Apple Pay

Simple Order API
Vero

cybersource
A Visa Solution

Developer Guide

Cybersource Contact Information

For general information about our company, products, and services, go to https://www.cybersource.com.

For sales questions about any Cybersource service, email sales@cybersource.com or call 650-432-7350 or 888-330-2300 (toll free in the United States).

For support information about any Cybersource service, visit the Support Center: https://www.cybersource.com/support

Copyright

© 2020. Cybersource Corporation. All rights reserved. Cybersource Corporation ("Cybersource") furnishes this document and the software described in this document under the applicable agreement between the reader of this document ("You") and Cybersource ("Agreement"). You may use this document and/or software only in accordance with the terms of the Agreement. Except as expressly set forth in the Agreement, the information contained in this document is subject to change without notice and therefore should not be interpreted in any way as a guarantee or warranty by Cybersource. Cybersource assumes no responsibility or liability for any errors that may appear in this document. The copyrighted software that accompanies this document is licensed to You for use only in strict accordance with the Agreement. You should read the Agreement carefully before using the software. Except as permitted by the Agreement, You may not reproduce any part of this document, store this document in a retrieval system, or transmit this document, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written consent of Cybersource.

Restricted Rights Legends

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

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 Unpublished rights reserved under the copyright laws of the United States.

Trademarks

Authorize.Net, eCheck.Net, and The Power of Payment are registered trademarks of Cybersource Corporation. Cybersource, Cybersource Payment Manager, Cybersource Risk Manager, Cybersource Decision Manager, and Cybersource Connect are trademarks and/or service marks of Cybersource Corporation. Visa, Visa International, Cybersource, the Visa logo, and the Cybersource logo are the registered trademarks of Visa International in the United States and other countries. All other trademarks, service marks, registered marks, or registered service marks are the property of their respective owners.

Confidentiality Notice

This document is furnished to you solely in your capacity as a client of Cybersource and as a participant in the Visa payments system.

By accepting this document, you acknowledge that the information contained herein (the "Information") is confidential and subject to the confidentiality restrictions contained in Visa's operating regulations and/or other confidentiality agreements, which limity our use of the Information. You agree to keep the Information confidential and not to use the Information for any purpose other than its intended purpose and in your capacity as a customer of Cybersource or as a participant in the Visa payments system. The Information may only be disseminated within your organization on a need-to-know basis to enable your participation in the Visa payments system. Please be advised that the Information may constitute material non-public information under U.S. federal securities laws and that purchasing or selling securities of Visa Inc. while being aware of material non-public information would constitute a violation of applicable U.S. federal securities laws.

Revision

Version: 24.01

Contents

Recent Revisions to This Document **5**
About This Guide **6**
Introduction to Apple Pay **7**
Requirements for Using Apple Pay 7
Supported Card Types and Optional Features 8
Apple Pay Integrations and Payload Decryption Methods 8
Flow of Apple Pay In-App Transactions Processed Using Cybersource Decryption 8
Flow of Apple Pay In-App Transactions Processed Using Merchant Decryption 9
Flow of Apple Pay Web Transactions Using Cybersource Decryption 10
Flow of Apple Pay Web Transactions Using Merchant Decryption 11
Configuring Apple Pay Processing **12**
Generating and Loading a New Certificate Signing Request 12
Configuring Apple Pay Response Handling 13
Processing Apple Pay Transactions **15**
Authorization Service 16
Authorizations Using Cybersource Decryption for Mastercard 16
Required Fields for Authorizing a Payment Using Cybersource Decryption for Mastercard 16
Simple Order XML Example: Cybersource Decryption for Mastercard 17
Authorizations Using Cybersource Decryption for Visa 18
Required Fields for Authorizing a Payment Using Cybersource Decryption for Visa 18
Simple Order XML Example: Cybersource Decryption for Visa 19
Authorizations Using Merchant Decryption for Mastercard 20
Required Fields for Authorizing a Payment Using Merchant Decryption for Mastercard 21
Simple Order XML Example: Merchant Decryption and Mastercard 22
Authorizations Using Merchant Decryption for Visa 23

Required Fields for Authorizing a Payment Using Merchant Decryption for Visa ... 23
Simple Order XML Example: Merchant Decryption and Visa ... 24
Authorization Reversals ... 25
Required Fields for Processing an Authorization Reversal ... 25
Simple Order Example: Processing an Authorization Reversal ... 26
Captures ... 26
Required Fields for Capturing an Authorization ... 26
Simple Order Example: Capturing an Authorization ... 27
Sale ... 27
Required Fields for Processing a Sale ... 27
Simple Order Example: Processing a Sale ... 28
Searching for Apple Pay Transactions ... 30

Recent Revisions to This Document

24.01

This revision contains only editorial changes and no technical updates.

23.04

This revision contains only editorial changes and no technical updates.

23.03

This revision contains only editorial changes and no technical updates.

23.02

This revision contains only editorial changes and no technical updates.

23.01

Authorizations using Cybersource decryption	**For each supported card type, these fields were removed from the list of required fields:** • **card_accountNumber** • **card_expirationMonth** • **card_expirationYear** **For each supported card type, this field was added to the list of required fields:** • **paymentNetworkToken_transactionType**

About This Guide

This section describes the audience and purpose of this guide and the conventions used in this guide.

Audience and Purpose

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

Conventions

The following special statement is used in this document:

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

Customer Support

For support information about any service, visit the Support Center:
http://www.cybersource.com/support

Introduction to Apple Pay

You can use the Cybersource platform to process and manage Apple Pay transactions.

Requirements for Using Apple Pay

In order to use the Cybersource platform to process Apple Pay transactions, you must have these things:

- A Cybersource account. If you do not already have a Cybersource account, contact your local Cybersource sales representative.
- A *merchant evaluation account* with a *supported processor*.
- An Admin or Team Agent user of the *Apple Pay Developer* account.

Important

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

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

If one of these statements is not true, you must take one of these actions before you can sign up for Apple Pay:

- Obtain a new merchant account with a processor that supports payment network tokens.
- Wait until your processor supports payment network tokens.

Supported Card Types and Optional Features

Merchant-initiated transactions, multiple partial captures, and subsequent authorizations are described in *Authorizations with Payment Network Tokens*. Recurring payments and split shipments are described in *Credit Card Services Optional Features*.

Processor	Card Types	Optional Features
Vero	Mastercard Visa Elo	Merchant-Initiated transactions Recurring payments Subsequent authorizations

Apple Pay Integrations and Payload Decryption Methods

In response to a payment request, Apple Pay returns an encrypted payload that contains sensitive payment information. Two methods can be used to extract and decrypt the payment information, and both methods support Apple Pay in-app and web transactions.

- With Cybersource decryption, Cybersource manages the generation of the payment encryption key and the decryption of the Apple Pay response.

  Important

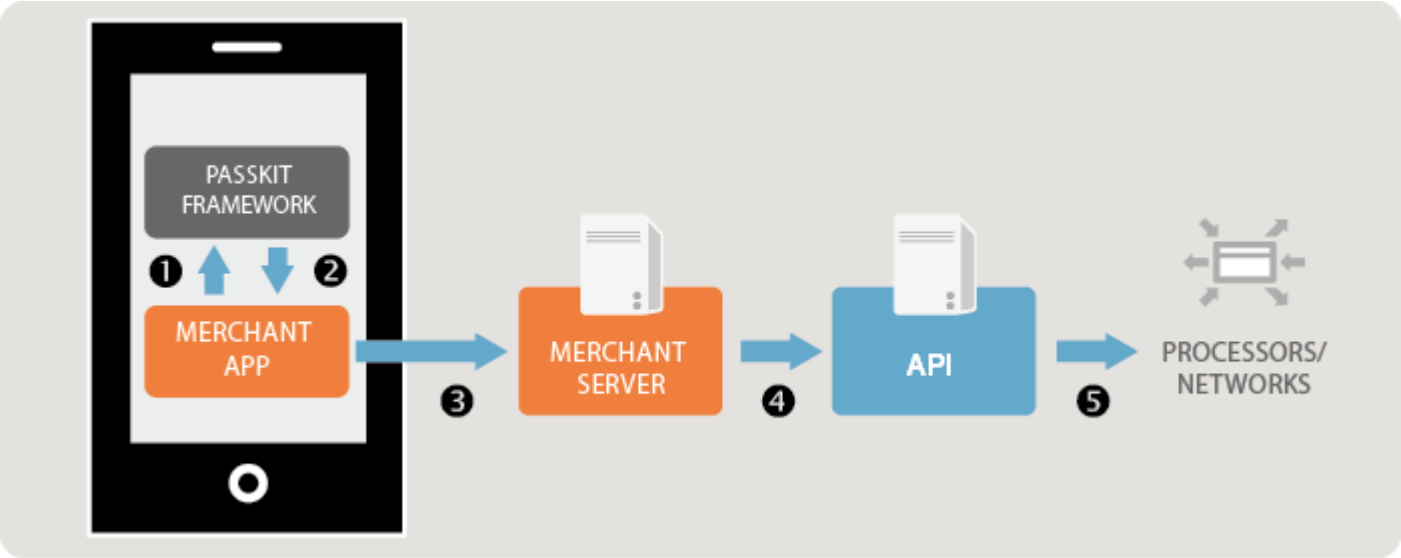
  This method reduces the exposure of the sensitive payment data to your system.

- With merchant decryption, you manage both the payment encryption key generation and the decryption of the Apple Pay response.

The remainder of this section summarizes the flow of Apple Pay transaction processing for each decryption method.

Flow of Apple Pay In-App Transactions Processed Using Cybersource Decryption

This diagram illustrates the flow of in-app transaction processing using Cybersource decryption.



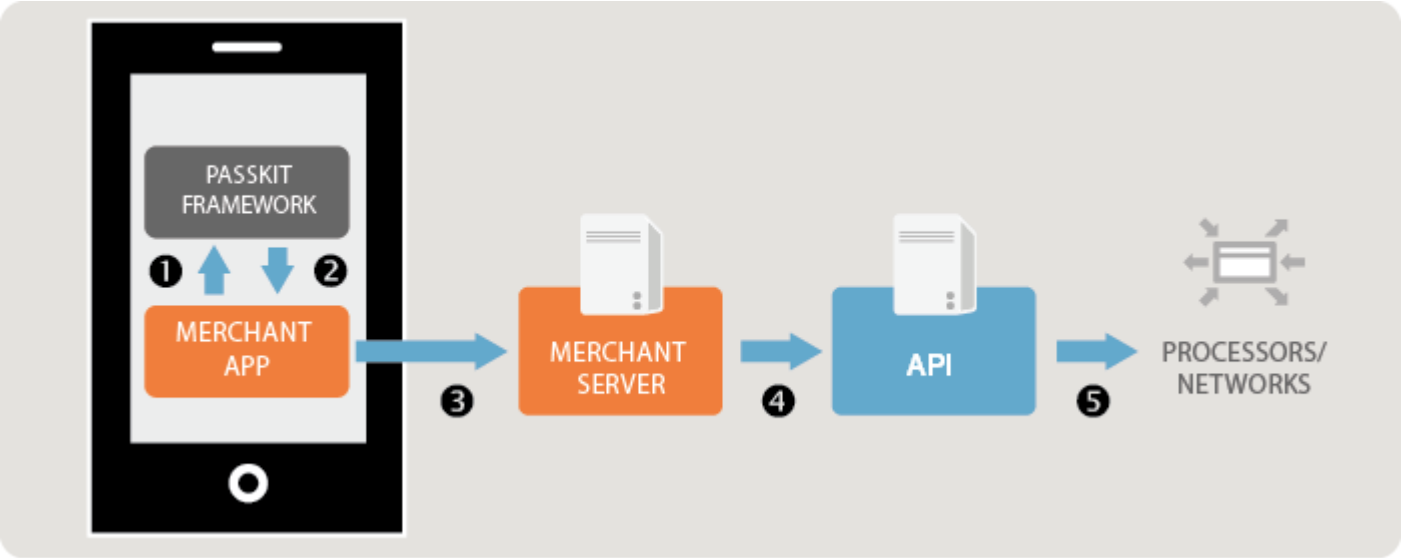
In-App Transaction Processing with Cybersource Decryption

1. When the customer chooses to pay with Apple Pay, you use the Apple PassKit Framework to request the encrypted payment data from Apple.
2. Apple uses the Secure Element to create a payment token and encrypt the token's payment data before it sends your application.
   - The payment token is the `PKPaymentToken` structure.
   - The payment data is the **paymentData** field of the `PKPaymentToken` structure.
3. You forward the encrypted payment data to your order management system.
4. Using the Cybersource API, you submit the authorization request. In the **encryptedPayment_data** field, include the Base64-encoded value obtained from the **paymentData** field of the `PKPaymentToken` structure.
5. Cybersource decrypts the payment data and forwards the information to the payment network, which includes your processor and the relevant payment card company.

> **Important**
>
> You must use the Business Center or one of the Cybersource API services to capture, credit, or void the authorization.

Flow of Apple Pay In-App Transactions Processed Using Merchant Decryption

This diagram illustrates the flow of in-app transaction processing using merchant decryption.



In-App Transaction Processing with Merchant Decryption

1. When the customer chooses to pay with Apple Pay, you use the Apple PassKit Framework to request the encrypted payment data from Apple.
2. Apple uses the Secure Element to create a payment token and encrypt the token's payment data before it sends your application.
   - The payment token is the `PKPaymentToken` structure.
   - The payment data is the **paymentData** field of the `PKPaymentToken` structure.
3. You forward the encrypted payment data to your order management system to decrypt. For information on decryption, see the *Payment Token Format Reference* article in the Apple Pay Developer Documentation.
4. Using the Cybersource API, you submit the authorization request and include the decrypted payment data.
5. Cybersource forwards the information to the payment network, which includes your processor and the relevant payment card company.

> **Important**
>
> You must use the Business Center or one of the Cybersource API services to capture, credit, or void the authorization.

Flow of Apple Pay Web Transactions Using Cybersource Decryption

1. When the customer chooses to pay with Apple Pay, you use the Apple Pay JavaScript to request the encrypted payment data from Apple.
2. Apple uses the Secure Element to create a payment token and encrypt the token's payment data before it sends your application using the `onpaymentauthorized` event handler.
   - The payment token is the `ApplePayPaymentToken` structure.

- The payment data is the **paymentData** field of the `ApplePayPaymentToken` structure.

For more information, see *Apple Pay on the Web Interactive Demo*.

3. You forward the encrypted payment data to your order management system.
4. Using the Cybersource API, you submit the authorization request. In the **encryptedPayment_data** field, include the Base64-encoded value obtained from the **paymentData** field of the `ApplePayPaymentToken` structure.
5. Cybersource decrypts the payment data and forwards the information to the payment network, which includes your processor and the relevant payment card company.

> **Important**
>
> Use the Business Center or one of the Cybersource API services to capture, credit, or void the authorization.

Flow of Apple Pay Web Transactions Using Merchant Decryption

1. When the customer chooses to pay with Apple Pay, you use the Apple Pay JavaScript to request the encrypted payment data from Apple.
2. Apple uses the Secure Element to create a payment token and encrypt the token's payment data before it sends your application using the `onpaymentauthorized` event handler.

   - The payment token is the `ApplePayPaymentToken` structure.
   - The payment data is the **paymentData** field of the `ApplePayPaymentToken` structure.

   For more information, see *Apple Pay on the Web Interactive Demo*.
3. You forward the encrypted payment data to your order management system to decrypt. For information on decryption, see the *Payment Token Format Reference* article in the Apple Pay Developer Documentation.
4. Using the Cybersource API, you submit the authorization request and include the decrypted payment data. See *Using the Merchant Decryption Method*.
5. Cybersource forwards the information to the payment network, which includes your processor and the relevant payment card company.

> **Important**
>
> Use the Business Center or one of the Cybersource API services to capture, credit, or void the authorization.

Configuring Apple Pay Processing

A successful Apple Pay response to a payment request returns an encrypted payload that contains sensitive payment information. The payment information is extracted and decrypted using Cybersource decryption or merchant decryption.
Before you can process Apple Pay transactions, you must complete these configuration tasks:

Generating and Loading a New Certificate Signing Request	For processing Apple Pay transactions using Cybersource decryption, you must first generate an Apple Pay encryption key on the business portal and load it into the Apple development portal. For merchant decryption, this configuration task is not needed.
Configuring Apple Pay Response Handling	After the payment token is received, the transaction is finalized by extracting and decrypting the payment data. Cybersource decryption and merchant decryption methods consist of different configuration steps.

Generating and Loading a New Certificate Signing Request

This task is required to process Apple Pay transactions using Cybersource decryption. It does not apply to merchant decryption.

Follow these steps in order to configure Apple Pay in the Cybersource Business Center:

1. Log in to the Business Center:
   - Test: *https://businesscentertest.cybersource.com*
   - Production: *https://businesscenter.cybersource.com*
2. On the left navigation panel, click the **Payment Configuration** icon.
3. Click **Digital Payment Solution**. The Digital Payments page appears. If you do not have the correct permissions enabled on your account, the Digital Payment Solution option does not appear on the left navigation panel.
4. Click **Configure**. The Apple Pay Registration panel opens.
5. Enter your Apple Merchant ID.
6. Click **Generate New CSR**.
7. To download your CSR, click the **Download** icon next to the key.
8. Follow your browser's instructions to save and open the file.
9. Complete the enrollment process by submitting your CSR to Apple.
10. For information about adding certificates to your Apple Merchant ID, refer to the Apple Pay PassKit:
    - In-App: *Setting Up Apple Pay*
    - Website: *Configuring Your Environment*
11. Test your software by following the steps in *Processing Apple Pay Transactions*.

> **Important**
>
> If you are using a Cybersource test account, you must connect to the *Apple sandbox* tester account and not to the Apple production system.
> After you complete your testing, you must create a new CSR for the Cybersource production system, and you must use that CSR for the Apple production system. Until you perform these steps, you cannot enable payments in your iOS application or website.

12. Repeat Steps 1 through 11 with your Cybersource production account and the Apple production account.

Configuring Apple Pay Response Handling

Configure the decryption method to handle the response payload of a successful Apple Pay transaction.

After the payment token is received, the transaction is finalized by extracting and decrypting the payment data.

- Cybersource Decryption:
  This method forwards the encrypted payment data to your order management system. Use the Cybersource API to submit the authorization request, and include the Base64-

encoded value obtained from the `paymentData` object in the **encryptedPayment_data** field.
Example:

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

- Merchant Decryption:
  This method forwards the encrypted payment data to your order management system to decrypt. Use the Cybersource API to submit the authorization request and include the decrypted payment data.
  For detailed information about decryption, see *Payment Token Format Reference* in the Apple Developer Center.

Processing Apple Pay Transactions

This section provides information about these Apple Pay transactions:

Authorizing a Payment Using Cybersource Decryption or Merchant Decryption	You can request the authorization service using these decryption methods: • Cybersource decryption: Within this integration, Cybersource facilitates all of the encryption key generation and decryption processes associated with implementing Apple Pay. This method reduces the exposure of the sensitive payment data to your system. • Merchant decryption: Within this integration, you manage all aspects of the Apple Pay integration, from generation of the payment encryption keys to decryption of the Apple Pay payload response. As a merchant, you submit the Apple Pay payment token and other payment information to Cybersource for processing.
Reversing an Authorization	This service 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.
Capturing a Payment	This service uses the request ID returned from the previous authorization. The request ID links the capture to the

	authorization. Use this service to transfer funds from the customer's account to your bank. The transaction typically completes in two to four days.
Authorizing and Capturing a Payment	A bundled authorization and capture is called a sale. Request the authorization and capture services at the same time. Cybersource processes the capture immediately.

Authorization Service

This section provides the steps for requesting the authorization service using these methods, which determine the request fields that are required to request the authorization service. In addition, different request fields are required in order to request the authorization service. The type of card used to process the transaction determines which request fields are used.
To request the authorization service, use the endpoint specified below.
After you send the request, verify the response messages to make sure that the request was successful. A value of `ACCEPT` for the **decision** field indicates success.
For information about response codes, see *Reason Codes for the Simple Order API*.

Endpoint

Set the **ccAuthService_run** field to `true`.
Send the request to `https://ics2ws.ic3.com/commerce/1.x/transactionProcessor`.

Authorizations Using Cybersource Decryption for Mastercard

This section provides the information you need in order to process an authorization using Cybersource decryption for Mastercard.

Required Fields for Authorizing a Payment Using Cybersource Decryption for Mastercard

billTo_city

billTo_country

billTo_email

billTo_firstName

billTo_lastName	
billTo_postalCode	
billTo_state	
billTo_street1	
ccAuthService_run	**Set this field to `true`.**
encryptedPayment_data	**Set this field to the Base64-encoded value obtained from the `paymentData` property of the `PKPaymentToken` object.**
encryptedPayment_descriptor	**Set this field to `RklEPUNPTU1PTi5BUFBMRS5JTkFQUC5QQVlNRU5U`.**
merchantID	
merchantReferenceCode	
paymentSolution	**Set this field to `001`.**
purchaseTotals_currency	
purchaseTotals_grandTotalAmount	**Either purchaseTotals_grandTotalAmount or item_#_unitPrice must be included in the request.**

Related Information

- *API Field Reference for the Simple Order API*

Simple Order XML Example: Cybersource Decryption for Mastercard

Request

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.121">
  <merchantID>demomerchant</merchantID>
  <merchantReferenceCode>demorefnum</merchantReferenceCode>
  <billTo>
    <firstName>Jane</firstName>
    <lastName>Smith</lastName>
    <street1>123 Main Street</street1>
    <city>Small Town</city>
    <state>CA</state>
    <postalCode>98765</postalCode>
    <country>US</country>
    <email>jsmith@example.com</email>
  </billTo>
  <purchaseTotals>
    <currency>USD</currency>
    <grandTotalAmount>5.00</grandTotalAmount>
  </purchaseTotals>
  <encryptedPayment>
```

```
    <descriptor>RklEPUNPTU1PTi5BUFBMRS5JTkFQUC5QQVlNRU5U</descriptor>
    <data>ABCDEFabcdefABCDEFabcdef0987654321234567</data>
    <encoding>Base64</encoding>
  </encryptedPayment>
  <card>
    <cardType>002</cardType>
  </card>
  <ccAuthService run="true"/>
  <paymentSolution>001</paymentSolution>
</requestMessage>
```

Response to a Successful Request

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

Authorizations Using Cybersource Decryption for Visa

This section provides the information you need in order to process an authorization using Cybersource decryption for Visa.

Required Fields for Authorizing a Payment Using Cybersource Decryption for Visa

billTo_city

billTo_country	
billTo_email	
billTo_firstName	
billTo_lastName	
billTo_postalCode	
billTo_state	
billTo_street1	
ccAuthService_run	**Set this field to `true`.**
encryptedPayment_data	**Set this field to the Base64-encoded value obtained from the `paymentData` property of the `PKPaymentToken` object.**
encryptedPayment_descriptor	**Set this field to `Rk1EPUNPTU1PTi5BUFBMRS5JTkFQUC5QQV1NRU5U`.**
merchantID	
merchantReferenceCode	
paymentSolution	**Set this field to `001`.**
purchaseTotals_currency	
purchaseTotals_grandTotalAmount	**Either purchaseTotals_grandTotalAmount or item_#_unitPrice must be included in the request.**

Related Information

- *API Field Reference for the Simple Order API*

Simple Order XML Example: Cybersource Decryption for Visa

Request

```
<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>
```

Response to a Successful Request

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

Authorizations Using Merchant Decryption for Mastercard

This section provides the information you need in order to process an authorization using merchant decryption for Mastercard.

Required Fields for Authorizing a Payment Using Merchant Decryption for Mastercard

billTo_city	
billTo_country	
billTo_email	
billTo_firstName	
billTo_lastName	
billTo_postalCode	
billTo_state	
billTo_street1	
card_accountNumber	**Set this field to the payment network token value.**
card_expirationMonth	**Set this field to the value from the payment network token expiration month.**
card_expirationYear	**Set this field to the value from the payment network token expiration year.**
ccAuthService_commerceIndicator	**Set this field to `internet`.**
ccAuthService_networkTokenCryptogram	**Set this field to the network token cryptogram.**
ccAuthService_run	**Set this field to `true`.**
merchantID	
merchantReferenceCode	
paymentNetworkToken_transactionType	**Set this field to `1`.**
paymentSolution	**Set this field to `001`.**
purchaseTotals_currency	
purchaseTotals_grandTotalAmount	**Either purchaseTotals_grandTotalAmount or item_#_unitPrice must be included in the request.**
ucaf_authenticationData	Set this field to the network token cryptogram.
ucaf_collectionIndicator	**Set this field to `2`.**

Related Information

- *API Field Reference for the Simple Order API*

Simple Order XML Example: Merchant Decryption and Mastercard

Request

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

Response to a Successful Request

```
<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>
```

Authorizations Using Merchant Decryption for Visa

This section provides the information you need in order to process an authorization using merchant decryption for Visa.

Required Fields for Authorizing a Payment Using Merchant Decryption for Visa

billTo_city	
billTo_country	
billTo_email	
billTo_firstName	
billTo_lastName	
billTo_postalCode	
billTo_state	
billTo_street1	
card_accountNumber	**Set this field to the payment network token value.**
card_expirationMonth	**Set this field to the value from the payment network token expiration month.**
card_expirationYear	**Set this field to the value from the payment network token expiration year.**
ccAuthService_cavv	**Set this field to the network token cryptogram.**
ccAuthService_commerceIndicator	**Set this field to the ECI value contained in the Apple Pay response payload.**
ccAuthService_networkTokenCryptogram	**Set this field to the network token cryptogram.**
ccAuthService_run	**Set this field to `true`.**

merchantID	
merchantReferenceCode	
paymentNetworkToken_transactionType	**Set this field to `1`.**
paymentSolution	**Set this field to `001`.**
purchaseTotals_currency	
purchaseTotals_grandTotalAmount	**Either purchaseTotals_grandTotalAmount or item_#_unitPrice must be included in the request.**

Related Information

- *API Field Reference for the Simple Order API*

Simple Order XML Example: Merchant Decryption and Visa

Request

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

Response to a Successful Request

```
<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>
```

Authorization Reversals

This section provides the information you need in order to process an authorization reversal.
Reversing an authorization releases the hold on the customer's payment card funds that the issuing bank placed when processing the authorization.

Endpoint

Set the **ccAuthReversalService_run** field to `true`.
Send the request to `https://ics2ws.ic3.com/commerce/1.x/transactionProcessor`.

Required Fields for Processing an Authorization Reversal

ccAuthReversalService_authRequestID	**Set this field to the request ID that was included in the authorization response message.**
ccAuthReversalService_run	**Set this field to `true`.**
merchantReferenceCode	
merchantTransactionIdentifier	
purchaseTotals_currency	
purchaseTotals_grandTotalAmount	**The amount of the reversal must be the same as the authorization amount that was included in the authorization response**

	message. Do not use the amount that was requested in the authorization request message.

Simple Order Example: Processing an Authorization Reversal

Request

```
ccAuthReversalService_authRequestID=6522033834410167772169
ccAuthReversalService_run=true
merchantReferenceCode=482046C3A7E94F5BD1FE3C66C
merchantTransactionIdentifier=Napa Valley Vacations
purchaseTotals_currency=USD
purchaseTotals_grandTotalAmount=49.95
```

Response to a Successful Request

```
requestID=1019827520348290570293
merchantReferenceCode=482046C3A7E94F5BD1FE3C66C
decision=ACCEPT
reasonCode=100
ccAuthReversalReply_amount=49.95
purchaseTotals_currency=USD
ccAuthReversalReply_reasonCode=100
ccAuthReversalReply_reconciliationID=1094820975023470
```

Captures

This section provides the information you need in order to capture an authorized transaction.

Endpoint

Set the **ccCaptureService_run** field to `true`.
Send the request to `https://ics2ws.ic3.com/commerce/1.x/transactionProcessor`.

Required Fields for Capturing an Authorization

ccCaptureService_authRequestID	
ccCaptureService_run	
merchantID	
merchantReferenceCode	**Set to `merchant_ref_number` value used in corresponding authorization request.**
purchaseTotals_currency	**Vero supports Brazilian real (BRL) currency only.**
purchaseTotals_grandTotalAmount	

Simple Order Example: Capturing an Authorization

Request

```
ccCaptureService_authRequestID=6629978499572480812782
ccCaptureService_run=true
merchantID=npr_paymentech
merchantReferenceCode=TC42703-1
purchaseTotals_grandTotalAmount=100.00
```

Response to a Successful Request

```
ccCaptureReply_amount=100.00
ccCaptureReply_requestDateTime=2022-09-12T173947Z
decision=ACCEPT
merchantReferenceCode=TC42703-1
purchaseTotals_currency=USD
requestID=6630043878211258349460
```

Sale

This section provides the information you need in order to process a sale transaction. A sale transaction combines an authorization and a capture into a single transaction.

Endpoint

Set the **ccAuthService_run** field to `true`, and the **ccCaptureService_run** field to `true`. Send the request to `https://ics2ws.ic3.com/commerce/1.x/transactionProcessor`.

Required Fields for Processing a Sale

Use these required fields for processing a sale.

billTo_city

billTo_country

billTo_email

billTo_firstName

billTo_lastName

billTo_postalCode

billTo_state

billTo_street1

card_accountNumber

card_cardType

card_expirationMonth

card_expirationYear	
ccAuthService_commerceIndicator	
ccAuthService_run	**Set this field to `true`.**
ccCaptureService_run	**Set this field to `true`.**
merchantID	
purchaseTotals_currency	**Vero supports Brazilian real (BRL) currency only.**
purchaseTotals_grandTotalAmount	

Related Information

- *API Field Reference for the Simple Order API*

Simple Order Example: Processing a Sale

Request

```
ccAuthService_run=true
ccCaptureService_run=true
merchantID=Napa Valley Vacations
merchantReferenceCode=482046C3A7E94F5
billTo_firstName=John
billTo_lastName=Doe
billTo_street1=1295 Charleston Rd.
billTo_city=Mountain View
billTo_state=CA
billTo_postalCode=94043
billTo_country=US
billTo_phoneNumber=650-965-6000
billTo_email=jdoe@example.com
item_0_unitPrice=49.95
item_0_quantity=1
purchaseTotals_currency=USD
card_expirationMonth=12
card_expirationYear=2015
card_accountNumber=4111111111111111
card_cardType=001
```

Response to a Successful Request

Most processors do not return all of the fields shown in this example.

```
requestID=0305782650000167905080
decision=ACCEPT
reasonCode=100
merchantReferenceCode=482046C3A7E94F5
purchaseTotals_currency=USD
ccAuthReply_reconciliationID=ABCDE12345FGHIJ67890
ccAuthReply_cardCategory=F^
ccAuthReply_cardGroup=0
```

```
ccAuthReply_reasonCode=100
ccAuthReply_amount=49.95
ccAuthReply_accountBalance=50.05
ccAuthReply_authorizationCode=123456
ccAuthReply_avsCode=Y
ccAuthReply_avsCodeRaw=YYY
ccAuthReply_processorResponse=A
ccAuthReply_paymentNetworkTransactionID=3312345
ccCaptureReply_amount=49.95
ccCaptureReply_reasonCode=100
ccCaptureReply_reconciliationID=1094820975023470
```

Searching for Apple Pay Transactions

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

For information about the Transaction Request Report, see the *Business Center Reporting User Guide*.