPUNPTU1PTi5BUFBMRS5JTkFQUC5QQV1NRU5U

Step 3 Set the paymentSolution field to 001.

Example 16 Authorization Request (Mastercard)

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.121">
  <merchantID>demomerchant</merchantID>
  <merchantReferenceCode>demorefnum</merchantReferenceCode>
  <billTo>
    <firstName>Jane</firstName>
    <lastName>Smith</lastName>
    <street1>123 Main Street</street1>
    <city>Small Town</city>
    <state>CA</state>
    <postalCode>98765</postalCode>
    <country>US</country>
    <email>jsmith@example.com</email>
  </billTo>
  <purchaseTotals>
    <currency>USD</currency>
    <grandTotalAmount>5.00</grandTotalAmount>
  </purchaseTotals>
  <encryptedPayment>
    <descriptor>RklEPUNPTU1PTi5BUFBMRS5JTkFQUC5QQV1NRU5U</descriptor>
    <data>ABCDEFabcdefABCDEFabcdef0987654321234567</data>
    <encoding>Base64</encoding>
  </encryptedPayment>
  <card>
    <cardType>002</cardType>
  </card>
  <ccAuthService run="true"/>
  <paymentSolution>001</paymentSolution>
</requestMessage>
```
Example 17  Authorization Response (Mastercard)

```xml
<c:replyMessage>
  <c:requestID>4465840340765000001541</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:requestToken>Ahj/7wSR5C/4Icd2fdAKakGLadfg5535r/ghx3Z90AoBj3u</c:requestToken>
  <c:token>
    <c:expirationMonth>07</c:expirationMonth>
    <c:expirationYear>2025</c:expirationYear>
    <c:prefix>239845</c:prefix>
    <c:suffix>2947</c:suffix>
  </c:token>
  <c:purchaseTotals>
    <c:currency>USD</c:currency>
  </c:purchaseTotals>
  <c:ccAuthReply>
    <c:reasonCode>100</c:reasonCode>
    <c:amount>5.00</c:amount>
    <c:authorizationCode>888888</c:authorizationCode>
    <c:avsCode>X</c:avsCode>
    <c:avsCodeRaw>I1</c:avsCodeRaw>
    <c:authorizedDateTime>2015-11-03T20:53:54Z</c:authorizedDateTime>
    <c:processorResponse>100</c:processorResponse>
    <c:reconciliationID>11267051CGJSMQDC</c:reconciliationID>
  </c:ccAuthReply>
</c:replyMessage>
```
American Express Transaction

To request an authorization for an American Express transaction:

**Step 1** Set the `encryptedPayment_data` field to the Base64-encoded value obtained from the `paymentData` property of the `PKPaymentToken` object. See "CyberSource Decryption," page 10.

**Step 2** Set the `encryptedPayment_descriptor` field to:

```
RklEPUNPTU1PTi5BUFBMRS5JTkPQUC5QQV1NRU5U
```

**Step 3** Set the `paymentSolution` field to 001.

Example 18  Authorization Request (American Express)

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.121">
  <merchantID>demomerchant</merchantID>
  <merchantReferenceCode>demorefnum</merchantReferenceCode>
  <billTo>
    <firstName>Jane</firstName>
    <lastName>Smith</lastName>
    <street1>123 Main Street</street1>
    <city>Small Town</city>
    <state>CA</state>
    <postalCode>98765</postalCode>
    <country>US</country>
    <email>jsmith@example.com</email>
  </billTo>
  <purchaseTotals>
    <currency>USD</currency>
    <grandTotalAmount>5.00</grandTotalAmount>
  </purchaseTotals>
  <encryptedPayment>
    <descriptor>RklEPUNPTU1PTi5BUFBMRS5JTkPQUC5QQV1NRU5U</descriptor>
    <data>ABCDEFabcdefABCDEFabcdef0987654321234567</data>
    <encoding>Base64</encoding>
  </encryptedPayment>
  <card>
    <cardType>003</cardType>
  </card>
  <ccAuthService run="true"/>
  <paymentSolution>001</paymentSolution>
</requestMessage>
```
Example 19  Authorization Response (American Express)

```xml
<c:replyMessage>
  <c:requestID>446584034076500001541</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:requestToken>Ahj/7wSR5C/4Icd2fdAKakGLadfg5535r/ghx3Z90AoBj3u</c:requestToken>
  <c:token>
    <c:expirationMonth>07</c:expirationMonth>
    <c:expirationYear>2025</c:expirationYear>
    <c:prefix>239845</c:prefix>
    <c:suffix>2947</c:suffix>
  </c:token>
  <c:purchaseTotals>
    <c:currency>USD</c:currency>
  </c:purchaseTotals>
  <c:ccAuthReply>
    <c:reasonCode>100</c:reasonCode>
    <c:amount>5.00</c:amount>
    <c:authorizationCode>8888888</c:authorizationCode>
    <c:avsCode>X</c:avsCode>
    <c:avsCodeRaw>I1</c:avsCodeRaw>
    <c:authorizedDateTime>2015-11-03T20:53:54Z</c:authorizedDateTime>
    <c:processorResponse>100</c:processorResponse>
    <c:reconciliationID>11267051CGJSMDQ</c:reconciliationID>
  </c:ccAuthReply>
</c:replyMessage>
```
Discover Transaction

To request an authorization for a Discover transaction:

See the Relaxed Requirements for Address Data and Expiration Date page and "Request Fields," page 46, for details and field descriptions.

Step 1 Set the encryptedPayment_data field to the Base64-encoded value obtained from the paymentData property of the PKPaymentToken object. See "CyberSource Decryption," page 10.

Step 2 Set the encryptedPayment_descriptor field to:
Rk1EPUNPTU1PTi5BUFBMRS5JTkFQUC5QQV1NRU5U

Step 3 Set the paymentSolution field to 001.
Example 20  Authorization Request (Discover)

```xml
:requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.121">
  <merchantID>demomerchant</merchantID>
  <merchantReferenceCode>demorefnum</merchantReferenceCode>
  <billTo>
    <firstName>Jane</firstName>
    <lastName>Smith</lastName>
    <street1>123 Main Street</street1>
    <city>Small Town</city>
    <state>CA</state>
    <postalCode>98765</postalCode>
    <country>US</country>
    <email>jsmith@example.com</email>
  </billTo>
  <purchaseTotals>
    <currency>USD</currency>
    <grandTotalAmount>5.00</grandTotalAmount>
  </purchaseTotals>
  <encryptedPayment>
    <descriptor>RklEPUNPTU1PTi5BUFBMRS5JTkFQUC5QQV1NRU5U</descriptor>
    <data>ABCDEFabcdefABCDEFabcdef0987654321234567</data>
    <encoding>Base64</encoding>
  </encryptedPayment>
  <card>
    <cardType>004</cardType>
  </card>
  <paymentNetworkToken>
    <transactionType>1</transactionType>
    <paymentSolution>001</paymentSolution>
    <ccAuthService run="true"/>
  </paymentNetworkToken>
</requestMessage>
```
Example 21  Authorization Response (Discover)

```xml
<replyMessage>
  <merchantReferenceCode>demorefnum</merchantReferenceCode>
  <requestID>4465840340765000001541</requestID>
  <decision>ACCEPT</decision>
  <reasonCode>100</reasonCode>
  <requestToken>Ahj/7wSR5C/4Icd2fdAKakGLadfg5535r/ghx3Z90AoBj3u</requestToken>
  <token>
    <expirationMonth>07</expirationMonth>
    <expirationYear>2025</expirationYear>
    <prefix>239845</prefix>
    <suffix>2947</suffix>
  </token>
  <purchaseTotals>
    <currency>USD</currency>
  </purchaseTotals>
  <ccAuthReply>
    <reasonCode>100</reasonCode>
    <amount>5.00</amount>
    <authorizationCode>888888</authorizationCode>
    <avsCode>X</avsCode>
    <avsCodeRaw>I1</avsCodeRaw>
    <authorizedDateTime>2015-11-03T20:53:54Z</authorizedDateTime>
    <processorResponse>100</processorResponse>
    <reconciliationID>11267051CGJSMQDC</reconciliationID>
  </ccAuthReply>
</replyMessage>
```

Note
See the Relaxed Requirements for Address Data and Expiration Date page and “Request Fields,” page 46, for details and field descriptions.

**JCB Transaction**

To request an authorization for a JCB transaction:

- **Step 1** Set the `encryptedPayment_data` field to the Base64-encoded value obtained from the `paymentData` property of the `PKPaymentToken` object.
- **Step 2** Set the `encryptedPaymentDescriptor` field to `Rk1EPUNPTU1PTi5BUFBMRS5JTkFQUC5QQV1NRU5U`.
- **Step 3** Set the `paymentSolution` field to 001.
Example 22  Authorization Request (JCB)

```xml
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.121">
  <merchantID>demomerchant</merchantID>
  <merchantReferenceCode>demorefnum</merchantReferenceCode>
  <billTo>
    <firstName>Jane</firstName>
    <lastName>Smith</lastName>
    <street1>123 Main Street</street1>
    <city>Small Town</city>
    <state>CA</state>
    <postalCode>98765</postalCode>
    <country>US</country>
    <email>jsmith@example.com</email>
  </billTo>
  <purchaseTotals>
    <currency>USD</currency>
    <grandTotalAmount>5.00</grandTotalAmount>
  </purchaseTotals>
  <encryptedPayment>
    <descriptor>RklEPUNPTU1PTi5BUFBMRS5JTkFQUC5QQV1NRU5U</descriptor>
    <data>ABCDEFabcdefABCDEFabcdef0987654321234567</data>
    <encoding>Base64</encoding>
  </encryptedPayment>
  <card>
    <cardType>001</cardType>
  </card>
  <ccAuthService run="true"/>
  <paymentSolution>001</paymentSolution>
</requestMessage>
```
Example 23  Authorization Reply (JCB)

```
<c:replyMessage>
  <c:requestID>446584034076500001541</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:requestToken>
    Ahj/7wSR5C/4Icd2fdAKakGLadfg5535r/ghx3Z90AoBj3u
  </c:requestToken>
  <c:token>
    <c:expirationMonth>07</c:expirationMonth>
    <c:expirationYear>2025</c:expirationYear>
    <c:prefix>239845</c:prefix>
    <c:suffix>2947</c:suffix>
  </c:token>
  <c:purchaseTotals>
    <c:currency>USD</c:currency>
  </c:purchaseTotals>
  <c:ccAuthReply>
    <c:reasonCode>100</c:reasonCode>
    <c:amount>5.00</c:amount>
    <c:authorizationCode>888888</c:authorizationCode>
    <c:avsCode>X</c:avsCode>
    <c:avsCodeRaw>I1</c:avsCodeRaw>
    <c:processorResponse>100</c:processorResponse>
    <c:reconciliationID>11267051CGJSMQDC</c:reconciliationID>
  </c:ccAuthReply>
</c:replyMessage>
```

Additional CyberSource Services

For information on how to request these follow-on services, refer to Credit Card Services Using the Simple Order API.

Table 2  CyberSource Services

<table>
<thead>
<tr>
<th>CyberSource Service</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capture</td>
<td>A follow-on service that uses the request ID returned from the previous authorization. The request ID links the capture to the authorization. This service transfers funds from the customer’s account to your bank and usually takes two to four days to complete.</td>
</tr>
<tr>
<td>Sale</td>
<td>A sale is a bundled authorization and capture. Request the authorization and capture services at the same time. CyberSource processes the capture immediately.</td>
</tr>
<tr>
<td>Auth Reversal</td>
<td>A follow-on service that uses the request ID returned from the previous authorization. An auth reversal releases the hold that the authorization placed on the customer’s credit card funds. Use this service to reverse an unnecessary or undesired authorization.</td>
</tr>
</tbody>
</table>
Data Type Definitions

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

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date and time</td>
<td>Format is YYYY-MM-DDThh:mm:ssZ, where:</td>
</tr>
<tr>
<td></td>
<td>• T separates the date and the time</td>
</tr>
<tr>
<td></td>
<td>• Z indicates Coordinated Universal Time (UTC), also known as Greenwich Mean Time (GMT)</td>
</tr>
<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>

Numbered Elements

The CyberSource XML schema includes several numbered elements. You can include these complex elements more than once in a request. For example, when a customer order includes more than one item, you must include multiple `<item>` elements in your request. Each item is numbered, starting with 0. The XML schema uses an `id` attribute in the item's opening tag to indicate the number. For example:

```xml
=item id="0">
```

As a name-value pair field name, this tag is represented as `item_0`. In this portion of the field name, the underscore before the number does not indicate hierarchy in the XML schema. The item fields are generically referred to as `item_#_<element name>` in the documentation.
Below is an example of the numbered `<item>` element and the corresponding name-value pair field names. If you are using the Simple Object Access Protocol (SOAP), the client contains a corresponding `Item` class.

**Example 24** Numbered XML Schema Element Names and Name-Value Pair Field Names

<table>
<thead>
<tr>
<th>XML Schema Element Names</th>
<th>Corresponding Name-Value Pair Field Names</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;item id=&quot;0&quot;&gt;</code></td>
<td><code>item_0_unitPrice</code></td>
</tr>
<tr>
<td><code>&lt;unitPrice&gt;</code></td>
<td></td>
</tr>
<tr>
<td><code>&lt;quantity&gt;</code></td>
<td><code>item_0_quantity</code></td>
</tr>
<tr>
<td><code>&lt;/item&gt;</code></td>
<td></td>
</tr>
<tr>
<td><code>&lt;item id=&quot;1&quot;&gt;</code></td>
<td><code>item_1_unitPrice</code></td>
</tr>
<tr>
<td><code>&lt;unitPrice&gt;</code></td>
<td></td>
</tr>
<tr>
<td><code>&lt;quantity&gt;</code></td>
<td><code>item_1_quantity</code></td>
</tr>
<tr>
<td><code>&lt;/item&gt;</code></td>
<td></td>
</tr>
</tbody>
</table>

---

**Important** When a request is in XML format and includes an `<item>` element, the element must include an `id` attribute. For example: `<item id="0">`.

---

**Relaxed Requirements for Address Data and Expiration Date**

To enable relaxed requirements for address data and expiration date, contact CyberSource Customer Support to have your account configured for this feature. For details about relaxed requirements, see the Relaxed Requirements for Address Data and Expiration Date page.
Appendix A  API Fields

Request Fields

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

Table 4  Request Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Used By: Required (R) or Optional (O)</th>
<th>Data Type (Length)</th>
</tr>
</thead>
<tbody>
<tr>
<td>billTo_city</td>
<td>City of the billing address.</td>
<td>ccAuthService (R)²</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 (R)²</td>
<td>String (2)</td>
</tr>
<tr>
<td>billTo_email</td>
<td>Customer’s email address.</td>
<td>ccAuthService (R)²</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 (R)²</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 (R)²</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>ccAuthService (O)</td>
<td>String (15)</td>
</tr>
<tr>
<td>billTo_postalCode</td>
<td>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</td>
<td>ccAuthService (R)²</td>
<td>String (9)</td>
</tr>
</tbody>
</table>

1 The TC 33 Capture file contains information about the purchases and refunds that a merchant submits to CyberSource. CyberSource through VisaNet creates the TC 33 Capture file at the end of the day and sends it to the merchant’s acquirer, who uses this information to facilitate end-of-day clearing processing with payment card companies.

2 This field is optional if your CyberSource account is configured for relaxed requirements for address data and expiration date. See “Relaxed Requirements for Address Data and Expiration Date,” page 45. Important It is your responsibility to determine whether a field is required for the transaction you are requesting.
### Table 4 Request Fields (Continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Used By: Required (R) or Optional (O)</th>
<th>Data Type (Length)</th>
</tr>
</thead>
<tbody>
<tr>
<td>billTo_state</td>
<td>State or province of the billing address. For an address in the U.S. or Canada, use the <em>State, Province, and Territory Codes for the United States and Canada.</em></td>
<td>ccAuthService (R)&lt;sup&gt;2&lt;/sup&gt;</td>
<td>String (2)</td>
</tr>
<tr>
<td>billTo_street1</td>
<td>First line of the billing street address.</td>
<td>ccAuthService (R)&lt;sup&gt;2&lt;/sup&gt;</td>
<td>String (60)</td>
</tr>
<tr>
<td>billTo_street2</td>
<td>Additional address information.</td>
<td>ccAuthService (O)</td>
<td>String (60)</td>
</tr>
<tr>
<td></td>
<td><strong>Example</strong> Attention: Accounts Payable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>card_accountNumber</td>
<td>The payment network token value.</td>
<td>ccAuthService (R)</td>
<td>Nonnegative integer (20)</td>
</tr>
<tr>
<td>card_cardType</td>
<td>Type of card to authorize. Possible values:</td>
<td>ccAuthService (R)</td>
<td>String (3)</td>
</tr>
<tr>
<td></td>
<td>- 001: Visa</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- 002: Mastercard</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- 003: American Express</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- 004: Discover</td>
<td></td>
<td></td>
</tr>
<tr>
<td>card_cvNumber</td>
<td>CVN.</td>
<td>cc AuthService (R)</td>
<td>Nonnegative integer (4)</td>
</tr>
<tr>
<td>card_expirationMonth</td>
<td>Two-digit month in which the payment network token expires.</td>
<td>ccAuthService (R)</td>
<td>String (2)</td>
</tr>
<tr>
<td></td>
<td>Format: MM.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Possible values: 01 through 12.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>card_expirationYear</td>
<td>Four-digit year in which the payment network token expires.</td>
<td>ccAuthService (R)</td>
<td>Nonnegative integer (4)</td>
</tr>
<tr>
<td></td>
<td>Format: YYYY.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

1. The TC 33 Capture file contains information about the purchases and refunds that a merchant submits to CyberSource. CyberSource through VisaNet creates the TC 33 Capture file at the end of the day and sends it to the merchant’s acquirer, who uses this information to facilitate end-of-day clearing processing with payment card companies.

2. This field is optional if your CyberSource account is configured for relaxed requirements for address data and expiration date. See "Relaxed Requirements for Address Data and Expiration Date," page 45. **Important** It is your responsibility to determine whether a field is required for the transaction you are requesting.
### ccAuthService_cavv

**Visa**
Cryptogram for authorizations with payment network tokens. The value for this field must be 28-character Base64 or 40-character hex binary. All cryptograms use one of these formats.

**American Express**
For a 20-byte cryptogram, set this field to the cryptogram for authorizations with payment network tokens. For a 40-byte cryptogram, set this field to block A of the cryptogram for authorizations with payment network tokens. The value for this field must be 28-character Base64 or 40-character hex binary. All cryptograms use one of these formats.

**Discover**
Cryptogram for authorizations with payment network tokens. The value for this field can be a 20 or 40-character hex binary. All cryptograms use one of these formats.

**CyberSource through VisaNet**
The value for this field corresponds to the following data in the TC 33 capture file¹:
- Record: CP01 TCR8
- Position: 77-78
- Field: CAVV version and authentication action.

### ccAuthService_commerceIndicator
For an authorization with a payment network token. The values are required for the merchant decryption method (see "Option 1: Merchant Decryption," page 21).

Possible values:
- aesk: American Express card type
- spa: Mastercard card type
- vbv: Visa card type mapped for Apple Pay transactions with e-commerce indicator of 5
- internet: Visa card type mapped for Apple Pay transactions with e-commerce indicator of 7
- dipb: Discover card type

---

¹ The TC 33 Capture file contains information about the purchases and refunds that a merchant submits to CyberSource. CyberSource through VisaNet creates the TC 33 Capture file at the end of the day and sends it to the merchant’s acquirer, who uses this information to facilitate end-of-day clearing processing with payment card companies.

² This field is optional if your CyberSource account is configured for relaxed requirements for address data and expiration date. See "Relaxed Requirements for Address Data and Expiration Date," page 45. **Important** It is your responsibility to determine whether a field is required for the transaction you are requesting.
### Table 4  Request Fields  (Continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Used By: Required (R) or Optional (O)</th>
<th>Data Type (Length)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ccAuthService_directoryServerTrans ID</td>
<td>Identifier generated during the authentication transaction by the Mastercard Directory Server and passed back with the authentication results.</td>
<td>ccAuthService (O)</td>
<td>String (36)</td>
</tr>
<tr>
<td>ccAuthService_eciRaw</td>
<td>Raw electronic commerce indicator (ECI).</td>
<td>ccAuthService (O)</td>
<td>String (2)</td>
</tr>
<tr>
<td>ccAuthService_networkTokenCryptogram</td>
<td>Token authentication verification value cryptogram. For token-based transactions with 3D Secure or SecureCode, you must submit both types of cryptograms: network token and 3D Secure/SecureCode. The value for this field must be 28-character Base64 or 40-character hex binary. All cryptograms use one of these formats.</td>
<td>ccAuthService (O)</td>
<td>String (40)</td>
</tr>
<tr>
<td>ccAuthService_paSpecificationVersion</td>
<td>The 3D Secure version that you used for Secured Consumer Authentication (SCA); for example, 3D Secure 1.0.2 or 2.0.0.</td>
<td>ccAuthService (O)</td>
<td>String (20)</td>
</tr>
</tbody>
</table>
| ccAuthService_run | Whether to include ccAuthService in your request. Possible values:  

- TRUE: Include the service in your request.
- FALSE (default): Do not include the service in your request. | ccAuthService (R) |
| ccAuthService_xid | Visa  

Cryptogram for authorizations with payment network tokens. The value for this field must be 28-character Base64 or 40-character hex binary. All cryptograms use one of these formats.  

American Express  

For a 20-byte cryptogram, set this field to the cryptogram for authorizations with payment network tokens. For a 40-byte cryptogram, set this field to block A of the cryptogram for authorizations with payment network tokens (see "American Express Transaction," page 25). The value for this field must be 28-character Base64 or 40-character hex binary. All cryptograms use one of these formats. | ccAuthService (R) | String (40) |

---

1. The TC 33 Capture file contains information about the purchases and refunds that a merchant submits to CyberSource. CyberSource through VisaNet creates the TC 33 Capture file at the end of the day and sends it to the merchant’s acquirer, who uses this information to facilitate end-of-day clearing processing with payment card companies.

2. This field is optional if your CyberSource account is configured for relaxed requirements for address data and expiration date. See "Relaxed Requirements for Address Data and Expiration Date," page 45. Important It is your responsibility to determine whether a field is required for the transaction you are requesting.
## Table 4 Request Fields (Continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Used By: Required (R) or Optional (O)</th>
<th>Data Type (Length)</th>
</tr>
</thead>
<tbody>
<tr>
<td>encryptedPayment_data</td>
<td>The encrypted payment data value. Populate this field with the encrypted payment data value obtained from the paymentData property of the PKPaymentToken object. See the PassKit Framework Reference.</td>
<td>ics_auth (R)</td>
<td></td>
</tr>
<tr>
<td>encryptedPayment_descriptor</td>
<td>Format of the encrypted payment data. The value for Apple Pay is: Rk1EPUNPTU1PTi5BUBFBMRS5JTkFQUC5QQV1NRU5U</td>
<td>ics_auth (R)</td>
<td>String (128)</td>
</tr>
<tr>
<td>encryptedPayment_encoding</td>
<td>Encoding method used to encrypt the payment data: Base64</td>
<td>ics_auth (R)</td>
<td>String (6)</td>
</tr>
<tr>
<td>item_#_productCode</td>
<td>Type of product. This value is used to determine the product category: electronic, handling, physical, service, or shipping. The default is default. See &quot;Numbered Elements,&quot; page 44.</td>
<td>ccAuthService (O)</td>
<td>String (255)</td>
</tr>
<tr>
<td>item_#_productName</td>
<td>Name of the product. This field is required when the item_#_productCode value is not default or one of the values related to shipping and/or handling. See &quot;Numbered Elements,&quot; page 44.</td>
<td>ccAuthService (See description)</td>
<td>String (255)</td>
</tr>
<tr>
<td>item_#_productSKU</td>
<td>Identification code for the product. This field is required when the item_#_productCode value is not default or one of the values related to shipping and/or handling. See &quot;Numbered Elements,&quot; page 44.</td>
<td>ccAuthService (See description)</td>
<td>String (255)</td>
</tr>
<tr>
<td>item_#_quantity</td>
<td>The default is 1. This field is required when the item_#_productCode value is not default or one of the values related to shipping and/or handling. See &quot;Numbered Elements,&quot; page 44.</td>
<td>ccAuthService (See description)</td>
<td>Integer (10)</td>
</tr>
<tr>
<td>item_#_taxAmount</td>
<td>Total tax to apply to the product. This value cannot be negative. See &quot;Numbered Elements,&quot; page 44.</td>
<td>ccAuthService (See description)</td>
<td>String (15)</td>
</tr>
</tbody>
</table>

1 The TC 33 Capture file contains information about the purchases and refunds that a merchant submits to CyberSource. CyberSource through VisaNet creates the TC 33 Capture file at the end of the day and sends it to the merchant’s acquirer, who uses this information to facilitate end-of-day clearing processing with payment card companies.

2 This field is optional if your CyberSource account is configured for relaxed requirements for address data and expiration date. See "Relaxed Requirements for Address Data and Expiration Date," page 45. Important It is your responsibility to determine whether a field is required for the transaction you are requesting.
Table 4  Request Fields  (Continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Used By: Required (R) or Optional (O)</th>
<th>Data Type (Length)</th>
</tr>
</thead>
<tbody>
<tr>
<td>item_#_unitPrice</td>
<td>Per-item price of the product. This value cannot be negative. You can include a decimal point (.), but you cannot include any other special characters.</td>
<td>ccAuthService (See description)</td>
<td>String (15)</td>
</tr>
<tr>
<td></td>
<td>See &quot;Numbered Elements,&quot; page 44.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>merchantID</td>
<td>Your CyberSource merchant ID. Use the same merchant ID for evaluation, testing, and production.</td>
<td>ccAuthService (R)</td>
<td>String (30)</td>
</tr>
<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 Getting Started with CyberSource Advanced for the Simple Order API.</td>
<td>ccAuthService (R)</td>
<td>String (50)</td>
</tr>
<tr>
<td>paymentNetworkToken_</td>
<td>Confidence level of the tokenization. This value is assigned by the token service provider.</td>
<td>ccAuthService (O)</td>
<td>String (2)</td>
</tr>
<tr>
<td>assuranceLevel</td>
<td><strong>Note</strong> This field is supported only for CyberSource through VisaNet and FDC Nashville Global.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>paymentNetworkToken_</td>
<td>Type of technology used in the device to store token data. Possible value:</td>
<td>ccAuthService (O)</td>
<td>Integer (3)</td>
</tr>
<tr>
<td>deviceTechType</td>
<td>001: Secure Element (SE)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Smart card or memory with restricted access and encryption to prevent data tampering. For storing payment credentials, a SE is tested against a set of requirements defined by the payment networks.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> This field is supported only for FDC Compass.</td>
<td></td>
<td></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>
<tr>
<td>requesterID</td>
<td><strong>Note</strong> This field is supported only for CyberSource through VisaNet, FDC Nashville Global, and Chase Paymentech Solutions.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1 The TC 33 Capture file contains information about the purchases and refunds that a merchant submits to CyberSource. CyberSource through VisaNet creates the TC 33 Capture file at the end of the day and sends it to the merchant’s acquirer, who uses this information to facilitate end-of-day clearing processing with payment card companies.
2 This field is optional if your CyberSource account is configured for relaxed requirements for address data and expiration date. See "Relaxed Requirements for Address Data and Expiration Date," page 45. **Important** It is your responsibility to determine whether a field is required for the transaction you are requesting.
Table 4  Request Fields  (Continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Used By: Required (R) or Optional (O)</th>
<th>Data Type (Length)</th>
</tr>
</thead>
<tbody>
<tr>
<td>paymentNetworkToken_transactionType</td>
<td>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.</td>
<td>ccAuthService (R)</td>
<td>String (1)</td>
</tr>
<tr>
<td></td>
<td>Set the value for this field to 1.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>paymentSolution</td>
<td>Identifies Apple Pay as the payment solution that is being used for the transaction:</td>
<td>ccAuthService (R)</td>
<td>String (3)</td>
</tr>
<tr>
<td></td>
<td>Set the value for this field to 001.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>  This unique ID differentiates digital solution transactions within the CyberSource platform for reporting purposes.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>purchaseTotals_currency</td>
<td>Currency used for the order. For the possible values, see the <a href="#">ISO Standard Currency Codes</a>.</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.</td>
<td>ccAuthService (R)</td>
<td>Decimal (60)</td>
</tr>
<tr>
<td>ucaf_authenticationData</td>
<td>Cryptogram for authorizations with payment network tokens with Mastercard.</td>
<td>ccAuthService (R)</td>
<td>String (32)</td>
</tr>
<tr>
<td>ucaf_collectionIndicator</td>
<td>Required field for authorizations with payment network tokens with Mastercard.</td>
<td>ccAuthService (R)</td>
<td>String with numbers only (1)</td>
</tr>
</tbody>
</table>

1. The TC 33 Capture file contains information about the purchases and refunds that a merchant submits to CyberSource. CyberSource through VisaNet creates the TC 33 Capture file at the end of the day and sends it to the merchant's acquirer, who uses this information to facilitate end-of-day clearing processing with payment card companies.

2. This field is optional if your CyberSource account is configured for relaxed requirements for address data and expiration date. See "Relaxed Requirements for Address Data and Expiration Date," page 45. **Important**  It is your responsibility to determine whether a field is required for the transaction you are requesting.
Appendix A  API Fields

Reply Fields

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 reply flag that it does not recognize.

Your payment processor can include additional API reply fields that are not documented in this guide. See Credit Card Services Using the Simple Order API for detailed descriptions of additional API reply fields.

Table 5  Reply Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Returned By</th>
<th>Data Type &amp; Length</th>
</tr>
</thead>
</table>
| card_suffix            | Last four digits of the cardholder’s account number. This field is returned only for tokenized transactions. You can use this value on the receipt that you give to the cardholder.  
**Note** This field is returned only for CyberSource through VisaNet and FDC Nashville Global.  
**CyberSource through VisaNet**  
The value for this field corresponds to the following data in the TC 33 capture file¹:
- Record: CP01 TCRB
- Position: 85
- Field: American Express last 4 PAN return indicator.  | ccAuthReply | String (4) |
| ccAuthReply_amount     | Amount that was authorized.                                                 | ccAuthReply   | String (15)        |
| ccAuthReply_authorizationCode | Authorization code. Returned only when the processor returns this value. | ccAuthReply   | String (7)         |
| ccAuthReply_authorizedDateTime | Time of authorization. Format: YYYY-MM-DDThh:mm:ssZ

¹ The TC 33 Capture file contains information about the purchases and refunds that a merchant submits to CyberSource.  
CyberSource through VisaNet creates the TC 33 Capture file at the end of the day and sends it to the merchant’s acquirer, who uses this information to facilitate end-of-day clearing processing with payment card companies.
### Table 5  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_avsCode</td>
<td>AVS results. See <a href="#">Credit Card Services Using the Simple Order API</a> 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_cvCode</td>
<td>CVN result code. See <a href="#">Credit Card Services Using the Simple Order API</a> for a detailed list of CVN codes.</td>
<td>ccAuthReply</td>
<td>String (1)</td>
</tr>
<tr>
<td>ccAuthReply_cvCodeRaw</td>
<td>CVN result code sent directly from the processor. Returned only when the processor returns this value.</td>
<td>ccAuthReply</td>
<td>String (10)</td>
</tr>
</tbody>
</table>
| ccAuthReply_paymentCardService | Mastercard service that was used for the transaction. Mastercard provides this value to CyberSource. Possible value: 53: Mastercard card-on-file token service  
**Note** This field is returned only for CyberSource through VisaNet. | ccAuthReply   | String (2)         |
| ccAuthReply_paymentCardServiceResult | Result of the Mastercard card-on-file token service. Mastercard provides this value to CyberSource. Possible values:  
- C: Service completed successfully.  
- F: One of the following:  
  - Incorrect Mastercard POS entry mode. The Mastercard POS entry mode should be 81 for an authorization or authorization reversal.  
  - Incorrect Mastercard POS entry mode. The Mastercard POS entry mode should be 01 for a tokenized request.  
  - Token requestor ID is missing or formatted incorrectly.  
- I: One of the following:  
  - Invalid token requestor ID.  
  - Suspended or deactivated token.  
  - Invalid token (not in mapping table).  
- T: Invalid combination of token requestor ID and token.  
- U: Expired token.  
- W: Primary account number (PAN) listed in electronic warning bulletin.  
**Note** This field is returned only for CyberSource through VisaNet. | ccAuthReply   | String (1)         |

1 The TC 33 Capture file contains information about the purchases and refunds that a merchant submits to CyberSource. CyberSource through VisaNet creates the TC 33 Capture file at the end of the day and sends it to the merchant’s acquirer, who uses this information to facilitate end-of-day clearing processing with payment card companies.
### Table 5  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 Credit Card Services Using the Simple Order API 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. This value is not returned for all processors.</td>
<td>ccAuthReply</td>
<td>String (60)</td>
</tr>
</tbody>
</table>
| ccAuthReply_transactionQualification | Type of authentication for which the transaction qualifies as determined by the Mastercard authentication service, which confirms the identity of the cardholder. Mastercard provides this value to CyberSource. Possible values:  
  - 1: Transaction qualifies for Mastercard authentication type 1.  
  
  **Note** This field is returned only for CyberSource through VisaNet. | ccAuthReply   | String (1)          |
| ccAuthReversalReply_paymentCardService | Mastercard service that was used for the transaction. Mastercard provides this value to CyberSource. Possible value:  
  - 53: Mastercard card-on-file token service  
  
  **Note** This field is returned only for CyberSource through VisaNet. | ccAuthReversalReply | String (2)         |

---

1 The TC 33 Capture file contains information about the purchases and refunds that a merchant submits to CyberSource. CyberSource through VisaNet creates the TC 33 Capture file at the end of the day and sends it to the merchant’s acquirer, who uses this information to facilitate end-of-day clearing processing with payment card companies.
### Table 5  Reply Fields  (Continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Returned By</th>
<th>Data Type &amp; Length</th>
</tr>
</thead>
</table>
| ccAuthReversalReply_paymentCardServiceResult | Result of the Mastercard card-on-file token service. Mastercard provides this value to CyberSource. Possible values:  
- **C**: Service completed successfully.  
- **F**: One of the following:  
  - Incorrect Mastercard POS entry mode. The Mastercard POS entry mode should be 81 for an authorization or authorization reversal.  
  - Incorrect Mastercard POS entry mode. The Mastercard POS entry mode should be 01 for a tokenized request.  
  - Token requestor ID is missing or formatted incorrectly.  
- **I**: One of the following:  
  - Invalid token requestor ID.  
  - Suspended or deactivated token.  
  - Invalid token (not in mapping table).  
- **T**: Invalid combination of token requestor ID and token.  
- **U**: Expired token.  
- **W**: Primary account number (PAN) listed in electronic warning bulletin.  
  **Note**: This field is returned only for CyberSource through VisaNet. | ccAuthReversalReply | String (1) |
| decision | Summarizes the result of the overall request. Possible values:  
- **ACCEPT**  
- **ERROR**  
- **REJECT**  
- **REVIEW**: Returned only when you use CyberSource Decision Manager. | ccAuthReply | String (6) |
| invalidField_0 through invalidField_N | Fields in the request that contained invalid data. For information about missing or invalid fields, see *Getting Started with CyberSource Advanced for the Simple Order API*. | ccAuthReply | String (100) |

---

1 The TC 33 Capture file contains information about the purchases and refunds that a merchant submits to CyberSource. CyberSource through VisaNet creates the TC 33 Capture file at the end of the day and sends it to the merchant’s acquirer, who uses this information to facilitate end-of-day clearing processing with payment card companies.
### Table 5  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>merchantReferenceCode</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 through missingField_N</td>
<td>Required fields that were missing from the request. For information about missing or invalid fields, see <em>Getting Started with CyberSource Advanced for the Simple Order API</em>.</td>
<td>ccAuthReply</td>
<td>String (100)</td>
</tr>
<tr>
<td>paymentNetworkToken_</td>
<td>Possible values:</td>
<td>ccAuthReply</td>
<td>String (1)</td>
</tr>
<tr>
<td>accountStatus</td>
<td>N: Nonregulated</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>R: Regulated</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Note</td>
<td>This field is returned only for CyberSource through VisaNet.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>paymentNetworkToken_</td>
<td>Confidence level of the tokenization. This value is assigned by the token service provider.</td>
<td>ccAuthReply</td>
<td>String (2)</td>
</tr>
<tr>
<td>assuranceLevel</td>
<td>Note This field is returned only for CyberSource through VisaNet and FDC Nashville Global.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>paymentNetworkToken_</td>
<td>Mastercard product ID associated with the primary account number (PAN). For the possible values, see “Mastercard Product IDs” in <em>Credit Card Services Using the Simple Order API</em>.</td>
<td>ccAuthReply</td>
<td>String (3)</td>
</tr>
<tr>
<td>originalCardCategory</td>
<td>CyberSource through VisaNet For the possible values, see “Mastercard Product IDs” in <em>Credit Card Services for CyberSource through VisaNet Using the Simple Order API</em>.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Note</td>
<td>This field is returned only for Mastercard transactions on CyberSource through VisaNet.</td>
<td></td>
<td></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. This value is returned only if the processor provides it.</td>
<td>ccAuthService</td>
<td>String (11)</td>
</tr>
<tr>
<td>requesterID</td>
<td>Note This field is supported only for CyberSource through VisaNet and FDC Nashville Global.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>purchaseTotals_currency</td>
<td>Currency used for the order. For the possible values, see the <em>ISO Standard Currency Codes</em>.</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 <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>
</tbody>
</table>

---

1  The TC 33 Capture file contains information about the purchases and refunds that a merchant submits to CyberSource. CyberSource through VisaNet creates the TC 33 Capture file at the end of the day and sends it to the merchant’s acquirer, who uses this information to facilitate end-of-day clearing processing with payment card companies.
Table 5  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>requestID</td>
<td>Identifier for the request.</td>
<td>ccAuthReply</td>
<td>String (26)</td>
</tr>
<tr>
<td>requestToken</td>
<td>Request token data created by CyberSource for each reply. The field is an encoded string that contains no confidential information such as an account or card verification number. The string can contain a maximum of 256 characters.</td>
<td>ccAuthReply</td>
<td>String (256)</td>
</tr>
<tr>
<td>token_expirationMonth</td>
<td>Month in which the token expires. CyberSource includes this field in the reply message when it decrypts the payment blob for the tokenized transaction. Format: MM. Possible values: 01 through 12.</td>
<td>ccAuthReply</td>
<td>String (2)</td>
</tr>
<tr>
<td>token_expirationYear</td>
<td>Year in which the token expires. CyberSource includes this field in the reply message when it decrypts the payment blob for the tokenized transaction. Format: YYYY.</td>
<td>ccAuthReply</td>
<td>String (4)</td>
</tr>
<tr>
<td>token_prefix</td>
<td>First 6 digits of token. CyberSource includes this field in the reply message when it decrypts the payment blob for the tokenized transaction.</td>
<td>ccAuthReply</td>
<td>String (6)</td>
</tr>
<tr>
<td>token_suffix</td>
<td>Last 4 digits of token. CyberSource includes this field in the reply message when it decrypts the payment blob for the tokenized transaction.</td>
<td>ccAuthReply</td>
<td>String (4)</td>
</tr>
</tbody>
</table>

1  The TC 33 Capture file contains information about the purchases and refunds that a merchant submits to CyberSource. CyberSource through VisaNet creates the TC 33 Capture file at the end of the day and sends it to the merchant’s acquirer, who uses this information to facilitate end-of-day clearing processing with payment card companies.