Cybersource Contact Information

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

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

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

Copyright

© 2021. Cybersource Corporation. All rights reserved. Cybersource Corporation ("Cybersource") furnishes this document and the software described in this document under the applicable agreement between the reader of this document ("You") and Cybersource ("Agreement"). You may use this document and/or software only in accordance with the terms of the Agreement. Except as expressly set forth in the Agreement, the information contained in this document is subject to change without notice and therefore should not be interpreted in any way as a guarantee or warranty by Cybersource. Cybersource assumes no responsibility or liability for any errors that may appear in this document. The copyrighted software that accompanies this document is licensed to You for use only in strict accordance with the Agreement. You should read the Agreement carefully before using the software. Except as permitted by the Agreement, You may not reproduce any part of this document, store this document in a retrieval system, or transmit this document, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written consent of Cybersource.

Restricted Rights Legends

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

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 Unpublished rights reserved under the copyright laws of the United States.

Trademarks

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

Revision: 2021.01
Contents

Recent Revisions to This Document  5

About This Guide  6
  Audience and Purpose  6
  Conventions  6
    Notes and Important Statements  6
    Text and Command Conventions  6
  Related Documents  7
  Customer Support  7

Chapter 1   Introduction  8
  Requirements  8
  Supported Processors, Acquirers, Card Types, and Optional Features  9
  Transaction Endpoints  10

Chapter 2   Registration  11
  Registering with Samsung  11
  Registering with Cybersource  12

Chapter 3   Integrating the Samsung SDK  13
  Creating a Project  13
  Integrating the Samsung Pay SDK  13
  Using the API Key  14
Chapter 4  Using the Samsung Pay SDK  15
  Eligibility  15
  Payment Request  16
    Initiating a Payment  16
    Requesting a Payment  18

Chapter 5  Authorizing a Payment  20
  Merchant Decryption  20
    Visa Transaction  20
    Mastercard Transaction  22
    American Express Transaction  23
    JCB Transaction  25
  Cybersource Decryption  26
    Visa Transaction  26
    Mastercard Transaction  28
    American Express Transaction  30
    JCB Transaction  32
  Additional Cybersource Services  34

Appendix A  API Fields  35
  Data Type Definitions  35
  Relaxed Requirements for Address Data  35
  Request Fields  36
  Offer-Level Fields  43
  Reply Fields  44
# Recent Revisions to This Document

<table>
<thead>
<tr>
<th>Release</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2021.01</td>
<td>Added the <code>merchant_url</code> field to “Request Fields,” page 36.</td>
</tr>
<tr>
<td>2020.03</td>
<td>Changed the name of CyberSource through VisaNet to Visa Platform Connect.</td>
</tr>
<tr>
<td>2020.01</td>
<td>Changed payment network tokenization to authorizations with payment network tokens throughout. Updated the Business Center procedure. See &quot;Registering with Cybersource,&quot; page 12.</td>
</tr>
<tr>
<td>2019.03</td>
<td>This revision contains only editorial changes and no technical updates.</td>
</tr>
<tr>
<td>2019.02</td>
<td>Changed the name of Mastercard SecureCode to Mastercard Identity Check.</td>
</tr>
</tbody>
</table>
About This Guide

Audience and Purpose

This document is written for merchants who want to enable customers to use Samsung Pay to pay for in-app purchases. This document provides an overview of integrating the Samsung Pay SDK and describes how to request the Cybersource API to process an authorization. See "Using the Samsung Pay SDK," page 15, and "Authorizing a Payment," page 20. Merchants must use the Samsung Pay SDK to receive the customer’s encrypted payment data before requesting the Cybersource API to process the transaction.

Conventions

Notes and Important Statements

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

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

Text and Command Conventions

<table>
<thead>
<tr>
<th>Convention</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>bold</strong></td>
<td>• Field and service names in text; for example: Include the <strong>customer_cc_number</strong> field. • Items that you are instructed to act upon; for example: Click <strong>Save</strong>.</td>
</tr>
<tr>
<td>Screen text</td>
<td>Code examples and samples.</td>
</tr>
</tbody>
</table>
Related Documents

Cybersource Documents:
- Getting Started with Cybersource Advanced for the SCMP API (PDF | HTML)
- Cybersource SCMP API Client Developer Guide
- Credit Card Services Using the SCMP API (PDF | HTML)
- Authorizations with Payment Network Tokens Using the SCMP API (PDF | HTML)

Samsung Pay documents:
- Samsung Pay Partner Portal

Refer to the Support Center for complete Cybersource technical documentation:

Customer Support

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

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

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

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

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

You must create:

- A Cybersource account. If you do not already have a Cybersource account, contact your local Cybersource sales representative:
  
  http://www.cybersource.com/locations/


- A profile on the Samsung Pay Partner Portal, and you must obtain a partner ID. See "Registration," page 11.
## Supported Processors, Acquirers, Card Types, and Optional Features

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

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

<table>
<thead>
<tr>
<th>Processors</th>
<th>Card Types</th>
<th>Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>American Express Direct</td>
<td>American Express</td>
<td>- Multiple partial captures</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Recurring payments</td>
</tr>
<tr>
<td>Barclays</td>
<td>Visa, Mastercard, JCB, Maestro (International), Maestro (UK Domestic)</td>
<td>- Multiple partial captures</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Recurring payments</td>
</tr>
<tr>
<td>Chase Paymentech Solutions</td>
<td>Visa, Mastercard, American Express, Discover, Diners Club, JCB, Carte Blanche, Maestro (International)</td>
<td>- Multiple partial captures</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Recurring payments</td>
</tr>
<tr>
<td>Elavon Americas</td>
<td>Visa, Mastercard, American Express, JCB, Diners Club, Discover, China UnionPay</td>
<td>- Merchant-Initiated transactions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Multiple partial captures</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Recurring payments</td>
</tr>
<tr>
<td>FDC Compass</td>
<td>Visa, Mastercard, American Express, Discover, Diners Club, JCB</td>
<td>- Multiple partial captures</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Recurring payments</td>
</tr>
<tr>
<td>FDC Nashville Global</td>
<td>Visa, Mastercard, American Express, Discover, Diners Club, JCB, China UnionPay</td>
<td>- Multiple partial captures</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Recurring payments</td>
</tr>
<tr>
<td>GPN</td>
<td>Visa, Mastercard, American Express, Discover, Diners Club, JCB</td>
<td>- Split shipments</td>
</tr>
</tbody>
</table>
### Table 1  Supported Processors, Card Types, and Optional Features (Continued)

<table>
<thead>
<tr>
<th>Processors</th>
<th>Card Types</th>
<th>Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>JCN Gateway</td>
<td>Visa, Mastercard, American Express, Diners Club, JCB, NICOS house card, ORICO house card</td>
<td>Multiple partial captures</td>
</tr>
<tr>
<td>OmniPay Direct</td>
<td>Visa, Mastercard, American Express, Diners Club, Maestro (UK Domestic), Maestro (International)</td>
<td>Multiple partial captures, Recurring payments</td>
</tr>
<tr>
<td>OmniPay Direct</td>
<td>Bank of America Merchant Services</td>
<td></td>
</tr>
<tr>
<td>OmniPay Direct</td>
<td>First Data Merchant Solutions (Europe)</td>
<td></td>
</tr>
<tr>
<td>OmniPay Direct</td>
<td>Global Payments International Acquiring</td>
<td></td>
</tr>
<tr>
<td>Streamline</td>
<td>Visa, Mastercard</td>
<td>Multiple partial captures</td>
</tr>
<tr>
<td>TSYS Acquiring Solutions</td>
<td>Visa, Mastercard, American Express</td>
<td>Multiple partial captures</td>
</tr>
<tr>
<td>Visa Platform Connect. The supported acquirer is:</td>
<td>Visa, Mastercard, American Express, Discover, JCB, Diners Club</td>
<td>Recurring payments, Split shipments</td>
</tr>
<tr>
<td>Visa Platform Connect. The supported acquirer is:</td>
<td>Vantiv</td>
<td></td>
</tr>
<tr>
<td>Visa Platform Connect is a single processor with multiple acquirers.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Transaction Endpoints

**Test transactions:**
- Akamai endpoint: [http://ics2testa.ic3.com](http://ics2testa.ic3.com)

**Live transactions:**
- Akamai endpoint: [http://ics2a.ic3.com](http://ics2a.ic3.com)
Registration

Registering with Samsung

To register with Samsung:

Step 1 Create a profile by completing the merchant application on the Samsung Pay Partner Portal. Samsung will contact you if clarification is needed.

Step 2 After your merchant application is approved, you receive a unique partner ID. Include this ID in your application.

You need the partner ID in order to generate the Certificate Signing Request (CSR) file in the Business Center. See "Registering with Cybersource," page 12. Samsung requires the CSR file in order to encrypt sensitive payment data; it contains an identifier and public key.

Step 3 Using the Samsung Pay Partner Portal, upload the CSR file.

Step 4 Enter an application name and a package name.

Step 5 When you associate the CSR file with the application, Samsung generates a product ID.

Step 6 Create login details for application developers on the Samsung Pay Partner Portal.

Step 7 Download and integrate the Samsung Pay SDK into your application. See "Using the Samsung Pay SDK," page 15.

The SDK contains:

- A Javadoc
- The Samsung Pay SDK files samsungpay.jar and sdk-v1.0.0.jar
- A sample app
- The branding guide
- Image files
Chapter 2  Registration

Step 8  Register a Samsung account ID and request a debug-api-key file using the Samsung Pay Partner Portal. The debug-api-key file is valid for three months. See "Using the API Key," page 14.

The Samsung account ID, the debug-api-key, and the product ID are used to validate your application so that you can use the Samsung Pay SDK for testing.

Step 9  Submit your application for approval using the Samsung Pay Partner Portal. Upload the final version of the Android Application Package (APK) file using the Samsung Pay Partner Portal, and include screenshots of your checkout page displaying the Samsung Pay logo.

Registering with Cybersource

To register with Cybersource:

Step 1  Log in to the Business Center:
- Create a CSR file for test transactions: https://ebctest.cybersource.com/ebc2/
- Create a CSR file for live transactions: https://ebc2.cybersource.com/ebc2/

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

Step 3  Click Digital Payment Solutions. The Digital Payments page appears.

Step 4  Click Configure. The Samsung Pay Registration panel opens.

Step 5  Enter your Samsung partner ID.

Step 6  Click Generate New CSR.

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

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

Only one CSR is permitted for each unique Samsung partner ID. If you modify your Samsung partner ID you must generate a new CSR.

Step 9  Complete the enrollment process by submitting your CSR to Samsung.
Integrating the Samsung SDK

Creating a Project

To create a project using Android Studio:

Step 1 Download Android Studio.
Step 2 Open Android Studio and click Start a new Android Studio project.
Step 3 In the New Project settings, enter the name of your application and the company domain.
Step 4 To change the package name, click Edit. By default, Android Studio sets the last element of the project’s package name to the name of your application.
Step 5 Click Next.
Step 6 In the Target Android Devices settings, choose the required API levels.
Step 7 Click Next.
Step 8 Choose the required activity and click Finish.

Integrating the Samsung Pay SDK

To integrate the Samsung Pay SDK:

Step 1 Add the samsungpay.jar and sdk-v1.0.0.jar files to the libs folder of your Android project.
Step 2 Choose Gradle Scripts > build.gradle and enter the dependencies shown below.

```groovy
dependencies {
    compile files('libs/samsungpay.jar')
    compile files('libs/sdk-v1.0.0.jar')
}
```
Step 3  Import the package.

```java
import com.samsung.android.sdk.samsungpay;
```

Using the API Key

The API key is used to verify that your app (in debug mode or release mode) can use the Samsung Pay SDK APIs with the Samsung Pay application. To get the API key, you must create a `debug-api-key` file (Step 8 in "Registering with Samsung," page 11) and include it in the `manifest` file.

To use the API key:

Step 1  Include the API key in the `manifest` file with a custom tag. This enables the merchant app `android manifest` file to provide the `DebugMode, spay_debug_api_key` values as metadata.

**Example 1  Debug Mode**

```xml
<meta-data
    android:name="debug_mode"
    android:value="Y" />
<meta-data
    android:name="spay_debug_api_key"
    android:value="asdfggkndkeie17283094858" />
```

**Example 2  Release Mode**

```xml
<meta-data
    android:name="debug_mode"
    android:value="N" />
```
Eligibility

Initialize the SSamsungPay class to verify that your application is eligible for Samsung Pay and to display the Samsung Pay button to the customer (refer to branding guidelines).

The SSamsungPay class provides the following API methods:

- **initialize()**—initializes the Samsung Pay SDK and verifies eligibility for Samsung Pay, including the device, software, and business area.

  Request the initialize() API method of the SSamsungPay class before using the Samsung Pay SDK.

- **getVersionCode()**—retrieves the version number of the Samsung Pay SDK as an integer.

- **getVersionName()**—retrieves the version name of the Samsung Pay SDK as a string.

After the initialize() API method request is successful, display the Samsung Pay button to the customer.

If the initialize() API method request fails, the method displays a SsdkUnsupportedException or NullPointerException error.

- **SsdkUnsupportedException**—the device is not a Samsung device or does not support the Samsung Pay package.

- **NullPointerException**—the context passed is null.
Example 3 Samsung Pay Class

```java
SSamsungPay spay = new SSamsungPay();
try {
    spay.initialize(mContext);
} catch (SsdkUnsupportedException e1) {
    e1.printStackTrace();
    pay_button.setVisibility(View.INVISIBLE);
}
```

Payment Request

Initiating a Payment

To initiate a payment:

**Step 1** Include the following fields in the PaymentInfo class:

- Merchant Name—the merchant name as it appears on the payment sheet of Samsung Pay and customer's bank statement. This field is required.
- Amount—this field is required.
- Payment Protocol—3D Secure. This field is required.
- Permitted Card Brands—specify the card brands that are supported such as Visa, Mastercard, or American Express. This field is required.
- Merchant ID
- Order Number
- Shipping Address—this field is required if SEND_SHIPPING or NEED_BILLING_AND_SEND_SHIPPING is set for AddressVisibilityOption.
- Address Visibility Option
- Card Holder Name
- Recurring Option

If the required fields are not included, you receive a NullPointerException error.
**Example 4  Transaction Request Structure**

```java
private PaymentInfo makeTransactionDetails() {
    // Supported card brands
    ArrayList<CardInfo.Brand> brandList = new ArrayList<CardInfo.Brand>();
    if (visaBrand.isChecked())
        brandList.add(CardInfo.Brand.VISA);
    if (mcBrand.isChecked())
        brandList.add(CardInfo.Brand.Mastercard);
    if (amexBrand.isChecked())
        brandList.add(CardInfo.Brand.AMERICANEXPRESS);

    // Basic payment information
    PaymentInfo paymentReq = new PaymentInfo.Builder()
        .setMerchantId("merchantID")
        .setMerchantName("Test")
        .setAmount(getAmount())
        .setShippingAddress(getShippingAddressInfo())
        .setOrderNumber(orderNoView.getText().toString())
        .setPaymentProtocol(PaymentProtocol.PROTOCOL_3DS)
        .setAddressInPaymentSheet(AddressInPaymentSheet.DO_NOT_SHOW)
        .setAllowedCardBrands(brandList)
        .setRecurringEnabled(isRecurring)
        .setCardHolderNameEnabled(isCardHolderNameRequired)
        .build();
    return paymentReq;
}

// Add shipping address details
private Address getShippingAddressInfo() {
    Address address = new Address.Builder()
        .setAddressee(name.getText().toString())
        .setAddressLine1(addLine1.getText().toString())
        .setAddressLine2(addline2.getText().toString())
        .setCity(city.getText().toString())
        .setState(state.getText().toString())
        .setCountryCode(country.getSelectedItem().toString())
        .setPostalCode(zip.getText().toString()).build();
    return address;
}

// Add amount details private Amount getAmount() {
    Amount amount = new Amount.Builder()
        .setCurrencyCode(currencyType.getSelectedItem().toString())
        .setItemTotalPrice(productPrice.getText().toString())
        .setShippingPrice(shippingPrice.getText().toString())
        .setTax(taxPrice.getText().toString())
        .setTotalPrice(totalAmount.getText().toString()).build();
    return amount;
}
```
Chapter 4 Using the Samsung Pay SDK

Requesting a Payment

To request a payment:

Step 1 Use the startSamsungPay() API method in the PaymentManager class.

The PaymentManager class includes the following API methods:

- startSamsungPay() — requests to initiate payment with Samsung Pay.
- updateAmount() — updates the transaction amount if shipping address or card information is updated by Samsung Pay.
- updateAmountFailed() — returns an error code when the new amount cannot be updated because of a wrong address.

Step 2 Request the startSamsungPay() API method and include the following data:

- PaymentInfo — contains payment information.
- PID — the product ID created in the Samsung Pay Partner Portal. See "Registration," page 11.
- StatusListener — the result of the payment request is delivered to StatusListener. This listener should be registered before you call the startSamsungPay() API method.

When you request the startSamsungPay() API method, the Samsung Pay online payment sheet is displayed on your application. The customer selects a registered card for payment and can also update the billing and shipping address.

The payment reply is delivered as one of the following events to StatusListener:

- onSuccess() — this event is requested when Samsung Pay confirms the payment. It includes encryptedPaymentCredential in JSON format. See Table 2, "Encrypted Payment Credential," on page 19.
- onFailure() — this event is requested when the transaction fails. It returns an error code and error message.
public void onPayButtonClicked(View v) {
    // Call startSamsungPay() method of PaymentManager class.
    // To create a transaction request for makeTransactionDetails() in
    // the following code, see Example 4, "Transaction Request Structure,"
    // on page 17.
    try {
        mPaymentManager.startSamsungPay(makeTransactionDetails(), "enter
        product ID",
        mStatusListener);
    } catch (NullPointerException e) {
        e.printStackTrace();
    }
}

private PaymentManager.StatusListener mStatusListener = new
    PaymentManager.StatusListener() {
        @Override
        public void onFailure(int errCode, String msg) {
            Log.d(TAG, " onFailed");
        }
        @Override
        public void onSuccess(PaymentInfo arg0, String result) {
            Log.d(TAG, "onSuccess ");
        };
    }

Table 2  Encrypted Payment Credential

<table>
<thead>
<tr>
<th>Payment Credential</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>method</td>
<td>Payment protocol: 3D Secure.</td>
</tr>
<tr>
<td>merchant_ref</td>
<td>Merchant reference code.</td>
</tr>
<tr>
<td>billing_address.street</td>
<td>Number, street name.</td>
</tr>
<tr>
<td>billing_address.state_province</td>
<td>Two-letter state code.</td>
</tr>
<tr>
<td>billing_address.zip_postal_code</td>
<td>Five-character zip code.</td>
</tr>
<tr>
<td>billing_address.city</td>
<td>City name.</td>
</tr>
<tr>
<td>billing_address.county</td>
<td>Two letter country code.</td>
</tr>
<tr>
<td>3ds.type</td>
<td>S for Samsung Pay.</td>
</tr>
<tr>
<td>3ds.version</td>
<td>Current version 100.</td>
</tr>
<tr>
<td>3ds.data</td>
<td>Base64-encoded payment data.</td>
</tr>
</tbody>
</table>

For information on how to decrypt the encrypted payment credential, see:

https://pay.samsung.com/developers
Your payment processor can include API reply fields that are not documented in this guide. See Credit Card Services Using the SCMP API (PDF | HTML) for detailed descriptions of additional API reply fields.

**Merchant Decryption**

**Visa Transaction**


To request an authorization for a Visa transaction:

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

**Step 2** Set the `customer_cc_expmo` and `customer_cc_expyr` fields to the payment network token expiration date values.

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

**Step 4** Set the `network_token_cryptogram` field to the network token cryptogram.

**Step 5** Set the `payment_network_token_transaction_type` field to 1.

**Step 6** Set the `e_commerce_indicator` field to `internet`.

**Step 7** Set the `payment_solution` field to `008`.
Example 6  Merchant Decryption Authorization Request (Visa)

```plaintext
bill_address1=111 S. Division St.
bill_address2=Suite 123
bill_city=Ann Arbor
bill_country=US
bill_state=MI
bill_zip=48104-2201
currency=usd
customer_cc_expmo=12
customer_cc_expyr=2021
customer_cc_number=xxxx1000000xxxx
customer_email=demo@example.com
customer_firstname=James
customer_ipaddress=66.123.123.2
customer_lastname=Smith
customer_phone=999-999-9999
e_commerce_indicator=internet
grand_total_amount=100.00
ics_applications=ics_auth
merchant_id=demomerchant
merchant_ref_number=demorefnum
cavv=ABCDEFabcdefABCDEFabcdef0987654321234567
payment_network_token_transaction_type=1
solution_type=008
```
Mastercard Transaction


To request an authorization for a Mastercard transaction:

**Step 1** Set the **customer_cc_number** field to the payment network token value.

**Step 2** Set the **customer_cc_expmo** and **customer_cc_expyr** fields to the payment network token expiration date values.

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

**Step 4** Set the **network_token_cryptogram** field to the network token cryptogram.

**Step 5** Set the **ucaf_collection_indicator** field to 2.

**Step 6** Set the **payment_network_token_transaction_type** field to 1.

**Step 7** Set the **e_commerce_indicator** field to spa.

**Step 8** Set the **payment_solution** field to 008.

Example 8  Merchant Decryption Authorization Request (Mastercard)

```
bill_address1=111 S. Division St.
bill_address2=Suite 123
bill_city=Ann Arbor
bill_country=US
bill_state=MI
billZip=48104-2201
currency=usd
customer_cc_expmo=12
customer_cc_expyr=2021
customer_cc_number=xxxx55555555xxxx
customer_email=demo@example.com
customer_firstname=James
customer_lastname=Smith
customer_phone=999-999-9999
grand_total_amount=100.00
ics_applications=ics_auth
merchant_id=demomerchant
merchant_ref_number=demorefnum
ucaf_authentication_data=ABCDEFabcdefABCDEFabcdef0987654321234567
ucaf_collection_indicator=2
payment_network_token_transaction_type=1
solution_type=008
```
Example 9  Merchant Decryption Authorization Reply (Mastercard)

```plaintext
request_token=Ahj/7wSR5C/p6oJEy1gKIkJGLNkwcsMrWHi1U5tGHST/hHgZdACT/hVB3c
currency=usd
request_id=4465838340055000001541
auth_rflag=SOK
ics_rmsg=Request was processed successfully.
auth_auth_amount=100.00
auth_rcode=1
auth_trans_ref_no=13209255CGJSMQCR
auth_auth_code=888888
auth_rmsg=Request was processed successfully.
ics_rflag=SOK
auth_auth_response=100
auth_avs_raw=I1
auth_auth_time=2015-11-03T205035Z
merchant_ref_number=demorefnum
ics_rcode=1
```

American Express Transaction


To request an authorization for an American Express transaction:

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

**Step 2** Set the `customer_cc_expmo` and `customer_cc_expyr` fields to the payment network token expiration date values.

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

Include the whole 20-byte cryptogram in the `cavv` field. For a 40-byte cryptogram, split the cryptogram into two 20-byte binary values (block A and block B). Set the `cavv` field to the block A value and set the `xid` field to the block B value.

**Step 4** Set the `network_token_cryptogram` field to the network token cryptogram.

**Step 5** Set the `payment_network_token_transaction_type` field to 1.

**Step 6** Set the `e_commerce_indicator` field to `aesk`.

**Step 7** Set the `payment_solution` field to 008.
Example 10  Merchant Decryption Authorization Request (American Express)

```text
bill_address1=111 S. Division St.
bill_address2=Suite 123
bill_city=Ann Arbor
bill_country=US
bill_state=MI
bill_zip=48104-2201
currency=usd
customer_cc_expmo=12
customer_cc_expyr=2021
customer_cc_number=xxxx82246310xxxx
customer_email=demo@example.com
customer_firstname=James
customer_ipaddress=66.123.123.2
customer_lastname=Smith
customer_phone=999-999-9999
grand_total_amount=100.00
ics_applications=ics_auth
merchant_id=demomerchant
merchant_ref_number=demorefnum
cavv=ABCDEFabcdefABCDEFabcdef0987654321234567
xid=1234567890987654321ABCDEFabcdefABCDE123
payment_network_token_transaction_type=1
solution_type=008
```

Example 11  Merchant Decryption Authorization Reply (American Express)

```text
request_token=Ahj/7wSR5C/wGKx1xAKIkGLNkwsmaH/1U5tGHaT/hHgzcDT/h6BBL
currency=usd
request_id=4465839210285000001541
auth_rflag=SOK
ics_rmsg=Request was processed successfully.
auth_auth_amount=100.00
auth_rcode=1
auth_trans_ref_no=13209256CGJSMQCZ
auth_auth_code=888888
auth_rmsg=Request was processed successfully.
ics_rflag=SOK
auth_auth_response=100
auth_avs_raw=I1
auth_auth_time=2015-11-03T205202Z
merchant_ref_number=demorefnum
ics_rcode=1
```
JCB Transaction


To request an authorization for a JCB transaction:

**Step 1** Set the **customer_cc_number** field to the payment network token value.

**Step 2** Set the **customer_cc_expmo** and **customer_cc_expyr** fields to the payment network token expiration date values.

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

**Step 4** Set the **network_token_cryptogram** field to the network token cryptogram.

**Step 5** Set the **payment_network_token_transaction_type** field to 1.

**Step 6** Set the **eci_raw** field to the ECI value contained in the Samsung Pay reply message.

**Step 7** Set the **payment_solution** field to 008.

**Example 12 Merchant Decryption Authorization Request (JCB)**

```plaintext
bill_address1=123 Main Street
bill_address2=Suite 12345
bill_city=Small Town
bill_country=US
bill_state=CA
bill_zip=98765
card_type=007
currency=usd
customer_cc_expmo=12
customer_cc_expyr=2031
customer_cc_number=xxxx11111111xxxx
customer_email=js@example.com
customer_firstname=Jane
customer_lastname=Smith
customer_phone=999-999-9999
currency=usd
customer_cc_expmo=12
customer_cc_expyr=2031
customer_cc_number=xxxx11111111xxxx
customer_email=js@example.com
customer_firstname=Jane
customer_lastname=Smith
customer_phone=999-999-9999
currency=usd
customer_cc_expmo=12
customer_cc_expyr=2031
customer_cc_number=xxxx11111111xxxx
customer_email=js@example.com
customer_firstname=Jane
customer_lastname=Smith
customer_phone=999-999-9999
currency=usd
customer_cc_expmo=12
customer_cc_expyr=2031
customer_cc_number=xxxx11111111xxxx
customer_email=js@example.com
customer_firstname=Jane
customer_lastname=Smith
customer_phone=999-999-9999
currency=usd
customer_cc_expmo=12
customer_cc_expyr=2031
customer_cc_number=xxxx11111111xxxx
customer_email=js@example.com
customer_firstname=Jane
customer_lastname=Smith
customer_phone=999-999-9999
currency=usd
customer_cc_expmo=12
customer_cc_expyr=2031
customer_cc_number=xxxx11111111xxxx
customer_email=js@example.com
customer_firstname=Jane
customer_lastname=Smith
customer_phone=999-999-9999
currency=usd
customer_cc_expmo=12
customer_cc_expyr=2031
customer_cc_number=xxxx11111111xxxx
customer_email=js@example.com
customer_firstname=Jane
customer_lastname=Smith
customer_phone=999-999-9999
currency=usd
customer_cc_expmo=12
customer_cc_expyr=2031
customer_cc_number=xxxx11111111xxxx
customer_email=js@example.com
customer_firstname=Jane
customer_lastname=Smith
customer_phone=999-999-9999
currency=usd
customer_cc_expmo=12
customer_cc_expyr=2031
customer_cc_number=xxxx11111111xxxx
customer_email=js@example.com
customer_firstname=Jane
customer_lastname=Smith
customer_phone=999-999-9999
currency=usd
customer_cc_expmo=12
customer_cc_expyr=2031
customer_cc_number=xxxx11111111xxxx
customer_email=js@example.com
customer_firstname=Jane
customer_lastname=Smith
customer_phone=999-999-9999
currency=usd
customer_cc_expmo=12
customer_cc_expyr=2031
customer_cc_number=xxxx11111111xxxx
customer_email=js@example.com
customer_firstname=Jane
customer_lastname=Smith
customer_phone=999-999-9999
currency=usd
customer_cc_expmo=12
customer_cc_expyr=2031
customer_cc_number=xxxx11111111xxxx
customer_email=js@example.com
customer_firstname=Jane
customer_lastname=Smith
customer_phone=999-999-9999
currency=usd
customer_cc_expmo=12
customer_cc_expyr=2031
customer_cc_number=xxxx11111111xxxx
customer_email=js@example.com
customer_firstname=Jane
customer_lastname=Smith
customer_phone=999-999-9999
currency=usd
customer_cc_expmo=12
customer_cc_expyr=2031
customer_cc_number=xxxx11111111xxxx
customer_email=js@example.com
customer_firstname=Jane
customer_lastname=Smith
customer_phone=999-999-9999
currency=usd
customer_cc_expmo=12
customer_cc_expyr=2031
customer_cc_number=xxxx11111111xxxx
customer_email=js@example.com
customer_firstname=Jane
customer_lastname=Smith
customer_phone=999-999-9999
currency=usd
customer_cc_expmo=12
customer_cc_expyr=2031
customer_cc_number=xxxx11111111xxxx
customer_email=js@example.com
customer_firstname=Jane
customer_lastname=Smith
customer_phone=999-999-9999
currency=usd
customer_cc_expmo=12
customer_cc_expyr=2031
customer_cc_number=xxxx11111111xxxx
customer_email=js@example.com
customer_firstname=Jane
customer_lastname=Smith
customer_phone=999-999-9999
currency=usd
customer_cc_expmo=12
customer_cc_expyr=2031
customer_cc_number=xxxx11111111xxxx
customer_email=js@example.com
customer_firstname=Jane
customer_lastname=Smith
customer_phone=999-999-9999
currency=usd
customer_cc_expmo=12
customer_cc_expyr=2031
customer_cc_number=xxxx11111111xxxx
customer_email=js@example.com
customer_firstname=Jane
customer_lastname=Smith
customer_phone=999-999-9999
currency=usd
customer_cc_expmo=12
customer_cc_expyr=2031
customer_cc_number=xxxx11111111xxxx
customer_email=js@example.com
customer_firstname=Jane
customer_lastname=Smith
customer_phone=999-999-9999
currency=usd
customer_cc_expmo=12
customer_cc_expyr=2031
customer_cc_number=xxxx11111111xxxx
customer_email=js@example.com
customer_firstname=Jane
customer_lastname=Smith
customer_phone=999-999-9999
currency=usd
custume
Example 13   Merchant Decryption Authorization Reply (JCB)

```
auth_auth_amount=100.00
auth_auth_avs=X
auth_auth_code=888888
auth_auth_response=100
auth_avs_raw=11
auth_rcode=1
auth_rflag=SOK
auth_rmsg=Request was processed successfully.
auth_trans_ref_no=15356268CR2XF23X
currency=USD
ics_rcode=1
ics_rflag=SOK
ics_rmsg=Request was processed successfully.
merchant_ref_number=ref123
request_id=469739268106124601541
request_token=Ahj/7wSR/UoVm1bMmziHS2jMECT/h+KjMHSB04gwGA2dJqoxQAA6xd
```

Cybersource Decryption

Visa Transaction


To request an authorization for a Visa transaction:

**Step 1**  Set the `encrypted_payment_data` field to the value that was returned from Samsung Pay in the `3ds.data` block.

- a Retrieve the payment data from Samsung Pay in JSON Web Encryption (JWE) format.
- b Encode it in Base64.
- c Retrieve the corresponding Key ID (KID) with encryption and set the values as:

```json
{
  "publicKeyHash": "kid",
  "version": "100",
  "data": "encoded data from step b above"
}
```
d  Encode the structure in Base64.

e  Add the value to the `encrypted_payment_data` field.

**Step 2** Set the `encrypted_payment_descriptor` field to `Rk1EPUNPTU1PTi5TQU1TVU5HLk1OQVBQL1BBWU1FT1Q=`.

**Step 3** Set the `payment_network_token_transaction_type` field to 1.

**Step 4** Set the `e_commerce_indicator` field to `internet`.

**Step 5** Set the `payment_solution` field to 008.

**Example 14  Cybersource Decryption Authorization Request (Visa)**

```plaintext
bill_address1=111 S. Division St.
bill_address2=Suite 123
bill_city=Ann Arbor
bill_country=US
bill_state=MI
bill_zip=48104-2201
currency=usd
customer_email=demo@example.com
customer_firstname=James
customer_ipaddress=66.123.123.2
customer_lastname=Smith
customer_phone=999-999-9999
e_commerce_indicator=internet
grand_total_amount=100.00
ics_applications=ics_auth
merchant_id=demomerchant
merchant_ref_number=demorefnum
encrypted_payment_data=ABCDEFabcdefABCDEFabcdef0987654321234567
encrypted_payment_descriptor=Rk1EPUNPTU1PTi5TQU1TVU5HLk1OQVBQL1BBWU1FT1Q=
payment_network_transaction_type=1
solution_type=008
```
Example 15  Cybersource Decryption Authorization Reply (Visa)

```
request_token=Ahj/7wSR5C/kX6302hAKIkGLNkwcmrSHl1U5tGHRT/hHgzc8BT/hHgk
currency=usd
request_id=4465837560045000001541
auth_rflag=SOK
ics_rmsg=Request was processed successfully.
auth_auth_amount=100.00
auth_rcode=1
auth_trans_ref_no=13209254CGJSMQCQ
auth_auth_code=888888
auth_rmsg=Request was processed successfully.
ics_rflag=SOK
auth_auth_response=100
auth_avs_raw=11
merchant_ref_number=demorefnum
ics_rcode=1
token_prefix=294672
token_suffix=4397
token_expirationMonth=08
token_expirationYear=2021
```

Mastercard Transaction


To request an authorization for a Mastercard transaction:

**Step 1** Set the `encrypted_payment_data` field to the value that was returned from Samsung Pay in the `3ds.data` block.

- a Retrieve the payment data from Samsung Pay in JSON Web Encryption (JWE) format.

- b Encode it in Base64.

- c Retrieve the corresponding Key ID (KID) with encryption and set the values as:

```
{
  "publicKeyHash": "kid",
  "version": "100",
  "data": "encoded data from step b above"
}
```
**d** Encode the structure in Base64.

**e** Add the value to the `encrypted_payment_data` field.

**Step 2** Set the `encrypted_payment_descriptor` field to

```
Rk1EPUNPTU1PTi5TQU1TVU5HLk1OQVBQLlBBWU1FTlQ=.
```

**Step 3** Set the `e_commerce_indicator` field to `spa`.

**Step 4** Set the `payment_network_token_transaction_type` field to `1`.

**Step 5** Set the `payment_solution` field to `008`.

**Example 16** Cybersource Decryption Authorization Request (Mastercard)

```plaintext
bill_address1=111 S. Division St.
bill_address2=Suite 123
bill_city=Ann Arbor
bill_country=US
bill_state=MI
bill_zip=48104-2201
currency=usd
customer_email=demo@example.com
customer_firstname=James
customer_lastname=Smith
customer_ipaddress=66.123.123.2
customer_phone=999-999-9999
e_commerce_indicator=spa
grand_total_amount=100.00
ics_applications=ics_auth
merchant_id=demomerchant
merchant_ref_number=demorefnum
encrypted_payment_data=ABCDEFabcdefABCDEFabcdef0987654321234567
encrypted_payment_descriptor=Rk1EPUNPTU1PTi5TQU1TVU5HLk1OQVBQLlBBWU1FTlQ=
payment_network_transaction_type=1
solution_type=008
```
Example 17  Cybersource Decryption Authorization Reply (Mastercard)

```plaintext
request_token=Ahj/7wSR5C/p6oJEy1gKIkGLNkwcmrWHH1U5tGHST/hHgzdACT/hVB3c
request_id=4465838340055000001541
auth_rflag=SOK
ics_rmsg=Request was processed successfully.
auth_auth_amount=100.00
auth_rcode=1
auth_trans_ref_no=13209255CGJSMQCR
auth_auth_code=888888
auth_rmsg=Request was processed successfully.
ics_rflag=SOK
auth_auth_response=100
auth_avs_raw=I1
auth_auth_time=2015-11-03T205035Z
merchant_ref_number=demorefnum
ics_rcode=1
token_prefix=128945
token_suffix=2398
token_expirationMonth=08
token_expirationYear=2021
```

American Express Transaction


To request an authorization for an American Express transaction:

**Step 1** Set the `encrypted_payment_data` field to the value that was returned from Samsung Pay in the `3ds.data` block.

- **a** Retrieve the payment data from Samsung Pay in JSON Web Encryption (JWE) format.
- **b** Encode it in Base64.
- **c** Retrieve the corresponding Key ID (KID) with encryption and set the values as:

```plaintext
{
    "publicKeyHash": "kid",
    "version": "100",
    "data": "encoded data from step b above"
}
```
d  Encode the structure in Base64.

e  Add the value to the `encrypted_payment_data` field.

**Step 2**  Set the `encrypted_payment_descriptor` field to

```
Rk1EPUNPT1PTt5QTV5HLk1QvQgQL1BBWU1FTlQ=.
```

**Step 3**  Set the `payment_network_token_transaction_type` field to 1.

**Step 4**  Set the `e_commerce_indicator` field to `aesk`.

**Step 5**  Set the `payment_solution` field to 008.

---

**Example 18  Cybersource Decryption Authorization Request (American Express)**

```plaintext
bill_address1=111 S. Division St.
bill_address2=Suite 123
bill_city=Ann Arbor
bill_country=US
bill_state=MI
bill_zip=48104-2201
card_type=003
currency=usd
customer_email=demo@example.com
customer_firstname=James
customer_ipaddress=66.123.123.2
customer_lastname=Smith
customer_phone=999-999-9999
e_commerce_indicator=aesk
grand_total_amount=100.00
ics_applications=ics_auth
merchant_id=demomerchant
merchant_ref_number=demorefnum
encrypted_payment_data=ABCDEFabcdefABCDEFabcdef0987654321234567
encrypted_payment_descriptor=Rk1EPUNPT1PTt5QTV5HLk1QvQgQL1BBWU1FTlQ=
payment_network_transaction_type=1
solution_type=008
```
Example 19  Cybersource Decryption Authorization Reply (American Express)

```plaintext
request_token=Ahj/7wSR5C/wGZKh1xAKIkGLNkwsmaHH1U5tGHaT/hHgzcDT/h6BBL
currency=usd
request_id=4465839210285000001541
auth_rflag=SOK
ics_rmsg=Request was processed successfully.
auth_auth_amount=100.00
auth_rcode=1
auth_trans_ref_no=13209256CGJSMQCU
auth_auth_code=888888
auth_rmsg=Request was processed successfully.
ics_rflag=SOK
auth_auth_response=100
auth_avs_raw=I1
auth_auth_time=2015-11-03T205202Z
merchant_ref_number=demorefnm
ics_rcode=1
token_prefix=593056
token_suffix=0842
token_expirationMonth=08
token_expirationYear=2021
```

JCB Transaction


To request an authorization for a JCB transaction:

**Step 1** Set the encrypted_payment_data field to the Base64-encoded value obtained from the paymentData property of the PKPaymentToken object.

**Step 2** Set the encrypted_payment_descriptor field to Rk1EPUNPTU1PTi5TQU1TVU5HLk1OQVBQLBBWU1FTI=.

**Step 3** Set the payment_solution field to 008.
Example 20  Cybersource Decryption Authorization Request (JCB)

```
bill_address1=123 Main Street
bill_address2=Suite 12345
bill_city=Small Town
bill_country=US
bill_state=CA
bill_zip=98765
card_type=007
currency=usd
customer_cc_expmo=12
customer_cc_expyr=2031
customer_cc_number=xxxx55555555xxxx
customer_email=js@example.com
customer_firstname=Jane
customer_lastname=Smith
customer_phone=999-999-9999
currency=USD
```
Additional Cybersource Services

Refer to *Credit Card Services Using the SCMP API* (PDF | HTML) for information on how to request these follow-on services.

### Table 3  Cybersource Services

<table>
<thead>
<tr>
<th>Cybersource Service</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capture</td>
<td>A follow-on service that uses the request ID returned from the previous authorization. The request ID links the capture to the authorization. This service transfers funds from the customer’s account to your bank and usually takes two to four days to complete.</td>
</tr>
<tr>
<td>Sale</td>
<td>A sale is a bundled authorization and capture. Request the authorization and capture services at the same time. Cybersource processes the capture immediately.</td>
</tr>
<tr>
<td>Authorization Reversal</td>
<td>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.</td>
</tr>
</tbody>
</table>
Data Type Definitions

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

Table 4  Data Type Definitions

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date and time</td>
<td>Format is YYYY-MM-DDThhmmssZ, where:</td>
</tr>
<tr>
<td></td>
<td>▪ T separates the date and the time</td>
</tr>
<tr>
<td></td>
<td>▪ Z indicates Coordinated Universal Time (UTC), also known as Greenwich Mean Time (GMT)</td>
</tr>
<tr>
<td>Decimal</td>
<td>Number that includes a decimal point</td>
</tr>
<tr>
<td></td>
<td><strong>Example</strong> 23.45, -0.1, 4.0, 90809.0468</td>
</tr>
<tr>
<td>Integer</td>
<td>Whole number {..., -3, -2, -1, 0, 1, 2, 3, ...}</td>
</tr>
<tr>
<td>Nonnegative integer</td>
<td>Whole number greater than or equal to zero {0, 1, 2, 3, ...}</td>
</tr>
<tr>
<td>Positive integer</td>
<td>Whole number greater than zero {1, 2, 3, ...}</td>
</tr>
<tr>
<td>String</td>
<td>Sequence of letters, numbers, spaces, and special characters</td>
</tr>
</tbody>
</table>

Relaxed Requirements for Address Data

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

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

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Used By: Required (R) or Optional (O)</th>
<th>Data Type (Length)</th>
</tr>
</thead>
<tbody>
<tr>
<td>bill_address1</td>
<td>First line of the billing street address.</td>
<td>ics_auth</td>
<td>String (60)</td>
</tr>
<tr>
<td></td>
<td><strong>Important</strong> It is your responsibility to determine whether a field is required for the transaction you are requesting.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>See &quot;Relaxed Requirements for Address Data,&quot; page 35.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bill_address2</td>
<td>Additional address information.</td>
<td>ics_auth (R)</td>
<td>String (60)</td>
</tr>
<tr>
<td></td>
<td><strong>Example</strong> Attention: Accounts Payable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bill_city</td>
<td>City of the billing address.</td>
<td>ics_auth</td>
<td>String (50)</td>
</tr>
<tr>
<td></td>
<td><strong>Important</strong> It is your responsibility to determine whether a field is required for the transaction you are requesting.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>See &quot;Relaxed Requirements for Address Data,&quot; page 35.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bill_country</td>
<td>Country of the billing address. Use the two-character ISO Standard Country Codes.</td>
<td>ics_auth</td>
<td>String (2)</td>
</tr>
<tr>
<td></td>
<td><strong>Important</strong> It is your responsibility to determine whether a field is required for the transaction you are requesting.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>See &quot;Relaxed Requirements for Address Data,&quot; page 35.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bill_state</td>
<td>State or province of the billing address. For an address in the U.S. or Canada, use the State, Province, and Territory Codes for the United States and Canada.</td>
<td>ics_auth</td>
<td>String (2)</td>
</tr>
<tr>
<td></td>
<td><strong>Important</strong> It is your responsibility to determine whether a field is required for the transaction you are requesting.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>See &quot;Relaxed Requirements for Address Data,&quot; page 35.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1 The TC 33 Capture file contains information about the purchases and refunds that a merchant submits to Cybersource. Visa Platform Connect creates the TC 33 Capture file at the end of the day and sends it to the merchant’s acquirer, who uses this information to facilitate end-of-day clearing processing with payment card companies.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Used By: Required (R) or Optional (O)</th>
<th>Data Type (Length)</th>
</tr>
</thead>
<tbody>
<tr>
<td>bill_zip</td>
<td>Postal code for the billing address. The postal code must consist of 5 to 9 digits.</td>
<td>ics_auth (See description)</td>
<td>String (9)</td>
</tr>
<tr>
<td></td>
<td>When the billing country is the U.S., the 9-digit postal code must follow this format: [5 digits][dash][4 digits]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Example</strong> 12345-6789</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>When the billing country is Canada, the 6-digit postal code must follow this format: [alpha][numeric][alpha][space] [numeric][alpha][numeric]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Example</strong> A1B 2C3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Important</td>
<td>It is your responsibility to determine whether a field is required for the transaction you are requesting.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>See &quot;Relaxed Requirements for Address Data,&quot; page 35.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cavv</td>
<td><strong>Visa</strong> Cryptogram for payment network tokens transactions. The value for this field must be 28-character base64 or 40-character hex binary. All cryptograms use one of these formats.</td>
<td>ics_auth (R)</td>
<td>String (40)</td>
</tr>
<tr>
<td></td>
<td><strong>American Express</strong> For a 20-byte cryptogram, set this field to the cryptogram for payment network tokens transactions. For a 40-byte cryptogram, set this field to block A of the cryptogram for payment network tokens transactions. The value for this field must be 28-character base64 or 40-character hex binary. All cryptograms use one of these formats.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>currency</td>
<td>Currency used for the order: USD</td>
<td>ics_auth (R)</td>
<td>String (5)</td>
</tr>
<tr>
<td>customer_cc_expmo</td>
<td>Two-digit month in which the payment network token expires. Format: MM. Possible values: 01 through 12.</td>
<td>ics_auth (R)</td>
<td>String (2)</td>
</tr>
<tr>
<td>customer_cc_expyr</td>
<td>Four-digit year in which the payment network token expires. Format: YYYY.</td>
<td>ics_auth (R)</td>
<td>Nonnegative integer (4)</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Used By: Required (R) or Optional (O)</th>
<th>Data Type (Length)</th>
</tr>
</thead>
<tbody>
<tr>
<td>customer_cc_number</td>
<td>The payment network token value. This value is obtained by decrypting the customer's encrypted payment data.</td>
<td>ics_auth (R)</td>
<td>Nonnegative integer (20)</td>
</tr>
<tr>
<td>customer_email</td>
<td>Customer’s email address.</td>
<td>ics_auth (See description)</td>
<td>String (255)</td>
</tr>
<tr>
<td>customer_firstname</td>
<td>Customer’s first name. For a credit card transaction, this name must match the name on the card.</td>
<td>ics_auth (See description)</td>
<td>String (60)</td>
</tr>
<tr>
<td>customer_lastname</td>
<td>Customer’s last name. For a credit card transaction, this name must match the name on the card.</td>
<td>ics_auth (See description)</td>
<td>String (60)</td>
</tr>
<tr>
<td>customer_phone</td>
<td>Customer’s phone number. It is recommended that you include the country code when the order is from outside the U.S.</td>
<td>ics_auth (O)</td>
<td>String (15)</td>
</tr>
<tr>
<td>eci_raw</td>
<td>Raw electronic commerce indicator (ECI).</td>
<td>ics_auth</td>
<td>String (2)</td>
</tr>
<tr>
<td>directory_server_</td>
<td>Identifier generated during the authentication transaction by the Mastercard Directory Server and passed back with the authentication results.</td>
<td>ics_auth (O)</td>
<td>String (36)</td>
</tr>
<tr>
<td>transaction_id</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1 The TC 33 Capture file contains information about the purchases and refunds that a merchant submits to Cybersource. Visa Platform Connect creates the TC 33 Capture file at the end of the day and sends it to the merchant’s acquirer, who uses this information to facilitate end-of-day clearing processing with payment card companies.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Used By:</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>e_commerce_indicator</td>
<td>For a payment network tokens transaction.</td>
<td>ics_auth (O)</td>
<td>String (20)</td>
</tr>
<tr>
<td></td>
<td>Possible values:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• aesk: American Express card type</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• spa: Mastercard card type</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• internet: Visa card type</td>
<td></td>
<td></td>
</tr>
<tr>
<td>encrypted_payment_data</td>
<td>The encrypted payment data value.</td>
<td>ics_auth (R)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>If you are using the Cybersource Decryption option, populate this field</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>with the encrypted payment data value returned from Samsung Pay in the</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3ds.data block.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>encrypted_payment_descriptor</td>
<td>Format of the encrypted payment data. The value for Samsung Pay is</td>
<td>ics_auth (R)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Rk1EPUNPTU1PTi5TQU1TVU5HLk10QVBQL1BBWU1FT1Q=</td>
<td></td>
<td></td>
</tr>
<tr>
<td>grand_total_amount</td>
<td>Grand total for the order. This value cannot be negative. You can include</td>
<td>ics_auth (R)</td>
<td>Decimal (15)</td>
</tr>
<tr>
<td></td>
<td>a decimal point (.), but you cannot include any other special</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>characters. Cybersource truncates the amount to the correct number of</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>decimal places.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ics_applications</td>
<td>Cybersource services to process for the request:</td>
<td>ics_auth (R)</td>
<td>String (255)</td>
</tr>
<tr>
<td></td>
<td>ics_auth</td>
<td></td>
<td></td>
</tr>
<tr>
<td>merchant_id</td>
<td>Your Cybersource merchant ID. Use the same merchant ID for evaluation,</td>
<td>ics_auth (R)</td>
<td>String (30)</td>
</tr>
<tr>
<td></td>
<td>testing, and production.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>merchant_ref_number</td>
<td>Merchant-generated order reference or tracking number. Cybersource</td>
<td>ics_auth (R)</td>
<td>String (50)</td>
</tr>
<tr>
<td></td>
<td>recommends that you send a unique value for each transaction so that you</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>can perform meaningful searches for the transaction.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>For information about tracking orders, see Getting Started with Cybersource Advanced for the SCMP API (PDF</td>
<td>HTML).</td>
<td></td>
</tr>
</tbody>
</table>

1 The TC 33 Capture file contains information about the purchases and refunds that a merchant submits to Cybersource. Visa Platform Connect creates the TC 33 Capture file at the end of the day and sends it to the merchant’s acquirer, who uses this information to facilitate end-of-day clearing processing with payment card companies.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Used By: Required (R) or Optional (O)</th>
<th>Data Type (Length)</th>
</tr>
</thead>
<tbody>
<tr>
<td>merchant_url</td>
<td>URL or reverse domain name for your business. This field is supported only for Mastercard digital secure remote payment (DSRP) transactions on FDC Compass and Visa Platform Connect.</td>
<td>ics_auth (R for Mastercard DSRP transactions on FDC Compass and Visa Platform Connect)</td>
<td>String (150)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Special symbols allowed:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Forward slash (/)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Colon (: )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Period (.)</td>
</tr>
<tr>
<td>network_token_cryptogram</td>
<td>Token authentication verification value cryptogram. For token-based transactions with 3D Secure or Identity Check, you must submit both types of cryptograms: network token and 3D Secure/Identity Check.</td>
<td>ics_auth (O)</td>
<td>String (40)</td>
</tr>
<tr>
<td></td>
<td>The value for this field must be 28-character Base64 or 40-character hex binary. All cryptograms use one of these formats.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pa_specification_version</td>
<td>The 3D Secure version that you used for strong customer authentication (SCA); for example, 3D Secure 1.0.2 or 2.0.0.</td>
<td>ics_auth (O)</td>
<td>String (20)</td>
</tr>
<tr>
<td>payment_network_token_assurance_level</td>
<td>Confidence level of the tokenization. This value is assigned by the token service provider.</td>
<td>ics_auth (O)</td>
<td>String (2)</td>
</tr>
<tr>
<td></td>
<td>This field is supported only for FDC Nashville Global.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>payment_network_token_device_tech_type</td>
<td>Type of technology used in the device to store token data. Possible value: 002: Host card emulation (HCE)</td>
<td>ics_auth (O)</td>
<td>Integer (3)</td>
</tr>
<tr>
<td></td>
<td>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.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>This field is supported only for FDC Compass.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Used By: Required (R) or Optional (O)</th>
<th>Data Type (Length)</th>
</tr>
</thead>
<tbody>
<tr>
<td>payment_network_token_requestor_id</td>
<td>Value that identifies your business and indicates that the cardholder’s account number is tokenized. This value is assigned by the token service provider and is unique within the token service provider’s database.</td>
<td>ics_auth (O)</td>
<td>Integer (1)</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> This field is supported only for FDC Nashville Global and Chase Paymentech Solutions.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>payment_network_token_transaction_type</td>
<td>Type of transaction that provided the token data. This value does not specify the token service provider; it specifies the entity that provided you with information about the token. Set the value for this field to 1.</td>
<td>ics_auth (R)</td>
<td>String (1)</td>
</tr>
<tr>
<td>payment_solution</td>
<td>Identifies Samsung Pay as the payment solution that is being used for the transaction: Set the value for this field to 008. <strong>Note</strong> This unique ID differentiates digital solution transactions within the Cybersource platform for reporting purposes.</td>
<td>ics_auth (R)</td>
<td>String (3)</td>
</tr>
<tr>
<td>surcharge_amount</td>
<td>The surcharge amount is included in the total transaction amount but is passed in a separate field to the issuer and acquirer for tracking. The issuer can provide information about the surcharge amount to the customer. This field is supported only for Visa Platform Connect.</td>
<td>ics_auth (O)</td>
<td>String (15)</td>
</tr>
<tr>
<td>ucaf_authentication_data</td>
<td>Cryptogram for payment network tokens transactions with Mastercard.</td>
<td>ics_auth (R)</td>
<td>String (32)</td>
</tr>
<tr>
<td>ucaf_collection_indicator</td>
<td>Required field for payment network tokens transactions with Mastercard. Set the value for this field to 2.</td>
<td>ics_auth (R)</td>
<td>String with numbers only (1)</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Used By: Required (R) or Optional (O)</th>
<th>Data Type (Length)</th>
</tr>
</thead>
<tbody>
<tr>
<td>xid</td>
<td><strong>American Express</strong>&lt;br&gt;For a 20-byte cryptogram, set this field to the cryptogram for payment network tokens transactions. For a 40-byte cryptogram, set this field to block A of the cryptogram for payment network tokens transactions. See &quot;Merchant Decryption,&quot; page 20. The value for this field must be 28-character base64 or 40-character hex binary. All cryptograms use one of these formats.</td>
<td>ics_auth (R)</td>
<td>String (40)</td>
</tr>
</tbody>
</table>

---

1 The TC 33 Capture file contains information about the purchases and refunds that a merchant submits to Cybersource. Visa Platform Connect 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.
# Offer-Level Fields

Table 6  Offer-Level Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Used By: Required (R) or Optional (O)</th>
<th>Data Type (Length)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>amount</strong></td>
<td>Per-item price of the product. This value cannot be negative. You can include a decimal point (.), but you cannot include any other special characters.</td>
<td>ics_auth (See description)</td>
<td>Decimal (15)</td>
</tr>
<tr>
<td><strong>merchant_product_sku</strong></td>
<td>Identification code for the product. This field is required when the product_code value is not default or one of the values related to shipping and/or handling.</td>
<td>ics_auth (See description)</td>
<td>String (255)</td>
</tr>
<tr>
<td><strong>product_code</strong></td>
<td>Type of product. This value is used to determine the product category: electronic, handling, physical, service, or shipping. The default is default.</td>
<td>ics_auth (See description)</td>
<td>String (255)</td>
</tr>
<tr>
<td><strong>product_name</strong></td>
<td>Name of the product. This field is required when the product_code value is not default or one of the values related to shipping and/or handling.</td>
<td>ics_auth (See description)</td>
<td>String (255)</td>
</tr>
<tr>
<td><strong>quantity</strong></td>
<td>The default is 1. This field is required when the product_code value is not default or one of the values related to shipping and/or handling.</td>
<td>ics_auth (See description)</td>
<td>Integer (10)</td>
</tr>
<tr>
<td><strong>tax_amount</strong></td>
<td>Total tax to apply to the product. This value cannot be negative.</td>
<td>ics_auth (See description)</td>
<td>String (15)</td>
</tr>
</tbody>
</table>
### Reply Fields

Because Cybersource can add reply fields, reply codes, and reply flags 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 reply codes and reply flags without problems.
- Your error handler should use the `ics_rcode` field to determine the result if it receives a reply flag that it does not recognize.

Your payment processor can include additional API reply fields that are not documented in this guide. See *Credit Card Services Using the SCMP API (PDF | HTML)* for detailed descriptions of additional API reply fields.

Table 7  Reply Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Returned By</th>
<th>Data Type &amp; Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>auth_auth_amount</td>
<td>Amount that was authorized.</td>
<td>ics_auth</td>
<td>Decimal (15)</td>
</tr>
<tr>
<td>auth_auth_avs</td>
<td>AVS result code. See *Credit Card Services Using the SCMP API (PDF</td>
<td>HTML)* for a detailed list of AVS values.</td>
<td>ics_auth</td>
</tr>
<tr>
<td>auth_auth_code</td>
<td>Authorization code. Returned only when the processor returns this value.</td>
<td>ics_auth</td>
<td>String (7)</td>
</tr>
<tr>
<td>auth_auth_response</td>
<td>For most processors, this value is the error message sent directly from the bank. Returned only when the processor returns this value.</td>
<td>ics_auth</td>
<td>String (10)</td>
</tr>
<tr>
<td>auth_auth_time</td>
<td>Time of authorization in UTC. See &quot;Data Type Definitions,&quot; page 35.</td>
<td>ics_auth</td>
<td>Date and time (20)</td>
</tr>
<tr>
<td>auth_avs_raw</td>
<td>AVS result code sent directly from the processor. Returned only when the processor returns this value.</td>
<td>ics_auth</td>
<td>String (10)</td>
</tr>
<tr>
<td>auth_rcode</td>
<td>Indicates whether the service request was successful. Possible values:</td>
<td>ics_auth</td>
<td>Integer (1)</td>
</tr>
<tr>
<td></td>
<td>- 1: An error occurred.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- 0: The request was declined.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- 1: The request was successful.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 7  Reply Fields (Continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Returned By</th>
<th>Data Type &amp; Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>auth_rflag</td>
<td>One-word description of the result of the entire request. See <em>Credit Card Services Using the SCMP API</em> (PDF</td>
<td>ics_auth</td>
<td>String (50)</td>
</tr>
<tr>
<td></td>
<td>Using the SCMP API* (PDF</td>
<td>HTML) for a detailed list of rflag values.</td>
<td></td>
</tr>
<tr>
<td>auth_rmsg</td>
<td>Message that explains the reply flag auth_rflag. Do not display this message to the customer, and do not use this field to write an error handler.</td>
<td>ics_auth</td>
<td>String (255)</td>
</tr>
<tr>
<td>auth_trans_ref_no</td>
<td>Reference number for the transaction. This value is not returned for all processors.</td>
<td>ics_auth</td>
<td>String (60)</td>
</tr>
<tr>
<td>card_suffix</td>
<td>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. This field is returned only for FDC Nashville Global.</td>
<td>ics_auth</td>
<td>String (4)</td>
</tr>
<tr>
<td>currency</td>
<td>Currency used for the order. For the possible values, see the <em>ISO Standard Currency Codes</em>.</td>
<td>ics_auth</td>
<td>String (5)</td>
</tr>
<tr>
<td>directory_server_transaction_id</td>
<td>Identifier generated during the authentication transaction by the Mastercard Directory Server and passed back with the authentication results.</td>
<td>pa_enroll (O)</td>
<td>String (36)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>pa_validate (O)</td>
<td></td>
</tr>
<tr>
<td>icsrcode</td>
<td>Indicates whether the service request was successful. Possible values:</td>
<td>ics_auth</td>
<td>Integer (1)</td>
</tr>
<tr>
<td></td>
<td>▪ -1: An error occurred.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>▪ 0: The request was declined.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>▪ 1: The request was successful.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ics_rflag</td>
<td>One-word description of the result of the entire request. See <em>Credit Card Services Using the SCMP API</em> (PDF</td>
<td>ics_auth</td>
<td>String (50)</td>
</tr>
<tr>
<td></td>
<td>Using the SCMP API* (PDF</td>
<td>HTML) for a detailed list of rflag values.</td>
<td></td>
</tr>
<tr>
<td>ics_rmsg</td>
<td>Message that explains the reply flag ics_rflag. Do not display this message to the customer, and do not use this field to write an error handler.</td>
<td>ics_auth</td>
<td>String (255)</td>
</tr>
</tbody>
</table>
### Table 7  Reply Fields (Continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Returned By</th>
<th>Data Type &amp; Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>merchant_ref_number</td>
<td>Order reference or tracking number that you provided in the request. If you included multi-byte characters in this field in the request, the returned value might include corrupted characters.</td>
<td>ics_auth</td>
<td>String (50)</td>
</tr>
<tr>
<td>request_id</td>
<td>Identifier for the request.</td>
<td>ics_auth</td>
<td>String (26)</td>
</tr>
<tr>
<td>request_token</td>
<td>Request token data created by Cybersource for each reply. The field is an encoded string that contains no confidential information such as an account or card verification number. The string can contain a maximum of 256 characters.</td>
<td>ics_auth</td>
<td>String (256)</td>
</tr>
</tbody>
</table>
| token_expiration_month | 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. | ics_auth      | String (2)        |
| token_expiration_year | 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. | ics_auth      | String (4)        |
| token_prefix        | First 6 digits of token. Cybersource includes this field in the reply message when it decrypts the payment blob for the tokenized transaction. | ics_auth      | String (6)        |
| token_suffix        | Last 4 digits of token. Cybersource includes this field in the reply message when it decrypts the payment blob for the tokenized transaction. | ics_auth      | String (4)        |