

# **Android Pay**

## **Using the Simple Order API**

January 2018



## 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

© 2018 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 Unpublished rights reserved under the copyright laws of the United States.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](#) 8

---

## **Chapter 1** [Introduction](#) 9

[Android Pay Overview](#) 9

[Payment Network Tokenization](#) 10

[Requirements](#) 10

[Supported Processors and Card Types](#) 11

[How Android Pay Works](#) 13

[Additional CyberSource Services](#) 15

[Transaction Endpoints](#) 15

[Merchant-Initiated Transactions](#) 16

[Terminology](#) 16

[Overview](#) 17

[Descriptions](#) 18

[Scenarios](#) 19

[Delayed Charge](#) 19

[Installment Payment](#) 19

[No-Show Transaction](#) 20

[Reauthorization](#) 21

[Recurring Payment](#) 21

[Resubmission](#) 22

[Unscheduled COF Transaction](#) 23

[API Field Descriptions](#) 23

---

<b>Chapter 2</b>	<b>Using the Google API for Android Pay</b>	<b>24</b>
	1. Obtain Your Client ID and Credentials	24
	2. Generate a Public/Private Key	24
	Option 1: Merchant Decryption	25
	Option 2: CyberSource Decryption	27
	3. Enable User	29
	4. Request Masked Wallet	29
	5. Request Full Wallet	30
	Option 1: Merchant Decryption	30
	Option 2: CyberSource Decryption	31
	6. Decrypt Encrypted Message Key	32
	Decrypted Payment Credential	33
	7. Request CyberSource Authorization	34

---

<b>Chapter 3</b>	<b>Requesting the Authorization Service</b>	<b>35</b>
	Option 1: Merchant Decryption	35
	Visa Transaction	35
	Mastercard Transaction	37
	American Express Transaction	40
	Discover Transaction	43
	JCB Transaction	45
	Option 2: CyberSource Decryption	49
	Visa Transaction	49
	Mastercard Transaction	51
	American Express Transaction	53
	Discover Transaction	55
	JCB Transaction	56
	Recurring Payments	58

---

<b>Appendix A</b>	<b>API Fields</b>	<b>60</b>
	Data Type Definitions	60
	Numbered Elements	60
	Relaxed Requirements for Address Data and Expiration Date	61
	API Request Fields	62
	API Reply Fields	73

# Recent Revisions to This Document

Release	Changes
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 and Card Types," page 11</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 62</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 and Card Types," page 11</a>).</li> <li>■ Added JCB transactions content (see <a href="#">page 45</a> and <a href="#">page 56</a>).</li> <li>■ CyberSource through VisaNet. Added Vantiv as a supported acquirer.</li> <li>■ Added a new section titled <a href="#">"Merchant-Initiated Transactions," page 16</a>.</li> <li>■ Added several merchant-initiated transaction fields (see <a href="#">Appendix A, "API Fields," on page 60</a>):               <ul style="list-style-type: none"> <li>• subsequentAuth</li> <li>• subsequentAuthFirst</li> <li>• subsequentAuthReason</li> <li>• subsequentAuthStoreCredential</li> <li>• subsequentAuthTransactionID</li> </ul> </li> </ul>
August 2017	Added the "Discover Transaction" sections. See <a href="#">"Discover Transaction," page 43</a> , and <a href="#">"Discover Transaction," page 55</a> .
June 2017	This revision contains only editorial changes and no technical updates.
May 2017	Updated the "Supported Processors and Card Types" section. See <a href="#">"Supported Processors and Card Types," page 11</a> .
April 2017	<ul style="list-style-type: none"> <li>■ Updated the merchant decryption examples:               <ul style="list-style-type: none"> <li>• Visa (see <a href="#">"Visa Transaction," page 35</a>).</li> <li>• Mastercard (see <a href="#">"Mastercard Transaction," page 37</a>).</li> <li>• American Express (see <a href="#">"American Express Transaction," page 40</a>).</li> </ul> </li> </ul> <p>Added the "Numbered Elements" section. See <a href="#">"Numbered Elements," page 60</a>. Added the item-level request fields. See <a href="#">Table 6, "Request Fields," on page 62</a>.</p>

# About This Guide

## Audience and Purpose

---

This document is written for merchants who want to enable customers to use Android Pay to pay for in-app purchases. This document provides an overview of integrating the Google API for Android Pay and describes how to request the CyberSource API to process an authorization.

This document describes both the Google API for Android Pay (see [Chapter 2, "Using the Google API for Android Pay," on page 24](#)) and the CyberSource API (see [page 35](#)). Merchants must request the Google API to receive the customer's encrypted payment data before requesting the CyberSource API to process the transaction.

## Conventions

---

## Note and Important Statements

---



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

---



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

---

## Text and Command Conventions

Convention	Usage
<b>Bold</b>	<ul style="list-style-type: none"> <li>Field and service names in text; for example: Include the <b>ics_applications</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>davService_run</b> field to <code>true</code>.</li> </ul>

## Related Documents

CyberSource Documents:

- [Getting Started with CyberSource Advanced for the Simple Order API](#)
- [Simple Order API and SOAP Toolkit API Documentation and Downloads page](#)
- [Credit Card Services Using the Simple Order API](#)

Android Pay documents:

- [Android Pay Developer Documentation](#)
- [Google API for Android Pay](#)
- [Android Pay Tutorial](#)
- [Android Pay Brand Guidelines](#)
- Android Pay Transaction Flow:  
<https://developers.google.com/android-pay/diagrams>
- Google API Reference:  
<https://developers.google.com/android/reference/com/google/android/gms/wallet/package-summary>

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>



# Introduction

## Android Pay Overview

---

Android Pay is a simple, secure in-app mobile payment solution. Merchants can choose CyberSource decryption or their own decryption to retrieve a customer's payment network token for processing Android Pay transactions.

### Merchant Decryption

You can decrypt the payment data from Google to retrieve the payment network token, the expiry date, the cryptogram, and other payment data associated with the transaction.

This method is best if your business has a fraud management solution or records management system that requires payment data relating to transactions.

- a** Using the Google API, the merchant requests the customer's encrypted pay data.
- b** The merchant **decrypts** the encrypted payment data from the Android Pay call back and extracts the payment network token, the cryptogram, and other payment data.
- c** Using the CyberSource API, the merchant constructs and submits the authorization request and includes the **decrypted** payment data.
- d** CyberSource processes the authorization request.

### CyberSource Decryption

You can simplify your payment processing by allowing CyberSource to decrypt the payment data for you during processing.

This method removes the complexity from your integration; you process transactions without seeing the payment network token and transaction data.

- a** Using the Google API, the merchant requests the customer's encrypted pay data.
- b** Using the CyberSource API, the merchant constructs and submits the authorization request and includes the **encrypted** payment data from the Android Pay call back.
- c** CyberSource **decrypts** the encrypted payment data to create the payment network token and processes the authorization request.

For an overview diagram showing how Android Pay works, see ["How Android Pay Works," page 13](#).

To get started, see ["1. Obtain Your Client ID and Credentials," page 24](#).

---

## Payment Network Tokenization

---

Payment network tokenization enables you to request a payment transaction with a payment network token instead of a customer's primary account number (PAN).

The payment network token is included in the customer's encrypted payment data, which is returned by the payment processor. There are two options to decrypt the customer's payment data to retrieve the payment network token: see "[Merchant Decryption](#)," page 9 and "[CyberSource Decryption](#)," page 9.

For in-app transactions, payment network tokenization uses some of the CyberSource payer authentication request fields. This approach simplifies your implementation if your order management system already uses payer authentication.

## Requirements

---

- Create a CyberSource merchant evaluation account if you do not have one already: <https://www.cybersource.com/register/>
- Have a merchant account with a supported processor (see "[Supported Processors and Card Types](#)," page 11).
- Install a CyberSource [Simple Order API client](#).
- Create an Android Pay client ID and credentials (see [Chapter 2, "Using the Google API for Android Pay](#)," on page 24).
- Submit your Android Pay enabled APK for [review](#).

## Supported Processors and Card Types

**Table 1** Supported Processors and Card Types

Processors	Card Types	Optional Feature
American Express Direct	American Express	Recurring Payments (see <a href="#">"Recurring Payments,"</a> page 58).
Barclays	<ul style="list-style-type: none"> <li>■ Visa</li> <li>■ Mastercard</li> </ul>	Recurring Payments (see <a href="#">"Recurring Payments,"</a> page 58).  Multiple partial captures. See <a href="#">Credit Card Services Using the Simple Order API</a> .
Chase Paymentech Solutions	<ul style="list-style-type: none"> <li>■ Visa</li> <li>■ Mastercard</li> <li>■ American Express</li> <li>■ Discover</li> </ul>	Recurring Payments (see <a href="#">"Recurring Payments,"</a> page 58).
CyberSource through VisaNet. The supported acquirers are: <ul style="list-style-type: none"> <li>■ Australia and New Zealand Banking Group Limited (ANZ)</li> <li>■ Vantiv</li> <li>■ Westpac</li> </ul>	<ul style="list-style-type: none"> <li>■ Visa</li> <li>■ Mastercard</li> </ul>	Recurring Payments (see <a href="#">"Recurring Payments,"</a> page 58).
FDC Compass	<ul style="list-style-type: none"> <li>■ Visa</li> <li>■ Mastercard</li> <li>■ American Express</li> </ul>	Recurring Payments (see <a href="#">"Recurring Payments,"</a> page 58).
FDC Nashville Global	<ul style="list-style-type: none"> <li>■ Visa</li> <li>■ Mastercard</li> <li>■ American Express</li> <li>■ Discover</li> </ul>	Recurring Payments (see <a href="#">"Recurring Payments,"</a> page 58).  Multiple partial captures. See <a href="#">Credit Card Services Using the Simple Order API</a> .
JCN Gateway	<ul style="list-style-type: none"> <li>■ JCB</li> </ul>	Multiple partial captures. See <a href="#">Credit Card Services Using the Simple Order API</a> .
GPN	<ul style="list-style-type: none"> <li>■ Visa</li> <li>■ Mastercard</li> <li>■ American Express</li> </ul>	Recurring Payments (see <a href="#">"Recurring Payments,"</a> page 58).

**Table 1 Supported Processors and Card Types (Continued)**

Processors	Card Types	Optional Feature
OmniPay Direct. The supported acquirers are: <ul style="list-style-type: none"> <li>■ Bank of America Merchant Services</li> <li>■ First Data Europe through OmniPay Direct</li> <li>■ Global Payments International Acquiring through OmniPay Direct</li> </ul>	<ul style="list-style-type: none"> <li>■ Visa</li> <li>■ Mastercard</li> </ul>	Recurring Payments (see <a href="#">"Recurring Payments," page 58</a> ).
SIX	<ul style="list-style-type: none"> <li>■ Visa</li> <li>■ Mastercard</li> </ul>	
Streamline	<ul style="list-style-type: none"> <li>■ Visa</li> <li>■ Mastercard</li> </ul>	Recurring Payments (see <a href="#">"Recurring Payments," page 58</a> ).
TSYS Acquiring Solutions	<ul style="list-style-type: none"> <li>■ Visa</li> <li>■ Mastercard</li> <li>■ American Express</li> </ul>	Recurring Payments (see <a href="#">"Recurring Payments," page 58</a> ).

## How Android Pay Works



- 1 The customer selects the *Buy with Android Pay* button. Using the Google API, the merchant requests the Masked Wallet object (see "[4. Request Masked Wallet](#)," page 29) to display customer details and masked card information to the customer.
- 2 The customer confirms the payment. Using the Google API, the merchant requests the Full Wallet object (see "[5. Request Full Wallet](#)," page 30) and includes the public key in the request.
- 3 Google requests the payment network token and cryptogram from the payment network.
  - a The payment network requests the payment network token and the cryptogram from the network token service provider.
  - b The network token service provider requests the payment network token from the card issuer.
  - c The card issuer returns the payment network token and the cryptogram to the network token service provider.
  - d The network token service provider returns the payment network token and the cryptogram to the payment network.
- 4 The payment network sends the payment network token and the cryptogram to Google.
- 5 Google creates an encrypted payload using the public key supplied in the Full Wallet request and includes it in the Google API response.

- 6 Android Pay App—call back returns the encrypted payment data to the merchant `server.onActivityResult()`.
- 7 The merchant chooses the Merchant decryption option or the CyberSource decryption option.
  - a CyberSource sends the authorization request to the acquirer.
  - b The acquirer processes the request from CyberSource and creates the payment network authorization request.
  - c The payment network processes the request from the acquirer and creates the issuer authorization request.
  - d The issuer processes the request from the payment network. The issuer looks up the PAN associated with the payment network token and returns an approved or declined authorization message to the payment network.
  - e The payment network returns the authorization response to the acquirer.
  - f The acquirer returns the authorization response to CyberSource.
- 8 CyberSource returns the authorization response to the merchant.
- 9 The merchant returns the authorization result to the Android app.
- 10 The Android app displays the confirmation or decline message to the customer.
  - a The acquirer submits the settlement request to the issuer for funds.
  - b The issuer supplies the funds to the acquirer for the authorized transactions.

## Additional CyberSource Services

Refer to [Credit Card Services Using the Simple Order API](#) for information on how to request these follow-on services.

**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.
Authorization Reversal	A follow-on service that uses the request ID returned from the previous authorization. An authorization 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.

## Transaction Endpoints

CAS (test transactions):

[https://ics2wstest.ic3.com/commerce/1.x/transactionProcessor/CyberSourceTransaction\\_1.104.xsd](https://ics2wstest.ic3.com/commerce/1.x/transactionProcessor/CyberSourceTransaction_1.104.xsd)

Production (live transactions):

[https://ics2ws.ic3.com/commerce/1.x/transactionProcessor/CyberSourceTransaction\\_1.104.xsd](https://ics2ws.ic3.com/commerce/1.x/transactionProcessor/CyberSourceTransaction_1.104.xsd)

## Merchant-Initiated Transactions

### Service:

- Authorization

### Processor:

- Chase Payment Solutions—the only scenarios supported on Chase Paymentech Solutions are reauthorizations and unscheduled card-on-file transactions.
- CyberSource through VisaNet

Most authorizations are initiated by a cardholder in person, on the phone, or on a web site. A *merchant-initiated transaction* (MIT) is an authorization that you initiate when the cardholder is not present.

## Terminology

**Table 3 Terminology for Merchant-Initiated Transactions**

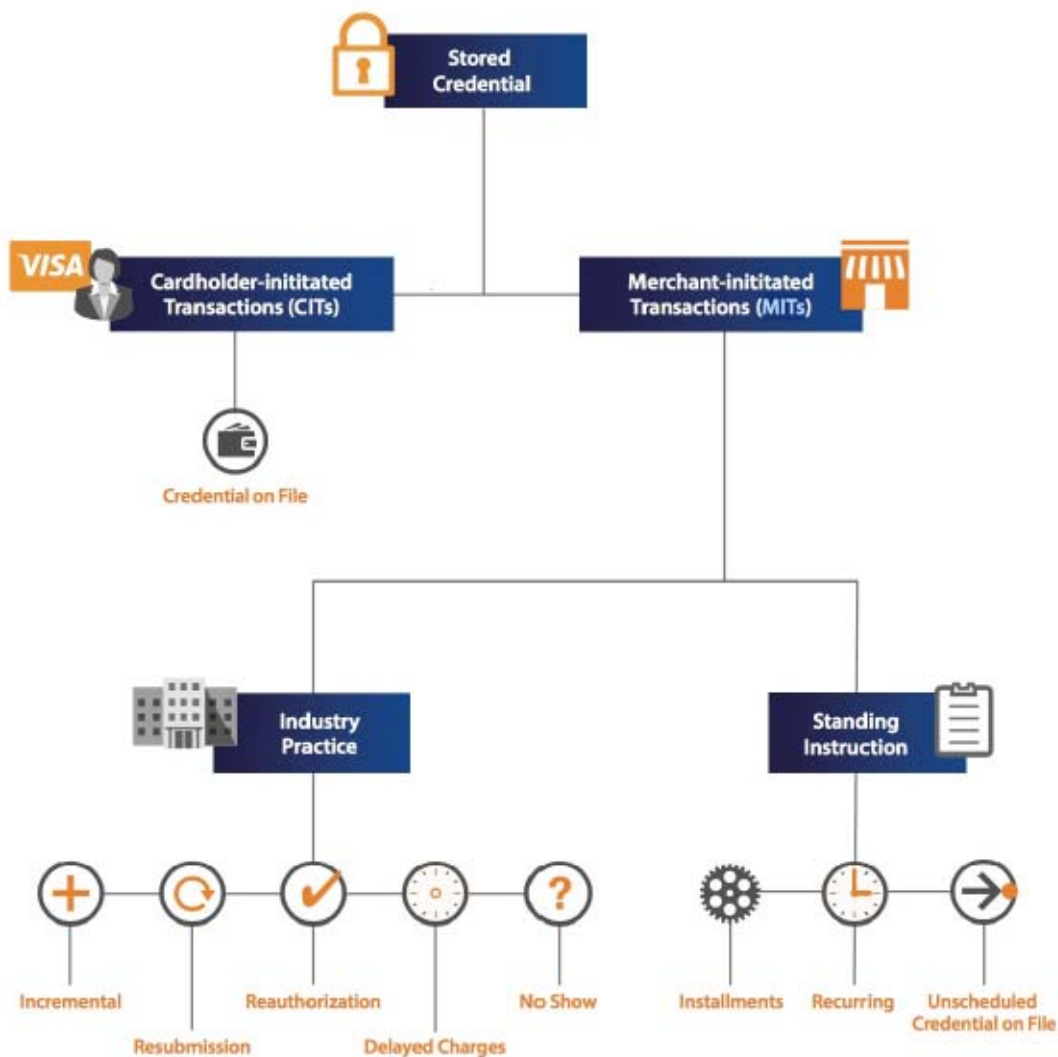
Term	Description
Cardholder-initiated transaction (CIT)	<p>Transaction that uses payment information provided by the cardholder. A CIT can be any of the following kinds of transactions:</p> <ul style="list-style-type: none"> <li>■ Card present: cardholder goes to a brick-and-mortar store in person to make a purchase and provides payment information in the store.</li> <li>■ COF: cardholder orders an item online and instructs you to use the payment information that is saved in your system.</li> <li>■ E-commerce: cardholder orders an item online and provides payment information during checkout.</li> <li>■ MOTO: cardholder orders an item over the telephone and provides payment information to the person who is taking the order.</li> </ul>
Card-on-file or credential-on-file (COF) transaction	Transaction that uses payment information that you saved in your system.



## Overview

Figure 1 illustrates the relationships between stored credentials, CITs, and MITs.

**Figure 1**     **Stored Credentials and Merchant-Initiated Transactions**



There are two main types of MITs:

- An *industry practice* transaction: a one-time MIT that derives payment information from a CIT.
- A *standing instruction*: one transaction in a series of repeated transactions or is a one-time, unscheduled transaction that uses COF payment information.

## Descriptions

- Account top-up—is the result of instructions between you and the cardholder to charge a specific or variable amount at specified or variable intervals. An account top-up is an unscheduled COF transaction.
- Delayed charge—is associated with an agreement between you and the cardholder for services rendered. Delayed charges are typical for lodging transactions and auto rental transactions.
- Final authorization—occurs when an estimated authorization was performed and you need to authorize the final amount.
- Incremental authorization—is a continuation of a purchase when the originally approved amount is modified to accommodate additional services. Incremental authorizations are typical for lodging transactions and auto rental transactions.
- Installment payment—is the result of instructions governed by a contract between you and a cardholder. The instructions enable you to charge a specific amount at specified intervals.
- No-show transaction—occurs when you and a cardholder have an agreement for a purchase, but the cardholder does not meet the terms of the agreement. No-show transactions are typically used in hotels and motels for a single-night stay.
- Reauthorization for split shipment—a split shipment occurs when goods are not available for shipment when the cardholder purchases them. When the goods become available to ship, a new authorization is performed, either by you or by CyberSource, to make sure that the cardholder's funds are still available. The reauthorization is performed in one of the following scenarios:
  - Before requesting a capture, you request an authorization using the saved cardholder credentials.
  - You use the CyberSource split shipments feature.



### Note

The CyberSource split shipments feature is not available on Chase Paymentech Solutions.

---

- Recurring payment—is the result of instructions governed by a contract between you and a cardholder. The instructions enable you to charge a specific or variable amount at specified intervals.
- Resubmission—occurs when a cardholder-initiated purchase occurred, but you could not obtain an authorization at that time. A resubmission is valid only when the original authorization was declined for insufficient funds and only for a limited number of days after the original purchase.

## Scenarios

### Delayed Charge

A delayed charge is associated with an agreement between you and the cardholder for services rendered. Delayed charges are typically used in lodging, cruise line, and auto rental environments to perform a supplemental charge after original services are rendered.

#### To create a delayed charge authorization request:

---

- Step 1** Include the following required fields in the authorization request:
- `subsequentAuth`—set the value for this field to `true`.
  - `subsequentAuthReason`—set the value for this field to 2.
  - `subsequentAuthTransactionID`—set the value for this field to the network transaction identifier.
- Step 2** If the payment information is COF information, include the following field in the authorization request:
- `subsequentAuthStoredCredential`—set the value for this field to `true`.
- 

### Installment Payment

An installment payment is a COF transaction. A series of installment payments consists of multiple transactions that you bill to a cardholder over a period of time agreed to by you and the cardholder for a single purchase of goods or services. The agreement enables you to charge a specific amount at specified intervals.

#### To create an installment payment authorization request:

---

- Step 1** Cardholder consents to terms and establishes service or obtains goods.
- Step 2** You charge the first installment payment as a CIT. Include the following field in the authorization request:
- `subsequentAuthFirst`—set the value for this field to `true`.

**Step 3** You charge subsequent installment payments on a regular basis. Include the following fields in each authorization request:

- `ccAuthService_commerceIndicator`—set the value for this field to `install`.
  - `subsequentAuthTransactionID`—set the value for this field to the network transaction identifier.
- 

## No-Show Transaction

A no-show transaction occurs when you and a cardholder have an agreement for a purchase, but the cardholder does not meet the terms of the agreement. No-show transactions are typically used in hotels and motels for a single-night stay.

### To create a no-show transaction authorization request:

---

**Step 1** Include the following required fields in the authorization request:

- `subsequentAuth`—set the value for this field to `true`.
- `subsequentAuthReason`—set the value for this field to 4.
- `subsequentAuthTransactionID`—set the value for this field to the network transaction identifier.

**Step 2** If the payment information is COF information, include the following field in the authorization request:

- `subsequentAuthStoredCredential`—set the value for this field to `true`.
-

## Reauthorization

A reauthorization is a purchase made after the original purchase and can reflect a number of specific conditions. Common scenarios include delayed shipments, split shipments, extended stays, and extended rentals.

### To create a reauthorization request:

---

- Step 1** Include the following required fields in the authorization request:
- `subsequentAuth`—set the value for this field to `true`.
  - `subsequentAuthReason`—set the value for this field to 3.
  - `subsequentAuthTransactionID`—set the value for this field to the network transaction identifier.
- Step 2** If the payment information is COF information, include the following field in the authorization request:
- `subsequentAuthStoredCredential`—set the value for this field to `true`.
- 

## Recurring Payment

A recurring payment is a COF transaction. A series of recurring payments consists of multiple transactions that you bill to a cardholder at fixed, regular intervals not to exceed one year between transactions. The series of recurring payments is the result of an agreement between you and the cardholder.

### To create a recurring payment authorization request:

---

- Step 1** Cardholder consents to terms and establishes service or obtains goods.
- Step 2** You charge the first recurring payment as a CIT. Include the following field in the authorization request:
- `subsequentAuthFirst`—set the value for this field to `true`.

**Step 3** You charge subsequent recurring payments on a regular basis. Include the following fields in each authorization request:

- `ccAuthService_commerceIndicator`—set the value for this field to `recurring`.
  - `subsequentAuthTransactionID`—set the value for this field to the network transaction identifier.
- 

## Resubmission

A resubmission occurs when a cardholder-initiated purchase occurred, but you could not obtain an authorization at that time. A resubmission is valid only when the original authorization was declined for insufficient funds and only for a limited number of days after the original purchase.

### To create a resubmission authorization request:

---

**Step 1** Include the following required fields in the authorization request:

- `subsequentAuth`—set the value for this field to `true`.
- `subsequentAuthReason`—set the value for this field to `1`.
- `subsequentAuthTransactionID`—set the value for this field to the network transaction identifier.

**Step 2** If the payment information is COF information, include the following field in the authorization request:

- `subsequentAuthStoredCredential`—set the value for this field to `true`.
-

## Unscheduled COF Transaction

An unscheduled COF transaction uses stored payment information for a fixed or variable amount that does not occur on a scheduled or regular basis.

### To create an unscheduled COF transaction authorization request:

---

- Step 1** Cardholder consents to terms and establishes service or obtains goods.
- Step 2** You charge the first payment. Include the following field in the authorization request:
- `subsequentAuthFirst`—set the value for this field to `true`.
- Step 3** You charge subsequent payments. Include the following fields in each authorization request:
- `subsequentAuth`—set the value for this field to `true`.
  - `subsequentAuthTransactionID`—set the value for this field to the network transaction identifier.
- 

## API Field Descriptions

For descriptions of the fields in the preceding scenarios, see ["API Fields," page 60](#).

# Using the Google API for Android Pay

**Important**

The Google API is not supported by CyberSource Customer Support. All examples in this chapter are for guidance only. Google developer resources can be accessed here:

<http://developer.android.com/support.html>

## 1. Obtain Your Client ID and Credentials

To access the Google API for Android Pay, you must obtain a client ID for OAuth 2.0 authorization in the Google Developers Console. The client ID is generated automatically when you register your app. You need the SHA1 fingerprint in your developer's key to generate a client ID.

To access the Google API for Android Pay, configure your Android Pay [client ID and credentials](#).

**Important**

You must submit your Android Pay enabled APK for [review](#) to ensure that it adheres to branding guidelines and it has tested all user payment flows.

## 2. Generate a Public/Private Key

You can generate a public/private key by choosing one of two methods:

- Merchant decryption—decrypt the encrypted payment data to retrieve the payment network token and include it in the authorization request to CyberSource.
- CyberSource decryption (see "[Option 2: CyberSource Decryption](#)," page 27)—include the encrypted payment data in the authorization request to CyberSource.



## Option 1: Merchant Decryption

This option is for merchants who decrypt the **encryptedMessage** key sent back as part of the Full Wallet request. When requesting the Masked Wallet (see ["4. Request Masked Wallet," page 29](#)) you must include a public key as an encoded base64 string.

### To generate a public/private key:

---

**Step 1** Generate the private key.

#### Example 1 Generating the Private Key

---

```
$ openssl ecparam -name prime256v1 -genkey -noout -out
merchant-key.pem
```

---

**Step 2** Generate the private and public key in hex form.

#### Example 2 Generating the Private and Public Key

---

```
$ openssl ec -in merchant-key.pem -pubout -text -noout
read EC key
PrivateKey:(256 bit)
priv:
  08:f4:ae:16:be:22:48:86:90:a6:b8:e3:72:11:cf:
  c8:3b:b6:35:71:5e:d2:f0:c1:a1:3a:4f:91:86:8a:
  f5:d7
pub:
  04:e7:68:5c:ff:bd:02:ae:3b:dd:29:c6:c2:0d:c9:
  53:56:a2:36:9b:1d:f6:f1:f6:a2:09:ea:e0:fb:43:
  b6:52:c6:6b:72:a3:f1:33:df:fa:36:90:34:fc:83:
  4a:48:77:25:48:62:4b:42:b2:ae:b9:56:84:08:0d:
  64:a1:d8:17:66
```

---

**Step 3** Decode the hex-encoded public key.

#### Example 3 Decoding the Public Key

---

```
cat <<EOF | xxd -r -p | base64
  04:e7:68:5c:ff:bd:02:ae:3b:dd:29:c6:c2:0d:c9:
  53:56:a2:36:9b:1d:f6:f1:f6:a2:09:ea:e0:fb:43:
  b6:52:c6:6b:72:a3:f1:33:df:fa:36:90:34:fc:83:
  4a:48:77:25:48:62:4b:42:b2:ae:b9:56:84:08:0d:
  64:a1:d8:17:66
```

```
EOF
```

---

- Step 4** Convert the hex-encoded public key to a base64 string and include the value in the **paymentMethodTokenizationParameters** parameter as part of the Masked Wallet request (see ["4. Request Masked Wallet," page 29](#)).

**Example 4 Google API Request**

---

```
.setPaymentMethodTokenizationParameters(PaymentMethodTokenizationParameters.newBuilder()  
    .setPaymentMethodTokenizationType(PaymentMethodTokenizationType.NETWORK_TOKEN)  
    .addParameter("publicKey", "[ENTER_YOUR_BASE64_ENCODED_PUBLIC_KEY_HERE]")  
    .build());
```

---

Next step: ["4. Request Masked Wallet," page 29](#).

---

## Option 2: CyberSource Decryption



This option is for merchants who do not decrypt the **encryptedMessage** key sent back as part of the Full Wallet request.

When requesting the Masked Wallet (see "4. Request Masked Wallet," page 29), you must include an elliptic curve point.

Choose one of two methods to generate the public key and elliptic curve point:

- Use the Business Center to generate and download the public key and elliptic curve point.
- Use the Business Center to generate and download the public key, and then create the elliptic curve point from the generated public key.

### To generate the public key and the elliptic curve point:

**Step 1** Log in to the Business Center:


- Live transactions: <https://ebc.cybersource.com>
- Test transactions: <https://ebctest.cybersource.com>

**Step 2** In the left navigation panel, choose **Account Management > Digital Payment Solutions**. The Digital Payment Solutions page displays the status of the Android Pay digital solution for your account:

- Signup: your account is eligible for integrating Android Pay.
- Enabled: Android Pay is already integrated into your account.
- Not Available: Android Pay is not available for your account.

**Step 3** Click **Enabled** to display the following page.

**Figure 2** Generating the Public Key and Elliptical Curve (EC) Point



Android Pay enables you to process payments from your mobile Android app. Provide your customers with a secure, seamless shopping experience when they use your Android app on their mobile devices.

[Learn more about the Android Pay integration with CyberSource here.](#)

**Step 1: Generate Public Key and EC Point**  
When processing Android Pay transactions, Google will send your app a unique Base64 encrypted message via the Android Pay API. Google requires an identifier, public key and Elliptic Curve Point in order to encrypt sensitive payment credentials. For further details see the [Google API reference](#).

Generated	Android Pay ID	Public Key	EC Point
08/03/2016 03:41:04 PM	2640632	<a href="#">Download</a>	<a href="#">Download</a>

[Generate Public Key and EC Point](#)

**Step 2: Generate a Transaction Security Key**  
This step is required only if you are using the SDK. It is not needed if you are using the API. When you are using the SDK, each transaction request that originates from your android application must include a unique signature.

You already have at least one CyberSource SOAP Toolkit API transaction security key.  
If you do not want to reuse an existing key, [generate a transaction security key here](#).

**IMPORTANT:** Android Pay uses payment network tokenization. To use the Android Pay, your processor must support payment network tokenization.

**Step 4** Click **Generate Public Key and EC Point**.

**Step 5** Download the public key and the EC point values.



The EC point value is required for the Masked Wallet request (see "4. Request Masked Wallet," page 29). The public key value is required for the CyberSource Decryption method (see "Option 2: CyberSource Decryption," page 31).

Next step: "4. Request Masked Wallet," page 29.

## 3. Enable User

---

Use the `isReadyToPay` method within the Google API to verify that the user has the Android pay app installed and is ready to pay. If the value `true` is returned, display the Android Pay button. If the value `false` is returned, display other checkout options and text to notify the user to install the Android Pay app.

## 4. Request Masked Wallet

---

Use the Google API to create a [Masked Wallet](#) request to retrieve the Masked Wallet information.

Include the elliptic curve point string (see ["Option 2: CyberSource Decryption," page 27](#)) in the `paymentMethodTokenizationParameters` parameter in the Masked Wallet request.

### Example 5 Google API Request

---

```
.setPaymentMethodTokenizationParameters(PaymentMethodTokenizationParameters.newBuilder()  
    .setPaymentMethodTokenizationType(PaymentMethodTokenizationType.NETWORK_TOKEN)  
    .addParameter("publicKey", <ENTER_YOUR_ELLIPTIC_CURVE_POINT_STRING>)  
    .build());
```

---

The Masked Wallet response contains customer details and masked card information. All of this information is displayed to the customer in your UI. The customer can change payment method and address or confirm the payment.

When the customer confirms the payment, you create a Full Wallet request (see ["5. Request Full Wallet," page 30](#)).

## 5. Request Full Wallet

---

### Option 1: Merchant Decryption

Use the Google API to create a [Full Wallet](#) request to access the customer's wallet and request their payment credentials.

When you receive the Full Wallet response, access the details of the customer's payment credentials by retrieving their payment token.

#### Example 6 Retrieving the Payment Token

---

```
PaymentMethodToken token = fullWallet.getPaymentMethodToken();
String tokenJSON = token.getToken();
```

---

The payment token is a UTF-8 encoded serialized JSON dictionary:

#### Example 7 Payment Token

---

```
{
  "encryptedMessage": "ZW5jcnlwdGVkTWVzc2FnZQ==",
  "ephemeralPublicKey": "ZXBoZWllcmFsUHVibGljS2V5",
  "signature": "c2lnbmF0dXJl"
}
```

---

The payment token contains the following:

- `encryptedMessage`—the encrypted message. String (base64).



You must decrypt this key (see ["6. Decrypt Encrypted Message Key," page 32](#)).

---

- `ephemeralPublicKey`—the public key associated with the private key that was used to encrypt the message. String (base64).
  - `signature`—the MAC key of the encrypted message.
- 

Next step: ["6. Decrypt Encrypted Message Key," page 32](#).

---

## Option 2: CyberSource Decryption

Use the Google API to create a [Full Wallet](#) request to access the customer's wallet and request their payment credentials.

When you receive the Full Wallet response, you must follow these steps to include the encrypted payment data in the CyberSource authorization request.

### To include the encrypted payment data:

---

**Step 1** Base64-encode the response from Google.

#### Example 8 Base64 Encoding the Google Response

---

```
Base64.encode( mFullWallet.getPaymentMethodToken().getToken().getBytes(), Base64.NO_WRAP)
.toString();
```

---

**Step 2** Create an SHA-256 hash of the public key that was generated within the Business Center (see ["Option 2: CyberSource Decryption," page 27](#)).

#### Example 9 SHA-256 Hash of the Public Key

---

```
MessageDigest digest = MessageDigest.getInstance("SHA-256");

byte[] pubKeyBytes = Base64.decode( ENTER_YOUR_PUBLIC_KEY, Base64.NO_WRAP );

byte[] publicKeyHash = digest.digest(pubKeyBytes);

String publicKeyHashString = new String(Base64.encode(publicKeyHash, Base64.NO_WRAP));
```

---

**Step 3** Create a JSON object containing the base64-encoded response from Google ([Step 1](#)), the public key hash ([Step 2](#)), and the version information.

#### Example 10 Creating the JSON Object

---

```
JSONObject jsonObject = new JSONObject();

jsonObject.put("publicKeyHash", getPublicKeyHash());

jsonObject.put("version", "1.0");

jsonObject.put("data", androidPayBlob);
```

---

**Step 4** Base64-encode the complete JSON object to a string.

#### Example 11 Base64 Encoding the JSON Object

---

```
Base64.encode(jsonObject.toString().getBytes(), Base64.NO_WRAP).toString();
```

---

**Step 5** Include the string in the **encryptedPayment\_data** request field to CyberSource.

---

## 6. Decrypt Encrypted Message Key

---



This step applies only to the Merchant Decryption option.

---

For reference you can download the Bouncy Castle [sample code](#).

### To decrypt the encryptedMessage key:

---

**Step 1** Use your private key and public key to derive a 256-bit shared key using ECIESKEM, as defined in [ISO 180332](#).

Use the following parameters:

- Elliptic curve—NIST P256 (prime256v1).
- **CheckMode**, **OldCofactorMode**, **SingleHashMode**, and **CofactorMode** are 0.
- Encoding function—uncompressed point format.
- Key derivation function—the HKDF key derivation function with SHA-256, as described in:

<https://tools.ietf.org/html/rfc5869>



Salt should not be provided (according to the RFC it should be equivalent to a salt of 32 zeroed bytes), and information should be **Android** encoded in ASCII.

---

**Step 2** Split the generated key into two 128-bit keys: a symmetric encryption key (**symmetricEncryptionKey**) and a MAC key (**macKey**).



**Step 3** Verify that the signature field contains a valid MAC key for the encrypted message key.

To generate the MAC key, use HMAC ([RFC 5869](#)) with hash function SHA-256 and the **macKey** key obtained in [Step 2](#).



**Note**

Use a constant time array comparison to avoid timing attacks.

**Step 4** Decrypt the encrypted message key using AES128 CTR mode with a zero IV, no padding, and decrypt the symmetric encryption key derived in [Step 2](#).

## Decrypted Payment Credential

The result of decrypting the encrypted message key is a UTF-8-encoded JSON object that includes the keys shown in [Example 12](#) and described in [Table 4](#).

### Example 12 Decrypted Payment Credential

```
{
  "dpan": "4444444444444444",
  "expirationMonth": 10,
  "expirationYear": 2015,
  "authMethod": "3DS",
  "3dsCryptogram": "AAAAAA...",
  "3dsEciIndicator": "eci indicator"
}
```

**Table 4** Decrypted Payment Credential

Key	Type	Description
dpan	string (digits only)	The payment network token enables you to request a CyberSource authorization with a token instead of a primary account number (PAN).
expirationMonth	number	The expiration month of the payment network token (1=jan, 2=feb, etc.).
expirationYear	number	The 4-digit expiration year of the payment network token.
authMethod	string	Returns 3DS. Additional methods may be added.
3dsCryptogram	string	3D Secure cryptogram.
3dsEciIndicator	string (optional)	ECI Indicator.

## 7. Request CyberSource Authorization

---

For Merchant Decryption examples, see ["Option 1: Merchant Decryption,"](#) page 35.

For CyberSource Decryption examples, see ["Option 2: CyberSource Decryption,"](#) page 49.

# Requesting the Authorization Service

## Option 1: Merchant Decryption

---

### Visa Transaction

To request an authorization for a Visa transaction:

---



**Note**

See ["API Request Fields," page 62](#), and ["API Reply Fields," page 73](#) 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 **ccAuthService\_xid** field to the 3D Secure cryptogram of the payment network token.
- Step 5** Set the **paymentNetworkToken\_transactionType** field to 1.
- Step 6** Set the **ccAuthService\_commerceIndicator** field to `internet`.
- Step 7** Set the **paymentSolution** field to 006.

**Example 13 Authorization Request (Visa)**


---

```

<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.121">
  <merchantID>demomerchant</merchantID>
  <merchantReferenceCode>demorefnum</merchantReferenceCode>
  <billTo>
    <firstName>James</firstName>
    <lastName>Smith</lastName>
    <street1>1295 Charleston Road</street1>
    <city>Test City</city>
    <state>CA</state>
    <postalCode>99999</postalCode>
    <country>US</country>
    <email>demo@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>
    <xid>1234567890987654321ABCDEFabcdefABCDEF123</xid>
  </ccAuthService>
  <paymentNetworkToken>
    <transactionType>1</transactionType>
  </paymentNetworkToken>
  <paymentSolution>006</paymentSolution>
</requestMessage>

```

---

**Example 14 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 ["API Request Fields," page 62](#), and ["API Reply Fields," page 73](#), 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 *spa*.
- Step 7** Set the **paymentSolution** field to 006.

**Example 15 Authorization Request (Mastercard)**


---

```

<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.121">
  <merchantID>demomerchant</merchantID>
  <merchantReferenceCode>demorefnum</merchantReferenceCode>
  <billTo>
    <firstName>James</firstName>
    <lastName>Smith</lastName>
    <street1>1295 Charleston Road</street1>
    <city>Test City</city>
    <state>CA</state>
    <postalCode>99999</postalCode>
    <country>US</country>
    <email>demo@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>ABCDEFabcdefABCDscdef0987654321</authenticationData>
    <collectionIndicator>2</collectionIndicator>
  </ucaf>
  <ccAuthService run="true">
    <commerceIndicator>spa</commerceIndicator>
  </ccAuthService>
  <paymentNetworkToken>
    <transactionType>1</transactionType>
  </paymentNetworkToken>
  <paymentSolution>006</paymentSolution>
</requestMessage>

```

---

**Example 16 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:

---



See ["API Request Fields," page 62](#), and ["API Reply Fields," page 73](#), 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.



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 `aesk`.
- Step 6** Set the **paymentSolution** field to 006.



**Example 17 Authorization Request (American Express)**


---

```

<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.121">
  <merchantID>demomerchant</merchantID>
  <merchantReferenceCode>demorefnum</merchantReferenceCode>
  <billTo>
    <firstName>James</firstName>
    <lastName>Smith</lastName>
    <street1>1295 Charleston Road</street1>
    <city>Test City</city>
    <state>CA</state>
    <postalCode>99999</postalCode>
    <country>US</country>
    <email>demo@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>
    <xid>1234567890987654321ABCDEFabcdefABCDEF123</xid>
  </ccAuthService>
  <paymentNetworkToken>
    <transactionType>1</transactionType>
  </paymentNetworkToken>
  <paymentSolution>006</paymentSolution>
</requestMessage>

```

---

**Example 18 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 ["API Request Fields," page 62](#), and ["API Reply Fields," page 73](#), 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 `dipb`.
- Step 6** Set the **paymentSolution** field to 006.

**Example 19 Authorization Request (Discover)**


---

```

<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.121">
  <merchantID>demomerchant</merchantID>
  <merchantReferenceCode>demorefnum</merchantReferenceCode>
  <billTo>
    <firstName>James</firstName>
    <lastName>Smith</lastName>
    <street1>1295 Charleston Road</street1>
    <city>Test City</city>
    <state>CA</state>
    <postalCode>99999</postalCode>
    <country>US</country>
    <email>demo@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>006</paymentSolution>
</requestMessage>

```

---

**Example 20 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>2017-01-03T20:53:54Z</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 ["API Request Fields," page 62](#), and ["API Reply Fields," page 73](#), 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 21 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 22 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 23 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=3566002020360505
card_expirationYear=2020
card_cvnNumber=123
cardType=001
ccAuthService_cavv=ABCDEFabcdefABCDEFabcdef0987654321234567
ccAuthService_cavv=5
paymentNetworkToken_transactionType=1
paymentSolution=001

```

---

**Example 24 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 "API Request Fields," page 62, and "API Reply Fields," page 73, for detailed field descriptions.

**Step 1** Set the **encryptedPayment\_data** field to the value of the **encryptedMessage** field that was returned in the Full Wallet response. See "5. Request Full Wallet," page 30.

**Step 2** Set the **paymentSolution** field to 006.

#### Example 25 Authorization Request (Visa)

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.121">
  <merchantID>demomerchant</merchantID>
  <merchantReferenceCode>demorefnum</merchantReferenceCode>
  <billTo>
    <firstName>James</firstName>
    <lastName>Smith</lastName>
    <street1>1295 Charleston Road</street1>
    <city>Test City</city>
    <state>CA</state>
    <postalCode>99999</postalCode>
    <country>US</country>
    <email>demo@example.com</email>
  </billTo>
  <purchaseTotals>
    <currency>USD</currency>
    <grandTotalAmount>5.00</grandTotalAmount>
  </purchaseTotals>
  <encryptedPayment>
    <data>ABCDEFabcdefABCDEFabcdef0987654321234567</data>
  </encryptedPayment>
  <card>
    <cardType>001</cardType>
  </card>
  <ccAuthService run="true"/>
  <paymentSolution>006</paymentSolution>
</requestMessage>
```

**Example 26 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:token>
    <c:prefix>294672</c:prefix>
    <c:suffix>4397</c:suffix>
    <c:expirationMonth>08</c:expirationMonth>
    <c:expirationYear>2021</c:expirationYear>
  </c:token>
</c:replyMessage>

```

---



---

## Mastercard Transaction

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



#### Note

See "API Request Fields," page 62, and "API Reply Fields," page 73, for detailed field descriptions.

**Step 1** Set the **encryptedPayment\_data** field to the value of the **encryptedMessage** field that was returned in the Full Wallet response. See "5. Request Full Wallet," page 30.

**Step 2** Set the **paymentSolution** field to 006.

#### Example 27 Authorization Request (Mastercard)

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.121">
  <merchantID>demomerchant</merchantID>
  <merchantReferenceCode>demorefnum</merchantReferenceCode>
  <billTo>
    <firstName>James</firstName>
    <lastName>Smith</lastName>
    <street1>1295 Charleston Road</street1>
    <city>Test City</city>
    <state>CA</state>
    <postalCode>99999</postalCode>
    <country>US</country>
    <email>demo@example.com</email>
  </billTo>
  <purchaseTotals>
    <currency>USD</currency>
    <grandTotalAmount>5.00</grandTotalAmount>
  </purchaseTotals>
  <encryptedPayment>
    <data>ABCDEFabcdefABCDEFabcdef0987654321234567</data>
  </encryptedPayment>
  <card>
    <cardType>002</cardType>
  </card>
  <ccAuthService run="true"/>
  <paymentSolution>006</paymentSolution>
</requestMessage>
```

**Example 28 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:token>
    <c:prefix>128945</c:prefix>
    <c:suffix>2398</c:suffix>
    <c:expirationMonth>08</c:expirationMonth>
    <c:expirationYear>2021</c:expirationYear>
  </c:token>
</c:replyMessage>

```

---

## American Express Transaction

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



See "API Request Fields," page 62, and "API Reply Fields," page 73, for detailed field descriptions.

#### Note

**Step 1** Set the **encryptedPayment\_data** field to the value of the **encryptedMessage** field that was returned in the Full Wallet response. See "5. Request Full Wallet," page 30.

**Step 2** Set the **paymentSolution** field to 006.

#### Example 29 Authorization Request (American Express)

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.121">
  <merchantID>demomerchant</merchantID>
  <merchantReferenceCode>demorefnum</merchantReferenceCode>
  <billTo>
    <firstName>James</firstName>
    <lastName>Smith</lastName>
    <street1>1295 Charleston Road</street1>
    <city>Test City</city>
    <state>CA</state>
    <postalCode>99999</postalCode>
    <country>US</country>
    <email>demo@example.com</email>
  </billTo>
  <purchaseTotals>
    <currency>USD</currency>
    <grandTotalAmount>5.00</grandTotalAmount>
  </purchaseTotals>
  <encryptedPayment>
    <data>ABCDEFabcdefABCDEFabcdef0987654321234567</data>
  </encryptedPayment>
  <card>
    <cardType>003</cardType>
  </card>
  <ccAuthService run="true"/>
  <paymentSolution>006</paymentSolution>
</requestMessage>
```

**Example 30 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:token>
    <c:prefix>593056</c:prefix>
    <c:suffix>0842</c:suffix>
    <c:expirationMonth>08</c:expirationMonth>
    <c:expirationYear>2021</c:expirationYear>
  </c:token>
</c:replyMessage>

```

---



---

## Discover Transaction

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



#### Note

See "API Request Fields," page 62, and "API Reply Fields," page 73, for detailed field descriptions.

- Step 1** Set the **encryptedPayment\_data** field to value of the **encryptedMessage** field that was returned in the Full Wallet response. See "5. Request Full Wallet," page 30.
- Step 2** Set the **paymentSolution** field to 006.

### Example 31 Authorization Request (Discover)

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.121">
  <merchantID>demomerchant</merchantID>
  <merchantReferenceCode>demorefnum</merchantReferenceCode>
  <billTo>
    <firstName>James</firstName>
    <lastName>Smith</lastName>
    <street1>1295 Charleston Road</street1>
    <city>Test City</city>
    <state>CA</state>
    <postalCode>99999</postalCode>
    <country>US</country>
    <email>demo@example.com</email>
  </billTo>
  <purchaseTotals>
    <currency>USD</currency>
    <grandTotalAmount>5.00</grandTotalAmount>
  </purchaseTotals>
  <encryptedPayment>
    <data>ABCDEFabcdefABCDEFabcdef0987654321234567</data>
  </encryptedPayment>
  <card>
    <cardType>004</cardType>
  </card>
  <ccAuthService run="true"/>
  <paymentSolution>006</paymentSolution>
</requestMessage>
```

**Example 32 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>2017-01-03T20:53:54Z</c:authorizedDateTime>
    <c:processorResponse>100</c:processorResponse>
    <c:reconciliationID>11267051CGJSMQDC</c:reconciliationID>
  </c:ccAuthReply>
  <c:token>
    <c:prefix>593056</c:prefix>
    <c:suffix>0842</c:suffix>
    <c:expirationMonth>08</c:expirationMonth>
    <c:expirationYear>2021</c:expirationYear>
  </c:token>
</c:replyMessage>

```

---

## JCB Transaction

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

**Note**

See ["API Request Fields," page 62](#), and ["API Reply Fields," page 73](#), 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 `Rk1EPUNPTU1PTi5BUFBMRS5JTtkFQUC5QQVlNRU5U`.
  - Step 3** Set the **paymentSolution** field to `001`.



**Example 33 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>RklEPUNPTU1PTi5BUFBMRS5JTkFQUC5QQVlNRU5U</descriptor>
    <data>ABCDEFabcdefABCDEFabcdef0987654321234567</data>
    <encoding>Base64</encoding>
  </encryptedPayment>
  <card>
    <cardType>001</cardType>
  </card>
  <ccAuthService run="true"/>
    <paymentSolution>001</paymentSolution>
</requestMessage>

```

---

**Example 34 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>

```

---

## Recurring Payments

---

### To request a recurring payment:

---

**Step 1** In the first authorization request:

- FDC Nashville Global—for Mastercard and American Express transactions, include the following fields and values:
  - `ccAuthService_commerceIndicator=recurring`
  - `ccAuthService_firstRecurringPayment=TRUE`
- OmniPay Direct—for all card types include **`ccAuthService_firstRecurringPayment=TRUE`**

- All other supported processors—do not include the following fields in the request:
  - `ccAuthService_commerceIndicator`
  - `ccAuthService_firstRecurringPayment`

**Step 2** In each subsequent authorization request, include the following fields and values:

- `ccAuthService_commerceIndicator=recurring`
- `paymentNetworkToken_transactionType=1`

With processors that support merchant-initiated transactions, your authorization request must include subsequent authorization fields as described in "[Merchant-Initiated Transactions](#)," page 16.

---

# 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 5** 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, if 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">
```

For the name-value pair field names, 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 35    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 [Relaxed Requirements for Address Data and Expiration Date page](#).

## API Request Fields



### Note

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

**Table 6 Request Fields**

Field	Description	Used By: Required (R) or Optional (O)	Data Type (Length)
billTo_city	City of the billing 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 (50)
billTo_country	Country of the billing address. Use the two-character <i>ISO Standard Country Codes</i> . <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)
billTo_email	Customer's email 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 (255)
billTo_firstName	Customer's first 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_ipAddress	Customer's IP address.	ccAuthService (O)	String (15)
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)

<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.

<sup>2</sup> 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 61. **Important** It is your responsibility to determine whether a field is required for the transaction you are requesting.

**Table 6 Request Fields (Continued)**

Field	Description	Used By: Required (R) or Optional (O)	Data Type (Length)
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	<p>Postal code for the billing address. The postal code must consist of 5 to 9 digits.</p> <p>When the billing country is the U.S., the 9-digit postal code must follow this format: [5 digits][dash][4 digits]</p> <p><b>Example</b> 12345-6789</p> <p>When the billing country is Canada, the 6-digit postal code must follow this format: [alpha][numeric][alpha][space] [numeric][alpha][numeric]</p> <p><b>Example</b> A1B 2C3</p> <p><b>Important</b> It is your responsibility to determine whether a field is required for the transaction you are requesting.</p>	ccAuthService (R) <sup>2</sup>	String (9)
billTo_state	<p>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>.</p> <p>It is your responsibility to determine whether a field is required for the transaction you are requesting.</p>	ccAuthService (R) <sup>2</sup>	String (2)
billTo_street1	<p>First line of the billing street address.</p> <p><b>Important</b> It is your responsibility to determine whether a field is required for the transaction you are requesting.</p>	ccAuthService (R) <sup>2</sup>	String (60)
billTo_street2	<p>Additional address information.</p> <p><b>Example</b> Attention: Accounts Payable</p>	ccAuthService (O)	String (60)
<p><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.</p> <p><sup>2</sup> 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 61. <b>Important</b> It is your responsibility to determine whether a field is required for the transaction you are requesting.</p>			

**Table 6 Request Fields (Continued)**

Field	Description	Used By: Required (R) or Optional (O)	Data Type (Length)
card_accountNumber	The payment network token value.  This value is obtained by decrypting the customer's encrypted payment data (see <a href="#">"6. Decrypt Encrypted Message Key," page 32</a> ). Populate this field with the decrypted dpan 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 61. **Important** It is your responsibility to determine whether a field is required for the transaction you are requesting.



Table 6 Request Fields (Continued)

Field	Description	Used By: Required (R) or Optional (O)	Data Type (Length)
<code>ccAuthService_cavv</code>	<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:</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)
<code>ccAuthService_commerceIndicator</code>	<p>For a payment network tokenization transaction.</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>internet</code>: Visa card type</li> <li><code>dipb</code>: Discover card type</li> </ul> <p><b>Important</b> For Visa in-app transactions, the <code>internet</code> value is mapped to the Visa ECI value 7.</p> <p><b>Note</b> For recurring payments, set this field to a value from the preceding list for the first payment and set this field to <code>recurring</code> for subsequent payments.</p>	ccAuthService (R for merchant decryption, O for CyberSource decryption)	String (20)
<code>ccAuthService_eciRaw</code>	Raw electronic commerce indicator (ECI).	ccAuthService	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 61. **Important** It is your responsibility to determine whether a field is required for the transaction you are requesting.

**Table 6 Request Fields (Continued)**

Field	Description	Used By: Required (R) or Optional (O)	Data Type (Length)
ccAuthService_run	Whether to include <b>ccAuthService</b> in your request. Possible values: <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	<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.  <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 (see <a href="#">"American Express Transaction," page 53</a> ). 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)
encryptedPayment_data	The encrypted payment data value.  If you are using the <a href="#">CyberSource Decryption</a> option, populate this field with the encrypted payment data value returned by the Full Wallet request. See <a href="#">page 30</a> .	ics_auth (R)	
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 <a href="#">"Numbered Elements," page 60</a> .	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 <a href="#">"Numbered Elements," page 60</a> .	ccAuthService (See description)	String (255)

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 61](#). **Important** It is your responsibility to determine whether a field is required for the transaction you are requesting.

**Table 6 Request Fields (Continued)**

Field	Description	Used By: Required (R) or Optional (O)	Data Type (Length)
item_#_productSKU	<p>Identification code for the product.</p> <p>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.</p> <p>See <a href="#">"Numbered Elements," page 60</a>.</p>	ccAuthService (See description)	String (255)
item_#_quantity	<p>The default is 1.</p> <p>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.</p> <p>See <a href="#">"Numbered Elements," page 60</a>.</p>	ccAuthService (See description)	Integer (10)
item_#_taxAmount	<p>Total tax to apply to the product. This value cannot be negative.</p> <p>See <a href="#">"Numbered Elements," page 60</a>.</p>	ccAuthService (See description)	String (15)
item_#_unitPrice	<p>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.</p> <p>See <a href="#">"Numbered Elements," page 60</a>.</p>	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	<p>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>.</p>	ccAuthService (R)	String (50)
paymentNetworkToken_assuranceLevel	Confidence level of the tokenization. This value is assigned by the token service provider.	ccAuthService (O)	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 61. **Important** It is your responsibility to determine whether a field is required for the transaction you are requesting.

**Table 6 Request Fields (Continued)**

Field	Description	Used By: Required (R) or Optional (O)	Data Type (Length)
paymentNetworkToken_ deviceTechType	Type of technology used in the device to store token data. Possible value:  002: Host card emulation (HCE)  Emulation of a smart card by using software to create a virtual and exact representation of the card. Sensitive data is stored in a database that is hosted in the cloud. For storing payment credentials, a database must meet very stringent security requirements that exceed PCI DSS.  <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)
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.  Possible value:  ■ 1: In-app transaction.  An application on the customer's mobile device provided the token data for an e-commerce transaction. For recurring transactions, use this value if the original transaction was an in-app e-commerce transaction.	ccAuthService (R)	String (1)
paymentSolution	Identifies Android Pay as the payment solution that is being used for the transaction:  Set the value for this field to 006.  <b>Note</b> This unique ID differentiates digital solution transactions within the CyberSource platform for reporting purposes.	ccAuthService (R)	String (3)
<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 61. <b>Important</b> It is your responsibility to determine whether a field is required for the transaction you are requesting.</p>			

**Table 6 Request Fields (Continued)**

Field	Description	Used By: Required (R) or Optional (O)	Data Type (Length)
pos_environment	<p>Operating environment. Possible values:</p> <ul style="list-style-type: none"> <li>0: No terminal used or unknown environment.</li> <li>1: On merchant premises, attended.</li> <li>2: On merchant premises, unattended, or cardholder terminal. Examples: oil, kiosks, self-checkout, home computer, mobile telephone, personal digital assistant (PDA). Cardholder terminal is supported only for Mastercard transactions on CyberSource through VisaNet.</li> <li>3: Off merchant premises, attended. Examples: portable POS devices at trade shows, at service calls, or in taxis.</li> <li>4: Off merchant premises, unattended, or cardholder terminal. Examples: vending machines, home computer, mobile telephone, PDA. Cardholder terminal is supported only for Mastercard transactions on CyberSource through VisaNet.</li> <li>5: On premises of cardholder, unattended.</li> <li>9: Unknown delivery mode.</li> <li>S: Electronic delivery of product. Examples: music, software, or eTickets that are downloaded over the internet.</li> <li>T: Physical delivery of product. Examples: music or software that is delivered by mail or by a courier.</li> </ul> <p>This field is supported only for American Express Direct and CyberSource through VisaNet.</p> <p><b>CyberSource through VisaNet</b> For Mastercard transactions, the only valid values are 2 and 4.</p>	ccAuthService (O)	String (1)
purchaseTotals_currency	Currency used for the order: USD	ccAuthService (R)	String (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.

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

**Table 6 Request Fields (Continued)**

Field	Description	Used By: Required (R) or Optional (O)	Data Type (Length)
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)
subsequentAuth	<p>Indicates whether the transaction is a merchant-initiated transaction. Possible values:</p> <ul style="list-style-type: none"> <li>■ true: Merchant-initiated transaction</li> <li>■ false: Not a merchant-initiated transaction</li> </ul> <p>This field is supported only for Visa transactions on Chase Paymentech Solutions and CyberSource through VisaNet.</p> <p><b>CyberSource through VisaNet</b> The value for this field does not correspond to any data in the TC 33 capture file<sup>1</sup>.</p> <p><b>All Processors</b> See <a href="#">"Merchant-Initiated Transactions," page 16</a>.</p>	ccAuthService (R for merchant-initiated transactions; otherwise, not used)	String (5)
subsequentAuthFirst	<p>Indicates whether the transaction is the first merchant-initiated transaction in a series, which means that the customer initiated the previous transaction. Possible values:</p> <ul style="list-style-type: none"> <li>■ true: First merchant-initiated transaction</li> <li>■ false: Not the first merchant-initiated transaction</li> </ul> <p>This field is supported only for Visa transactions on Chase Paymentech Solutions and CyberSource through VisaNet.</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 TCR1</li> <li>■ Position: 136</li> <li>■ Field: POS Environment</li> </ul> <p><b>All Processors</b> See <a href="#">"Merchant-Initiated Transactions," page 16</a>.</p>	ccAuthService (R for merchant-initiated transactions; otherwise, not used)	String (5)

<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.

<sup>2</sup> 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 61. **Important** It is your responsibility to determine whether a field is required for the transaction you are requesting.

**Table 6 Request Fields (Continued)**

Field	Description	Used By: Required (R) or Optional (O)	Data Type (Length)
subsequentAuthReason	<p>Reason for the merchant-initiated transaction. Possible values:</p> <ul style="list-style-type: none"> <li>■ 1: Resubmission</li> <li>■ 2: Delayed charge</li> <li>■ 3: Reauthorization for split shipment</li> <li>■ 4: No show</li> <li>■ 5: Account top up</li> </ul> <p>This field is required only for the five kinds of transactions in the preceding list.</p> <p>This field is supported only for Visa transactions on Chase Paymentech Solutions and CyberSource through VisaNet.</p> <p><b>CyberSource through VisaNet</b></p> <p>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 TCR0</li> <li>■ Position: 160-163</li> <li>■ Field: Message Reason Code</li> </ul> <p><b>All Processors</b></p> <p>See <a href="#">"Merchant-Initiated Transactions," page 16.</a></p>	ccAuthService (See description)	String (1)
subsequentAuthStored Credential	<p>Indicates whether the transaction uses card-on-file (COF) payment information for a merchant-initiated transaction. Possible values:</p> <ul style="list-style-type: none"> <li>■ true: Transaction uses COF information</li> <li>■ false: Transaction does not use COF information</li> </ul> <p>This field is supported only for Visa transactions on Chase Paymentech Solutions and CyberSource through VisaNet.</p> <p>See <a href="#">"Merchant-Initiated Transactions," page 16.</a></p>	ccAuthService (R for merchant-initiated transactions; otherwise, not used)	String (5)

<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.

<sup>2</sup> 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 61. **Important** It is your responsibility to determine whether a field is required for the transaction you are requesting.

**Table 6 Request Fields (Continued)**

Field	Description	Used By: Required (R) or Optional (O)	Data Type (Length)
subsequentAuthTransactionID	<p>Network transaction identifier that was returned in the <b>ccAuthReply_paymentNetworkTransactionID</b> field in the reply message for either the original merchant-initiated authorization in the series or the previous merchant-initiated authorization in the series.</p> <p>If the current authorization request includes a token instead of an account number, the following time limits apply for the value of this field:</p> <ul style="list-style-type: none"> <li>■ For a resubmission, the transaction ID must be less than 14 days old.</li> <li>■ For a delayed charge or reauthorization, the transaction ID must be less than 30 days old.</li> </ul> <p>This field is supported only for Visa transactions on Chase Paymentech Solutions and CyberSource through VisaNet.</p> <p><b>CyberSource through VisaNet</b></p> <p>The value for this field does not correspond to any data in the TC 33 capture file<sup>1</sup>.</p> <p><b>All Processors</b></p> <p>See <a href="#">"Merchant-Initiated Transactions," page 16</a>.</p>	ccAuthService (R for merchant-initiated transactions; otherwise, not used)	String (15)
ucaf_authenticationData	Cryptogram for payment network tokenization transactions with Mastercard.	ccAuthService (R)	String (32)
ucaf_collectionIndicator	<p>Required field for payment network tokenization transactions with Mastercard.</p> <p>Set the value for this field to 2.</p>	ccAuthService (R)	String with numbers only (1)
<p><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.</p> <p><sup>2</sup> 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 61. <b>Important</b> It is your responsibility to determine whether a field is required for the transaction you are requesting.</p>			



## 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 7** 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>CyberSource through VisaNet</b></p> <p>The value for this field corresponds to the following data in the TC 33 capture file:</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> <p><b>Note</b> This field is returned only for CyberSource through VisaNet and FDC Nashville Global.</p>	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	<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 22:47:57 (10:47:57 p.m.). The T separates the date and the time. The Z indicates UTC.</p>	ccAuthReply	String (20)
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)

**Table 7** Reply Fields (Continued)

Field	Description	Returned By	Data Type & Length
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_ 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)
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)

**Table 7     Reply Fields (Continued)**

Field	Description	Returned By	Data Type & Length
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)
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)
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)

**Table 7     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"> <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)
decision	<p>Summarizes the result of the overall request. Possible values:</p> <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	<p>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>.</p>	ccAuthReply	String (100)
merchantReferenceCode	<p>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.</p>	ccAuthReply	String (50)

**Table 7     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> 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 “Mastercard Product IDs” 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)

**Table 7     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)