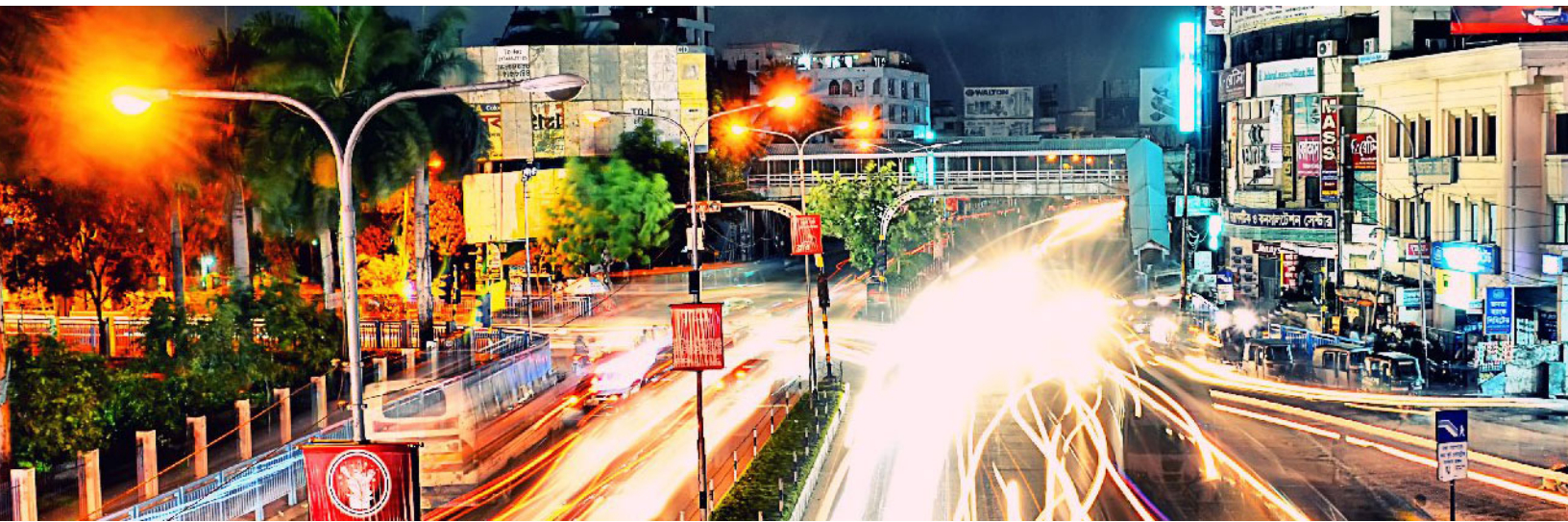


Apple Pay

Using the Simple Order API



CyberSource[®]
A Visa Solution

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

© 2020. 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. Visa, Visa International, CyberSource, the Visa logo, and the CyberSource logo are the registered trademarks of Visa International in the United States and other countries. All other trademarks, service marks, registered marks, or registered service marks are the property of their respective owners.

Revision: May 2020

Contents

Recent Revisions to This Document 5

About This Guide 6

Audience and Purpose 6

Conventions 6

 Important Statements 6

 Text and Command Conventions 6

Related Documents 7

Customer Support 7

Chapter 1 **Getting Started** 8

Requirements 8

Supported Processors, Card Types, and Optional Features 9

Enrolling in Apple Pay 11

 Generating a New CSR 12

Transaction Reports 12

Chapter 2 **Apple Pay Integrations** 13

In-App Transactions Using the CyberSource API 13

 Merchant Decryption 13

 CyberSource Decryption 14

Web Transactions 15

 Merchant Decryption 15

 CyberSource Decryption 15

 Requirements 16

 Apple Pay JavaScript 17

 Apple Pay Button 17

 ApplePaySession Class 17

 Create ApplePaySession Object 18

 Merchant Validation 18

 Payment Confirmation 18

Merchant Decryption	18
CyberSource Decryption	19

Chapter 3 Requesting the Authorization Service 20

Option 1: Merchant Decryption	20
Visa Transaction	20
Mastercard Transaction	22
American Express Transaction	24
Discover Transaction	26
JCB Transaction	28
Option 2: CyberSource Decryption	31
Visa Transaction	31
Mastercard Transaction	34
American Express Transaction	36
Discover Transaction	38
JCB Transaction	39
Additional CyberSource Services	42

Appendix A API Fields 43

Data Type Definitions	43
Numbered Elements	43
Relaxed Requirements for Address Data and Expiration Date	44
Request Fields	45
Reply Fields	52

Recent Revisions to This Document

Release	Changes
May 2020	Updated information about recurring payments. See "Supported Processors, Card Types, and Optional Features," page 9 .
February 2020	Added support for the processor <i>Moneris</i> . See "Supported Processors, Card Types, and Optional Features," page 9 .
January 2020	OmniPay Direct: added support for acquirer Cardnet International. See "Supported Processors, Card Types, and Optional Features," page 9 . Updated the purchaseTotals_grandTotalAmount field length. See purchaseTotals_grandTotalAmount, page 51 .
November 2019	Changed <i>payment network tokenization</i> to <i>authorizations with payment network tokens</i> throughout. SIX: added an important note. See SIX, page 10 .
October 2019	Updated Related Documents. See "Related Documents," page 7 . Updated Business Center procedures. See "Enrolling in Apple Pay," page 11 , and "Generating a New CSR," page 12 . Updated information about reports. See "Transaction Reports," page 12 .
September 2019	This revision contains only editorial changes and no technical updates.

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

Important Statements



An *Important* statement contains information essential to successfully completing a task or learning a concept.

Text and Command Conventions

Convention	Usage
Bold	<ul style="list-style-type: none">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	<ul style="list-style-type: none">XML elements.Code examples and samples.Text that you enter in an API environment; for example: Set the ccAuthService_run field to <code>true</code>.

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>

Getting Started

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 9).
- You must have an *Admin* or *Team Agent* user of the [Apple Pay Developer account](#).



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

Merchant-initiated transactions, multiple partial captures, and subsequent authorizations are described in *Authorizations with Payment Network Tokens Using the Simple Order API* ([PDF](#) | [HTML](#)). Recurring payments and split shipments are described in *Credit Card Services Using the Simple Order API* ([PDF](#) | [HTML](#)).

Table 1 Processors, Card Types, and Optional Features

Processor	Card Types	Optional Features
American Express Direct	American Express	<ul style="list-style-type: none"> ■ Multiple partial captures ■ Recurring payments
Barclays	Visa, Mastercard, Maestro (International), Maestro (UK Domestic)	<ul style="list-style-type: none"> ■ Multiple partial captures ■ Recurring payments
Chase Paymentech Solutions	Visa, Mastercard, American Express, Discover, Maestro (International)	<ul style="list-style-type: none"> ■ Merchant-Initiated transactions ■ Multiple partial captures ■ Recurring payments
Credit Mutuel-CIC	Visa, Mastercard, Cartes Bancaires	Recurring payments
CyberSource through VisaNet. The supported acquirers are: <ul style="list-style-type: none"> ■ Australia and New Zealand Banking Group Ltd. (ANZ) ■ CitiBank Singapore Ltd. ■ Global Payments Asia Pacific ■ Vantiv ■ Westpac 	Visa, Mastercard	<ul style="list-style-type: none"> ■ Merchant-Initiated transactions ■ Recurring payments ■ Split shipments
Elavon Americas	Visa, Mastercard, American Express, JCB, Discover	<ul style="list-style-type: none"> ■ Merchant-Initiated transactions ■ Multiple partial captures ■ Recurring payments
FDC Compass	Visa, Mastercard, American Express	<ul style="list-style-type: none"> ■ Multiple partial captures ■ Recurring payments
FDC Nashville Global	Visa, Mastercard, American Express, Discover	<ul style="list-style-type: none"> ■ Multiple partial captures ■ Recurring payments ■ Subsequent authorizations
GPN	Visa, Mastercard, American Express	Split shipments
JCN Gateway	JCB	<ul style="list-style-type: none"> ■ Multiple partial captures ■ Subsequent authorizations

Table 1 Processors, Card Types, and Optional Features (Continued)

Processor	Card Types	Optional Features
Moneris	Visa, Mastercard, American Express	<ul style="list-style-type: none"> ■ Merchant-initiated transactions ■ Recurring payments
OmniPay Direct. The supported acquirers are: <ul style="list-style-type: none"> ■ Bank of America Merchant Services ■ Cardnet International ■ First Data Merchant Solutions (Europe) ■ Global Payments International Acquiring 	Visa, Mastercard, Maestro (UK Domestic), Maestro (International)	<ul style="list-style-type: none"> ■ Multiple partial captures ■ Recurring payments
SIX Important SIX is supported only for card-present processing.	Visa, Mastercard, Maestro (UK Domestic), Maestro (International)	Recurring payments
Streamline	Visa, Mastercard, Maestro (UK Domestic), Maestro (International)	<ul style="list-style-type: none"> ■ Multiple partial captures ■ Recurring payments ■ Subsequent authorizations
TSYS Acquiring Solutions	Visa, Mastercard, American Express	Multiple partial captures
Worldpay VAP Worldpay VAP was previously called Litle. Litle was purchased by Vantiv, which was then purchased by Worldpay VAP. If you have any questions about this situation, contact your account manager at Worldpay VAP.	Visa, Mastercard	Recurring payments

Enrolling in Apple Pay

To enroll in Apple Pay:

- Step 1** Log in to the Business Center:
- Test: <https://ebctest.cybersource.com/ebc2/>
 - Live: <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:
<https://developer.apple.com/documentation/passkit>
- Step 11** Test your software. See "[Requesting the Authorization Service](#)," page 20.



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



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: <https://ebctest.cybersource.com/ebc2/>
 - Live: <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

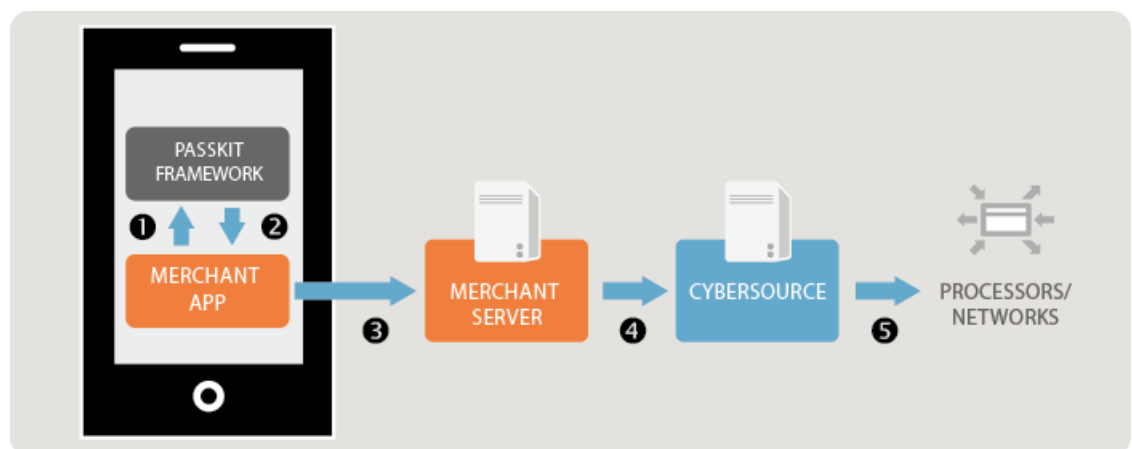
Go to the Business Center and use the Transaction Request Report to obtain information about your transactions:

- 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](#))

Apple Pay Integrations

In-App Transactions Using the CyberSource API

Merchant Decryption



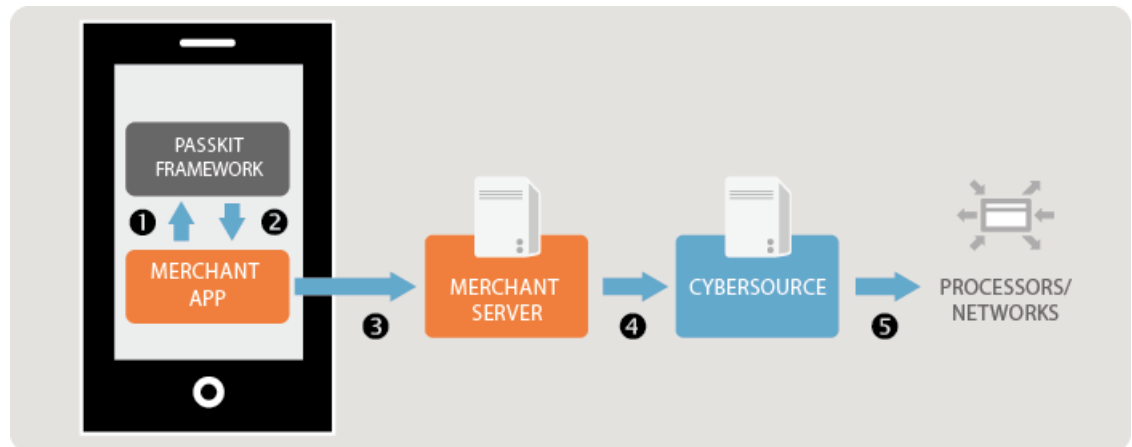
- 1 When the customer chooses to pay with Apple Pay, you use the Apple PassKit Framework to request the encrypted payment data from Apple.
- 2 Apple uses the Secure Element to create a payment token (the **PKPaymentToken** structure) and encrypt the token's payment data (the **paymentData** field of the **PKPaymentToken** structure) before it sends it your application.
- 3 You forward the encrypted payment data to your order management 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 20.

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



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



- When the customer chooses to pay with Apple Pay, you use the Apple PassKit Framework to request the encrypted payment data from Apple.
- 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.
- You forward the encrypted payment data to your order management system.
- 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 31.
- CyberSource decrypts the payment data and forwards the information to the payment network, including your processor and the relevant payment card company.



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 using the **onpaymentauthorized** callback function.
- 3 You forward the encrypted payment data to your order management 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 2: CyberSource Decryption](#)," page 31.
- 5 CyberSource forwards the information to the payment network, including your processor and the relevant payment card company.



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 order management 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 31.

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



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:

<https://developer.apple.com/support/apple-pay-domain-verification/>

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 20](#)), create a CSR.

If you are using the CyberSource decryption method (see "[Option 2: CyberSource Decryption](#)," [page 31](#)), upload the CSR that you created in the Business Center (see "[Enrolling in Apple Pay](#)," [page 11](#)).

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.
-

All optional features are described in *Authorizations with Payment Network Tokens Using the Simple Order API* ([PDF](#) | [HTML](#)).

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, you must ensure that it invokes the Apple Pay payment sheet.

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

<https://developer.apple.com/apple-pay/Apple-Pay-Identity-Guidelines.pdf>

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 17) or creating an Apple Pay session (see "[Create ApplePaySession Object](#)," page 18), 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.

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 order management 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 20.

CyberSource Decryption

Forward the encrypted payment data to your order management 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 31.

Example 1 JavaScript for Obtaining a Base64-Encoded Value

```
session.onpaymentauthorized = function (event) {  
  
  var paymentDataString = JSON.stringify(event.payment.token.paymentData);  
  
  var paymentDataBase64 = btoa(paymentDataString);  
  
  ...  
  
}
```

Requesting the Authorization Service

Option 1: Merchant Decryption

Visa Transaction

See the [Relaxed Requirements for Address Data and Expiration Date](#) page and "Request Fields," [page 45](#), for details and field descriptions.

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=`vbv` and 7=`internet`).
- Step 7** Set the `paymentSolution` field to 001.

Example 2 Authorization Request (Visa)

```

<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)

```

<c:replyMessage>
  <c:merchantReferenceCode>demorefnum</c:merchantReferenceCode>
  <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: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

See the [Relaxed Requirements for Address Data and Expiration Date](#) page and "Request Fields," page 45, for details and field descriptions.

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.

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>ABCDEFabcdefABCDEFscdef0987654321234567</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)

```

<c:replyMessage>
  <c:merchantReferenceCode>demorefnum</c:merchantReferenceCode>
  <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: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

See the [Relaxed Requirements for Address Data and Expiration Date](#) page and "Request Fields," page 45, for details and field descriptions.

To request an authorization for an American Express 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.



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)

```

<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)

```

<c:replyMessage>
  <c:merchantReferenceCode>demorefnum</c:merchantReferenceCode>
  <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: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>

```

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 45, for details and field descriptions.

- 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 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>

```

Example 9 Authorization Response (Discover)

```

<c:replyMessage>
  <c:merchantReferenceCode>demorefnum</c:merchantReferenceCode>
  <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: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:

See the [Relaxed Requirements for Address Data and Expiration Date](#) page and "Request Fields," page 45, for details and field descriptions.

- Step 1** Set the **card_accountNumber** field to the payment network token value.
- Step 2** Set the **card_expiration_Month** 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 **paymentNetworkToken_transactionType** field to 1.
- Step 5** Set the **ccAuthService_eciRaw** field to the ECI value contained in the Apple Pay response payload.
- Step 6** Set the **PaymentSolution** field to 001.

Example 10 Authorization Request (JCB)

```

<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>3566111111111111xxxx</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:merchantReferenceCode>demorefnum</c:merchantReferenceCode>
  <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: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
purchastTotals_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=4465840340765000001541
decision=accept
reasonCode=100
requestToken=Ahj/7wSR5C/4Icd2fdAKakGLadfg5535r/ghx3Z90AoBj3u
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 45](#), 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 14](#).
- Step 2** Set the **encryptedPayment_descriptor** field to:
Rk1EPUNPTU1PTi5BUFBMRS5JTkFQUC5QQV1NRU5U
- Step 3** Set the **paymentSolution** field to 001.

Example 14 Authorization Request (Visa)

```
<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 15 Authorization Response (Visa)

```

<c:replyMessage>
  <c:merchantReferenceCode>demorefnum</c:merchantReferenceCode>
  <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: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:

See the [Relaxed Requirements for Address Data and Expiration Date](#) page and ["Request Fields,"](#) page 45, 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 14.
- 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)

```

<c:replyMessage>
  <c:merchantReferenceCode>demorefnum</c:merchantReferenceCode>
  <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:

See the [Relaxed Requirements for Address Data and Expiration Date](#) page and "Request Fields," page 45, 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 14.
- Step 2** Set the `encryptedPayment_descriptor` field to:
RklEPUNPTU1PTi5BUFBMRS5JTkFQUC5QQVlNRU5U
- 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>RklEPUNPTU1PTi5BUFBMRS5JTkFQUC5QQVlNRU5U</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)

```

<c:replyMessage>
  <c:merchantReferenceCode>demorefnum</c:merchantReferenceCode>
  <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>

```

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 45, 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 14.
- Step 2** Set the **encryptedPayment_descriptor** field to:
Rk1EPUNPTU1PTi5BUFBMRS5JTkFQUC5QQV1NRU5U
- Step 3** Set the **paymentSolution** field to 001.

Example 20 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>
  <encryptedPayment>
    <descriptor>Rk1EPUNPTU1PTi5BUFBMRS5JTkFQUC5QQV1NRU5U</descriptor>
    <data>ABCDEFabcdefABCDEFabcdef0987654321234567</data>
    <encoding>Base64</encoding>
  </encryptedPayment>
  <card>
    <cardType>004</cardType>
  </card>
  <paymentNetworkToken>
    <transactionType>1</transactionType>
  </paymentNetworkToken>
  <paymentSolution>001</paymentSolution>
  <ccAuthService run="true"/>
</requestMessage>
```

Example 21 Authorization Response (Discover)

```

<c:replyMessage>
  <c:merchantReferenceCode>demorefnum</c:merchantReferenceCode>
  <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>

```

JCB Transaction

To request an authorization for a JCB transaction:

See the [Relaxed Requirements for Address Data and Expiration Date](#) page and "Request Fields," page 45, 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.
- Step 2** Set the `encryptedPaymentdescriptor` field to `Rk1EPUNPTU1PTi5BUFBMRS5JTkFQUc5QQV1NRU5U`.
- Step 3** Set the `paymentSolution` field to `001`.

Example 22 Authorization Request (JCB)

```
<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:merchantReferenceCode>demorefnum</c:merchantReferenceCode>
  <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: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

CyberSource Service	Description
Capture	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.
Sale	A sale is a bundled authorization and capture. Request the authorization and capture services at the same time. CyberSource processes the capture immediately.
Auth Reversal	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.

API Fields

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 3 Data Type Definitions

Data Type	Description
Date and time	Format is YYYY-MM-DDThh:mm:ssZ, where: <ul style="list-style-type: none">■ T separates the date and the time■ Z indicates Coordinated Universal Time (UTC), also known as Greenwich Mean Time (GMT) <p>Example 2019-08-11T22:47:57Z equals August 11, 2019, at 22:47:57 (10:47:57 p.m.).</p>
Integer	Whole number {..., -3, -2, -1, 0, 1, 2, 3, ...}
String	Sequence of letters, numbers, spaces, and special characters

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:

```
<item id="0">
```

As a name-value pair field name, this tag is called `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**

XML Schema Element Names	Corresponding Name-Value Pair Field Names
<pre><item id="0"> <unitPrice> <quantity> </item></pre>	<pre>item_0_unitPrice item_0_quantity</pre>
<pre><item id="1"> <unitPrice> <quantity> </item></pre>	<pre>item_1_unitPrice item_1_quantity</pre>



When a request in XML format 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](#).

Request Fields

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

Table 4 Request Fields

Field	Description	Used By: Required (R) or Optional (O)	Data Type (Length)
billTo_city	City of the billing address.	ccAuthService (R) ²	String (50)
billTo_country	Country of the billing address. Use the two-character ISO Standard Country Codes .	ccAuthService (R) ²	String (2)
billTo_email	Customer's email address.	ccAuthService (R) ²	String (255)
billTo_firstName	Customer's first name. For a credit card transaction, this name must match the name on the card.	ccAuthService (R) ²	String (60)
billTo_lastName	Customer's last name. For a credit card transaction, this name must match the name on the card.	ccAuthService (R) ²	String (60)
billTo_phoneNumber	Customer's phone number. CyberSource recommends that you include the country code when the order is from outside the U.S.	ccAuthService (O)	String (15)
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	ccAuthService (R) ²	String (9)
billTo_state	State or province of the billing address. For an address in the U.S. or Canada, use the State, Province, and Territory Codes for the United States and Canada .	ccAuthService (R) ²	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.
- 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 44. **Important** It is your responsibility to determine whether a field is required for the transaction you are requesting.

Table 4 Request Fields (Continued)

Field	Description	Used By: Required (R) or Optional (O)	Data Type (Length)
billTo_street1	First line of the billing street address.	ccAuthService (R) ²	String (60)
billTo_street2	Additional address information. Example Attention: Accounts Payable	ccAuthService (O)	String (60)
card_accountNumber	The payment network token value.	ccAuthService (R)	Nonnegative integer (20)
card_cardType	Type of card to authorize. Possible values: <ul style="list-style-type: none"> ■ 001: Visa ■ 002: Mastercard ■ 003: American Express ■ 004: Discover 	ccAuthService (R)	String (3)
card_cvNumber	CVN.	ccAuthService (R)	Nonnegative integer (4)
card_expirationMonth	Two-digit month in which the payment network token expires. Format: MM. Possible values: 01 through 12.	ccAuthService (R)	String (2)
card_expirationYear	Four-digit year in which the payment network token expires. Format: YYYY.	ccAuthService (R)	Nonnegative integer (4)

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 44. **Important** It is your responsibility to determine whether a field is required for the transaction you are requesting.

Table 4 Request Fields (Continued)

Field	Description	Used By: Required (R) or Optional (O)	Data Type (Length)
ccAuthService_cavv	<p>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.</p> <p>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.</p> <p>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.</p> <p>CyberSource through VisaNet The value for this field corresponds to the following data in the TC 33 capture file¹:</p> <ul style="list-style-type: none"> ■ Record: CP01 TCR8 ■ Position: 77-78 ■ Field: CAVV version and authentication action. 	ccAuthService (R)	String (40)
ccAuthService_commerceIndicator	<p>For an authorization with a payment network token. The values are required for the merchant decryption method (see "Option 1: Merchant Decryption," page 20).</p> <p>Possible values:</p> <ul style="list-style-type: none"> ■ <code>aesk</code>: American Express card type ■ <code>spa</code>: Mastercard card type ■ <code>vbv</code>: Visa card type mapped for Apple Pay transactions with e-commerce indicator of 5 ■ <code>internet</code>: Visa card type mapped for Apple Pay transactions with e-commerce indicator of 7 ■ <code>dipb</code>: Discover card type 	ccAuthService (See description)	String (20)

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 44. **Important** It is your responsibility to determine whether a field is required for the transaction you are requesting.

Table 4 Request Fields (Continued)

Field	Description	Used By: Required (R) or Optional (O)	Data Type (Length)
ccAuthService_ directoryServerTrans actionID	Identifier generated during the authentication transaction by the Mastercard Directory Server and passed back with the authentication results.	ccAuthService (O)	String (36)
ccAuthService_eciRaw	Raw electronic commerce indicator (ECI).	ccAuthService (O)	String (2)
ccAuthService_ networkTokenCryptogram	Token authentication verification value cryptogram. For token-based transactions with 3D Secure, you must submit both types of cryptograms: network token and 3D Secure. The value for this field must be 28-character Base64 or 40-character hex binary. All cryptograms use one of these formats.	ccAuthService (O)	String (40)
ccAuthService_ paSpecificationVersion	The 3D Secure version that you used for strong customer authentication (SCA); for example, 3D Secure 1.0.2 or 2.0.0.	ccAuthService (O)	String (20)
ccAuthService_run	Whether to include ccAuthService in your request. Possible values: <ul style="list-style-type: none">■ 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 24). 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 44. **Important** It is your responsibility to determine whether a field is required for the transaction you are requesting.

Table 4 Request Fields (Continued)

Field	Description	Used By: Required (R) or Optional (O)	Data Type (Length)
encryptedPayment_data	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 .	ics_auth (R)	
encryptedPayment_descriptor	Format of the encrypted payment data. The value for Apple Pay is: Rk1EPUNPTU1PTi5BUFBMRS5JTkFQUC5QQV1NRU5U	ics_auth (R)	String (128)
encryptedPayment_encoding	Encoding method used to encrypt the payment data: Base64	ics_auth (R)	String (6)
item_#_productCode	Type of product. This value is used to determine the product category: electronic, handling, physical, service, or shipping. The default is <code>default</code> . See "Numbered Elements," page 43.	ccAuthService (O)	String (255)
item_#_productName	Name of the product. This field is required when the item_#_productCode value is not <code>default</code> or one of the values related to shipping and/or handling. See "Numbered Elements," page 43.	ccAuthService (See description)	String (255)
item_#_productSKU	Identification code for the product. This field is required when the item_#_productCode value is not <code>default</code> or one of the values related to shipping and/or handling. See "Numbered Elements," page 43.	ccAuthService (See description)	String (255)
item_#_quantity	The default is 1. This field is required when the item_#_productCode value is not <code>default</code> or one of the values related to shipping and/or handling. See "Numbered Elements," page 43.	ccAuthService (See description)	Integer (10)
item_#_taxAmount	Total tax to apply to the product. This value cannot be negative. See "Numbered Elements," page 43.	ccAuthService (See description)	String (15)

- 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 44. **Important** It is your responsibility to determine whether a field is required for the transaction you are requesting.

Table 4 Request Fields (Continued)

Field	Description	Used By: Required (R) or Optional (O)	Data Type (Length)
item_#_unitPrice	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. See "Numbered Elements," page 43.	ccAuthService (See description)	String (15)
merchantID	Your CyberSource merchant ID. Use the same merchant ID for evaluation, testing, and production.	ccAuthService (R)	String (30)
merchantReferenceCode	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 .	ccAuthService (R)	String (50)
paymentNetworkToken_assuranceLevel	Confidence level of the tokenization. This value is assigned by the token service provider. Note This field is supported only for CyberSource through VisaNet and FDC Nashville Global.	ccAuthService (O)	String (2)
paymentNetworkToken_deviceTechType	Type of technology used in the device to store token data. Possible value: 001: Secure Element (SE) 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. Note This field is supported only for FDC Compass.	ccAuthService (O)	Integer (3)
paymentNetworkToken_requestorID	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. Note This field is supported only for CyberSource through VisaNet, FDC Nashville Global, and Chase Paymentech Solutions.	ccAuthService (O)	String (11)

- 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 44. **Important** It is your responsibility to determine whether a field is required for the transaction you are requesting.

Table 4 Request Fields (Continued)

Field	Description	Used By: Required (R) or Optional (O)	Data Type (Length)
paymentNetworkToken_ transactionType	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. Set the value for this field to 1.	ccAuthService (R)	String (1)
paymentSolution	Identifies Apple Pay as the payment solution that is being used for the transaction: Set the value for this field to 001. Note This unique ID differentiates digital solution transactions within the CyberSource platform for reporting purposes.	ccAuthService (R)	String (3)
purchaseTotals_ currency	Currency used for the order. For the possible values, see the ISO Standard Currency Codes .	ccAuthService (R)	String (5)
purchaseTotals_ grandTotalAmount	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.	ccAuthService (R)	Decimal (15)
ucaf_authenticationData	Cryptogram for authorizations with payment network tokens with Mastercard.	ccAuthService (R)	String (32)
ucaf_collectionIndicator	Required field for authorizations with payment network tokens with Mastercard. Set the value for this field to 2.	ccAuthService (R)	String with numbers only (1)
<p>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.</p> <p>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 44. Important It is your responsibility to determine whether a field is required for the transaction you are requesting.</p>			

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

Field	Description	Returned By	Data Type & Length
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 file ¹ : <ul style="list-style-type: none"> ■ 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.	ccAuthReply	Date and time (20)
ccAuthReply_avsCode	AVS results. See Credit Card Services Using the Simple Order API for a detailed list of AVS codes.	ccAuthReply	String (1)
ccAuthReply_avsCodeRaw	AVS result code sent directly from the processor. Returned only when the processor returns this value.	ccAuthReply	String (10)

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)

Field	Description	Returned By	Data Type & Length
ccAuthReply_cvCode	CVN result code. See Credit Card Services Using the Simple Order API for a detailed list of CVN codes.	ccAuthReply	String (1)
ccAuthReply_cvCodeRaw	CVN result code sent directly from the processor. Returned only when the processor returns this value.	ccAuthReply	String (10)
ccAuthReply_paymentCardService	<p>Mastercard service that was used for the transaction. Mastercard provides this value to CyberSource. Possible value:</p> <p>53: Mastercard card-on-file token service</p> <p>Note This field is returned only for CyberSource through VisaNet.</p>	ccAuthReply	String (2)
ccAuthReply_paymentCardServiceResult	<p>Result of the Mastercard card-on-file token service. Mastercard provides this value to CyberSource. Possible values:</p> <ul style="list-style-type: none"> ■ C: Service completed successfully. ■ E: One of the following: <ul style="list-style-type: none"> ● 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: <ul style="list-style-type: none"> ● 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. <p>Note This field is returned only for CyberSource through VisaNet.</p>	ccAuthReply	String (1)
ccAuthReply_processorResponse	For most processors, this is the error message sent directly from the bank. Returned only when the processor returns this value.	ccAuthReply	String (10)

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)

Field	Description	Returned By	Data Type & Length
ccAuthReply_reasonCode	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.	ccAuthReply	Integer (5)
ccAuthReply_reconciliationID	Reference number for the transaction. This value is not returned for all processors.	ccAuthReply	String (60)
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: <ul style="list-style-type: none"> ■ 1: Transaction qualifies for Mastercard authentication type 1. ■ 2: Transaction qualifies for Mastercard authentication type 2. <p>Note This field is returned only for CyberSource through VisaNet.</p>	ccAuthReply	String (1)
ccAuthReversalReply_paymentCardService	Mastercard service that was used for the transaction. Mastercard provides this value to CyberSource. Possible value: <p>53: Mastercard card-on-file token service</p> <p>Note This field is returned only for CyberSource through VisaNet.</p>	ccAuthReversal Reply	String (2)
<p>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.</p>			

Table 5 Reply Fields (Continued)

Field	Description	Returned By	Data Type & Length
ccAuthReversalReply_paymentCardServiceResult	<p>Result of the Mastercard card-on-file token service. Mastercard provides this value to CyberSource. Possible values:</p> <ul style="list-style-type: none"> ■ C: Service completed successfully. ■ F: One of the following: <ul style="list-style-type: none"> ● 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: <ul style="list-style-type: none"> ● 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. <p>Note This field is returned only for CyberSource through VisaNet.</p>	ccAuthReversal Reply	String (1)
decision	<p>Summarizes the result of the overall request. Possible values:</p> <ul style="list-style-type: none"> ■ ACCEPT ■ ERROR ■ REJECT ■ REVIEW: Returned only when you use CyberSource Decision Manager. 	ccAuthReply	String (6)
invalidField_0 through invalidField_N	<p>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.</p>	ccAuthReply	String (100)
<p>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.</p>			

Table 5 Reply Fields (Continued)

Field	Description	Returned By	Data Type & Length
merchantReferenceCode	Order reference or tracking number that you provided in the request. If you included multi-byte characters in this field in the request, the returned value might include corrupted characters.	ccAuthReply	String (50)
missingField_0 through missingField_N	Required fields that were missing from the request. For information about missing or invalid fields, see Getting Started with CyberSource Advanced for the Simple Order API .	ccAuthReply	String (100)
paymentNetworkToken_accountStatus	Possible values: <ul style="list-style-type: none"> ■ N: Nonregulated ■ R: Regulated Note This field is returned only for CyberSource through VisaNet.	ccAuthReply	String (1)
paymentNetworkToken_assuranceLevel	Confidence level of the tokenization. This value is assigned by the token service provider. Note This field is returned only for CyberSource through VisaNet and FDC Nashville Global.	ccAuthReply	String (2)
paymentNetworkToken_originalCardCategory	Mastercard product ID associated with the primary account number (PAN). For the possible values, see "Mastercard Product IDs" in <i>Credit Card Services Using the Simple Order API</i> . CyberSource through VisaNet For the possible values, see "Mastercard Product IDs" in <i>Credit Card Services for CyberSource through VisaNet Using the Simple Order API</i> . Note This field is returned only for Mastercard transactions on CyberSource through VisaNet.	ccAuthReply	String (3)
paymentNetworkToken_requestorID	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. Note This field is supported only for CyberSource through VisaNet and FDC Nashville Global.	ccAuthService	String (11)
purchaseTotals_currency	Currency used for the order. For the possible values, see the ISO Standard Currency Codes .	ccAuthReply	String (5)
reasonCode	Numeric value corresponding to the result of the overall request. See Credit Card Services Using the Simple Order API for a detailed list of reason codes.	ccAuthReply	Integer (5)

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)

Field	Description	Returned By	Data Type & Length
requestID	Identifier for the request.	ccAuthReply	String (26)
requestToken	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.	ccAuthReply	String (256)
token_expirationMonth	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.	ccAuthReply	String (2)
token_expirationYear	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.	ccAuthReply	String (4)
token_prefix	First 6 digits of token. CyberSource includes this field in the reply message when it decrypts the payment blob for the tokenized transaction.	ccAuthReply	String (6)
token_suffix	Last 4 digits of token. CyberSource includes this field in the reply message when it decrypts the payment blob for the tokenized transaction.	ccAuthReply	String (4)

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.