

# Apple Pay

## Using the Simple Order API

February 2019

**CyberSource<sup>®</sup>**  
the power of payment

## 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](mailto: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 Unpublished rights reserved under the copyright laws of the United States.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](#) 6

[Audience and Purpose](#) 6

[Conventions](#) 6

[Note and Important Statements](#) 6

[Text and Command Conventions](#) 7

[Related Documents](#) 7

[Customer Support](#) 7

---

## **Chapter 1** [Apple Pay Integrations](#) 8

[In-App Transactions](#) 8

[CyberSource API Integration](#) 8

[Merchant Decryption](#) 8

[CyberSource Decryption](#) 9

[Web Transactions](#) 10

[Integration Types](#) 10

[Merchant Decryption](#) 10

[CyberSource Decryption](#) 10

[Requirements](#) 11

[Apple Pay JavaScript](#) 12

[Apple Pay Button](#) 12

[ApplePaySession Class](#) 12

[Create ApplePaySession Object](#) 13

[Merchant Validation](#) 13

[Payment Confirmation](#) 13

[Merchant Decryption](#) 13

[CyberSource Decryption](#) 14

---

<b>Chapter 2</b>	<b>Getting Started</b>	<b>15</b>
	Requirements	15
	Supported Processors, Card Types, and Optional Features	16
	Enrolling for Apple Pay	17
	Generating a New CSR	18
	Transaction Report	19

---

<b>Chapter 3</b>	<b>Requesting the Authorization Service</b>	<b>20</b>
	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	41

---

<b>Appendix A</b>	<b>API Fields</b>	<b>42</b>
	Data Type Definitions	42
	Numbered Elements	42
	Relaxed Requirements for Address Data and Expiration Date	43
	API Request Fields	44
	API Reply Fields	52

# Recent Revisions to This Document

Release	Changes
February 2019	<p>Updated content about the Apple Pay response payload value for the <b>ccAuthService_commerceIndicator</b> field. See <a href="#">"Option 1: Merchant Decryption," page 20</a> and <a href="#">"ccAuthService_commerceIndicator," page 48</a>.</p> <p>Updated content about the JavaScript for obtaining a Base64-encoded value. See <a href="#">"CyberSource Decryption," page 14</a>.</p>
August 2018	This revision contains only editorial changes and no technical updates.
July 2018	<p>All processors: updated information about optional features. See <a href="#">"Supported Processors, Card Types, and Optional Features," page 16</a>.</p> <p>Added support for the processor Worldpay VAP. See <a href="#">"Supported Processors, Card Types, and Optional Features," page 16</a>.</p>
April 2018	All processors that support merchant-initiated transactions: added Visa card type for merchant-initiated transactions. See <a href="#">"Supported Processors, Card Types, and Optional Features," page 16</a> .
January 2018	<ul style="list-style-type: none"> <li>■ Added Discover card to the list of supported cards for FDC Nash Global (see <a href="#">"Supported Processors, Card Types, and Optional Features," page 16</a>).</li> <li>■ Updated the <b>ccAuthService_cavv</b> description to state that the value could be a 20 or 40-character hex binary (see <a href="#">"API Request Fields," page 44</a>).</li> </ul>
October 2017	<ul style="list-style-type: none"> <li>■ Added JCN Gateway to the list of supported processors (see <a href="#">"Supported Processors, Card Types, and Optional Features," page 16</a>).</li> <li>■ Added JCB transactions content (see <a href="#">page 28</a> and <a href="#">page 39</a>).</li> <li>■ CyberSource through VisaNet. Added Vantiv as a supported acquirer.</li> <li>■ Removed Moneris from the list of supported processors.</li> <li>■ Added a new section titled <a href="#">"Supported Processors, Card Types, and Optional Features," page 16</a>.</li> <li>■ Added several Merchant-Initiated Transaction fields (see <a href="#">Appendix A, "API Fields," on page 42</a>): <ul style="list-style-type: none"> <li>● subsequentAuth</li> <li>● subsequentAuthFirst</li> <li>● subsequentAuthReason</li> <li>● subsequentAuthStoredCredential</li> <li>● subsequentAuthTransactionID</li> </ul> </li> </ul>

# 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



**Note**

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



**Important**

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

## Text and Command Conventions

Convention	Usage
<b>Bold</b>	<ul style="list-style-type: none"> <li>Field and service names in text; for example: Include the <b>card_accountNumber</b> field.</li> <li>Items that you are instructed to act upon; for example: Click <b>Save</b>.</li> </ul>
Screen text	<ul style="list-style-type: none"> <li>XML elements.</li> <li>Code examples and samples.</li> <li>Text that you enter in an API environment; for example: Set the <b>ccAuthService_run</b> field to <code>true</code>.</li> </ul>

## Related Documents

CyberSource Documents:

- *Business Center Overview* ([PDF](#) | [HTML](#))
- *Classic Reporting Developer Guide* ([PDF](#) | [HTML](#))
- *Credit Card Services Using the Simple Order API* ([PDF](#) | [HTML](#))
- *Credit Card Services for CyberSource through VisaNet Using the Simple Order API*—contact CyberSource Customer Support to obtain this guide.
- *Payment Network Tokenization 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](http://www.cybersource.com/support_center/support_documentation)

## Customer Support

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

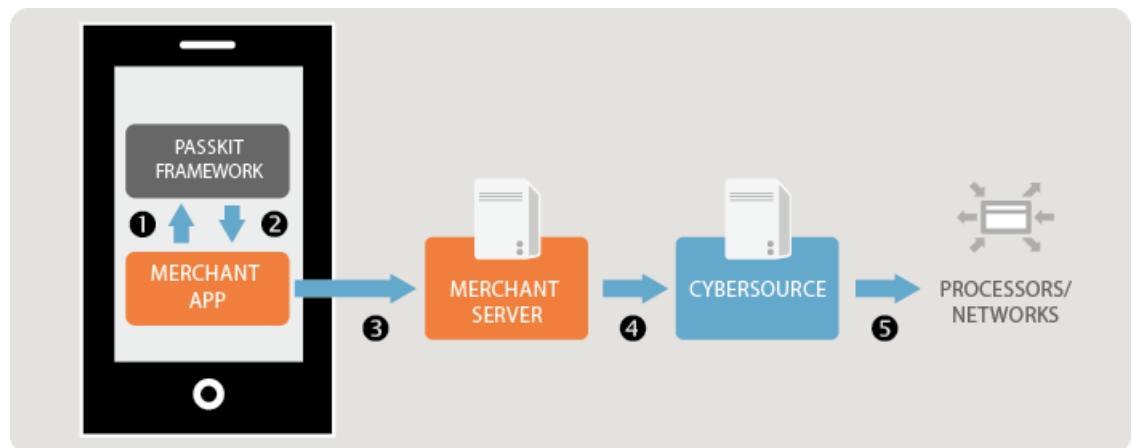
<http://www.cybersource.com/support>

# Apple Pay Integrations

## In-App Transactions

### CyberSource API Integration

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

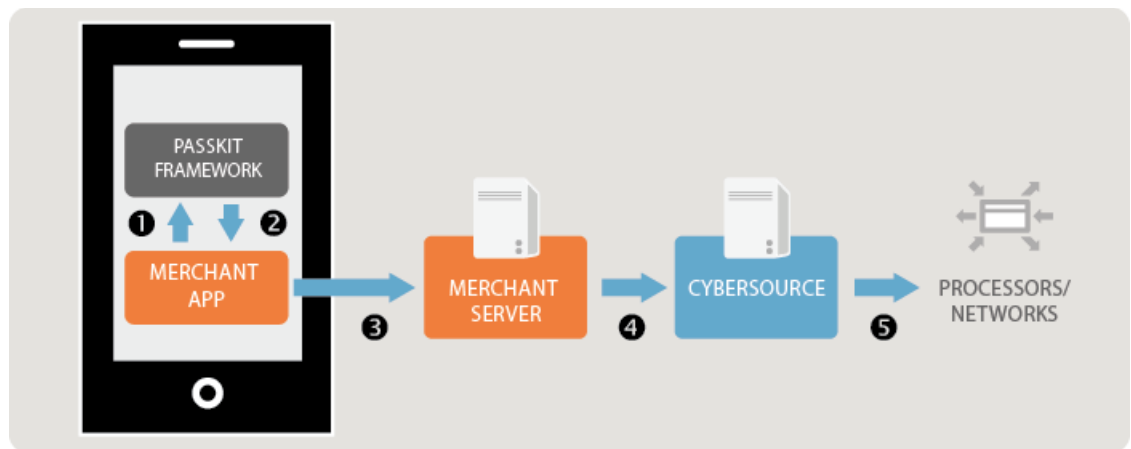


- 4 Using the CyberSource API, you submit the authorization request and include the decrypted payment data. See ["Option 1: Merchant Decryption," page 20](#).
- 5 CyberSource forwards the information to the payment network, including your processor and the relevant payment card company.

**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 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 31](#).
- 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

---

## Integration Types

### 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 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](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.



**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 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 for Apple Pay](#)," [page 17](#)).



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 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:

<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 12) or creating an Apple Pay session (see "[Create ApplePaySession Object](#)," page 13), 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 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](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 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 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);  
  
  ...  
}
```

---

# 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 16).
- You must have an *Admin* or *Team Agent* user of the [Apple Pay Developer account](#).



Apple Pay relies on payment network tokenization. You can sign up for Apple Pay only if both of the following statements are true:

- Your processor supports payment network tokenization.
- CyberSource supports payment network tokenization 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 tokenization.
  - Wait until your processor supports payment network tokenization.
-

## Supported Processors, Card Types, and Optional Features



### Note

All optional features, except split shipments, are described in the Payment Network Tokenization Guide. See *Payment Network Tokenization Using the Simple Order API* ([PDF](#) | [HTML](#)). Split shipments are described in the Credit Card Guide. See *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"> <li>■ Multiple partial captures</li> <li>■ Recurring payments</li> </ul>
Barclays	Visa, Mastercard	<ul style="list-style-type: none"> <li>■ Multiple partial captures</li> <li>■ Recurring payments</li> </ul>
Chase Paymentech Solutions	Visa, Mastercard, American Express, Discover	<ul style="list-style-type: none"> <li>■ Multiple partial captures</li> <li>■ Recurring payments</li> </ul>
CyberSource through VisaNet. The supported acquirers are: <ul style="list-style-type: none"> <li>■ Australia and New Zealand Banking Group Ltd. (ANZ)</li> <li>■ CitiBank Singapore Ltd.</li> <li>■ Global Payments Asia Pacific</li> <li>■ Vantiv</li> <li>■ Westpac</li> </ul>	Visa, Mastercard	<ul style="list-style-type: none"> <li>■ Split shipments</li> <li>■ Recurring payments</li> </ul>
FDC Compass	Visa, Mastercard, American Express	<ul style="list-style-type: none"> <li>■ Multiple partial captures</li> <li>■ Recurring payments</li> </ul>
FDC Nashville Global	Visa, Mastercard, American Express, Discover	<ul style="list-style-type: none"> <li>■ Recurring payments</li> <li>■ Multiple partial captures</li> </ul>
GPN	Visa, Mastercard, American Express	<ul style="list-style-type: none"> <li>■ Split shipments</li> <li>■ Recurring payments</li> </ul>
JCN Gateway	JCB	<ul style="list-style-type: none"> <li>■ Multiple partial captures</li> </ul>
OmniPay Direct. The supported acquirers are: <ul style="list-style-type: none"> <li>■ Bank of America Merchant Services</li> <li>■ First Data Merchant Solutions (Europe)</li> <li>■ Global Payments International Acquiring</li> </ul>	Visa, Mastercard	<ul style="list-style-type: none"> <li>■ Multiple partial captures</li> <li>■ Recurring payments</li> </ul>



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

Processor	Card Types	Optional Features
SIX	Visa, Mastercard	
Streamline	Visa, Mastercard	<ul style="list-style-type: none"> <li>■ Multiple partial captures</li> <li>■ Recurring payments</li> <li>■ Subsequent authorizations</li> </ul>
TSYS Acquiring Solutions	Visa, Mastercard, American Express	<ul style="list-style-type: none"> <li>■ Multiple partial captures</li> <li>■ Recurring payments</li> </ul>
Worldpay VAP	Visa, Mastercard	<ul style="list-style-type: none"> <li>■ Recurring payments</li> </ul>

Worldpay VAP was previously called *Little*.

## Enrolling for Apple Pay

### To integrate Apple Pay:

- Step 1** Log in to the Business Center:
- Test transactions: <https://ebctest.cybersource.com>
  - Live transactions: <https://ebc.cybersource.com>
- a** Under **Account Management** in the left navigation panel, choose **Digital Payment Solutions**.
- b** Click **Sign Up**. Follow the steps to verify your account information and accept the agreement on the Apple Pay Developers web site.

- Step 2** Generate a Certificate Signing Request (CSR).

- a** Enter your **Apple Merchant ID** that you registered in the Certificates, Identifiers and Profiles area of the Member Center on the Apple web site.



**Important**

CyberSource decryption method—[Step b](#) and [Step c](#) are required.

Merchant decryption method—[Step b](#) is required only for saving your Apple Pay merchant ID. The CSR must be obtained directly from Apple.

- b** Click **Generate CSR** to save your Apple Pay merchant ID and to generate a CSR that is associated with your merchant ID.

- c Submit the CSR to Apple.

Go to the Apple [web site](#) and upload the CSR. Apple provides you with an Apple Pay Certificate for your Apple Merchant ID. For information about adding certificates to your Apple Merchant ID, see the *PassKit Framework Reference*.



A CSR submitted to Apple expires after 25 months. CyberSource recommends generating and submitting a new CSR prior to the expiration date. See "[Generating a New CSR](#)," page 18.

---

- Step 3** Obtain the Apple Pay Certificate.

If you do not have the Apple Pay Certificate, complete the process that is described in the *PassKit Framework Reference*. The Apple Pay Certificate is required for creating an iOS application. The Apple Pay Certificate is not needed for payment processing with CyberSource.

- Step 4** 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 5** Repeat Steps 1, 2, 3, and 5 with your CyberSource production account and the Apple production account.
- 

## Generating a New CSR

### To generate a new CSR:

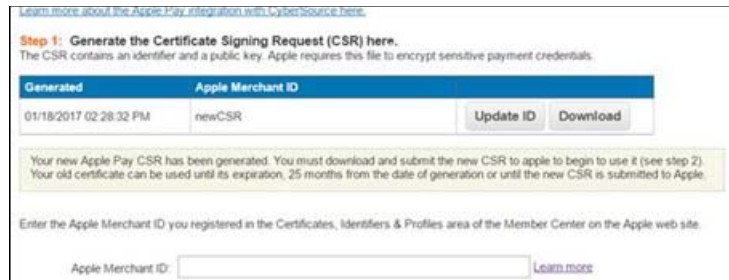
---

- Step 1** Log in to the Business Center:
  - Test transactions: <https://ebctest.cybersource.com>
  - Live transactions: <https://ebc.cybersource.com>
- Step 2** Under **Account Management** in the left navigation panel, choose **Digital Payment Solutions**.

**Step 3** Click **Enabled**.

**Step 4** Generate a New CSR:

- a Enter the Apple Merchant ID that you registered in the Certificates, Identifiers and Profiles area of the Member Center on the Apple web site.
- b Click **Generate New CSR**.



The new CSR replaces the previous CSR in the list above. The previous CSR continues to be active until its expiration date (25 months from the date it was generated.)

- c Download and submit the new CSR to Apple.

---

## Transaction Report

Use the Business Center and the Single Transaction 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 Single Transaction Report, see the [Classic Reporting Developer Guide](#).

# Requesting the Authorization Service

## Option 1: Merchant Decryption

---

### Visa Transaction

To request an authorization for a Visa transaction:

---



**Note**

See ["Relaxed Requirements for Address Data and Expiration Date,"](#) page 43, and ["API Request Fields,"](#) page 44, for detailed 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 payment network token expiration date fields.
- 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\_commerceIndicator** field to the ECI value contained in the Apple Pay response payload (5=*vbv* and 7=*internet*).
- Step 6** 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>4650100000000839</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

### To request an authorization for a Mastercard transaction:

**Note**

See "Relaxed Requirements for Address Data and Expiration Date," page 43, and "API Request Fields," page 44, for detailed 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 payment network token expiration date fields.
  - Step 3** Set the **ucaf\_authenticationData** field to the 3D Secure cryptogram of the payment network token.
  - Step 4** Set the **ucaf\_collectionIndicator** field to 2.
  - 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 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>5555555555554444</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)**


---

```

<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

### To request an authorization for an American Express transaction:

**Note**

See "Relaxed Requirements for Address Data and Expiration Date," page 43, and "API Request Fields," page 44, for detailed 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 payment network token expiration date fields.
  - 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 `paymentNetworkToken_transactionType` field to 1.
- Step 5** Set the `ccAuthService_commerceIndicator` field to ECI value contained in the Apple Pay response payload.
- Step 6** 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>378282246310005</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:

**Note**

See "Relaxed Requirements for Address Data and Expiration Date," page 43, and "API Request Fields," page 44, for detailed 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 payment network token expiration date fields.
  - 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\_commerceIndicator** field to ECI value contained in the Apple Pay response payload.
  - Step 6** 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>6011111111111117</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:

**Note**

See ["Relaxed Requirements for Address Data and Expiration Date,"](#) page 43, and ["API Request Fields,"](#) page 44, for detailed field descriptions.

- 
- Step 1** Set the **card\_accountNumber** field to the payment network token value.
  - Step 2** Set the **cardexpiration\_Month** and **card\_expirationYear** fields to the payment network token expiration date values.
  - 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 **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>3566111111111113</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>446584034076500001541</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=3566111111111113
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:

---



**Note**

See "Relaxed Requirements for Address Data and Expiration Date," page 43, and "API Request Fields," page 44, for detailed field descriptions.

---

- Step 1** Set the **encryptedPayment\_data** field to the Base64 encoded value obtained from the **paymentData** property of the **PKPaymentToken** object. See [page 9](#).
- 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>RklEPUNPTU1PTi5BUFBMRS5JTkFQUC5QQVlNRU5U</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:



#### Note

See "Relaxed Requirements for Address Data and Expiration Date," page 43, and "API Request Fields," page 44, for detailed field descriptions.

- Step 1** Set the `encryptedPayment_data` field to the Base64 encoded value obtained from the `paymentData` property of the `PKPaymentToken` object. See page 9.
- Step 2** Set the `encryptedPayment_descriptor` field to:  
RklEPUNPTU1PTi5BUFBMRS5JTkFQUC5QQVlNRU5U
- 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>RklEPUNPTU1PTi5BUFBMRS5JTkFQUC5QQVlNRU5U</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:



#### Note

See "Relaxed Requirements for Address Data and Expiration Date," page 43, and "API Request Fields," page 44, for detailed field descriptions.

- Step 1** Set the `encryptedPayment_data` field to the Base64 encoded value obtained from the `paymentData` property of the `PKPaymentToken` object. See page 9.
- 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:



#### Note

See "Relaxed Requirements for Address Data and Expiration Date," page 43, and "API Request Fields," page 44, for detailed field descriptions.

- Step 1** Set the `encryptedPayment_data` field to the Base64 encoded value obtained from the `paymentData` property of the `PKPaymentToken` object. See [page 9](#).
- Step 2** Set the `encryptedPayment_descriptor` field to:  
RklEPUNPTU1PTi5BUFBMRS5JTkFQUC5QQVlNRU5U
- 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>RklEPUNPTU1PTi5BUFBMRS5JTkFQUC5QQVlNRU5U</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:

**Note**

See ["Relaxed Requirements for Address Data and Expiration Date,"](#) page 43, and ["API Request Fields,"](#) page 44, for detailed 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 `Rk1EPUNPTU1PTi5BUFBMRS5JTtkFQUC5QQV1NRU5U`.
- 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>Rk1EPUNPTU1PTi5BUFBMRS5JTtkFQUC5QQV1NRU5U</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
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 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 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>&lt;item id="0"&gt;   &lt;unitPrice&gt;   &lt;quantity&gt; &lt;/item&gt;</pre>	<pre>item_0_unitPrice item_0_quantity</pre>
<pre>&lt;item id="1"&gt;   &lt;unitPrice&gt;   &lt;quantity&gt; &lt;/item&gt;</pre>	<pre>item_1_unitPrice item_1_quantity</pre>



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.

## API 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) <sup>2</sup>	String (50)
	<b>Important</b> It is your responsibility to determine whether a field is required for the transaction you are requesting.		
billTo_country	Country of the billing address. Use the two-character <i>ISO Standard Country Codes</i> .	ccAuthService (R) <sup>2</sup>	String (2)
	<b>Important</b> It is your responsibility to determine whether a field is required for the transaction you are requesting.		
billTo_email	Customer's email address.	ccAuthService (R) <sup>2</sup>	String (255)
	<b>Important</b> It is your responsibility to determine whether a field is required for the transaction you are requesting.		
billTo_firstName	Customer's first name. For a credit card transaction, this name must match the name on the card.	ccAuthService (R) <sup>2</sup>	String (60)
	<b>Important</b> It is your responsibility to determine whether a field is required for the transaction you are requesting.		

- 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 43. **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_lastName	Customer's last name. For a credit card transaction, this name must match the name on the card.  <b>Important</b> It is your responsibility to determine whether a field is required for the transaction you are requesting.	ccAuthService (R) <sup>2</sup>	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] <b>Example</b> 12345-6789  When the billing country is Canada, the 6-digit postal code must follow this format: [alpha][numeric][alpha][space] [numeric][alpha][numeric] <b>Example</b> A1B 2C3  <b>Important</b> It is your responsibility to determine whether a field is required for the transaction you are requesting.	ccAuthService (R) <sup>2</sup>	String (9)
billTo_state	State or province of the billing address. For an address in the U.S. or Canada, use the <a href="#">State, Province, and Territory Codes for the United States and Canada</a> .  <b>Important</b> It is your responsibility to determine whether a field is required for the transaction you are requesting.	ccAuthService (R) <sup>2</sup>	String (2)

---

- 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.
- 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 43. **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.  <b>Important</b> It is your responsibility to determine whether a field is required for the transaction you are requesting.	ccAuthService (R) <sup>2</sup>	String (60)
billTo_street2	Additional address information.  <b>Example</b> 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"> <li>■ 001: Visa</li> <li>■ 002: Mastercard</li> <li>■ 003: American Express</li> <li>■ 004: Discover</li> </ul>	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 43. **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><b>Visa</b> Cryptogram for payment network tokenization transactions. The value for this field must be 28-character Base64 or 40-character hex binary. All cryptograms use one of these formats.</p> <p><b>American Express</b> For a 20-byte cryptogram, set this field to the cryptogram for payment network tokenization transactions. For a 40-byte cryptogram, set this field to block A of the cryptogram for payment network tokenization transactions. The value for this field must be 28-character Base64 or 40-character hex binary. All cryptograms use one of these formats.</p> <p><b>Discover</b> Cryptogram for payment network tokenization transactions. The value for this field can be a 20 or 40-character hex binary. All cryptograms use one of these formats.</p> <p><b>CyberSource through VisaNet</b> The value for this field corresponds to the following data in the TC 33 capture file<sup>1</sup>:</p> <ul style="list-style-type: none"> <li>■ Record: CP01 TCR8</li> <li>■ Position: 77-78</li> <li>■ Field: CAVV version and authentication action.</li> </ul>	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 43. **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_ commerceIndicator	<p>For a payment network tokenization transaction.</p> <p>The values are required for the merchant decryption method (see "<a href="#">Option 1: Merchant Decryption</a>," <a href="#">page 20</a>).</p> <p>Possible values:</p> <ul style="list-style-type: none"> <li>■ <code>aesk</code>: American Express card type</li> <li>■ <code>spa</code>: Mastercard card type</li> <li>■ <code>vbv</code>: Visa card type mapped for Apple Pay transactions with eCommerce commerce indicator of 5</li> <li>■ <code>internet</code>: Visa card type mapped for Apple Pay transactions with eCommerce commerce indicator of 7</li> <li>■ <code>dipb</code>: Discover card type</li> </ul>	ccAuthService (See description)	String (20)
ccAuthService_eciRaw	Raw electronic commerce indicator (ECI).	ccAuthService	String (2)
ccAuthService_run	<p>Whether to include <b>ccAuthService</b> in your request.</p> <p>Possible values:</p> <ul style="list-style-type: none"> <li>■ <code>TRUE</code>: Include the service in your request.</li> <li>■ <code>FALSE</code> (default): Do not include the service in your request.</li> </ul>	ccAuthService (R)	
ccAuthService_xid	<p><b>Visa</b></p> <p>Cryptogram for payment network tokenization transactions. The value for this field must be 28-character Base64 or 40-character hex binary. All cryptograms use one of these formats.</p> <p><b>American Express</b></p> <p>For a 20-byte cryptogram, set this field to the cryptogram for payment network tokenization transactions. For a 40-byte cryptogram, set this field to block A of the cryptogram for payment network tokenization transactions (see <a href="#">page 24</a>). The value for this field must be 28-character Base64 or 40-character hex binary. All cryptograms use one of these formats.</p>	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 43](#). **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 <b>paymentData</b> property of the <b>PKPaymentToken</b> object. See the <a href="#">PassKit Framework Reference</a> .	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 42.	ccAuthService (O)	String (255)
item_#_productName	Name of the product.  This field is required when the <b>item_#_productCode</b> value is not <code>default</code> or one of the values related to shipping and/or handling.  See "Numbered Elements," page 42.	ccAuthService (See description)	String (255)
item_#_productSKU	Identification code for the product.  This field is required when the <b>item_#_productCode</b> value is not <code>default</code> or one of the values related to shipping and/or handling.  See "Numbered Elements," page 42.	ccAuthService (See description)	String (255)
item_#_quantity	The default is 1.  This field is required when the <b>item_#_productCode</b> value is not <code>default</code> or one of the values related to shipping and/or handling.  See "Numbered Elements," page 42.	ccAuthService (See description)	Integer (10)
item_#_taxAmount	Total tax to apply to the product. This value cannot be negative.  See "Numbered Elements," page 42.	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 43. <b>Important</b> 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 <a href="#">"Numbered Elements," page 42.</a>	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 <a href="#">Getting Started with CyberSource Advanced for the Simple Order API.</a>	ccAuthService (R)	String (50)
paymentNetworkToken_assuranceLevel	Confidence level of the tokenization. This value is assigned by the token service provider.  <b>Note</b> 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.  <b>Note</b> 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.  <b>Note</b> This field is supported only for CyberSource through VisaNet, FDC Nashville Global, and Chase Paymentech Solutions.	ccAuthService (O)	String (11)
<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 <a href="#">"Relaxed Requirements for Address Data and Expiration Date," page 43.</a> <b>Important</b> It is your responsibility to determine whether a field is required for the transaction you are requesting.</p>			

**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.  <b>Note</b> 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: USD	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 (60)
ucaf_ authenticationData	Cryptogram for payment network tokenization transactions with Mastercard.	ccAuthService (R)	String (32)
ucaf_ collectionIndicator	Required field for payment network tokenization transactions with Mastercard.  Set the value for this field to 2.	ccAuthService (R)	String with numbers only (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.		
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 43. <b>Important</b> It is your responsibility to determine whether a field is required for the transaction you are requesting.		

## API Reply Fields



### Important

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.



### Note

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	<p>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.</p> <p><b>Note</b> This field is returned only for CyberSource through VisaNet and FDC Nashville Global.</p> <p><b>CyberSource through VisaNet</b> The value for this field corresponds to the following data in the TC 33 capture file<sup>1</sup>:</p> <ul style="list-style-type: none"> <li>■ Record: CP01 TCRB</li> <li>■ Position: 85</li> <li>■ Field: American Express last 4 PAN return indicator.</li> </ul>	ccAuthReply	String (4)
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><b>Note</b> This field is returned only for CyberSource through VisaNet.</p>	ccAuthReply	String (2)

<sup>1</sup> 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_ paymentCardService Result	<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"> <li>■ C: Service completed successfully.</li> <li>■ F: One of the following: <ul style="list-style-type: none"> <li>● Incorrect Mastercard POS entry mode. The Mastercard POS entry mode should be 81 for an authorization or authorization reversal.</li> <li>● Incorrect Mastercard POS entry mode. The Mastercard POS entry mode should be 01 for a tokenized request.</li> <li>● Token requestor ID is missing or formatted incorrectly.</li> </ul> </li> <li>■ I: One of the following: <ul style="list-style-type: none"> <li>● Invalid token requestor ID.</li> <li>● Suspended or deactivated token.</li> <li>● Invalid token (not in mapping table).</li> </ul> </li> <li>■ T: Invalid combination of token requestor ID and token.</li> <li>■ U: Expired token.</li> <li>■ W: Primary account number (PAN) listed in electronic warning bulletin.</li> </ul> <p><b>Note</b> This field is returned only for CyberSource through VisaNet.</p>	ccAuthReply	String (1)
ccAuthReply_ transactionQualification	<p>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:</p> <ul style="list-style-type: none"> <li>■ 1: Transaction qualifies for Mastercard authentication type 1.</li> <li>■ 2: Transaction qualifies for Mastercard authentication type 2.</li> </ul> <p><b>Note</b> This field is returned only for CyberSource through VisaNet.</p>	ccAuthReply	String (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>			

**Table 5 Reply Fields (Continued)**

Field	Description	Returned By	Data Type & Length
ccAuthReversalReply_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><b>Note</b> This field is returned only for CyberSource through VisaNet.</p>	ccAuthReversalReply	String (2)
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"> <li>■ C: Service completed successfully.</li> <li>■ F: One of the following: <ul style="list-style-type: none"> <li>● Incorrect Mastercard POS entry mode. The Mastercard POS entry mode should be 81 for an authorization or authorization reversal.</li> <li>● Incorrect Mastercard POS entry mode. The Mastercard POS entry mode should be 01 for a tokenized request.</li> <li>● Token requestor ID is missing or formatted incorrectly.</li> </ul> </li> <li>■ I: One of the following: <ul style="list-style-type: none"> <li>● Invalid token requestor ID.</li> <li>● Suspended or deactivated token.</li> <li>● Invalid token (not in mapping table).</li> </ul> </li> <li>■ T: Invalid combination of token requestor ID and token.</li> <li>■ U: Expired token.</li> <li>■ W: Primary account number (PAN) listed in electronic warning bulletin.</li> </ul> <p><b>Note</b> This field is returned only for CyberSource through VisaNet.</p>	ccAuthReversalReply	String (1)
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	<p>Time of authorization.</p> <p>Format: YYYY-MM-DDThh:mm:ssZ</p> <p>Example: 2016-08-11T22:47:57Z equals August 11, 2016, at 22:47 (10:47:57 p.m.). The T separates the date and the time. The Z indicates UTC.</p>	ccAuthReply	String (20)

<sup>1</sup> 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_avsCode	AVS results. See <a href="#">Credit Card Services Using the Simple Order API</a> 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)
ccAuthReply_cvCode	CVN result code. See <a href="#">Credit Card Services Using the Simple Order API</a> 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_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)
ccAuthReply_reasonCode	Numeric value corresponding to the result of the credit card authorization request. See <a href="#">Credit Card Services Using the Simple Order API</a> 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)
decision	Summarizes the result of the overall request. Possible values: <ul style="list-style-type: none"> <li>■ ACCEPT</li> <li>■ ERROR</li> <li>■ REJECT</li> <li>■ REVIEW: Returned only when you use CyberSource Decision Manager.</li> </ul>	ccAuthReply	String (6)
invalidField_0 through invalidField_N	Fields in the request that contained invalid data. For information about missing or invalid fields, see <a href="#">Getting Started with CyberSource Advanced for the Simple Order API</a> .	ccAuthReply	String (100)
merchantReferenceCode	Order reference or tracking number that you provided in the request. If you included multi-byte characters in this field in the request, the returned value might include corrupted characters.	ccAuthReply	String (50)

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
missingField_0 through missingField_N	Required fields that were missing from the request. For information about missing or invalid fields, see <a href="#">Getting Started with CyberSource Advanced for the Simple Order API</a> .	ccAuthReply	String (100)
paymentNetworkToken_accountStatus	Possible values: <ul style="list-style-type: none"> <li>■ N: Nonregulated</li> <li>■ R: Regulated</li> </ul> <b>Note</b> 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. <b>Note</b> 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 “ <a href="#">Mastercard Product IDs</a> ” in <i>Credit Card Services Using the Simple Order API</i> . For the possible values, see “ <a href="#">Mastercard Product IDs</a> ” in <i>Credit Card Services for CyberSource through VisaNet Using the Simple Order API</i> . <b>Note</b> 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. <b>Note</b> 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 <a href="#">ISO Standard Currency Codes</a> .	ccAuthReply	String (5)
reasonCode	Numeric value corresponding to the result of the overall request. See <a href="#">Credit Card Services Using the Simple Order API</a> for a detailed list of reason codes.	ccAuthReply	Integer (5)
requestID	Identifier for the request generated by the client.	ccAuthReply	String (26)

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
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 six 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 four 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.		