Payments

Simple Order API
Credit Mutuel-CIC

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

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 Unpublished rights reserved under the copyright laws of the United States.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

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

25.05.01

Contents

Payments Developer Guide

This section describes how to use this guide and where to find further information.

Audience and Purpose

This guide is written for application developers who want to use the Simple Order API to integrate payment card processing into an order management system.
Implementing the Cybersource payment services requires software development skills. You must write code that uses the API request and response fields to integrate the credit card services into your existing order management system.

Conventions

These statements appear in this document:

Important

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

Warning

A Warning contains information or instructions, which, if not heeded, can result in a security risk, irreversible loss of data, or significant cost in time or revenue or both.

Related Documentation	Visit the *Cybersource documentation hub* to find additional processor-specific versions of this guide and additional technical documentation.
Customer Support	**For support information about any service, visit the Support Center:** *http://support.visaacceptance.com*

Recent Revisions to This Document

25.05.01

International Transaction Compliance	**Added a section about international transaction compliance. See *Compliance* on page 13.**

25.04.01

This revision contains only editorial changes and no technical updates.

25.03

This revision contains only editorial changes and no technical updates.

25.02

This revision contains only editorial changes and no technical updates.

25.01

Added a testing section. See *Testing the Payment Services* on page 21.

Credentialed Transactions	**Removed Mastercard required field for retrieving customer credentials during a CIT request. See *Card-Specific Required Field for Retrieving Customer Credentials During a CIT* on page 79.**

24.14

This revision contains only editorial changes and no technical updates.

24.13

This revision contains only editorial changes and no technical updates.

24.12

This revision contains only editorial changes and no technical updates.

24.11

This revision contains only editorial changes and no technical updates.

24.10

This revision contains only editorial changes and no technical updates.

24.09

This revision contains only editorial changes and no technical updates.

24.08

This revision contains only editorial changes and no technical updates.

24.07

Pre-Authorizations	**Added pre-authorization processing. See *Pre-Authorizations* on page 41.**

Introduction to Payments

This introduction provides the basic information that you will need to successfully process payment transactions. It also provides an overview of the payments industry and provides workflows for each process.
With Cybersource payment services, you can process payment cards (tokenized or non-tokenized), digital payments such as Apple Pay and Google Pay, and customer ID transactions. You can process payments across the globe and across multiple channels with scalability and security. Cybersource supports a large number of payment cards and offers a wide choice of gateways and financial institutions, all through one connection.
Visit the *Cybersource documentation hub* to find additional processor-specific versions of this guide and additional technical documentation.

Financial Institutions and Payment Networks

Financial institutions and payment networks enable payment services. These entities work together to complete the full payment cycle.

Merchant Financial Institutions (Acquirers)

A merchant financial institution, also known as an acquirer, offers accounts to businesses that accept payment cards. Before you can accept payments, you must have a merchant account from an acquirer. Your merchant account must be configured to process card-not-present, card-present, or mail-order/telephone-order (MOTO) transactions.
Each acquirer has connections to a limited number of payment processors. You must choose a payment processor that your acquirer supports.
You can expect your acquirer to charge these fees:

- Discount rates: your acquirer charges a fee and collects a percentage of every transaction. The combination of the fee and the percentage is called the discount rate. These charges can be bundled (combined into a single charge) or unbundled (charged separately).

- Interchange fees: payment networks, such as Visa or Mastercard, each have a base fee, called the interchange fee, for each type of transaction. Your acquirer and processor can show you ways to reduce this fee.
- Chargebacks: when cardholders dispute charges, you can incur chargebacks. A chargeback occurs when a charge on a customer's account is reversed. Your acquirer removes the money from your account and could charge you a fee for processing the chargeback.

Take these precautions to prevent chargebacks:

- Use accurate merchant descriptors so that customers can recognize the transactions on their statements.
- Provide good customer support.
- Ensure rapid problem resolution.
- Maintain a high level of customer satisfaction.
- Minimize fraudulent transactions.

If excessive chargebacks or fraudulant changes occur, these actions might be taken:

- You might be required to change your business processes to reduce the number chargebacks, fraud, or both.
- Your acquiring institution might increase your discount rate.
- Your acquiring institution might revoke your merchant account.

Contact your sales representative for information about products that can help prevent fraud.

Customer Financial Institutions (Issuers)

A customer financial institution, also known as an issuer, provides payment cards to and underwrites lines of credit for their customers. The issuer provides monthly statements and collects payments. The issuer must follow the rules of the payment card companies to which they belong.

Payment Networks

Payment networks manage communications between acquiring financial institutions and issuing financial institutions. They also develop industry standards, support their brands, and establish fees for acquiring institutions.
Some payment networks, such as Visa, Mastercard, and UnionPay International, are trade associations that do not issue cards. Issuers are members of these associations, and they issue cards under license from the association.
Other networks, such as Discover and American Express, issue their own cards. Before you process cards from these companies, you must sign agreements with them.

Payment Processors

Payment processors connect with acquirers. Before you can accept payments, you must register with a payment processor. An acquirer might require you to use a payment processor with an existing relationship with the acquirer.

Your payment processor assigns one or more merchant IDs (MIDs) to your business. These unique codes identify your business during payment transactions.
This table lists the processors and corresponding card types that are supported for payment services.

Important

Only the card types explicitly listed here are supported.

Payment Processors and Supported Card Types

Payment Processor	Supported Card Types	Notes
Credit Mutuel-CIC	Visa, Mastercard, Cartes Bancaires	

Card Types

You can process payments with these kinds of cards:

- Credit cards
- Debit cards

Credit Cards

Cardholders use credit cards to borrow money from issuing banks to pay for goods and services offered by merchants that accept credit cards.

Debit Cards

A debit card is linked to a cardholder's checking account. A merchant who accepts the debit card can deduct funds directly from the account.

Transaction Types

This topic provides information about transaction types that are supported by your processor, such as card-present, card-not-present, and international transactions.

Card-Not-Present Transactions

When a customer provides a card number, but the card and the customer are not physically present at the merchant's location, the purchase is known as a card-not-present transaction. Typical card-not-present transactions are internet and phone transactions. Card-not-present transactions pose an additional level of risk to your

business because the customer's identification cannot be verified. You can reduce that risk by using features such as the Address Verification System (AVS) and Card Verification Numbers (CVNs). The AVS and CVNs provide additional protection from fraud by verifying the validity of the customer's information and notifying you when discrepancies occur.

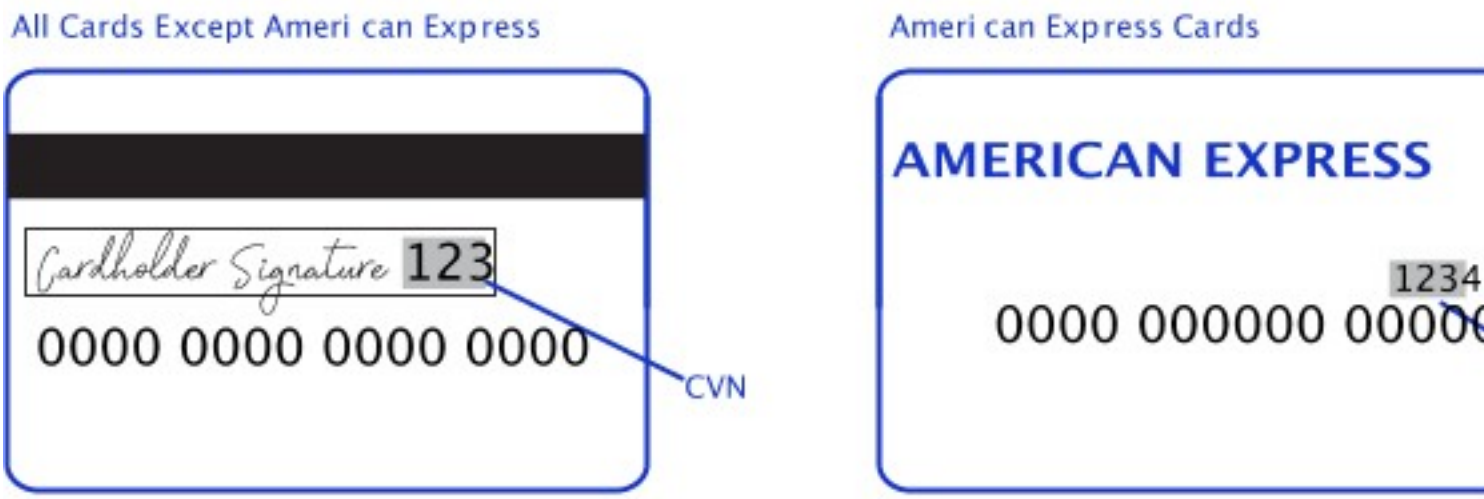
Authorizations with Card Verification Numbers

Card verification numbers (CVNs) are a required feature for the authorization service. The CVN is printed on a payment card, and only the cardholder can access it. The CVN is used in card-not-present transactions as a verification feature. Using the CVN helps reduce the risk of fraud.
CVNs are not included in payment card track data and cannot be obtained from a card swipe, tap, or dip.
CVNs must not be stored after authorization.

> **Important**
>
> In Europe, Visa mandates that you not include a CVN for mail-order transactions and not record a CVN on any physical format such as a mail-order form.

CVN Locations and Terminology

For most cards, the CVN is a three-digit number printed on the back of the card, to the right of the signature field. For American Express, the CVN is a four-digit number printed on the front of the card above the card number.



CVN Locations

Each payment card company has its own name for the CVN value:

- American Express and Discover call it the Card Identification Number (CID).
- JCB calls it the Card Authentication Value (CAV2).

- Mastercard calls it the Card Validation Code (CVC2).
- Visa calls it the Card Verification Value (CVV2).

International Transactions

Consider compliance and merchant remittance funding when processing international transactions.

Compliance

Accepting payments from a country other than your own requires that you observe the processing rules and practices of the payment systems in that country. The following list describes areas of compliance that are especially important.

- Merchant descriptor requirements—A merchant descriptor communicates merchant information to customers to remind them of the circumstances that triggered a payment. Merchant descriptors reduce the possibility of a chargeback. Accordingly, the merchant descriptor displayed on a customer's statement should be a close match to the name on your website. It is not good practice to consolidate multiple websites into a single merchant account and use a generic descriptor that more-or-less covers all offerings.
- Excessive chargebacks—To prevent an excessive number of chargebacks, you must maintain good customer support, rapid problem resolution, a high level of customer satisfaction, and transaction management processes that minimize fraudulent transactions. When payment card chargebacks become excessive, you must change business processes to reduce chargebacks. If chargebacks are not reduced to a satisfactory level, your account can be terminated.

Merchant Remittance Funding

You can request that the transaction proceeds be converted to another currency. Currency conversion uses a foreign exchange rate to calculate the conversion to the requested currency. The foreign exchange rate might be explicitly stated as a rate or implicitly stated as a transaction amount. The funded amount and can vary from day to day. The foreign exchange rate might also include an increase for the foreign exchange risk, sales commissions, and handling costs.

Token Management Service

The Token Management Service (TMS) tokenizes, securely stores, and manages customer and payment data. TMS enables you to:

- Securely store a customer's payment details and their billing and shipping addresses.
- Create a network token of a customer's payment card.

TMS simplifies your PCI DSS compliance. TMS passes back to you tokens that represent this data. You then store these tokens in your environment and databases instead of customer payment details.
TMS Token Types

- Customer — Stores the buyer's email address and the merchant's account ID for that buyer plus any other custom fields.
- Shipping Address — Stores a shipping address for a specific customer.
- Instrument Identifier — Stores either a payment card number or a bank account number and routing number
  This resource creates either:
  - An Instrument Identifier token using details of a payment card or an ACH bank account.
  - A payment network token using the details of a payment card; also uses the card expiration date and billing address, which are pass-through only fields.
- Payment Instrument — Stores a Payment Instrument using an Instrument Identifier token. It does not store the card number and cannot exist without an associated Instrument Identifier. It stores:
  - Card expiration date
  - Billing address

  You can also choose to store this information yourself instead and store only the card number or bank account and routing number in an Instrument Identifier object.
- Customer Payment Instrument — Creates and stores a payment instrument for a specific customer ID and an Instrument Identifier token.

TMS Features

- Create, retrieve, update, and delete tokens.
- Set a default payment instrument and shipping address for a customer.
- Process follow-on payment transactions with token IDs.
- Create and update tokens through bundled payment transactions.

Important

Due to mandates from the Reserve Bank of India, Indian merchants cannot store personal account numbers (PAN). Use network tokens instead. For more information on network tokens, see the Network Tokenization section of the *Token Management Service Guide.*

Payment Services

This section describes various services for processing payments.
These services enable customers to purchase goods and services. They also enable merchants to receive payments from customer accounts, to provide refunds, and to void transactions.

Authorizations

An authorization confirms that a payment card account holds enough funds to pay for a purchase. Authorizations can be made online or offline.

Online Authorizations

Online authorizations provide immediate confirmation of funds availability. The customer's financial institution also reduces the amount of credit available in the customer's account, setting aside the authorized funds for the merchant to capture at a later time. Authorizations for most payment cards are processed online. Typically, it is safe to start fulfilling the order when you receive an authorization confirmation.
An online authorization confirmation and the subsequent hold on funds expire after a specific length of time. Therefore it is important to capture funds in a timely manner. The issuing bank sets the expiration time interval, but most authorizations expire within 5 to 7 days.
The issuing bank does not inform Cybersource when an authorization confirmation expires. By default, the authorization information for each transaction remains in the Cybersource database for 180 days after the authorization date. To capture an authorization that expired with the issuing bank, you can resubmit the authorization request.

Offline Authorizations

Online transactions require an internet connection. In situations where the internet is not available, for example, due to an outage, merchants can continue to take credit card payments using offline transactions. An offline authorization is an authorization request for which you do not receive an immediate confirmation about the availability of funds. Offline authorizations have a higher level of risk than online transactions because they do not confirm funds availability or set aside the funds for later capture. Further, it can take up to 5 days to receive payment confirmations for offline transactions. To mitigate this risk, merchants may choose to fulfill orders only after receiving payment confirmation.

Pre-Authorizations

A pre-authorization enables you to authorize a payment when the final amount is unknown. It is typically used for lodging, auto rental, e-commerce, and restaurant transactions.
For a pre-authorization:

- The authorization amount must be greater than zero.
- The authorization must be submitted for capture within 30 calendar days of its request.
- When you do not capture the authorization, you must reverse it.
  In the U.S., Canada, Latin America, and Asia Pacific, Mastercard charges an additional fee for a pre-authorization that is not captured and not reversed.
  In Europe, Russia, Middle East, and Africa, Mastercard charges fees for all pre-authorizations.
- Chargeback protection is in effect for 30 days after the authorization.

Payment Network Token Authorizations

You can integrate authorizations with payment network tokens into your existing order management system. For an incremental authorization, you do not need to include any payment network tokenization fields in the authorization request because Cybersource obtains the payment network tokenization information from the original authorization request.

Authorization Workflow

This image and description show the authorization workflow:

1. The customer purchases goods or services from the merchant using a payment card.
2. You send an authorization request over secure internet connection to Cybersource. When the customer buys a digitally delivered product or service, you can request both the authorization and the capture at the same time. When the customer buys a physically fulfilled product, do not request the capture until you ship the product.
3. Cybersource validates the order information then contacts your payment processor and requests authorization.
4. The processor sends the transaction to the payment card company, which routes it to the issuing bank for the customer's payment card. Some card companies, including Discover and American Express, act as their own issuing banks.
5. The issuing bank approves or declines the request.
   - If funds are available, the issuing bank reserves the amount of the authorization request and returns an authorization approval to Cybersource.
   - If the issuing bank denies the request, it returns an authorization denial to Cybersource.
6. Cybersource runs its own tests then tells you whether the authorization succeeded.

Sales

A sale is a bundled authorization and capture. Some processors and acquirers require a sale transaction instead of using separate authorization and capture requests. For other processors and acquirers, you can request a sale instead of a separate authorization and capture when you provide the goods or services immediately after taking an order. There are two types of sale processing: dual-message processing and single-message processing.

Dual-Message Processing

Dual-message processing is a two-step process. The authorization is processed first. If the authorization is successful, the capture is processed immediately afterward. The response includes the authorization and the capture information. If the authorization

is declined, the capture is not processed, and the response message includes only the authorization information.

Partial Authorizations

All debit and prepaid card processors as well as a limited number of credit card processors support partial authorizations when dual-message processing is in place.

When partial authorization is enabled, the issuing financial institution can approve a partial amount when the balance on the card is less than the requested amount. When a partial amount is authorized, the capture is not processed. The merchant can then use a second card to cover the balance, adjust the total cost, or void the transaction.

Single-Message Processing

Single-message processing treats the authorization and capture as a single transaction. There are important differences between dual-message processing and single-message processing:

- Single-message processing treats the request as a full-financial transaction, and with a successful transaction, funds are immediately transferred from the customer account to the merchant account.
- Authorization and capture amounts must be the same.
- Some features cannot be used with single-message processing.

Authorization Reversals

The authorization reversal service releases the hold that an authorization placed on a customer's payment card funds.

Each card-issuing financial institution has its own rules for deciding whether an authorization reversal succeeds or fails. When a reversal fails, contact the card-issuing financial institution to learn whether there is a different way to reverse the authorization.

If your processor supports authorization reversal after void (ARAV), you can reverse an authorization after you void the associated capture. If your processor does not support ARAV, you can use the authorization reversal service only for an authorization that has not been captured and settled.

An authorization reversal is a follow-on transaction that uses the request ID returned from an authorization. The main purpose of a follow-on transaction is to link two transactions. The request ID links the follow-on transaction to the original transaction. The authorization request ID is used to look up the customer's billing and account information in the Cybersource database. You are not required to include those fields in the full authorization reversal request. The original transaction and follow-on transaction are linked in the database and in the Business Center.

For processors that support debit cards and prepaid cards, the full authorization reversal service works for debit cards and prepaid cards in addition to credit cards.

Important

You cannot perform an authorization reversal if a transaction is in a review state, which can occur if you use a fraud management service. You must reject the

transaction prior to authorization reversal. For more information, see the fraud management documentation in the Business Center.

Credits

Credits are payment refunds from a merchant to the cardholder after a cardholder pays for a product or service and that payment is captured by the merchant. When a credit request is successful, the issuer transfers funds from the merchant bank (acquirer) account to the customer's account. It typically takes 2 to 4 days for the acquirer to transfer funds from your merchant account.

Warning

You should carefully control access to the credit service. Do not request this service directly from your customer interface. Instead, incorporate this service as part of your customer service process. This process reduces the potential for fraudulent transactions.

There are two basic types of credits: follow-on credits and stand-alone credits.

Follow-On Credits

Follow-on credits, also known as refunds, use the capture request ID to link the refund to a specific transaction. This request ID is returned during the capture request (also known as a settlement) and is used in all subsequent refunds associated with the original capture. The request ID links the transaction to the customer's billing and account information, so you are not required to include those fields in the credit request. However, when you combine a request for a refund with a request for another service, such as the tax calculation service, you must provide the customer's billing and account information. Unless otherwise specified, refunds must be requested within 180 days of a settlement. You can request multiple refunds against a single capture. To perform multiple refunds, use the same request ID in each request.

Stand-Alone Credits

Stand-alone credits are not tied to an original transaction. Stand-alone credits do not have a time restriction, and they can be used to issue refunds more than 180 days after a transaction settlement.

Credit Workflow

The credit workflow begins when you send a request for a credit.
A credit does not happen in real time. All of the credit requests for a day are typically placed in a file and sent to the processor as a single batch transaction. In most cases, the batch transaction is settled overnight.

1. The merchant sends a request for a credit to Cybersource.
2. For online credits, Cybersource validates the order information then sends an online credit to the payment processor. For offline credits, Cybersource stores the credit

request in a batch file and sends the batch file to the payment processor after midnight.
3. The processor validates the request and forwards it to the acquiring bank.
4. The acquiring bank transfers funds to the issuing bank.

Voids

A void cancels a capture or credit request that was submitted but not yet processed by the processor.
Capture and credit requests are usually submitted once a day. A void request is declined when the capture or credit request has already been sent to the processor.
After a void is processed, you cannot credit or capture the funds. You must perform a new transaction to capture or credit the funds. Further, when you void a capture, a hold remains on the authorized funds. If you are not going to re-capture the authorization, and if your processor supports authorization reversal after void (ARAV), you should request an authorization reversal to release the hold on the unused funds.
A void uses the capture or credit request ID to link the transactions. The authorization request ID is used to look up the customer's billing and account information, so there is no need to include those fields in the void request. You cannot perform a follow-on credit against a capture that has been voided.

Payment Features

You can apply features to different payment services to enhance the customer payment processing experience. This section includes an overview of these features:

- *Debit and Prepaid Card Payments* on page 19
- *Payer Authentication* on page 20
- *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 20

Debit and Prepaid Card Payments

Debit cards are linked to a cardholder's checking account. A merchant who accepts the debit card can deduct funds directly from the linked cardholder's account.
You can process debit cards using these services:

- Credit card services
- PIN debit services

Related Information

- See *Standard Payment Processing* on page 23 for information that shows you how to use credit card services.
- See *Debit and Prepaid Card Processing* on page 57 for information that shows you how to process authorizations that use a debit or prepaid card.

Payer Authentication

Payer authentication is run before a transaction is submitted for authorization. Most of the time payer authentication is bundled with authorization so that after payer authentication happens, the transaction is automatically submitted for authorization. Payer authentication and authorization can be configured to occur as separate operations. This section shows you how to run payer authentication as a separate process and pass the payer authentication data when seeking authorization for a transaction. Payer authentication consists of a two-step verification process that adds an extra layer of fraud protection during the payment process. During transactions, the transaction device, location, past purchasing habits, and other factors are analyzed for indications of fraud. This process collects customer data during the transaction from at least two of these three categories:

- Something you have: A payment card or a payment card number
- Something you know: A password or pin
- Something you are: Facial recognition or fingerprint

Each of these payment card companies has its own payer authentication product:

- Discover: ProtectBuy
- JCB: J/Secure
- Mastercard: Identity Check
- Visa: Visa Secure

Payer authentication can be used to satisfy the Strong Customer Authentication (SCA) requirement of the Payment Services Directive (PSD2). SCA applies to the European Economic Area (EEA) and the United Kingdom. SCA requires banks to perform additional checks when customers make payments to confirm their identity.

Related Information

- See the *Payer Authentication Developer Guide* for more information about payer authentication.
- See *Payer Authentication Processing* on page 65 for information about how to process payments with payer authentication.

Relaxed Requirements for Address Data and Expiration Date in Payment Transactions

With relaxed requirements for address data and the expiration date, not all standard payment request fields are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required.

Related Information

- See *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 73 for information about how to process payments with relaxed requirements for address data and expiration date.

Testing the Payment Services

To ensure that requests are processed correctly, you must test the basic success and error conditions for each service you plan to use.

Requirements for Testing

Important

Before you can test, you must contact customer support to activate the credit card services and configure your account for testing. You must also contact your processor to set up your processor account.

Important

When building your connection to the Cybersource payment gateway, ensure that you have implemented controls to prevent card testing or card enumeration attacks on your platform. For more information, see the *best practices guide*. When we detect suspicious transaction activity associated with your merchant ID, including a card testing or card enumeration attack, Cybersource reserves the right to enable fraud management tools on your behalf in order to mitigate the attack. The fraud team might also implement internal controls to mitigate attack activity. These controls block traffic that is perceived as fraudulent. Additionally, if you are using one of our fraud tools and experience a significant attack, our internal team might modify or add rules to your configuration to help prevent the attack and minimize the threat to our infrastructure. However, any actions taken by Cybersource would not replace the need for you to follow industry standard best practices to protect your systems, servers, and platforms.

Follow these requirements when you test your system:

- Use your regular merchant ID.
- Use a real combination for the city, state, and postal code.
- Use a real combination for the area code and telephone number.
- Use a nonexistent account and domain name for the customer's email address.
- Simple Order API test server: `https://ics2wstesta.ic3.com/commerce/1.x/transactionProcessor`

Test Card Numbers

Use these payment card numbers to test the authorization, capture, and credit services. Remove the spaces from the test card numbers when sending them to the test system. Do not use real payment card numbers. To test card types that are not included in the list, use an account number that is in the card's BIN range. For best results, try each test with a different service request and with different test payment card numbers.

- American Express—3782 8224 6310 005
- Discover—6011 1111 1111 1117
- JCB—3566 1111 1111 1113
- Maestro (International)
  - 5033 9619 8909 17
  - 5868 2416 0825 5333 38
- Maestro (UK Domestic)—the issue number is not required for Maestro (UK Domestic) transactions.
  - 6759 4111 0000 0008
  - 6759 5600 4500 5727 054
  - 5641 8211 1116 6669
- Mastercard
  - 2222 4200 0000 1113
  - 2222 6300 0000 1125
  - 5555 5555 5555 4444
- UATP—1354 1234 5678 911
- Visa—4111 1111 1111 1111

Using Amounts to Simulate Errors

You can simulate error messages by requesting authorization, capture, or credit services with specific amounts that trigger the error messages. These triggers work only on the test server, not on the production server.
Each payment processor uses its own error messages. For more information, see: *Simple Order API Testing Information*.

Test American Express Card Verification

Before using CVN with American Express, it is strongly recommended that you follow these steps:

1. Contact customer support to have your account configured for CVN. Until you do this, you will receive a `1` in the **ccAuthReply_cvCode** response field.
2. Test your system in production using a small currency amount, such as one currency unit. Instead of using the test account numbers, use a real payment card account number, and send an incorrect CVN in the request for authorization. The card should be refused and the request declined.

Standard Payment Processing

This section shows you how to process various authorization, capture, credit, and sales transactions.

Basic Authorizations

This section provides the information you need in order to process a basic authorization.

Endpoint

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

Declined Authorizations

If an authorization is declined, you can use response categories to help you decide whether to retry or block a declined transaction. These response fields provide additional information:

- **ccAuthReply_paymentInsightsInformation_responseInsightsCategory**
- **ccAuthReply_paymentInsightsInformation_responseInsightsCategoryCode**
  These fields are available starting in the XML schema version 1.193.

Category codes have possible values (such as `01`) each of which corresponds to a category that contains a description.
You cannot retry this category code and category:

- `01` `ISSUER_WILL_NEVER_APPROVE`

For these values, you can retry the transaction a maximum of 15 times over a period of 30 days:

- `02` `ISSUER_CANNOT_APPROVE_AT_THIS_TIME`

- `03 ISSUER_CANNOT_APPROVE_WITH_THESE_DETAILS`: Data quality issue. Revalidate data prior to retrying the transaction.
- `04 GENERIC_ERROR`
- `97 PAYMENT_INSIGHTS_INTERNAL_ERROR`
- `98 OTHERS`
- `99 PAYMENT_INSIGHTS_RESPONSE_CATEGORY_MATCH_NOT_FOUND`

Required Fields for Processing a Basic Authorization

Use these required fields for processing a basic authorization.

Important

When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 73.

billTo_city	
billTo_country	
billTo_email	
billTo_firstName	
billTo_lastName	
billTo_postalCode	
billTo_state	
billTo_street1	
card_accountNumber	
card_expirationMonth	
card_expirationYear	
ccAuthService_run	**Set the value to `true`.**
merchantID	
merchantReferenceCode	
purchaseTotals_currency	
purchaseTotals_grandTotalAmount	

Related Information

- *API Field Reference for the Simple Order API*

Simple Order Example: Processing a Basic Authorization

Request

```
billTo_city=Ann Arbor
billTo_country=US
billTo_email=null@cybersource.com
billTo_firstname=John
billTo_lastname=Smith
billTo_postalCode=48104-2201
billTo_state=MI
billTo_street1=201 S. Division St.
card_accountNumber=41111111XXXXXXXX
card_expirationMonth=12
card_expirationYear=2023
ccAuthService_run=true
merchant_id=npr_paymentech
merchant_referenceCode=TC42703-1
purchaseTotals_currency=usd
purchaseTotals_grandTotalAmount=100
```

Response to a Successful Request

```
requestID=6629977932421985593067
decision=ACCEPT
reasonCode=100
merchantReferenceCode=TC42703-1
purchaseTotals_currency=usd
ccAuthService_reconciliationID=57953165A7YFPS77
ccAuthReply_amount=100.00
ccAuthReply_avsCode=5
ccAuthReply_authorizationCode=570110
ccAuthReply_processorResponse=1
ccAuthReply_authorizedDateTime=2022-09-12T154953Z
ccAuthReply_paymentNetworkTransactionID=123456789619999
```

Response to a Declined Request

```
requestID=6629977932421985593067
merchantReferenceCode=Merchant_REF
decision=REJECT
ccAuthReply_avsCode=Y
ccAuthReply_avsCodeRaw=Y
ccAuthReply_paymentNetworkTransactionID=111222
ccAuthReply_transactionID=111222
ccAuthReply_paymentInsightsInformation_responseInsightsCategory=
  ISSUER_CANNOT_APPROVE_WITH_THESE_DETAILS
ccAuthReply_paymentInsightsInformation_responseInsightsCategoryCode=03
ccAuthReply_processorResponse=183
ccAuthReply_reasonCode=233
```

Authorizations with Line Items

This section shows you how to process an authorization with line items.

The main difference between a basic authorization and an authorization that includes line items is that the **purchaseTotals_grandTotalAmount** field, which is included in a basic authorization, is substituted with one or more line items that are included in the **item_#_** fields, starting with the **item_0_** fields.

Fields Specific to this Use Case

These fields are required for each line item that you use:

item_#_unitPrice	
item_#_quantity	
item_#_productCode	
item_#_productSKU	Optional when **item_#_productCode** is set to `default`, `shipping_only`, `handling_only`, or `shipping_and_handling`
item_#_productName	Optional when **item_#_productCode** is set to `default`, `shipping_only`, `handling_only`, or `shipping_and_handling`

At a minimum, you must include the **item_#_unitPrice** field in order to include a line item in an authorization. When this field is the only field included in the authorization, the system sets:

- **item_#_productCode**: `default`
- **item_#_quantity**: `1`

For example, these three line items are valid.

```
item_0_unitPrice=10.00
item_1_unitPrice=5.99
item_1_quantity=3
item_1_productCode=shipping_only
item_2_unitPrice=29.99
item_2_quantity=3
item_2_productCode=electronic_good
item_2_productSKU=12384569
item_2_productName=receiver
```

Endpoint

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

Optional Line Item Fields

These fields can be used to provide more line item information. For more information on each field, see the field reference guide:

- **item_#_buyerRegistration**
- **item_#_commodityCode**
- **item_#_nationalTax**
- **item_#_orderAcceptanceCity**
- **item_#_orderAcceptanceCountry**
- **item_#_orderAcceptancePostalCode**
- **item_#_orderAcceptanceState**
- **item_#_orderOriginCity**
- **item_#_orderOriginCountry**
- **item_#_orderOriginPostalCode**
- **item_#_orderOriginState**
- **item_#_otherTax_#_passengerFirstName**
- **item_#_otherTax_#_passengerLastName**
- **item_#_productCode**
- **item_#_productDescription**
- **item_#_productName**
- **item_#_productSKU**
- **item_#_quantity**
- **item_#_shippingDestinationType**
- **item_#_unitPrice**

Required Fields for Processing an Authorization with Line Items

Use these required fields for processing an authorization that includes line items.

Important

When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 73.

billTo_city

billTo_country

billTo_email

billTo_firstName

billTo_lastName

billTo_postalCode

billTo_state	
billTo_street1	
card_accountNumber	
card_expirationMonth	
card_expirationYear	
ccAuthService_run	**Set the value to `true`.**
merchantID	
merchantReferenceCode	**Required when billTo_personalID is included in the request.**
purchaseTotals_currency	
purchaseTotals_grandTotalAmount	**Either purchaseTotals_grandTotalAmount or item_#_unitPrice must be included in the request.**

Related Information

- *API field reference guide for the Simple Order API*

Simple Order Example: Processing an Authorization with Line Items

Request

```
billTo_city=Palo Alto
billTo_country=US
billTo_email=null@cybersource.com
billTo_firstname=Julia
billTo_lastname=Fernandez
billTo_postalCode=94043
billTo_state=CA
billTo_street1=123 Main St.
card_accountNumber=41111111XXXXXXXX
card_expirationMonth=12
card_expirationYear=2023
ccAuthService_run=true
dcc_dccIndicator=1
merchant_id=MID23
merchant_referenceCode=Merchant_REF
purchaseTotals_currency=usd
item_0_unitPrice=10.00
item_1_unitPrice=5.99
item_1_quantity=3
item_1_productCode=shipping_only
item_2_unitPrice=29.99
item_2_quantity=3
item_2_productCode=electronic_good
item_2_productSKU=12384569
item_2_productName=receiver
```

```
purchaseTotals_exchangeRate=0.91
purchaseTotals_originalAmount=107.33
purchaseTotals_originalCurrency=eur
```

Response to a Successful Request

```
additional_processor_response=e1cdcafc-cdbb-4ef7-8788-a1234e844805
request_id=6461515866500167772420
decision=ACCEPT
reasonCode=100
merchantReferenceCode=Merchant_REF
purchaseTotals_currency=usd
cardCategory=FccAuthService_reconciliationID=ZUDCXJO8KZRFXQJJ
ccAuthReply_amount=117.94
ccAuthReply_avsCode=5
ccAuthReply_authorizationCode=570110
ccAuthReply_processorResponse=1
ccAuthReply_authorizedDateTime=2022-03-01T161947Z
ccAuthReply_paymentNetworkTransactionID=111222
```

Authorizations with Payment Network Tokens

This section shows you how to successfully process an authorization with payment network tokens.

> **Important**
>
> Due to mandates from the Reserve Bank of India, Indian merchants cannot store personal account numbers (PAN). Use network tokens instead. For more information on network tokens, see *Network Tokenization* in the Token Management Service Developer Guide.

Endpoint

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

Required Fields for Authorizations with Payment Network Tokens

Use these required fields for processing an authorization with payment network tokens.

> **Important**
>
> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required.

> For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 73.

billTo.email

billTo.firstName

billTo.lastName

billTo.street1

ccAuthService.networkTokenCryptogram

purchaseTotals.currency

purchaseTotals.grandTotalAmount

token.expirationMonth

token.expirationYear

Related Information

- *API field reference guide for the Simple Order API*

Optional Fields for Authorizations with Payment Network Tokens

billTo.city	
billTo.country	
billTo.email	
billTo.firstName	
billTo.lastName	
billTo.postalCode	**Required only for transactions in the U.S. and Canada.**
billTo.state	**Required only for transactions in the U.S. and Canada.**
billTo.street1	
card.accountNumber	**Set to the token value that you received from the token service provider.**
card.cardType	**It is strongly recommended that you send the card type even if it is optional for your processor. Omitting the card type can cause the transaction to be processed with the wrong card type.**
card.expirationMonth	**Set to the token expiration month that you received from the token service provider.**

card.expirationYear	**Set to the token expiration year that you received from the token service provider.**
ccAuthService.cavv	**For 3-D Secure in-app transactions for Visa and JCB, set to the 3-D Secure cryptogram. Otherwise, set to the network token cryptogram.**
ccAuthService.commerceIndicator	
ccAuthService.networkTokenCryptogram	
ccAuthService.run	**Set the value to `true`.**
merchantID	
merchantReferenceCode	
purchaseTotals.currency	
purchaseTotals.grandTotalAmount or item_#.unitPrice	
paymentNetworkToken.transactionType	
paymentNetworkToken.requestorID	
ucaf.authenticationData	**For Mastercard requests, set this field to the Identity Check cryptogram.**
ucaf.collectionIndicator	**For Mastercard requests, set the value to `2`.**

Related Information

- *API field reference guide for the Simple Order API*

Simple Order API Example: Authorizations with Payment Network Tokens

Request

```
<requestMessage>
  <purchaseTotals>
    <currency>USD</currency>
    <grandTotalAmount>16.00</grandTotalAmount>
  </purchaseTotals>
  <card>
    <accountNumber>4111111111111111</accountNumber>
    <expirationMonth>12</expirationMonth>
    <expirationYear>2031</expirationYear>
  </card>
  <ccAuthService run="true">
    <networkTokenCryptogram>qE5juRwDzAUFBAkEHuWW9PiBkWv=</networkTokenCryptogram>
  </ccAuthService>
  <paymentNetworkToken>
    <transactionType>1</transactionType>
```

```
  </paymentNetworkToken>
</requestMessage>
```

Successful Response

```
<c:replyMessage>
  <c:merchantReferenceCode>Postman-1684858432</c:merchantReferenceCode>
  <c:requestID>6848584316126969103007</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:purchaseTotals>
    <c:currency>USD</c:currency>
  </c:purchaseTotals>
  <c:ccAuthReply>
    <c:reasonCode>100</c:reasonCode>
    <c:amount>16.00</c:amount>
    <c:authorizationCode>888888</c:authorizationCode>
    <c:avsCode>X</c:avsCode>
    <c:avsCodeRaw>I1</c:avsCodeRaw>
    <c:authorizedDateTime>2023-05-23T16:13:51Z</c:authorizedDateTime>
    <c:processorResponse>100</c:processorResponse>
    <c:reconciliationID>78849228NHPFQCKD</c:reconciliationID>
    <c:paymentNetworkTransactionID>123456789619999</c:paymentNetworkTransactionID>
   </c:ccAuthReply>
   <c:card>
     <c:cardType>001</c:cardType>
   </c:card>
</c:replyMessage>
```

Authorizations with a Card Verification Number

This section shows you how to process an authorization with a Card Verification Number (CVN).

CVN Results

The response includes a raw response code and a mapped response code:

- The raw response code is the value returned by the processor. This value is returned in the **ccAuthReply_cvCodeRaw** field. Use this value only for debugging purposes; do not use it to determine the card verification response.
- The mapped response code is the pre-defined value that corresponds to the raw response code. This value is returned in the **ccAuthReply_cvCode** field.

Even when the CVN does not match the expected value, the issuing bank might still authorize the transaction. You will receive a CVN decline, but you can still capture the transaction because it has been authorized by the bank. However, you must review the order to ensure that it is legitimate.

Settling authorizations that fail the CVN check might have an impact on the fees charged by your bank. Contact your bank for details about how card verification management might affect your discount rate.
When a CVN decline is received for the authorization in a sale request, the capture request is not processed unless you set the **businessRules_ignoreCVResult** field to `true`.

CVN Results for American Express	A value of `1` in the **ccAuthReply_cvCode** field indicates that your account is not configured to use card verification. Contact customer support to have your account enabled for this feature.
CVN Results for Discover	
CVN Results for Visa and Mastercard	A CVN code of `D` or `N` causes the request to be declined with a reason code value of `230`. You can still capture the transaction, but you must review the order to ensure that it is legitimate. Cybersource, not the issuer, assigns the CVN decline to the authorization. You can capture any authorization that has a valid authorization code from the issuer, even when the request receives a CVN decline. When the issuer does not authorize the transaction and the CVN does not match, the request is declined because the card is refused. You cannot capture the transaction.

Fields Specific to this Use Case

Include this field with a standard authorization request when processing an authorization with a CVN:

- **card_cvNumber**

Endpoint

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

Required Fields for Processing an Authorization with a Card Verification Number

Use these required fields for processing an authorization that includes a Card Verification Number (CVN).

When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 73.

billTo_city

billTo_country

billTo_email

billTo_firstName

billTo_lastName

billTo_postalCode

billTo_state

billTo_street1

card_accountNumber

card_cvNumber

card_expirationMonth

card_expirationYear

ccAuthService_run — **Set the value to `true`.**

merchantID

merchantReferenceCode

purchaseTotals_currency

purchaseTotals_grandTotalAmount

Related Information

- *API field reference guide for the Simple Order API*

Optional Fields for Processing an Authorization with a Card Verification Number

You can use these optional fields to include additional information when processing an authorization with a card verification number.

businessRules_ignoreCVResult

card_cvIndicator

Simple Order Example: Processing an Authorization with a Card Verification Number

Request

```
ccAuthService_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=2031
card_accountNumber=4111111111111111
card_cvNumber=999
card_cardType=001
```

Response to a Successful Request

```
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_authorizationCode=123456
ccAuthReply_avsCode=Y
ccAuthReply_avsCodeRaw=YYY
ccAuthReply_processorResponse=A
ccAuthReply_paymentNetworkTransactionID=3312345
```

Authorizations with Strong Customer Authentication Exemption

This section shows you how to process an authorization with a strong customer authentication (SCA) exemption.

You can use SCA exemptions to streamline the payment process. SCA exemptions are part of the European second Payment Services Directive (PSD2) and allow certain types of low-risk transactions to bypass additional authentication steps while still remaining compliant with PSD2. You can choose which exemption can be applied to a transaction, but the card-issuing bank actually grants an SCA exemption during card authentication.
You can process an authorization with two types of SCA exemptions:

- Exemption on Authorization: Send an authorization without payer authentication and request an SCA exemption on the authorization. If it is not approved, you may be required to request further authentication upon retry.
- Exemption on Authentication: Request an SCA exemption during payer authentication and if successful, send an authorization including the SCA exemption details.

Depending on your processor, use one of these exemption fields:

Important

If you send more than one SCA exemption field with a single authentication, the transaction is denied.

- Authentication Outage: Payer authentication is not available for this transaction due to a system outage.
- B2B Corporate Card: Payment cards specifically for business-to-business transactions are exempt.
- Delegated Authentication: Payer authentication was performed outside of the authorization workflow.
- Follow-On Installment Payment: Installment payments of a fixed amount are exempt after the first transaction.
- Follow-On Recurring Payment: Recurring payments of a fixed amount are exempt after the first transaction.
- Low Risk: The average fraud levels associated with this transaction are considered low.
- Low Value: The transaction value does not warrant SCA.
- Merchant Initiated Transactions: As follow-on transactions, merchant-initiated transactions are exempt.
- Stored Credential Transaction: Credentials are authenticated before storing, so stored credential transactions are exempt.
- Trusted Merchant: Merchants registered as trusted beneficiaries.

Exemption Fields Specific to the Strong Customer Authentication Use Case

Use one of these fields to request an SCA exemption:

Types of SCA Exemptions

Exemption Type	Field	Value
Authentication outage	ccAuthService_authenticationOutageExemptionIndicator	1
Delegated authentication	ccAuthService_delegatedAuthenticationExemptionIndicator	1
Low-risk transaction	ccAuthService_riskAnalysisExemptionIndicator	1
Low-value transaction	ccAuthService_lowValueExemptionIndicator	1
Trusted merchant transaction	ccAuthService_trustedMerchantExemptionIndicator	1

Processor Support for SCA Exemptions

You can send an authorization without payer authentication and request an SCA exemption on the authorization. If it is not approved, you may be required to request further authentication upon retry. Use this table to determine which processors support SCA exemptions on authorization:

Processor Support for SCA Exemption on Authorization

	Authentication Outage	Follow-On Recurring Payment	Low Value	Transaction Risk Analysis
Credit Mutuel-CIC				

You can request an SCA exemption during payer authentication and if successful, send an authorization including the SCA exemption details. Use this table to determine which processors support SCA exemptions on authentication:

Processor Support for SCA Exemption on Authentication

	B2B Corporate Card	Delegated Authentication	Low Risk	Low Value	Trusted Merchant
Credit Mutuel-CIC					

For more information, see the *Exemption Test Cases* section of the Payer Authentication Developer Guide.

Endpoint

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

Required Fields for Processing an Authorization with an SCA Exemption

Use these required fields for processing an authorization that includes an SCA exemption.

> **Important**
>
> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 73.

billTo_city

billTo_country

billTo_email

billTo_firstName

billTo_lastName

billTo_postalCode

billTo_state

billTo_street1

card_accountNumber

card_expirationMonth

card_expirationYear

ccAuthService_run — **Set the value to `true`.**

merchantID

merchantReferenceCode

purchaseTotals_grandTotalAmount

Related Information

- *API field reference guide for the Simple Order API*

Simple Order Example: Processing an Authorization with an SCA Exemption for Low Value Transactions

Request

```
<requestMessage>
 <merchantID>{{merchantID}}</merchantID>
 <merchantReferenceCode>Postman-{{$timestamp}}</merchantReferenceCode>
 <billTo>
```

```
   <firstName>John</firstName>
   <lastName>Doe</lastName>
   <street1>1295 Charleston Road</street1>
   <city>Mountain View</city>
   <state>CA</state>
   <postalCode>94043</postalCode>
   <country>US</country>
   <email>null@cybersource.com</email>
  </billTo>
  <purchaseTotals>
   <currency>USD</currency>
   <grandTotalAmount>1.01</grandTotalAmount>
  </purchaseTotals>
  <card>
   <accountNumber>4111111111111111</accountNumber>
   <expirationMonth>12</expirationMonth>
   <expirationYear>2023</expirationYear>
   <cardType>001</cardType>
  </card>
  <ccAuthService run="true"/>
</requestMessage>
```

Response to a Successful Request

```
<c:replyMessage>
 <c:merchantReferenceCode>Postman-1666374834</c:merchantReferenceCode>
 <c:requestID>6663748348516429203007</c:requestID>
 <c:decision>ACCEPT</c:decision>
 <c:reasonCode>100</c:reasonCode>
 <c:purchaseTotals>
  <c:currency>USD</c:currency>
 </c:purchaseTotals>
 <c:ccAuthReply>
  <c:reasonCode>100</c:reasonCode>
  <c:amount>1.01</c:amount>
  <c:authorizationCode>888888</c:authorizationCode>
  <c:avsCode>X</c:avsCode>
  <c:avsCodeRaw>I1</c:avsCodeRaw>
  <c:authorizedDateTime>2022-10-21T17:53:54Z</c:authorizedDateTime>
  <c:processorResponse>100</c:processorResponse>
  <c:reconciliationID>66737280B9CGUCCP</c:reconciliationID>
  <c:paymentNetworkTransactionID>123456789619999</c:paymentNetworkTransactionID>
 </c:ccAuthReply>
 <c:card>
  <c:cardType>001</c:cardType>
 </c:card>
</c:replyMessage>
```

Zero Amount Authorizations

This section provides the information that you need in order to process a zero amount authorization.

Authorizing a payment for a zero amount shows whether a payment card account is valid and whether the card is lost or stolen. You cannot capture a zero amount authorization.

Processor-Specific Information

Credit Mutuel-CIC	**CVN is supported.** **Card types: co-badged Cartes Bancaires and Mastercard, co-badged Cartes Bancaires and Visa**

Endpoint

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

Required Fields for Processing a Zero Amount Authorization

Use these required fields for processing a zero amount authorization.

Important

When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 73.

billTo_city	
billTo_country	
billTo_email	
billTo_firstName	
billTo_lastName	
billTo_postalCode	
billTo_state	
billTo_street1	
card_accountNumber	
card_expirationMonth	
card_expirationYear	
ccAuthService_run	**Set the value to `true`.**
merchantID	
merchantReferenceCode	

purchaseTotals_currency

purchaseTotals_grandTotalAmount

Related Information

- *API field reference guide for the Simple Order API*

Simple Order Example: Processing a Zero Amount Authorization

Request

```
billTo_city=Sao Paulo
billTo_country=BR
billTo_email=null@cybersource.com
billTo_firstname=Julia
billTo_lastname=Fernandez
billTo_postalCode=01310-000
billTo_state=SP
billTo_street1=R. Augusta
card_accountNumber=41111111XXXXXXXX
card_expirationMonth=12
card_expirationYear=2023
ccAuthService_run=true
merchant_id=MID23
merchant_referenceCode=Merchant_REF
purchaseTotals_currency=mxn
purchaseTotals_grandTotalAmount=0
```

Response to a Successful Request

```
additional_processor_response=e1cdcafc-cdbb-4ef7-8788-a1234e844805
request_id=6461515866500167772420
decision=ACCEPT
reasonCode=100
merchantReferenceCode=Merchant_REF
purchaseTotals_currency=mxn
cardCategory=FccAuthService_reconciliationID=ZUDCXJO8KZRFXQJJ
ccAuthReply_amount=0
ccAuthReply_avsCode=5
ccAuthReply_authorizationCode=570110
ccAuthReply_processorResponse=1
ccAuthReply_authorizedDateTime=2022-03-01T161947Z
ccAuthReply_paymentNetworkTransactionID=111222
```

Pre-Authorizations

This section provides the information you need in order to process a pre-authorization. A pre-authorization enables you to authorize a payment when the final amount is unknown. It is typically used for lodging, auto rental, e-commerce, and restaurant transactions.

For a pre-authorization:

- The authorization amount must be greater than zero.
- The authorization must be submitted for capture within 30 calendar days of its request.
- When you do not capture the authorization, you must reverse it.
  In the U.S., Canada, Latin America, and Asia Pacific, Mastercard charges an additional fee for a pre-authorization that is not captured and not reversed.
  In Europe, Russia, Middle East, and Africa, Mastercard charges fees for all pre-authorizations.
- Chargeback protection is in effect for 30 days after the authorization.

Endpoint

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

Required Fields for a Pre-Authorization

Use these required fields for processing a pre-authorization.

Important

When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 73.

billTo_city	
billTo_country	
billTo_email	
billTo_firstName	
billTo_lastName	
billTo_postalCode	
billTo_state	
billTo_street1	
card_accountNumber	
card_expirationMonth	
card_expirationYear	
ccAuthService_run	**Set the value to `true`.**
merchantID	
merchantReferenceCode	

purchaseTotals_currency

purchaseTotals_grandTotalAmount

Related Information

- *API field reference guide for the Simple Order API*

Simple Order Example: Processing a Pre-Authorization

Request

```
billTo_city=Ann Arbor
billTo_country=US
billTo_email=null@cybersource.com
billTo_firstname=John
billTo_lastname=Smith
billTo_postalCode=48104-2201
billTo_state=MI
billTo_street1=201 S. Division St.
card_accountNumber=41111111XXXXXXXX
card_expirationMonth=12
card_expirationYear=2023
ccAuthService_run=true
merchant_id=npr_paymentech
merchant_referenceCode=TC42703-1
purchaseTotals_currency=usd
purchaseTotals_grandTotalAmount=100
```

Response to a Successful Request

```
requestID=6629977932421985593067
decision=ACCEPT
reasonCode=100
merchantReferenceCode=TC42703-1
purchaseTotals_currency=usd
ccAuthService_reconciliationID=57953165A7YFPS77
ccAuthReply_amount=100.00
ccAuthReply_avsCode=5
ccAuthReply_authorizationCode=570110
ccAuthReply_processorResponse=1
ccAuthReply_authorizedDateTime=2022-09-12T154953Z
ccAuthReply_paymentNetworkTransactionID=123456789619999
```

Response to a Declined Request

```
requestID=6629977932421985593067
merchantReferenceCode=Merchant_REF
decision=REJECT
ccAuthReply_avsCode=Y
ccAuthReply_avsCodeRaw=Y
ccAuthReply_paymentNetworkTransactionID=111222
ccAuthReply_transactionID=111222
ccAuthReply_paymentInsightsInformation_responseInsightsCategory=
  ISSUER_CANNOT_APPROVE_WITH_THESE_DETAILS
```

```
ccAuthReply_paymentInsightsInformation_responseInsightsCategoryCode=03
ccAuthReply_processorResponse=183
ccAuthReply_reasonCode=233
```

Authorization Reversal

This section provides the information about how 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.
For a debit card or prepaid card in which only a partial amount was approved, the amount of the reversal must be the amount that was authorized, not the amount that was requested.

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 the value 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=10198275203482905702 93
merchantReferenceCode=482046C3A7E94F5BD1FE3C66C
decision=ACCEPT
reasonCode=100
ccAuthReversalReply_amount=49.95
purchaseTotals_currency=USD
ccAuthReversalReply_reasonCode=100
ccAuthReversalReply_reconciliationID=10948209750234 70
```

Sales

This section provides the information you need in order to process a sale transaction. A sale 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

Important

When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 73.

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 the value to `true`.**
ccCaptureService_run	**Set the value to `true`.**
merchantID	
purchaseTotals_currency	
purchaseTotals_grandTotalAmount	

Related Information

- *API field reference guide 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=2031
card_accountNumber=4111111111111111
card_cardType=001
```

Response to a Successful Request

```
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
```

Sales with Payment Network Tokens

This section shows you how to successfully process a sale with payment network tokens.

Important

Due to mandates from the Reserve Bank of India, Indian merchants cannot store personal account numbers (PAN). Use network tokens instead. For more information on network tokens, see *Network Tokenization* in the Token Management Service Developer Guide.

Endpoint

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

Required Fields for Sales with Payment Network Tokens

Use these required fields for processing a sale with payment network tokens.

Important

When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 73.

billTo_email

billTo_firstName

billTo_lastName

billTo_street1

ccAuthService_networkTokenCryptogram

ccCaptureService_run — **Set the value to `true`.**

purchaseTotals_currency

purchaseTotals_grandTotalAmount

token_expirationMonth

token_expirationYear

Related Information

- *API field reference guide for the Simple Order API*

Optional Fields for Sales with Payment Network Tokens

billTo.city	
billTo.country	
billTo.email	
billTo.firstName	
billTo.lastName	
billTo.postalCode	**Required only for transactions in the U.S. and Canada.**
billTo.state	**Required only for transactions in the U.S. and Canada.**
billTo.street1	
card.accountNumber	**Set to the token value that you received from the token service provider.**
card.cardType	**It is strongly recommended that you send the card type even if it is optional for your processor. Omitting the card type can cause the transaction to be processed with the wrong card type.**
card.expirationMonth	**Set to the token expiration month that you received from the token service provider.**
card.expirationYear	**Set to the token expiration year that you received from the token service provider.**
ccAuthService.cavv	**For 3-D Secure in-app transactions for Visa and JCB, set to the 3-D Secure cryptogram. Otherwise, set to the network token cryptogram.**
ccAuthService.commerceIndicator	
ccAuthService.networkTokenCryptogram	
ccAuthService.run	**Set the value to `true`.**
merchantID	

merchantReferenceCode	
purchaseTotals.currency	
purchaseTotals.grandTotalAmount or item_#.unitPrice	
paymentNetworkToken.transactionType	
paymentNetworkToken.requestorID	
ucaf.authenticationData	**For Mastercard requests, set this field to the Identity Check cryptogram.**
ucaf.collectionIndicator	**For Mastercard requests, set the value to 2.**

Related Information

- *API field reference guide for the Simple Order API*

Simple Order API Example: Authorizations with Payment Network Tokens

Request

```
<requestMessage>
  <merchantID>Foster_City_Flowers</merchantID>
  <merchantReferenceCode>12345678</merchantReferenceCode>
  <billTo>
    <firstName>Jane</firstName>
    <lastName>Smith</lastName>
    <street1>100 Main Street</street1>
    <street2>Suite 1234</street2>
    <city>Foster City</city>
    <state>CA</state>
    <postalCode>94404</postalCode>
    <country>US</country>
    <email>test@cybs.com</email>
  </billTo>
  <purchaseTotals>
    <currency>USD</currency>
    <grandTotalAmount>16.00</grandTotalAmount>
  </purchaseTotals>
  <card>
    <accountNumber>4111111111111111</accountNumber>
    <expirationMonth>12</expirationMonth>
    <expirationYear>2031</expirationYear>
  </card>
  <ccAuthService run="true">
    <networkTokenCryptogram>qE5juRwDzAUFBAkEHuWW9PiBkWv=</networkTokenCryptogram>
  </ccAuthService>
  <paymentNetworkToken>
    <transactionType>1</transactionType>
  </paymentNetworkToken>
</requestMessage>
```

Successful Response

```
<c:replyMessage>
  <c:merchantReferenceCode>Postman-1684858432</c:merchantReferenceCode>
  <c:requestID>6848584316126969103007</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:purchaseTotals>
    <c:currency>USD</c:currency>
  </c:purchaseTotals>
  <c:ccAuthReply>
    <c:reasonCode>100</c:reasonCode>
    <c:amount>16.00</c:amount>
    <c:authorizationCode>888888</c:authorizationCode>
    <c:avsCode>X</c:avsCode>
    <c:avsCodeRaw>I1</c:avsCodeRaw>
    <c:authorizedDateTime>2023-05-23T16:13:51Z</c:authorizedDateTime>
    <c:processorResponse>100</c:processorResponse>
    <c:reconciliationID>78849228NHPFQCKD</c:reconciliationID>
    <c:paymentNetworkTransactionID>123456789619999</c:paymentNetworkTransactionID>
   </c:ccAuthReply>
   <c:card>
     <c:cardType>001</c:cardType>
   </c:card>
</c:replyMessage>
```

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

Use these required fields for capturing an authorization.

ccCaptureService_authRequestID	
ccCaptureService_run	
merchantID	
merchantReferenceCode	**Set the value to `merchant_ref_number` value used in corresponding authorization request.**
purchaseTotals_currency	
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
```

Follow-On Credits

This section provides the information you need in order to process a follow-on credit, which is linked to a capture or sale. You must request a follow-on credit within 180 days of the authorization.

When your account is enabled for credit authorizations, also known as purchase return authorizations, Cybersource authenticates the card and customer during a credit request. Every credit request is automatically authorized.

Credit authorization results are returned in these response fields:

- **ccCreditReply_authorizationCode**
- **ccCreditReply_paymentNetworkTransactionID**
- **ccCreditReply_processorResponse**

When you request a void for the credit and the credit is voided. If your account is enabled for credit authorizations, the credit authorization is also reversed.

Endpoint

Set the **ccCreditService_run** field to `true`.

Send the request to `https://ics2ws.ic3.com/commerce/1.x/transactionProcessor`.

Required Fields for Processing a Follow-On Credit

Use these required fields for processing a follow-on credit.

ccCreditService_captureRequestID	
ccCreditService_run	**Set the value to `true`.**
merchantID	

merchantReferenceCode	**Set to merchantReferenceCode value used in corresponding capture or sale request.**
purchaseTotals_currency	
purchaseTotals_grandTotalAmount	

Related Information

- *API field reference guide for the Simple Order API*

Simple Order Example: Processing a Follow-On Credit

Request

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.86">
 <merchantID>merchantID</merchantID>
 <merchantReferenceCode>merchantRefCode</merchantReferenceCode>
 <purchaseTotals>
  <currency>USD</currency>
  <grandTotalAmount>1.01</grandTotalAmount>
 </purchaseTotals>
 <ccCreditService run="true">
  <captureRequestID>captureRequestID</captureRequestID>
 </ccCreditService>
</requestMessage>
```

Response to a Successful Request

```
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.86">
 <c:merchantReferenceCode>Postman-1666641056</c:merchantReferenceCode>
 <c:requestID>6666410568976150003010</c:requestID>
 <c:decision>ACCEPT</c:decision>
 <c:reasonCode>100</c:reasonCode>
 <c:purchaseTotals>
  <c:currency>USD</c:currency>
 </c:purchaseTotals>
 <c:ccCreditReply>
  <c:reasonCode>100</c:reasonCode>
  <c:requestDateTime>2022-10-24T19:50:57Z</c:requestDateTime>
  <c:amount>1.01</c:amount>
  <c:reconciliationID>6691571329CM5P99</c:reconciliationID>
  <c:authorizationCode>831111</c:authorizationCode>
  <c:processorResponse>00</c:processorResponse>
  <c:paymentNetworkTransactionID>222222222222222</c:paymentNetwork>
 </c:ccCreditReply>
</c:replyMessage>
```

Stand-Alone Credits

This section shows you how to process a stand-alone credit, which is not linked to a capture or sale. There is no time limit for requesting a stand-alone credit.

When your account is enabled for credit authorizations, also known as purchase return authorizations, Cybersource authenticates the card and customer during a credit request. Every credit request is automatically authorized.
Credit authorization results are returned in these response fields:

- **ccCreditReply_authorizationCode**
- **ccCreditReply_paymentNetworkTransactionID**
- **ccCreditReply_processorResponse**

When you request a void for the credit and the credit is voided. If your account is enabled for credit authorizations, the credit authorization is also reversed.

Endpoint

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

Required Fields for Processing a Stand-Alone Credit

Use these required fields for processing a stand-alone credit.

Important

When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 73.

billTo_city	
billTo_country	
billTo_email	
billTo_firstName	
billTo_lastName	
billTo_postalCode	
billTo_state	
billTo_street1	
card_accountNumber	
card_expirationMonth	
card_expirationYear	
ccCreditService	**Set the value to `true`. For example `ccCreditService run="true"`.**
merchantID	

merchantReferenceCode	**Set to merchantReferenceCode value used in corresponding capture request.**
purchaseTotals_currency	
purchaseTotals_grandTotalAmount	

Simple Order Example: Processing a Stand-Alone Credit

Request

```
<requestMessage>
  <billTo>
    <firstName>John</firstName>
    <lastName>Doe</lastName>
    <street1>1295 Charleston Road</street1>
    <city>Mountain View</city>
    <state>CA</state>
    <postalCode>94043</postalCode>
    <country>US</country>
    <email>null@cybersource.com</email>
  </billTo>
  <card>
    <accountNumber>4111111111111111</accountNumber>
    <expirationMonth>12</expirationMonth>
    <expirationYear>2023</expirationYear>
  </card>
  <merchantID>lrsebctest</merchantID>
  <merchantReferenceCode>Postman-1666381004</merchantReferenceCode>
  <purchaseTotals>
    <currency>USD</currency>
    <grandTotalAmount>1.01</grandTotalAmount>
  </purchaseTotals>
  <ccCreditService run="true"/>
</requestMessage>
```

Response to a Successful Request

```
<c:replyMessge>
  <c:merchantReferenceCode>Postman-1666374834</c:merchantReferenceCode>
  <c:requestID>6663748348516429203007</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:purchaseTotals>
    <c:currency>USD</c:currency>
  </c:purchaseTotals>
  <c:ccAuthReply>
    <c:reasonCode>100</c:reasonCode>
    <c:amount>1.01</c:amount>
    <c:authorizationCode>888888</c:authorizationCode>
    <c:avsCode>X</c:avsCode>
    <c:avsCodeRaw>I1</c:avsCodeRaw>
    <c:authorizedDateTime>2022-10-21T17:53:54Z</c:authorizedDateTime>
    <c:processorResponse>100</c:processorResponse>
    <c:reconciliationID>66737280B9CGUCCP</c:reconciliationID>
    <c:paymentNetworkTransactionID>123456789619999</c:paymentNetworkTransactionID>
```

```
  </c:ccAuthReply>
  <c:card>
    <c:cardType>001</c:cardType>
  </c:card>
</c:replyMessge>
```

Voids for a Capture or Credit

This section describes how to void a capture or credit that was submitted but not yet processed by the processor.

Endpoint

Void a Capture
Void a Credit
Set the **VoidService_run** field to `true`.
Send the request to `https://ics2ws.ic3.com/commerce/1.x/transactionProcessor`.

Required Fields for Voiding a Capture or Credit

merchantID	
merchantReferenceCode	
voidService_voidRequestID	**Set this field to the request ID that was included in the authorization response message.**
voidService_run	**Set the value to `true`.**

Simple Order API Example: Voiding a Capture or Credit

Request

```
merchantID=Napa Valley Vacations
merchantReferenceCode=482046C3A7E94F5
voidService_run
voidService_voidRequestID=6522033834410167772169
```

Response to a Successful Request

```
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
```

Debit and Prepaid Card Processing

This section shows you how to process authorizations that use a debit or prepaid card.

Related Information

- See *Debit and Prepaid Card Payments* on page 19 for a description of the debit or prepaid card transactions you can process.

Additional Resources for Debit and Prepaid Payments

For more information, see these guides:

- *API field reference guide for the Simple Order API*
- Github repositories: *https://github.com/Cybersource*

Processing Debit and Prepaid Authorizations

This section shows you how to process an authorization using debit and prepaid cards.

Endpoint

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

Required Fields for Processing Debit and Prepaid Authorizations

Use these required fields for processing debit and prepaid authorizations.

Important

When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 73.

billTo_city	
billTo_country	
billTo_email	
billTo_firstName	
billTo_lastName	
billTo_postalCode	
billTo_state	
billTo_street1	
card_accountNumber	
card_expirationMonth	
card_expirationYear	
ccAuthService_run	**Set the value to `true`.**
merchantID	
merchantReferenceCode	
purchaseTotals_currency	
purchaseTotals_grandTotalAmount	

Related Information

- *API field reference guide for the Simple Order API*

Optional Field for Processing Debit and Prepaid Authorizations

You can use this optional field to include additional information when processing debit and prepaid authorizations.

linkToRequest	**Set this field to the request ID that was returned in the response message from the original authorization request.**

Related Information

- *API field reference guide for the Simple Order API*

Simple Order Example: Processing Debit and Prepaid Authorizations

Request

```
billTo_city=Foster City
billTo_country=US
billTo_email=null@cybersource.com
billTo_firstname=John
billTo_lastname=Smith
billTo_postalCode=40500
billTo_state=CA
billTo_street1=901 Metro Center Blvd
card_accountNumber=41111111XXXXXXXX
card_expirationMonth=12
card_expirationYear=2031
ccAuthService_run=true
merchant_id=pa_ctv_sg101
merchantReferenceCode=rts_6595481893301034778276
purchaseTotals_currency=usd
purchaseTotals_grandTotalAmount=100
```

Response to a Successful Request

```
additionalData=ABC
ccAuthReply_amount=100.00
ccAuthReply_avsCode=Y
ccAuthReply_authorizationCode=831000
ccAuthReply_processorResponse=00
ccAuthReply_authorizedDateTime=2022-08-30T165039Z
ccAuthReply_avsCodeRaw=Y
ccAuthReply_cavvResponseCode=2
ccAuthReply_cavvResponseCodeRaw=2
ccAuthReply_merchantAdviceCode=01
ccAuthReply_merchantAdviceCodeRaw=M001
ccAuthReply_paymentNetworkTransactionID=016153570198200
ccAuthReply_reconciliationReferenceNumber=224216876457
apAuthReply_reconciliationID=6618782389070178232890
card_cardType=001
payerAuthEnrollReply_cardTypeName=VISA
purchaseTotals_currency=usd
merchantReferenceCode=rts_6595481893301034778276
receiptNumber=876457
requestID=6618782389070178232890
```

Enabling Debit and Prepaid Partial Authorizations

Partial authorizations and balance responses are special features that are available for debit cards and prepaid cards. This section shows you how to enable partial authorizations for a specific transaction.

You must use version 1.52 or later of the XML schema to implement partial authorizations or balance responses.

Field Specific to this Use Case

Include this field in addition to the fields required for a standard authorization request:

- Indicate that this request is a partial authorization.
  Set the `ccAuthService_partialAuthIndicator` to `true`.

Endpoint

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

Required Fields for Enabling Debit and Prepaid Partial Authorizations

Use these required fields for enabling debit and prepaid partial authorizations.

> **Important**
>
> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 73.

billTo_city	
billTo_country	
billTo_email	
billTo_firstName	
billTo_lastName	
billTo_postalCode	
billTo_state	
billTo_street1	
card_accountNumber	
card_expirationMonth	
card_expirationYear	
ccAuthService_partialAuthIndicator	**Set the value to `true`.**
ccAuthService_run	**Set the value to `true`.**
merchantID	
merchantReferenceCode	

purchaseTotals_currency

purchaseTotals_grandTotalAmount

Related Information

- *API field reference guide for the Simple Order API*

Optional Field for Enabling Debit and Prepaid Partial Authorizations

You can use these optional fields to include additional information when enabling debit and prepaid partial authorizations.

linkToRequest	**Set this field to the request ID that was returned in the response message from the original authorization request.**

Related Information

- *API field reference guide for the Simple Order API*

Simple Order Example: Enabling Debit and Prepaid Partial Authorizations

Request

```
billTo_street1=201 S. Division St
billTo_city=Ann Arbor
billTo_country=US
billTo_state=MI
billTo_postalCode=48104-2201
billTo_email=test@cybs.com
billTo_firstname=John
billTo_lastname=Deo
card_expirationMonth=12
card_expirationYear=2031
card_accountNumber=5555555555554444
ccAuthService_partialAuthIndicator=true
merchant_id=pa_ctv_sg101
merchantReferenceCode=TC50171_3
purchaseTotals_currency=usd
purchaseTotals_grandTotalAmount=1000.00
```

Response to a Successful Request

```
apCaptureService_authRequestID=6618807769750178232890
apAuthReply_reconciliationID=661880776975017823289O
card_cardType=002
ccAuthReply_amount=1000.00
ccAuthReply_avsCode=Y
ccAuthReply_authorizationCode=831000
ccAuthReply_authorizedDateTime=2022-08-30T173257Z
```

```
ccAuthReply_avsCodeRaw=Y
ccAuthReply_cavvResponseCode=2
ccAuthReply_cavvResponseCodeRaw=2
ccAuthReply_merchantAdviceCode=01
ccAuthReply_merchantAdviceCodeRaw=M001
ccAuthReply_processorResponse=00
ccAuthReply_reconciliationReferenceNumber=224217876503
ccCreditReply_paymentNetworkTransactionID=MCC9689130830
merchantReferenceCode=TC50171_3
payerAuthEnrollReply_cardTypeName=MASTERCARD
purchaseTotals_currency=usd
receiptNumber=876503
requestID=6618807769750178232890
```

Disabling Debit and Prepaid Partial Authorizations

This topic shows you how to successfully disable partial authorizations for specific transactions.

Field Specific to this Use Case

Include this field in addition to the fields required for a standard authorization request:

- Indicate that this request is not a partial authorization.
  Set the `ccAuthService_partialAuthIndicator` to `false`.

Endpoint

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

Required Field for Disabling Debit and Prepaid Partial Authorizations

Use these required fields for disabling debit and prepaid partial authorizations.

Important

When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 73.

billTo_city

billTo_country

billTo_email

billTo_firstName	
billTo_lastName	
billTo_postalCode	
billTo_state	
billTo_street1	
card_accountNumber	
card_expirationMonth	
card_expirationYear	
ccAuthService_partialAuthIndicator	Set the value to `false`.
ccAuthService_run	Set the value to `true`.
merchantID	
merchantReferenceCode	
purchaseTotals_currency	
purchaseTotals_grandTotalAmount	

Related Information

- *API field reference guide for the Simple Order API*

Optional Field for Disabling Debit and Prepaid Partial Authorizations

You can use this optional field to include additional information when disabling debit and prepaid partial authorizations.

linkToRequest	**Set this field to the request ID that was returned in the response message from the original authorization request.**

Related Information

- *API field reference guide for the Simple Order API*

Simple Order Example: Disabling Debit and Prepaid Partial Authorizations

Request

```
billTo_street1=201 S. Division St
billTo_city=Ann Arbor
billTo_country=US
billTo_state=MI
billTo_postalCode=48104-2201
billTo_email=test@cybs.com
```

```
billTo_firstname=John
billTo_lastname=Deo
card_expirationMonth=12
card_expirationYear=2031
card_accountNumber=5555555555554444
ccAuthService_partialAuthIndicator=false
merchant_id=pa_ctv_sg101
merchantReferenceCode=TC50171_3
purchaseTotals_currency=usd
purchaseTotals_grandTotalAmount=1000.00
```

Response to a Successful Request

```
apCaptureService_authRequestID=6643889552520668668655
apAuthReply_reconciliationID=6643889552520668668655
card_cardType=002
ccAuthReply_amount=1000.00
ccAuthReply_avsCode=Y
ccAuthReply_authorizationCode=831000
ccAuthReply_authorizedDateTime=2022-09-28T173257Z
ccAuthReply_avsCodeRaw=Y
ccAuthReply_cavvResponseCode=2
ccAuthReply_cavvResponseCodeRaw=2
ccAuthReply_merchantAdviceCode=01
ccAuthReply_merchantAdviceCodeRaw=M001
ccAuthReply_processorResponse=00
ccAuthReply_reconciliationReferenceNumber=227118876340
ccCreditReply_paymentNetworkTransactionID=MCC8605090928
merchantReferenceCode=TC50171_3
payerAuthEnrollReply_cardTypeName=MASTERCARD
purchaseTotals_currency=usd
receiptNumber=876340
requestID=6618807769750178232890
```

Payer Authentication Processing

This section shows you how to process authorizations that use these payer authentication methods:

- Mastercard: Identity Check
- Visa: Visa Secure

Related Information

- See the *Payer Authentication Developer Guide* for details about payer authentication.

Additional Resources for Payer Authentication

For more information, see these guides:

- *API field reference guide for the Simple Order API*
- Github repositories: *https://github.com/Cybersource*

Providing Payer Authentication Information for Authorization

The values that are returned from payer authentication must be provided when seeking authorization for the transaction. Authentication information that is not included when considering authorization may cause the transaction to be refused or downgraded and prevent the normal liability shift from occurring.

The level of security in payer authentication is denoted by the two digit Electronic Commerce Indicator (ECI) that is assigned to the transaction. These digital values have text equivalents which are assigned to the **e_commerce_indicator** field.

The American Express, Diners, Discover, UPI, and Visa card brands use 05, 06, and 07 digit values to express the authentication level for a 3-D Secure transaction.

Text Values for ECI Values

ECI Value	Meaning	Visa	Diners	Discover	UPI	Amex
05	Authenticated	vbv	pb	dipb	up3ds	aesk
06	Attempted authentication with a cryptogram	vbv_attempted	pb_attempted	dipb_attempted	up3ds_attempted	aesk_attempted
07	Internet, not authenticated	vbv_failure/internet	internet	internet	up3ds_failure/internet	internet

Mastercard and Maestro cards use 00, 01, 02, 06, and 07 digit values to indicate the authentication level of the transaction.

Mastercard/Maestro Text Values for ECI Values

ECI Value	Meaning	Mastercard/Maestro
00	Internet, not authenticated	spa/internet
01	Attempted authentication	spa
02	Authenticated	spa
06	Exemption from authentication or network token without 3#D Secure	spa
07	Authenticated merchant-initiated transaction	spa

The payer authentication response contains other information that needs to be passed on for successful authorization. Be sure to include these fields when requesting a separate authorization:

- **ccAuthService_directoryServerTransactionID** (Mastercard, Maestro, UPI only)
- **ccAuthService_eciRaw**
- **ccAuthService_paresStatus**
- **ccAuthService_paSpecificationVersion**
- **payerAuthEnrollReply_ucafAuthenticationData** (Mastercard/Maestro only)
- **payerAuthValidateReply_ucafCollectionIndicator**(Mastercard/Maestro only)

- **ccAuthService_cavv**
- **ccAuthService_xid**

Mastercard Identity Check

Mastercard Identity Check is the authentication service in the Mastercard card network that uses the 3-D Secure protocol in online transactions to authenticate customers at checkout.
Mastercard Identity Check generates a unique, 32-character transaction token, called the account authentication value (AAV) each time a Mastercard Identity Check-enabled account holder makes an online purchase. The AAV binds the account holder to a specific transaction. Mastercard Identity Check transactions use the universal cardholder authentication field (UCAF) as a standard to collect and pass AAV data.
Before implementing payer authentication for Mastercard Identity Check, contact customer support to have your account configured for this feature.

Fields Specific to the Mastercard Identity Check Use Case

These API fields are required specifically for this use case.

	Set this field to the transaction ID returned by Mastercard Identity Check during the authentication process. Set this field to the Mastercard Identity Check version returned by Mastercard Identity Check during the authentication process.
ucaf_collectionIndicator	Set to the last digit of the raw ECI value returned from authentication. For example, if ECI=02, this value should be 2. Set this field to one of these values: • `spa`: Successful authentication (3-D Secure value of `02`). • `spa`: Authentication was attempted (3-D Secure value of `01`). • `spa` or `internet`: Authentication failed or was not attempted (3-D Secure value of `00`)

Endpoint

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

Required Fields for Processing an Authorization Using Mastercard Identity Check

Use these required fields to process an authorization using Mastercard Identity Check.

Important

When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 73.

billTo_city	
billTo_country	
billTo_email	
billTo_firstName	
billTo_lastName	
billTo_postalCode	
billTo_state	
billTo_street1	
card_accountNumber	
card_expirationMonth	
card_expirationYear	
ccAuthService_run	Set the value to `true`.
ccAuthService_cavv	
ccAuthService_commerceIndicator	Set this field to one of these values: • `spa`: Successful authentication (3-D Secure value of `02`). • `spa`: Authentication was attempted (3-D Secure value of `01`). • `spa` or `internet`: Authentication failed or was not attempted (3-D Secure value of `00`).
ccAuthService_directoryServerTransactionID	
ccAuthService_paSpecificationVersion	
mercahnt_id	
merchant_referenceCode	

purchaseTotals_currency	
purchaseTotals_grandTotalAmount	
ucaf_collectionIndicator	Set to the last digit of the raw ECI value returned from authentication. For example, if ECI=02, this value should be 2.

Related Information

- *API field reference guide for the Simple Order API*

Simple Order Example: Processing an Authorization Using Mastercard Identity Check

Request

```
billTo_city=Sao Paulo
billTo_country=BR
billTo_email=null@cybersource.com
billTo_firstname=Julia
billTo_lastname=Fernandez
billTo_postalCode=01310-000
billTo_state=SP
billTo_street1=R. Augusta
card_accountNumber=41111111XXXXXXXX
card_expirationMonth=12
card_expirationYear=2023
ccAuthService_run=true
ccAuthService_cavv=ABCDEFabcdefABCDEFabcdef0987654321234567
ccAuthService_commerceIndicator=spa
ccAuthService_paSpecificationVersion=1
merchant_id=MID23
merchant_referenceCode=Merchant_REF
ucaf_collectionIndicator=1
purchaseTotals_currency=mxn
purchaseTotals_grandTotalAmount=100
```

Response to a Successful Request

```
merchantReferenceCode=Merchant_REF
request_id=6461515866500167772420
decision=ACCEPT
reasonCode=100
purchaseTotals_currency=mxn
ccAuthReply_cardCategory=F
ccAuthService_reconciliationID=ZUDCXJO8KZRFXQJJ
ccAuthReply_reasonCode=100
ccAuthReply_amount=100.00
ccAuthReply_avsCode=5
ccAuthReply_authorizationCode=570110
ccAuthReply_processorResponse=1
ccAuthReply_authorizedDateTime=2022-03-01T161947Z
ccAuthReply_paymentNetworkTransactionID=111222
```

Visa Secure

Visa Secure is the authentication service in the Visa card network that uses the 3-D Secure protocol to authenticate customers at checkout. This authentication is a two-step process. First, the cardholder is authenticated by 3-D Secure. Then, the transaction is authorized based on the 3-D Secure evaluation. This section explains how to authorize a card payment based on the 3-D Secure evaluation.
Before implementing Visa Secure, contact customer support to have your account configured for this feature.

Fields Specific to the Visa Secure Use Case

These API fields are required specifically for this use case.

ccAuthService_commerceIndicator	Set the value to `vbv` for a successful authentication (3-D Secure value of `05`), `vbv_attempted` if authentication was attempted but did not succeed (3-D Secure value of `06`), or `vbv_failure` if authentication failed (3-D Secure value of `07`).
ccAuthService_cavv	Required when payer authentication is successful.

Endpoint

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

Related Information

- *API field reference guide for the Simple Order API*

Required Fields for Processing an Authorization Using Visa Secure

Use these required fields to process an authorization using Visa Secure.

Important

When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. Refer to the Payments guide for more information about relaxed requirements in payment transactions.

Required Fields

billTo_city	
billTo_country	
billTo_email	
billTo_firstName	
billTo_lastName	
billTo_postalCode	
billTo_state	
billTo_street1	
card_accountNumber	
card_expirationMonth	
card_expirationYear	
ccAuthService_cavv	This field is required when payer authentication is successful. Otherwise, this field is optional.
ccAuthService_commerceIndicator	Set the value of this field to one of these values: • `vbv`: Successful authentication (EMV 3-D Secure value of `05`). • `vbv_attempted`: Authentication was attempted (EMV 3-D Secure value of `06`). • `vbv_failure`: or `internet`: Authentication failed or was not attempted (EMV 3-D Secure value of `07`)
ccAuthService_run	Set the value of this field to `true`.
ccAuthService_xid	
merchant_referenceCode	
purchaseTotals_currency	
purchaseTotals_grandTotalAmount	

Related Information

- *API field reference guide for the Simple Order API*

Simple Order Example: Validating and Authorizing an Authorization

Request

```
billTo_city=Sao Paulo
billTo_country=BR
billTo_email=julia@example.com
billTo_firstname=Julia
billTo_lastname=Fernandez
billTo_postalCode=01310-000
billTo_state=SP
billTo_street1=R. Augusta
card_accountNumber=41111111XXXXXXXX
card_expirationMonth=12
card_expirationYear=2023
ccAuthService_run=true
ccAuthService_cavv=ABCDEFabcdefABCDEFabcdef0987654321234567
ccAuthService_commerceIndicator=vbv
ccAuthService_xid=MID23
merchant_referenceCode=Merchant_REF
purchaseTotals_currency=mxn
purchaseTotals_grandTotalAmount=100
```

Response to a Successful Request

```
merchantReferenceCode=Merchant_REF
request_id=6461515866500167772420
decision=ACCEPT
reasonCode=100
purchaseTotals_currency=mxn
ccAuthReply_cardCategory=F
ccAuthService_reconciliationID=ZUDCXJO8KZRFXQJJ
ccAuthReply_reasonCode=100
ccAuthReply_amount=100.00
ccAuthReply_avsCode=5
ccAuthReply_authorizationCode=570110
ccAuthReply_processorResponse=1
ccAuthReply_authorizedDateTime=2022-03-01T161947Z
ccAuthReply_paymentNetworkTransactionID=111222
```

Relaxed Requirements for Address Data and Expiration Date in Payment Transactions

With relaxed requirements for address data and the expiration date, not all standard payment request fields are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required.

Requirements

You must contact customer support in order to enable relaxed requirements for address data and expiration date.

Services

Relaxed requirements for address data and expiration date are supported for these services:

- Authorization
- Capture
- Stand-alone credit
- Subscription create
- Subscription update

Relaxed Fields

Important

When relaxed requirements for address data and expiration date are enabled for your Cybersource account, and your service request does not include one or more of the fields in the following list, you increase the risk of declined transactions and fraud depending on your location, your processor, and the cardholder's issuing bank.

It is your responsibility to determine whether a field is required for the transaction you are requesting. For example, an issuing bank can decline an authorization request for a recurring transaction with a Visa Europe card if the expiration date is incorrect, invalid, or missing. If you do not provide the correct expiration date for a recurring transaction the authorization request may be declined.

billTo_city	
billTo_country	
billTo_email	
billTo_firstName	
billTo_lastName	
billTo_postalCode	**When you include this field in your request, you must also include billTo_country**
billTo_state	
billTo_street1	
card_expirationMonth	**When you include this field in your request, you must also include card_expirationYear.** **This field is required for payment network token transactions and subscription creation requests.**
card_expirationYear	**When you include this field in your request, you must also include card_expirationMonth.** **This field is required for payment network token transactions and subscription creation requests.**

Processing Payments Using Credentials

This section provides the information you need in order to process payments using credentials.

Additional Resources for Credentialed Transactions

For more information, see these guides:

- *API field reference guide for the Simple Order API*
- Github repositories: *https://github.com/Cybersource*

Customer-Initiated Transactions with Credentials on File

A customer-initiated transaction (CIT) is a transaction initiated by the customer. There are two types of CITs:

- Customer transactions during which the credentials are stored for future customer-initiated transactions.
- Customer transactions during which the credentials are stored for future merchant-initiated transactions.

Customers can initiate a CIT at a merchant payment terminal, through an online purchase transaction, or by making a purchase using a previously stored credential. When storing cardholder data for a CIT, you must also include 3-D Secure authentication credentials to ensure that the CIT can successfully process. Authentication credentials can be stored for future use with the card credentials by doing a non-payment authentication (NPA).

Business Center

You can create a new customer-initiated transaction in the Business Center by going to the One-Time Payments section and requesting a new authorization. When you have entered the customer's information, you can store the customer's credentials with the customer's permission in the Payment Information section. By doing so, you can perform merchant-initiated transactions for payments that the customer has pre-approved.

Storing Customer Credentials with a CIT and PAN

Before you can perform a merchant-initiated transaction (MIT) or a customer-initiated transaction (CIT) with credentials-on-file (COF), you must store the customer's credentials for later use. Further, before you can store the user's credentials, you must get the customer's consent to store their private information. This is also known as establishing a relationship with the customer.

Endpoint

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

Required Fields for Storing Customer Credentials During a CIT

Use these required fields for storing customer credentials during a customer-initiated transaction.

> **Important**
>
> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 73.

billTo_city

billTo_country

billTo_email

billTo_firstName

billTo_lastName

billTo_postalCode

billTo_state

billTo_street1

card_accountNumber

card_expirationMonth

card_expirationYear

ccAuthService_run	**Set the value to `true`.**
merchantID	
merchantReferenceCode	
purchaseTotals_currency	
purchaseTotals_grandTotalAmount	
subsequentAuthFirst	**Set the value to `true`.**

Simple Order Example: Storing Customer Credentials During a CIT

Request

```
billTo_city=Foster City
billTo_country=US
billTo_email=null@cybersource.com
billTo_firstname=John
billTo_lastname=Smith
billTo_state=CA
billTo_postalCode=94404
billTo_street1=201 S. Division St.
card_expirationMonth=12
card_expirationYear=2031
card_accountNumber=4111111111111111
ccAuthService_run=true
merchantId=pa_ctv_sg101
merchantReferenceCode=33557799
purchaseTotals_currency=USD
purchaseTotals_grandTotalAmount=100.00
subsequentAuthFirst=True
```

Response to a Successful Request

```
additional_processor_response=e1cdcafc-cdbb-4ef7-8788-a1234e844805
request_id=6461515866500167772420
decision=ACCEPT
reasonCode=100
merchantReferenceCode=Merchant_REF
purchaseTotals_currency=mxn
cardCategory=FccAuthService_reconciliationID=ZUDCXJO8KZRFXQJJ
ccAuthReply_amount=100.00
ccAuthReply_avsCode=5
ccAuthReply_authorizationCode=570110
ccAuthReply_processorResponse=1
ccAuthReply_authorizedDateTime=2022-03-01T161947Z
ccAuthReply_paymentNetworkTransactionID=111222
```

Retrieving Stored Customer Credentials During a CIT

After customers store their credentials on file, you can retrieve these credentials to use with subsequent transactions.

Endpoint

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

Required Fields for Retrieving Customer Credentials During a Customer-Initiated Transaction

Use these required fields to retrieve customer credentials during a customer-initiated transaction.

> **Important**
>
> When relaxed requirements for address data and the expiration date are being used, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 73.

billTo_city	
billTo_country	
billTo_email	
billTo_firstName	
billTo_lastName	
billTo_postalCode	
billTo_state	
billTo_street1	
card_accountNumber	
card_expirationMonth	
card_expirationYear	
ccAuthService_run	**Set the value to `true`.**
merchantID	
merchantReferenceCode	
purchaseTotals_currency	
purchaseTotals_grandTotalAmount	
subsequentAuthStoredCredential	**Set the value to `true`.**

Card-Specific Required Field for Retrieving Customer Credentials During a CIT

Discover

Discover requires the authorization amount from the original transaction in addition to the above required fields.

subsequentAuthOriginalAmount

Simple Order Example: Retrieving Customer Credentials During a CIT

Request

```
billTo_city=Foster City
billTo_country=US
billTo_email=null@cybersource.com
billTo_firstname=John
billTo_lastname=Smith
billTo_state=CA
billTo_postalCode=94404
billTo_street1=201 S. Division St.
card_expirationMonth=12
card_expirationYear=2031
card_accountNumber=4111111111111111
ccAuthService_run=true
merchantId=pa_ctv_sg101
merchantReferenceCode=33557799
purchaseTotals_currency=USD
purchaseTotals_grandTotalAmount=100.00
subsequentAuthStoredCredential=True
```

Response to a Successful Request

```
additional_processor_response=e1cdcafc-cdbb-4ef7-8788-a1234e844805
request_id=6461515866500167772420
decision=ACCEPT
reasonCode=100
merchantReferenceCode=Merchant_REF
purchaseTotals_currency=mxn
cardCategory=FccAuthService_reconciliationID=ZUDCXJO8KZRFXQJJ
ccAuthReply_amount=100.00
ccAuthReply_avsCode=5
ccAuthReply_authorizationCode=570110
ccAuthReply_processorResponse=1
ccAuthReply_authorizedDateTime=2022-03-01T161947Z
ccAuthReply_paymentNetworkTransactionID=111222
```