Payments

Simple Order API
FDC Nashville Global

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 Unpublished rights reserved under the copyright laws of the United States.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

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

Confidentiality Notice

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

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

Revision

Version: 24.04

Contents

Recent Revisions to This Document

24.04

Relaxed Requirements for Address Data and Expiration Date	**Added a new section *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 33.**

24.03

Foreign Retail Indicator	**Added new authorization and capture use cases based on the foreign merchant mandate by Visa. See *Authorizations with Foreign Merchants* on page 49 and *Captures with Foreign Merchants* on page 65.**
Timeout Authorization Reversals	**Added an important note about the wait time before requesting a timeout authorization reversal. See *Timeout Authorization Reversals* on page 62.** **Added an important note about the wait time before requesting a timeout authorization reversal. See *Timeout Authorization Reversals* on page 62.**
Token-Based Processing	**Added support for authorizing a payment ignoring a network token. See *Authorize a Payment While Ignoring Network Token*.**

24.02

Follow-on Credits	**The time limit for follow-on credits has been updated. See these topics:**

	• *Credits* on page 22 in the section on follow-on credits • *Timeout Authorization Reversals* on page 62 • *Follow-On Credits* on page 70 • *Timeout Voids for a Capture, Sale, Follow-On Credit, or Stand-Alone Credit* on page 77

24.01

Processing Payments Using Credentials	**Updated the Mastercard Standing Order Payments and the Mastercard Subscription Payments to include TMS information. See** *Mastercard Standing Order Payments* **on page 143 and** *Mastercard Subscription Payments* **on page 147.**
Timeout Reversals	**Added timeout authorization reversals. See** *Timeout Authorization Reversals* **on page 62.**
Timeout Voids	**Added timeout voids. See** *Timeout Voids for a Capture, Sale, Follow-On Credit, or Stand-Alone Credit* **on page 77.**

23.10

This revision contains only editorial changes and no technical updates.

23.09

Zero Amount Authorizations	**Added information about supported card types for** *Zero Amount Authorizations* **on page 55.**
Authorizations with Payment Network Tokens	**Added support for** *Authorizations with Payment Network Tokens* **on page 42.**
Authorizations with Strong Customer Authentication Exemption	**Added support for** *Authorizations with Strong Customer Authentication Exemption* **on page 52.**
Processing Airline Data	**Added support for processing airline data. See** *Airline Data Processing* **on page 87.**

23.08

This revision contains only editorial changes and no technical updates.

23.07

Added notes about merchants in India not being permitted to store PANs. For details, see *Authorizations with Payment Network Tokens* on page 42.
23.07.01: Fixed broken links.

23.06

This revision contains only editorial changes and no technical updates.

23.05

This revision contains only editorial changes and no technical updates.

23.04

Authorization Reversals	**Updated *Authorization Reversals* on page 20 and the section *Authorization Reversals* on page 61.**
Micropayment Authorizations	**Added information about payment authorizations for amounts less than one unit in the transaction's currency. See *Micropayment Authorizations* on page 18.**

23.03

Stored Credential Processing	**Updated the section *Stored Credentials* on page 33.**
Introduction to Payments	**Made changes to the section *Card Types* on page 14.**
Stand-Alone Credits	**Removed the ccCreditService_captureRequestID field from stand-alone credit required fields list and example. See *Required Fields for Processing a Stand-Alone Credit* on page 72 and *Simple Order Example: Processing a Stand-Alone Credit* on page 72.**

About This 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.

Visit the *Cybersource documentation hub* to find additional processor-specific versions of this guide and additional technical documentation.

Conventions

These special statements are used 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.

Customer Support

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

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.

Issuing (Customer) Financial Institutions

An issuing (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 and Mastercard, 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.

Card Types

You can process payments with these card types:

- Co-badged cards
- Co-branded cards
- Credit cards
- Debit cards
- Prepaid cards
- Private label cards
- Quasi-cash

Co-Badged Cards

Co-badged cards are credit and debit cards that integrate two or more payment networks.

Co-Branded Cards

Co-branded cards are credit cards that are branded with a merchant's logo, brand, or other identifier as well as the payment network logo. These cards are not limited for use at the branded merchant and can be used at any merchant that accepts credit 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.

Prepaid Cards

Prepaid cards enable cardholders to pay for goods and services using money stored directly on the card.

Private Label Cards

Private label cards are issued by private companies. They enable cardholders to borrow money to pay for goods exclusively at the issuing company’s stores.

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.

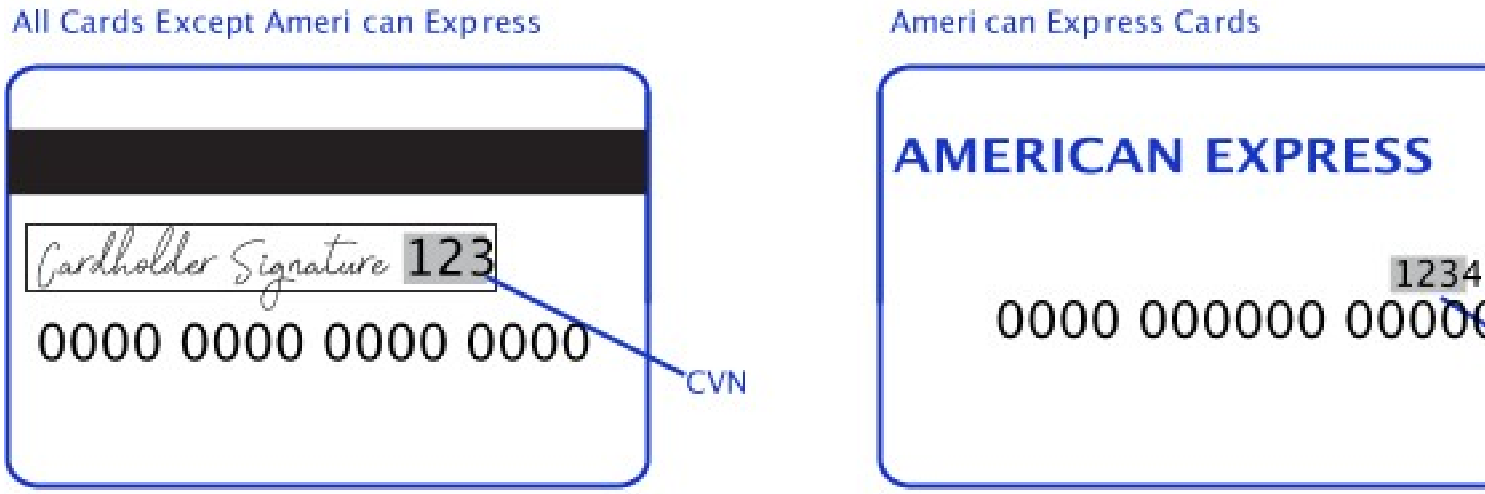
Endpoint

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

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

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 these dynamic currency conversion and merchant remittance funding when processing international transactions.

Dynamic Currency Conversion

Dynamic Currency Conversion (DCC) is a service that enables users to conver the price of a transaction from a merchant's local currency to the customer's billing currency in real time. It is regulated by Mastercard and Visa.
The DCC Gateway service allows users to choose thier own service provider for DCC while complying with Mastercard and Visa payment processing rules for DCC transactions. The currency conversion is performed directly between the user and a third-party provider prior to authorizing a network compliant DCC transaction on their processor connection. Using the DCC Gateway option, you can comply with card mandates and other regulations while using a third-party currency conversion service.

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

Various services are use to process payments. These services enable customers to purchase goods and services, merchants to receive payments from the customer's accounts, merchants to provide refunds, and merchants 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.

Micropayment Authorizations

Micropayments are payments for less than one unit in the transaction’s currency.
For FDC Nashville Global, Cybersource supports micropayment authorizations for these cards:

- American Express
- Discover (in the US only)
- JCB (in the US only)
- Mastercard
- Visa

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 expires after a specific length of time. Thus 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.

Incremental Authorizations

Incremental authorizations are useful when a customer adds products and services to a purchase. After a successful initial authorization, you can request subsequent authorizations and request one capture for the initial authorization and the incremental authorizations.

The incremental authorization service is not the same as the incremental authorization scenario for a merchant-initiated transaction.

Example: Incremental Authorization

This sequence is an example of how incremental authorizations work:

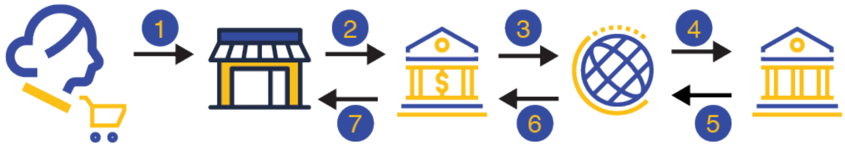
1. The customer reserves a hotel room for two nights at a cost of 200.00 per night. You request an authorization for 400.00. The authorization request is approved.
2. The customer orders dinner through room service the first night. You request an incremental authorization of 50.00 for the dinner.
3. The customer decides to stay an extra night. You request an incremental authorization of 200.00 for the additional night.
4. The customer uses items from the mini-bar. The cost of the mini-bar items is 50.00. You request an incremental authorization of 50.00.
5. When the customer checks out, they sign a receipt for 700.00, which is the total of all costs incurred.
6. You request a capture for 700.00.

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 service from the merchant using a payment card.
2. The merchant sends an authorization request to the acquiring (merchant) bank.
3. The acquiring (merchant) bank forwards the authorization request to the payment network.

4. The payment network forwards the authorization request to the issuer (customer) bank.
5. If funds are available, the issuer (customer) bank reserves the amount of the authorization request and returns an authorization approval to the payment network. If the issuer (customer) bank denies the request, it returns an authorization denial.
6. The payment network forwards the message to the acquiring (merchant) bank.
7. The acquiring (merchant) bank forwards the message to the merchant.

Authorization Reversals

An authorization reversal 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.

Automatic Partial Authorization Reversals

Automatic partial authorization reversals are supported for:

- Credit cards
- Debit cards and prepaid cards.
- Quasi-cash.

If the capture amount is less than the authorization amount, Cybersource automatically performs a partial authorization reversal before it sends the capture request to the processor. The results of a successful partial authorization reversal are:

- The capture amount matches the new authorization amount at the payment card company.
- The hold on the unused credit card funds might be released. The issuing bank decides whether or not to release the hold on unused funds.
  Not all issuers act on a request for a partial authorization reversal. Therefore, Cybersource cannot guarantee that the funds will be released.

Captures

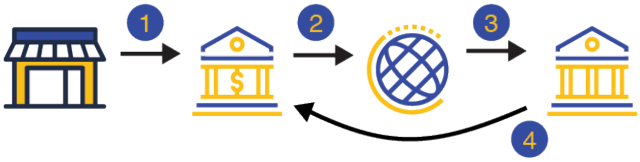
A capture is a follow-on transaction to an authorization. It is used to transfer the authorized funds from the customer's account to the merchant account. To link the authentication transaction to the capture transaction, you include a request ID in your capture request. This request ID is returned to you in the authentication response.
Captures are typically not performed in real time. They are placed in a batch file and sent to the processor, and the processor settles all of the captures at one time. In most cases, these batch files are sent and processed outside of the merchant's business hours. It usually takes 2 to 4 days for the acquiring financial institution to deposit the funds into the merchant account.
When fulfilling only part of a customer's order, do not capture the full amount of the authorization. Capture only the cost of the delivered items. When you deliver the remaining items, request a new authorization, and then capture the new authorization.

Important

It is not possible to perform a capture if a transaction is in a review state, which can occur if you use a fraud management service. You must accept the transaction prior to capture. For more information, see the fraud management documentation in the Business Center.

Capture Workflow

This image and description show the capture workflow:



1. The merchant sends one or more transaction capture requests to the merchant bank (acquirer).
2. The merchant bank sends the capture package to the payment network.
3. The payment network forwards the capture package to the customer bank (issuer).
4. The customer bank settles the transactions and transfers the money to the merchant bank (acquirer).

Important

> The payment processor does not notify Cybersource that the money has been transferred. To ensure that all captures are processed correctly, you should reconcile your capture requests with the capture reports from your processor.

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.

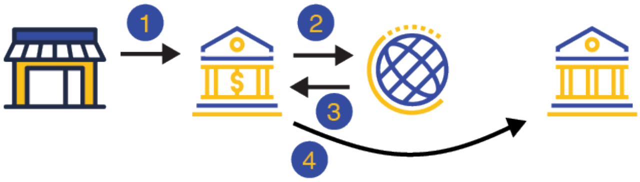
Stand-Alone Credits

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

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.

This image and description show the credit workflow:



1. The merchant sends the credit request to Cybersource.
2. The order information is validated by Cybersource.
3. The credit request is sent to the acquirer.
4. The acquirer transfers the requested funds to the issuer.

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.

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.

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 24
- *Payer Authentication* on page 32
- *Processing Payments Using Credentials* on page 126
- *Relaxed Requirements for Address Data and Expiration Date in Payment Transactions* on page 33
- *Visa Bill Payments* on page 34
- *Visa Consumer Bill Payments* on page 35

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
- Partial authorizations, which are a special feature available for debit cards
- Balance inquiries, which are a special feature available for debit cards

Requirements

In Canada, to process domestic debit transactions on Visa Platform Connect with Mastercard, you must contact customer support to have your account configured for this feature.
When using the Simple Order API in XML format, partial authorizations or balance responses require the XML schema version 1.52 or later.

Related Information

- See *Standard Payment Processing* on page 36 for information that shows you how to use credit card services.
- See *Debit and Prepaid Card Processing* on page 79 for information that shows you how to process authorizations that use a debit or prepaid card.

Airline Data

Airline data processing goes beyond basic payment transactions by allowing you to process specific travel data. This requires you to submit additional information, such as:

- Carrier
- Departure Date
- Destination Airport
- Purchase Date
- Originating Airport
- Ticket Class
- Trip Legs

Supported Card Types

- American Express
- Discover
- Mastercard
- Visa

Supported Acquirers

These Visa Platform Connect acquirers are supported for airline data processing:

- Agricultural Bank of China (ABC)
- Ahli United Bank in Bahrain
- Arab African International Bank (AAIB)
- Asia Commercial Bank (ACB)
- Auckland Savings Bank (ASB)
- Axis Bank Ltd. of India
- Bangkok Bank Ltd.
- Bank Muscat of Oman
- Bank of Ayudhya (BAY)
- Bank of China (BOC)
- Bank of Communications
- Bank Sinarmas (Omise Ltd.)
- Banque Pour Le Commerce Exterieur Lao (BCEL)
- Barclays Bank Mauritius Ltd.
- Barclays Bank Botswana

- Barclays Bank of Ghana Ltd., Barclays Bank of Tanzania Ltd., and Barclays Bank of Uganda Ltd.
- Barclays Bank of Kenya
- Barclays Bank of Zambia
- Barclays Bank Seychelles
- BC Card Co., Ltd.
- BLOM Bank
- Cathay United Bank (CUB)
- Citibank Hongkong and Macau
- Citibank Singapore Ltd.
- Commercial Bank of Qatar
- CrediMax (Bahrain)
- CTBC Bank Ltd.
- FirstRand Bank
- Global Payments Asia Pacific
- Habib Bank Ltd. (HBL)
- HDFC Bank Ltd. of India
- I&M Bank
- ICICI of India
- Korea Exchange Bank (KEB)
- Mashreq
- National Bank of Abu Dhabi (NBAD)
- National Bank of Kuwait (NBK)
- National Commercial Bank
- Network International
- Overseas Chinese Banking Corp (OCBC)
- Promerica in Honduras and Nicaragua
- Qatar National Bank (QNB Group)
- Raiffeisenbank
- Rosbank
- Taishin Bank Ltd.
- United Overseas Bank (UOB) in Singapore and Vietnam
- United Overseas Bank (UOB) in Thailand
- Vietcombank
- VTB24
- Wing Lung Bank

Requirement

When you are ready to go live with airline data processing, contact Cybersource Customer Support to have your account configured to process airline data. If your account is not enabled, and you try to send airline transactions, you will receive an error for invalid data.

Related Information

- See *Airline Data Processing* on page 87 for information that shows you how to process payments that include airline data.

Cybersource Airline Data Processing

Cybersource does not store airline data. Instead, it functions as a pass-through service for the data. Cybersource enforces only the minimal level of field validation.

When you request an airline service, Cybersource responds with certain fields and values to indicate whether the airline data was processed. The response fields for each service are:

- Authorization: **ccAuthReply_enhancedDataEnabled**
- Capture: **ccCaptureReply_enhancedDataEnabled**
- Credit: **ccCreditReply_enhancedDataEnabled**

The possible values for the response fields are:

- `Y`: the airline data was included in the request to the processor.
- `N`: the airline data was not included in the request to the processor.

Cybersource temporarily disables your account's airline data processing capability and contacts you if your airline data transactions produce batching errors when the information is sent to the processor. If this happens, your request is not rejected, but you receive one of the above listed fields with the `N` value in the response indicating that airline data in the request has been ignored and not sent to the processor.

Airline Travel Legs

This section shows you how to process an airline transaction with travel legs for this processor:

- Visa Platform Connect

Using Travel Legs

To include travel legs in an airline transaction, include one or more **airline_leg#_** fields, starting with the **airline_leg_0_** fields.

The **<leg id="#">** fields must start at 0, and you must use consecutive numbers for any additional legs.

For example, these three travel legs are valid:

```
<leg id="0">
  <carrierCode>XX</carrierCode>
</leg>
<leg id="1">
  <carrierCode>XZ</carrierCode>
</leg>
<leg id="2">
  <carrierCode>XX</carrier<code>
</leg>
```

Important

If you skip a number, Cybersource ignores the legs that follow the skipped number.

Travel Leg Limitations

Some processors limit the amount of travel legs for each trip based on card type. For more information, see the Leg Limitations table in the capture section of the processor you are using.

Airline Data Reference Information

This section contains reference information that is useful when using Airline Data.

Airline Document Type Codes

To indicate the purpose of a purchase, set the **airlineData_documentType** field to a value listed in the Code column.

Airline Document Type Codes

Code	Description
01	Passenger ticket
02	Additional collection
03	Excess baggage
04	Miscellaneous charge order (MCO) or prepaid ticket authorization
05	Special service ticket
06	Supported refund
07	Unsupported refund
08	Lost ticket application
09	Tour order voucher
10	Ticket by mail
11	Undercharge adjustment
12	Group ticket
13	Exchange adjustment
14	SPD or air freight
15	In-flight adjustment
16	Agency passenger ticket
17	Agency tour order or voucher
18	Agency miscellaneous charge order (MCO)

Code	Description
19	Agency exchange order
20	Agency group ticket
21	Debit adjustment for duplicate refund or use
22	In-flight merchandise order
23	Catalogue merchandise order
24	In-flight phone charges
25	Frequent flyer fee or purchase
26	Kennel charge
27	Animal transportation charge
28	Firearms case
29	Upgrade charge
30	Credit for unused transportation
31	Credit for class of service adjustment
32	Credit for denied boarding
33	Credit for miscellaneous refund
34	Credit for lost ticket refund
35	Credit for exchange refund
36	Credit for overcharge adjustment
37	Credit for multiple Unused tickets
38	Exchange order
39	Self-service ticket
41	In-flight duty-free purchase
42	Senior citizen discount booklets
43	Club membership fee
44	Coupon book
45	In-flight charges
46	Tour deposit
47	Frequent flyer overnight delivery charge
48	Frequent flyer fulfillment
49	Small package delivery

Code	Description
50	Vendor sale
51	Miscellaneous taxes or fees
52	Travel agency fee
60	Vendor refund or credit
64	Duty free sale
65	Preferred seat upgrade
66	Cabin upgrade
67	Lounge or club access or day pass
68	Agent assisted reservation or ticketing fee
69	Ticket change or cancel fee
70	Trip insurance
71	Unaccompanied minor
72	Standby fee
73	Curbside baggage
74	In-flight medical equipment
75	Ticket or pass print fee
76	Checked sporting or special equipment
77	Dry ice fee
78	Mail or postage fee
79	Club membership fee or temporary trial
80	Frequent flyer activation or reinstatement
81	Gift certificate
82	Onboard or in-flight prepaid voucher
83	Optional services fee
84	Advance purchase for excess baggage
85	Advance purchase for preferred seat upgrade
86	Advance purchase for cabin upgrade
87	Advance purchase for optional services
88	Wi-Fi
89	Packages

Code	Description
90	In-flight entertainment or internet access
91	Overweight bag fee
92	Sleep sets
93	Special purchase fee

Ancillary Service Category Codes

To indicate the service provided in an ancillary purchase, set the **ancillaryData_service_#_categoryCode** and **ancillaryData_service_#_subcategoryCode** fields to a value listed in the Ancillary Service Category Code column.

Ancillary Service Category Codes

Ancillary Service Category Codes	Description
BF	Bundled service
BG	Baggage fee
CF	Change fee
CG	Cargo
CO	Carbon offset
FF	Frequent flyer
GF	Gift card
GT	Ground transport
IE	In-flight entertainment
LG	Lounge
MD	Medical
ML	Meal or beverage
OT	Other
PA	Passenger assist fee
PT	Pets
SA	Seat fees
SB	Standby
SF	Service fee
ST	Store
TS	Travel service

Ancillary Service Category	Codes	Description
UN		Unaccompanied travel
UP		Upgrades
WI		Wi-Fi

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:

- American Express: SafeKey
- 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 112 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 124 for information about how to process payments with relaxed requirements for address data and expiration date.

Stored Credentials

Credentialed transactions are transactions that involve either storing a customer's payment credentials for future transactions or using a customer's already stored payment credentials. When processing a credentialed transaction, you must indicate the type of credentialed transaction and the reason for the transaction.
There are several types of credentialed transactions:

- Customer-Initiated Transactions (CITs): Any transaction a customer is actively participating in such as making a card-present payment, completing an online checkout, or by using a stored credential.
- Merchant-Initiated Transactions (MITs): Any transaction a merchant initiates without the customer's participation such as an industry practice transaction or a standing instruction transaction.
  - Industry Practice Transactions: MITs that are performed as subsequent transactions to a CIT because the initial transaction could not be completed in one transaction. Not every industry practice transaction involves a stored credential. If a stored credential is used only for one transaction, that transaction is not considered a credentialed transaction.
  - Standing Instruction Transactions: MITs that are performed to follow agreed-upon instructions from the customer for the provision of goods and services.

Supported Services

These are the supported merchant-initiated services:

- Installment Transactions
- Mastercard Standing Order Transactions
- Mastercard Subscription Transactions
- Reauthorization
- Recurring Transactions
- Unscheduled Credentials-on-File Transactions

Related Information

- See *Processing Payments Using Credentials* on page 126 for information that shows you how to process transactions using credentials.

Visa Bill Payments

Visa provides a Bill Payment program that enables customers to use their Visa cards to pay their bills. When you participate in this program, Visa requests that you flag the bill payments and credits so they can be easily identified.
The bill payment indicator is accepted no matter which processor you are using, do not use this indicator if you have not signed up with Visa to participate in the program.

> **Important**
>
> This feature is not the same as the Visa Consumer Bill Payment Service.

Supported Services

You can process Visa Bill Payments using these services:

- Authorization
- Credit

Fields Specific to this Use Case

Authorization: include this field in a standard authorization when processing a Visa Bill Payment:

ccAuthService_billPayment

Credit: include this field in a standard credit when processing a Visa Bill Payment:

Include this field in a credit when processing a Visa bill payment:

ccCreditService_billPayment	**Include in a credit.**

Endpoints

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

Requirement

Before you process Visa Bill Payments, you must sign up with Visa to participate in the program.

Related Information

- See *Visa Bill Payments Processing* on page 154 for information that shows you how to process Visa Bill Payment transactions.

Visa Consumer Bill Payments

The Visa Consumer Bill Payment Service (CBPS) is an optional service that enables you to provide bill payment services to consumers using a Visa card. Cardholders can pay one or more of their bills in one transaction using a third-party intermediary rather than paying each bill separately. When you request a Visa CBPS authorization, you must include the **invoiceHeader_businessApplicationID** field set to CB.

> **Important**
>
> This feature is not the same as Visa Bill Payments.

Supported Service

You can process Visa Consumer Bill Payments using the authorization service.

Requirements

> **Important**
>
> As a Visa CBPS provider you must comply with all Visa rules. For the Visa CBPS rules, see Rule ID# 0030635 of the Visa Rules "Acquirer Responsibilities Related to Consumer Bill Payment Service Providers" at *https://usa.visa.com/dam/VCOM/download/about-visa/visa-rules-public.pdf*.

Before you process Visa CBPS payments, you must:

- Contact your acquirer to register your account for the Visa Consumer Bill Pay Service.
- Contact Cybersource customer support to have your account configured for this feature.

Related Information

- See *Visa Consumer Bill Payment Processing* on page 158 for a instructions that show you how to process a Visa Consumer Bill Payment.

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 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 using relaxed requirements for address data and the expiration date, 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 124.

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 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

```
additional_processor_response=e1cdcafc-cdbb-4ef7-8788-a1234e844805
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
requestToken=Axj/7wSTZ2ltSmjDPebrABEg1buWrNi2awW9mNQpt26a
      VBa2DsAU0qC1sHZpA6cQIz4ZNuOAZxjceFoE5NnaW1KaMM95usAAgxhp
```

Response to a Declined Request

```
requestID=6629977932421985593067
requestToken=Axj77w5434574i9uAabDoL
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_#_alternativeTaxAmount**
- **item_#_alternativeTaxID**
- **item_#_alternativeTaxTypeApplied**
- **item_#_buyerRegistration**
- **item_#_commodityCode**
- **item_#_DiscountAmount**
- **item_#_discountIndicator**
- **item_#_discountRate**
- **item_#_grossNetIndicator**
- **item_#_invoiceNumber**
- **item_#_localTax**
- **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_#_taxAmount**
- **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 using relaxed requirements for address data and the expiration date, 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 124.

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=94053
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 using relaxed requirements for address data and the expiration date, 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 124.

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	**Required on Visa Platform Connect**
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

```
   <soap:Body>
      <c:replyMessage>
         <c:merchantReferenceCode>Postman-1684858432</c:merchantReferenceCode>
         <c:requestID>6848584316126969103007</c:requestID>
         <c:decision>ACCEPT</c:decision>
         <c:reasonCode>100</c:reasonCode>

 <c:requestToken>Axj/7wSTc0MzLimjhoKfABEg3cOGjlkycTpFCNRhy4iiOX4Rn4QFRHL8Iz8OkB8xiPDJpJl6MWWg4kCcm5oZmXFNH
+AAA3QNj</c:requestToken>
         <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>
   </soap:Body>
```

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 can be returned for any of the card types on the Discover Network. When CVN results are returned to you, it is your responsibility to decide whether or not to accept the transaction.
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).

> **Important**
>
> When using relaxed requirements for address data and the expiration date, 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 124.

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 Foreign Merchants

Visa mandates marketplaces identify domestic marketplace transactions where the marketplace and issuer are in the same country, but the retailer is in a different country. For marketplaces in the European Economic Area (EEA) and the UK (and Gibraltar), this includes transactions where the marketplace and the issuer are within the EEA, UK, and Gibraltar, but the retailer is not located within the EEA, UK, and Gibraltar. This is flagged using the Foreign Retail Indicator (FRI).

> **Important**
>
> This feature is intended to be used during a settlement. While you can include this information in an authorization, this is not the preferred method.

When you include the **invoiceHeader_merchantDescriptorCountry** and **invoiceHeader_submerchantCountry** fields and the merchant and submerchant are located in separate locations, within the authorization request, the transaction includes the foreign retail indicator flag. However, this flag is only submitted during settlement process and if the same fields are included in the capture request, the capture request will override the information included in the authorization request.

Fields Specific to this Use Case

These fields are required for this use case:

invoiceHeader_merchantDescriptorCountry	If the Merchant country is not submitted, the merchant country associated with the transaction MID will be used.
invoiceHeader_submerchantCountry	If the Merchant country is not submitted, the merchant country associated with the transaction MID will be used.

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 Basic Authorization

Use these required fields for processing a basic authorization.

Important

When using relaxed requirements for address data and the expiration date, 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 124.

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`.
invoiceHeader_merchantDescriptorCountry	
invoiceHeader_submerchantCountry	
merchantID	
merchantReferenceCode	
purchaseTotals_currency	
purchaseTotals_grandTotalAmount	

Related Information

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

Simple Order Example: Processing a Authorization with a Foreign Merchant

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
invoiceHeader_merchantDescriptorCountry
invoiceHeader_submerchantCountry
merchant_id=npr_paymentech
merchant_referenceCode=TC42703-1
purchaseTotals_currency=usd
purchaseTotals_grandTotalAmount=100
```

Response to a Successful Request

```
additional_processor_response=e1cdcafc-cdbb-4ef7-8788-a1234e844805
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
requestToken=Axj/7wSTZ2ltSmjDPebrABEg1buWrNi2awW9mNQpt26a
        VBa2DsAU0qC1sHZpA6cQIz4ZNuOAZxjceFoE5NnaW1KaMM95usAAgxhp
```

Response to a Declined Request

```
requestID=6629977932421985593067
requestToken=Axj77w5434574i9uAabDoL
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 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
B2B Corporate Card Transaction	ccAuthService_ secureCorporatePaymentIndicator	1
Delegated Authentication	ccAuthService_ delegatedAuthenticationExemptionIndicator	1
Follow-On Installment	ccAuthService_ commerceIndicator	installment
Low-Risk Transaction	ccAuthService_ riskAnalysisExemptionIndicator	1
Low-Value Transaction	ccAuthService_ lowValueExemptionIndicator	1
Merchant-Initiated Transaction	subsequentAuthReason	
Stored Credential Transaction	subsequentAuthStoredCredential	1
Trusted Merchant Transaction	ccAuthService_ trustedMerchantExemptionIndicator	1

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 using relaxed requirements for address data and the expiration date, 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 124.

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

```
<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"/>
```

Response to a Successful Request

```
<c:merchantReferenceCode>Postman-1666374834</c:merchantReferenceCode>
<c:requestID>6663748348516429203007</c:requestID>
<c:decision>ACCEPT</c:decision>
<c:reasonCode>100</c:reasonCode>
<c:requestToken>
  Axj/7wSTaT4V6Ox9sAY/ABEg2bN2bdk4YQnMOPVhw6Cft2LvhsgKft2Lvhs
  6QHyke0MmkmXoxZaDiQJybSfCvR2PtgDH4AAAyhRH
</c:requestToken>
<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:pos>
 <c:terminalID>111111
</c:terminalID></c:pos>
```

Zero Amount Authorizations

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. This section provides information you need to process a zero amount authorization.

Processor-Specific Information

FDC Nashville Global	**AVS and CVN are supported. CVN is not supported for American Express.** **Card types: American Express, Diners Club, Discover, Mastercard, Visa** **Discover and Diners Club—only USD is supported for zero amount authorizations.** **American Express, Mastercard, and Visa—all currencies that are supported for standard**

authorizations are also supported for zero amount authorizations.

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 using relaxed requirements for address data and the expiration date, 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 124.

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
```

Final Authorization Indicator

The purpose of this feature is to ensure that unused funds are reversed, so that customer's funds are available again when an order is not fulfilled.

For an authorization with an amount greater than zero, indicate whether the authorization is a final authorization, a preauthorization, or an undefined authorization.

You can set a default authorization type in your account. To set the default authorization type in your account, contact customer support.

Chargeback protection is in effect for seven days after the authorization.

Supported Services

- Authorization
- Incremental authorization

Supported Card Types

- Co-badged Mastercard and mada. You must identify the card type as Mastercard. Supported only on Visa Platform Connect.
- Maestro (International)
- Maestro (UK Domestic)
- Mastercard

Requirements for Final Authorizations

For a final authorization:

- The authorization amount must be greater than zero.
- The authorization amount must be the final amount that the customer agrees to pay.
- The authorization should not be cancelled after it is approved except when a system failure occurs.
- The authorization must be submitted for capture within seven calendar days of its request.
- The capture amount and currency must be the same as the authorization amount and currency.

Pre-authorizations

A pre-authorization enables merchants to authorize a charge when the final amount is unknown, which is typical for hotel, 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 preauthorization 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.

Unmarked Authorizations

An authorization is unmarked when the default authorization type is not set in your account and you do not include the **authIndicator** field in the authorization request. To set the default authorization type in your account, contact customer support.
Unmarked authorizations are supported only in the US, Canada, Latin America, and Asia Pacific. They are not supported in Europe, Russia, Middle East, and Africa.
Cybersource does not set a mark or indicator for the type of authorization in the request that is sent to the processor.

Important

> Your acquirer processes an unmarked authorization as a final authorization, a preauthorization, or an undefined authorization. Contact your acquirer to learn how they process unmarked authorizations.

Requirements for Unmarked Authorizations

For an unmarked authorization:

- The authorization amount must be greater than zero.
- The authorization amount can be different from the final transaction amount.

Undefined Authorizations

An authorization is undefined when you set the default authorization type in your account to undefined and do not include the **authIndicator** field in the authorization request. To set the default authorization type in your account, contact customer support.

Undefined authorizations are supported only in the U.S., Canada, Latin America, and Asia Pacific. They are not supported in Europe, Russia, Middle East, and Africa.

Chargeback protection is in effect for seven days after the authorization.

Requirements for Undefined Authorizations

For an undefined authorization:

- The authorization amount must be greater than zero.
- The authorization amount can be different from the final transaction amount.
- The authorization should not be cancelled after it is approved except when a system failure occurs.
- The authorization must be submitted for capture within seven calendar days of its request.
- When you do not capture the authorization, you must reverse it; otherwise, Mastercard charges an additional fee for the transaction.

Required Fields for Final Authorizations

Use these required fields for final authorizations and preauthorizations.

> **Important**
>
> When using relaxed requirements for address data and the expiration date, 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 124.

authIndicator	**Set the value to `0` for preauthorizations, or to `1` for final authorizations. Do not include this field for unmarked or undefined authorizations.**

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	

Simple Order Example: Final Authorizations

Request

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.{{WSDLVersion}}">
 <merchantID>testrest</merchantID>
 <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>test@cybs.com</email>
 </billTo>
 <purchaseTotals>
   <currency>USD</currency>
   <grandTotalAmount>1.02</grandTotalAmount>
 </purchaseTotals>
 <card>
   <accountNumber>4111111111111111</accountNumber>
   <expirationMonth>12</expirationMonth>
   <expirationYear>2023</expirationYear>
   <cardType>001</cardType>
 </card>
 <ccAuthService run="true"/>
 <authIndicator>1</authIndicator>
</requestMessage>
```

Response to a Successful Request

```
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.142">
  <c:merchantReferenceCode>Postman-1691009216</c:merchantReferenceCode>
  <c:requestID>6910092160816328603011</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:requestToken>Axj/7wSTdpjK8QZssfWDABEg3cMXLRu4cVpdGrRnuaKiOZMuP9AFRHMmXH
+mkB8xkdDJpJl6MWWg4kCcm7TGV4gzZY+sGAAAJA8c</c:requestToken>
  <c:purchaseTotals>
    <c:currency>USD</c:currency>
  </c:purchaseTotals>
  <c:ccAuthReply>
    <c:reasonCode>100</c:reasonCode>
    <c:amount>1.02</c:amount>
    <c:authorizationCode>888888</c:authorizationCode>
    <c:avsCode>X</c:avsCode>
    <c:avsCodeRaw>I1</c:avsCodeRaw>
    <c:authorizedDateTime>2023-08-02T20:46:56Z</c:authorizedDateTime>
    <c:processorResponse>100</c:processorResponse>
    <c:reconciliationID>78194788VKQUQO9Q</c:reconciliationID>
    <c:paymentNetworkTransactionID>123456789619999</c:paymentNetworkTransactionID>
  </c:ccAuthReply>
  <c:card>
    <c:cardType>001</c:cardType>
  </c:card>
  <c:pos>
    <c:terminalID>111111</c:terminalID>
  </c:pos>
</c:replyMessage>
```

Authorization Reversals

This section provides the information you need in order to process an authorization reversal.

Reversing an authorization releases the hold on the customer’s payment card funds that the issuing bank placed when processing the authorization.

Endpoint

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

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

Required Fields for Processing an Authorization Reversal

ccAuthReversalService_authRequestID	**Set this field to the request ID that was included in the authorization response message.**
ccAuthReversalService_run	**Set 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=1019827520348290570293
merchantReferenceCode=482046C3A7E94F5BD1FE3C66C
decision=ACCEPT
reasonCode=100
ccAuthReversalReply_amount=49.95
purchaseTotals_currency=USD
ccAuthReversalReply_reasonCode=100
ccAuthReversalReply_reconciliationID=1094820975023470
```

Timeout Authorization Reversals

When you do not receive a response message after sending an authorization request, this feature enables you to reverse the authorization that you requested.

Important

> Wait 60 seconds before requesting a timeout authorization reversal.

Include the **merchantTransactionIdentifier** field in the original request for an authorization. The value of the merchant transaction ID must be unique for 180 days. When the original transaction fails, the response message for the reversal request includes these fields:

- **originalTransaction_amount**
- **originalTransaction_reasonCode**

Requirements

Unless your processor supports authorization reversal after void (ARAV), timeout authorization reversals are supported only for authorizations that have not been captured and settled.

Required Fields for Processing a Timeout Authorization Reversal

Use these required fields for processing a timeout 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	**Identifier that links the reversal request to the original request.**
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.**

Related Information

- *API Simple Order API Field Reference*

Simple Order Example: Processing a Merchant-Initiated Authorization Reversal

Request

```
ccAuthReversalService_authRequestID=6522033834410167772169
ccAuthReversalService_run=true
merchantReferenceCode=482046C3A7E94F5BD1FE3C66C
```

```
merchantTransactionIdentifier=987654321
purchaseTotals_currency=USD
purchaseTotals_grandTotalAmount=49.95
```

Response to a Successful Request

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

Captures

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

Endpoint

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

Required Fields for Capturing an Authorization

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
```

Captures with Foreign Merchants

Visa mandates marketplaces identify domestic marketplace transactions where the marketplace and issuer are in the same country, but the retailer is in a different country. For marketplaces in the European Economic Area (EEA) and the UK (and Gibraltar), this includes transactions where the marketplace and the issuer are within the EEA, UK, and Gibraltar, but the retailer is not located within the EEA, UK, and Gibraltar. This is flagged using the Foreign Retail Indicator (FRI).

When you include the **invoiceHeader_merchantDescriptorCountry** and **invoiceHeader_submerchantCountry** fields and the merchant and submerchant are located in separate locations, within the capture request, the transaction includes the foreign retail indicator flag.

If you include the **invoiceHeader_merchantDescriptorCountry** and **invoiceHeader_submerchantCountry** fields in both an authorization request and a capture request, the information set in the capture request will override the information in the authorization request.

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
invoiceHeader_merchantDescriptorCountry=GB
invoiceHeader_submerchantCountry=AU
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
```

Multiple Partial Captures

This section shows you how to process multiple partial captures for an authorization.

This feature enables you to request multiple partial captures for one authorization. A multiple partial capture allows you to incrementally settle authorizations over time. Ensure that the total amount of all the captures does not exceed the authorized amount.

Fields Specific to This Use Case

These API request fields and values are specific to this use case:

ccCaptureService_sequence

ccCaptureService_totalCount

Limitations

Multiple partial captures are supported only for card-not-present transactions; they are not supported for card-present transactions.

Endpoint

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

Required Fields for Processing Multiple Partial Captures

ccCaptureService_authRequestID

ccCaptureService_run	
ccCaptureService_sequence	For the final capture request, set this field and **ccCaptureService_totalCount** to the same value. **For Visa and Mastercard, the value for this field is sent to the processor. For all card types, the value for this field is used to determine whether to initiate an automatic partial authorization reversal.**
ccCaptureService_totalCount	**When you do not know the total number of captures that you are going to request, set this field to an estimated value or** `99` **for all capture requests except the final one. For the final capture request, set this field and ccCaptureService_sequence to the same value.** **For Visa and Mastercard, the value for this field is sent to the processor. For all card types, the value for this field is used to determine whether to initiate an automatic partial authorization reversal.**
merchantID	
merchantReferenceCode	**Set the value to** `merchantReferenceCode` **value used in corresponding authorization request.**
purchaseTotals_currency	
purchaseTotals_grandTotalAmount	

Related Information

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

Simple Order Example: Processing Multiple Partial Captures

Request

This example includes special fields that are not required by all processors.

```
ccCaptureService_authRequestID=6744978508916643203008
ccCaptureService_run=true
ccCaptureService_sequence=2
ccCaptureService_totalCount=5
merchantID=npr_paymentech
merchantReferenceCode=Postman-1674497851
purchaseTotals_grandTotalAmount=1.01
purchaseTotals_currency=USD
```

Response to a Successful Request

```
merchantReferenceCode=Postman-1674497851
requestID=6630043878211258349460
decision=ACCEPT
reasonCode=100reasonCode
requestToken=Axj//wSTbaVhaW6UcgfAABEg3bMGLZk2ZzojObHi1IiiOVW
qhpAFRHKrZAzmkB8pefDJpJl6MWWg4gYE5NtpFwS3YNrBQIAAOBLU
purchaseTotals_currency=USD
ccCaptureReply_reasonCode=100
ccCaptureReply_requestDateTime=2023-01-23T18:17:31Z
ccCaptureReply_amount=1.01
ccCaptureReply_reconciliationID=76016263ND3MGETD
```

Forced Captures

This feature allows merchants to process authorizations obtained through an organization other than Cybersource. For example, a merchant might call their processor to request a manual authorization, at which point they can request a forced capture of the authorization.
A manual authorization cannot be captured for more than the original authorization amount, and the authorization expires after seven days.

Supported Services

- Authorization

Required Fields for Forced Captures

Use these required fields for processing forced captures.

authIndicator	**Set the value to `0` for preauthorizations, or to `1` for final authorizations..**
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: Forced Captures

Request

```
<requestMessage
  xmlns="urn:schemas-cybersource-com:transaction-data-1.{{WSDLVersion}}">
  <merchantID>testrest</merchantID>
  <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>test@cybs.com</email>
  </billTo>
  <purchaseTotals>
    <currency>USD</currency>
    <grandTotalAmount>1.02</grandTotalAmount>
  </purchaseTotals>
  <card>
    <accountNumber>4111111111111111</accountNumber>
    <expirationMonth>12</expirationMonth>
    <expirationYear>2023</expirationYear>
    <cardType>001</cardType>
  </card>
  <ccAuthService run="true">
    <authType>verbal</authType>
    <verbalAuthCode>ABC123</verbalAuthCode>
  </ccAuthService>
</requestMessage>
```

Response to a Successful Request

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd">
```

```
      <wsu:Timestamp xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd" wsu:Id="Timestamp-576203304">
        <wsu:Created>2023-08-08T16:45:33.955Z</wsu:Created>
      </wsu:Timestamp>
    </wsse:Security>
  </soap:Header>
  <soap:Body>
    <c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.142">
      <c:merchantReferenceCode>Postman-1691513134</c:merchantReferenceCode>
      <c:requestID>6915131335966844703009</c:requestID>
      <c:decision>ACCEPT</c:decision>
      <c:reasonCode>100</c:reasonCode>
      <c:requestToken>Axj77wSTdt65sCA0LtEhABFRHMq+xfYCojmVfYvvSA
+YyUhk0ky9GLLQcSBOTdt65sCA0LtEhAAA2gjZ</c:requestToken>
      <c:purchaseTotals>
        <c:currency>USD</c:currency>
      </c:purchaseTotals>
      <c:ccAuthReply>
        <c:reasonCode>100</c:reasonCode>
        <c:amount>1.02</c:amount>
        <c:authorizationCode>ABC123</c:authorizationCode>
        <c:authorizedDateTime>2023-08-08T16:45:33Z</c:authorizedDateTime>
      </c:ccAuthReply>
      <c:card>
        <c:cardType>001</c:cardType>
      </c:card>
      <c:pos>
        <c:terminalID>111111</c:terminalID>
      </c:pos>
    </c:replyMessage>
  </soap:Body>
</soap:Envelope>
```

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.

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	**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	

Related Information

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

Simple Order Example: Processing a Follow-On Credit

Request

```
<merchantID>merchantID</merchantID>
<merchantReferenceCode>merchantRefCode</merchantReferenceCode>
<purchaseTotals>
  <currency>USD</currency>
  <grandTotalAmount>1.01</grandTotalAmount>
</purchaseTotals>
<ccCreditService run="true">
  <captureRequestID>captureRequestID</captureRequestID>
</ccCreditService>
```

Response to a Successful Request

```
<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:ccCreditReply>
```

Stand-Alone Credit

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.

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 using relaxed requirements for address data and the expiration date, 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 124.

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

```
<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"/>
```

Response to a Successful Request

```
<c:merchantReferenceCode>Postman-1666374834</c:merchantReferenceCode>
<c:requestID>6663748348516429203007</c:requestID>
<c:decision>ACCEPT</c:decision>
<c:reasonCode>100</c:reasonCode>
<c:requestToken>
  Axj/7wSTaT4V6Ox9sAY/ABEg2bN2bdk4YQnMOPVhw6Cft2LvhsgKft
  2Lvhs6QHyke0MmkmXoxZaDiQJybSfCvR2PtgDH4AAAyhRH
</c:requestToken>
<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:pos>
  <c:terminalID>111111</c:terminalID>
</c:pos>
```

Sales

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

Endpoint

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

Required Fields for Processing a Sale

Use these required fields for processing a sale.

> **Important**
>
> When using relaxed requirements for address data and the expiration date, 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 124.

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=2015
card_accountNumber=4111111111111111
card_cardType=001
```

Response to a Successful Request

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

```
requestID=0305782650000167905080
decision=ACCEPT
reasonCode=100
merchantReferenceCode=482046C3A7E94F5
purchaseTotals_currency=USD
ccAuthReply_reconciliationID=ABCDE12345FGHIJ67890
ccAuthReply_cardCategory=F^
ccAuthReply_cardGroup=0
ccAuthReply_reasonCode=100
ccAuthReply_amount=49.95
ccAuthReply_accountBalance=50.05
ccAuthReply_authorizationCode=123456
ccAuthReply_avsCode=Y
ccAuthReply_avsCodeRaw=YYY
ccAuthReply_processorResponse=A
ccAuthReply_paymentNetworkTransactionID=3312345
ccCaptureReply_amount=49.95
ccCaptureReply_reasonCode=100
ccCaptureReply_reconciliationID=1094820975023470
```

Void

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 Processing a Void

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: Processing a Void

Request

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

Response to a Successful Request

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

```
requestID=0305782650000167905080
decision=ACCEPT
reasonCode=100
merchantReferenceCode=482046C3A7E94F5
purchaseTotals_currency=USD
ccAuthReply_reconciliationID=ABCDE12345FGHIJ67890
ccAuthReply_cardCategory=F^
ccAuthReply_cardGroup=0
ccAuthReply_reasonCode=100
ccAuthReply_amount=49.95
ccAuthReply_accountBalance=50.05
ccAuthReply_authorizationCode=123456
ccAuthReply_avsCode=Y
ccAuthReply_avsCodeRaw=YYY
ccAuthReply_processorResponse=A
```

```
ccAuthReply_paymentNetworkTransactionID=3312345
```

Timeout Voids for a Capture, Sale, Follow-On Credit, or Stand-Alone Credit

When you do not receive a response message after sending a capture, sale, or credit request, this feature enables you to void the transaction that you requested.

Include the **merchantTransactionIdentifier** field in the original request for a capture, sale, follow-on credit, or stand-alone credit. The value of the merchant transaction ID must be unique for 180 days.

When the original transaction fails, the response message for the reversal request includes these fields:

- **originalTransaction_amount**
- **originalTransaction_reasonCode**

Endpoint

Required Fields for Processing a Timeout Void for a Capture, Sale, Follow-On Credit, or Stand-Alone Credit

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

Related Information

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

Simple Order Example: Processing a Timeout Void for a Capture, Sale, Follow-on Credit, or Stand-Alone Credit

Request

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

```
voidService_voidRequestID=6522033834410167772169
```

Response to a Successful Request

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

```
requestID=0305782650000167905080
decision=ACCEPT
reasonCode=100
merchantReferenceCode=482046C3A7E94F5
purchaseTotals_currency=USD
ccAuthReply_reconciliationID=ABCDE12345FGHIJ67890
ccAuthReply_cardCategory=F^
ccAuthReply_cardGroup=0
ccAuthReply_reasonCode=100
ccAuthReply_amount=49.95
ccAuthReply_accountBalance=50.05
ccAuthReply_authorizationCode=123456
ccAuthReply_avsCode=Y
ccAuthReply_avsCodeRaw=YYY
ccAuthReply_processorResponse=A
ccAuthReply_paymentNetworkTransactionID=3312345
```

Debit and Prepaid Card Processing

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

Requirements

In Canada, to process domestic debit transactions on Visa Platform Connect with Mastercard, you must contact customer support to have your account configured for this feature.

When using the Simple Order API in XML format, partial authorizations or balance responses require the XML schema version 1.52 or later.

Related Information

- See *Debit and Prepaid Card Payments* on page 24 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?q=rest-sample&type=all&language=&sort=*

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 using relaxed requirements for address data and the expiration date, 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 124.

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 using relaxed requirements for address data and the expiration date, 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 124.

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=6618807769750178232890
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 using relaxed requirements for address data and the expiration date, 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 124.

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
```

Airline Data Processing

This section describes how to process airline payments.

Requirement

When you are ready to go live with airline data processing, contact Cybersource Customer Support to have your account configured to process airline data. If your account is not enabled, and you try to send airline transactions, you will receive an error for invalid data.

Related Information

- See *Airline Data* on page 25 for information about and requirements for processing payments that include airline data.
- See *Airline Data Reference Information* on page 28 for a list and description of the different document type and ancillary service category codes that are used when processing payments that include airline data.

Airline Travel Legs

Some processors require travel legs in the API service request. This section describes how to successfully include travel legs in an API request.

Using Travel Legs

To include travel legs in an airline transaction, include one or more **airlineData_leg_#_** fields, starting with the **airlineData_leg_0_** fields.
Direct flights only use the **airlineData_leg_0_** fields.
For multiple leg trips, the **airlineData_leg_#_** fields must start at 0, and you must use consecutive numbers for the additional legs.
For example, these three travel legs are valid:

```
<leg id="0">
  <carrierCode>XX</carrierCode>
</leg>
```

```
<leg id="1">
  <carrierCode>XZ</carrierCode>
</leg>
<leg id="2">
  <carrierCode>XX</carrier<code>
</leg>
```

> **Important**
>
> If you skip a number, Cybersource ignores the legs that follow the skipped number.

Travel Leg Limitations

Some processors limit the number of travel legs for each trip based on the card type.

Authorizations

This section shows the fields necessary to make an airline authorization.

Authorization Restrictions

Ticket purchases that include multiple passengers may be included in a single authorization request, however, Cybersource recommends sending a separate capture requests for each passenger.

If any ancillary purchases are made at the same time as the ticket purchase, you may include all the items in a single authorization request, however, Cybersource recommends sending separating the ancillary and ticket purchases into individual capture requests.

If any ancillary purchases are not made at the same time as the ticket purchase, Cybersource recommends separating the authorization and capture requests for the ancillary and ticket purchases.

Endpoint

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

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

Required Fields for an Airline Authorization

Include these fields to process an airline authorization.

> **Important**
>
> When using relaxed requirements for address data and the expiration date, 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 124.

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

Related Information:

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

Card-Specific Fields for an Airline Authorization

This section includes card-specific information required for an airline authorization.

American Express

Include these fields in addition to the required fields when using an American Express card. American Express requires leg fields in the authorization request. For more information about leg fields, see *Airline Travel Legs* on page 87.

airlineData_leg_#_carrierCode	**Replace the `#` character with the number `0`, and consecutive numbers for every additional leg.**
airlineData_leg_#_departureDate	**Replace the `#` character with the number `0`, and consecutive numbers for every additional leg.**

airlineData_leg_#_destination	**Replace the `#` character with the number `0`, and consecutive numbers for every additional leg.**
airlineData_leg_#_fareBasis	**Replace the `#` character with the number `0`, and consecutive numbers for every additional leg.**
airlineData_leg_#_originatingAirportCode	**Replace the `#` character with the number `0`, and consecutive numbers for every additional leg.**
airlineData_passengerName	
shipTo_firstname	

Related Information:

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

Simple Order API Example: American Express Airline Authorization

This example shows a successful airline authorization using an American Express card.

Request

```
<merchantID>fdiglobal_cert_no_cbps</merchantID>
<merchantReferenceCode>TLPANI6-28-1</merchantReferenceCode>
<billTo>
  <firstName>John</firstName>
  <lastName>Smith</lastName>
  <street1>201 S. Division St.</street1>
  <city>Ann Arbor</city>
  <state>MI</state>
  <postalCode>48104-2201</postalCode>
  <country>US</country>
  <email>null@cybersource.com</email>
</billTo>
<shipTo>
<firstName>CPS</firstName>
</shipTo>
<purchaseTotals>
  <currency>USD</currency>
  <grandTotalAmount>50.00</grandTotalAmount>
</purchaseTotals>
<card>
  <accountNumber>3739531923510-04</accountNumber>
  <expirationMonth>12</expirationMonth>
  <expirationYear>2025</expirationYear>
  <cardType>003</cardType>
</card>
<ccAuthService run="true">
  <industryDatatype>airline</industryDatatype>
</ccAuthService>
```

```
<airlineData>
 <passengerName>John</passengerName>
 <leg id="0">
  <carrierCode>AA</carrierCode>
  <originatingAirportCode>PDX</originatingAirportCode>
  <departureDate>20231010</departureDate>
  <destination>SJ</destination>
  <fareBasis>AB1234</fareBasis>
 </leg>
</airlineData>
```

Response to a Successful Request

```
<merchantReferenceCode>gZZE9w03mnzQef4ePI1CnJeu7e+d0lTRmBZOSpG6gheF3JyRat</
merchantReferenceCode>
<requestID>6859960691616692203007</requestID>
<decision>ACCEPT</decision>
<reasonCode>100</reasonCode>
<requestToken>Axj/7wSTc+EUL2Vb7m3/
ABEg3YMmjFkxaNpMmxQoN3CiOYF36MwNRHMC79GekDZxBipDJpJl6MWWh/8G8m58IoXsq33Nv+AAGDzm</
requestToken>
<purchaseTotals>
  <currency>USD</currency>
</purchaseTotals>
<ccAuthReply>
  <reasonCode>100</reasonCode>
  <amount>50.00</amount>
  <authorizationCode>888888</authorizationCode>
  <avsCode>X</avsCode>
  <avsCodeRaw>I1</avsCodeRaw>
  <authorizedDateTime>2023-06-05T20:14:29Z</authorizedDateTime>
  <processorResponse>100</processorResponse>
  <reconciliationID>702412146IIXPP78</reconciliationID>
  <accountBalance>70.00</accountBalance>
</ccAuthReply>
<card>
  <cardType>003</cardType>
</card>
<pos>
  <terminalID>111111</terminalID>
</pos>
```

Simple Order Example: Discover, Mastercard, and Visa Airline Authorization

This example shows a successful airline authorization using a Discover, Mastercard, and Visa card.

Request

```
<merchantID>fdiglobal_cert_no_cbps</merchantID>
<merchantReferenceCode>TLPANI6-28-1</merchantReferenceCode>
<billTo>
 <firstName>John</firstName>
 <lastName>Smith</lastName>
```

```
  <street1>201 S. Division St.</street1>
  <city>Ann Arbor</city>
  <state>MI</state>
  <postalCode>48104-2201</postalCode>
  <country>US</country>
  <email>null@cybersource.com</email>
</billTo>
<purchaseTotals>
  <currency>USD</currency>
  <grandTotalAmount>50.00</grandTotalAmount>
</purchaseTotals>
<card>
  <accountNumber>40120000333300-26</accountNumber>
  <expirationMonth>12</expirationMonth>
  <expirationYear>2025</expirationYear>
  <cardType>001</cardType>
</card>
<ccAuthService run="true">
  <industryDatatype>airline</industryDatatype>
</ccAuthService>
```

Response to a Successful Request

```
<merchantReferenceCode>gZZE9w03mnzQef4ePI1CnJeu7e+d0lTRmBZOSpG6gheF3JyRat</
merchantReferenceCode>
<requestID>6859958674966249203009</requestID>
<decision>ACCEPT</decision>
<reasonCode>100</reasonCode>
<requestToken>Axj/7wSTc+ENBUNWo
+VBABEg3YMmjBs5bw5MmxOl2GiiOYF3w4gFRHMC74cWkDZxBipDJpJl6MWWh/8Dcm58IaCoatR8qCAAxwAL</
requestToken>
<purchaseTotals>
   <currency>USD</currency>
</purchaseTotals>
<ccAuthReply>
   <reasonCode>100</reasonCode>
   <amount>50.00</amount>
   <authorizationCode>888888</authorizationCode>
   <avsCode>X</avsCode>
   <avsCodeRaw>I1</avsCodeRaw>
   <authorizedDateTime>2023-06-05T20:11:07Z</authorizedDateTime>
   <processorResponse>100</processorResponse>
   <reconciliationID>70240697CIIXNKX4</reconciliationID>
   <paymentNetworkTransactionID>123456789619999</paymentNetworkTransactionID>
</ccAuthReply>
<card>
   <cardType>001</cardType>
</card>
<pos>
   <terminalID>111111</terminalID>
</pos>
```

Captures for Ticket Payments

This section shows the fields necessary to capture an airline ticket payment.

Capture Restrictions

For a ticket purchase with multiple passengers, Cybersource recommends sending separate capture requests for each passenger. The original authorization may be a single request.

When ancillary purchases are made at the same time as the ticket purchase, Cybersource recommends sending separate capture requests for the ticket and ancillary purchases.

Airline Travel Legs

FDC Nashville Global requires you to use travel leg fields for ticket purchases, even for direct flights. For more information about how to use travel leg fields, see *Airline Travel Legs* on page 87.

Leg Limitations

FDC Nashville Global limits the maximum number of legs for each trip based on card type. This table shows the maximum number of legs for each trip based on processor and card type.

FDC Nashville Global Leg Limitations

Supported Card Types	Maximum Number of Trip Legs
American Express	4
Discover	4
Mastercard	99
Visa	4

Endpoint

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

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

Required Fields for Capturing a Ticket Payment

Include these required fields to capture an airline payment for tickets.

airlineData_leg_#_carrierCode	**Replace the `#` character with the number `0` for the first leg and consecutive numbers for any additional legs.**

airlineData_leg_#_class	**Replace the `#` character with the number `0` for the first leg and consecutive numbers for any additional legs.**
airlineData_leg_#_departureDate	**Replace the `#` character with the number `0` for the first leg and consecutive numbers for any additional legs.**
airlineData_leg_#_destination	**Replace the `#` character with the number `0` for the first leg and consecutive numbers for any additional legs.**
airlineData_leg_#_originatingAirportCode	**Replace the `#` character with the number `0` for the first leg and consecutive numbers for any additional legs.**
airlineData_passengerName	
airlineData_ticketNumber	
ccCaptureService_authRequestID	
ccCaptureService_run	**Set the value to `true`.**
ccCapture_industryDataType	**Set the value to `airline`.**
merchantID	
merchantReferenceCode	**Set to the `merchant_ref_number` value used in corresponding authorization request.**
purchaseTotals_currency	
purchaseTotals_grandTotalAmount	

Related Information:

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

Card-Specific Fields for Capturing a Ticket Payment

This section includes card-specific information required to capture a payment for tickets.

Discover

Include these fields in addition to the required fields when using a Discover card.

airlineData_agentCode	
airlineData_arrivalDate	
airlineData_leg_#_arrivalTime	**Replace the `#` character with the number `0` for the first leg and consecutive numbers for any additional legs.**

airlineData_leg_#_arrivalTimeSegment	Replace the `#` character with the number `0` for the first leg and consecutive numbers for any additional legs. Set the value to `A` or `P`.
airlineData_leg_#_conjunctionTicket	Replace the `#` character with the number `0` for the first leg and consecutive numbers for any additional legs.
airlineData_leg_#_departureTime	Replace the `#` character with the number `0` for the first leg and consecutive numbers for any additional legs.
airlineData_leg_#_departureTimeSegment	Replace the `#` character with the number `0` for the first leg and consecutive numbers for any additional legs. Set the value to `A` or `P`.
airlineData_leg_#_fareBasis	Replace the `#` character with the number `0` for the first leg and consecutive numbers for any additional legs.
airlineData_leg_#_fare	Replace the `#` character with the number `0` for the first leg and consecutive numbers for any additional legs.
airlineData_leg_#_flightNumber	Replace the `#` character with the number `0` for the first leg and consecutive numbers for any additional legs.
airlineData_ticketIssueDate	
airlineData_ticketIssuerAddress	

Mastercard

Include these fields in addition to the required fields when using a Mastercard card.

airlineData_leg_#_departureTime	Replace the `#` character with the number `0` for the first leg and consecutive numbers for any additional legs.
airlineData_leg_#_fareBasis	Replace the `#` character with the number `0` for the first leg and consecutive numbers for any additional legs.
airlineData_leg_#_flightNumber	Replace the `#` character with the number `0` for the first leg and consecutive numbers for any additional legs.

Visa

Include this field in addition to the required fields when using a Visa card.

airlineData_restrictedTicketIndicator	**Set the value to 0 for refundable.** **Set the value to 1 for nonrefundable.**

Related Information:

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

Optional Fields for an Airline Capture

Choose from these optional fields to include additional information when capturing an airline payment for direct flight tickets.

airlineData_agentName	
airlineData_clearingCount	
airlineData_clearingSequence	
airlineData_customerCode	
airlineData_exchangeTicketAmount	
airlineData_exchangeTicketFee	
airlineData_leg_#_couponNumber	**Replace the # character with the number 0 for the first leg and consecutive numbers for any additional legs.**
airlineData_leg_#_endorsementsRestrictions	**Replace the # character with the number 0 for the first leg and consecutive numbers for any additional legs.**
airlineData_leg_#_exchangeTicket	**Replace the # character with the number 0 for the first leg and consecutive numbers for any additional legs.**
airlineData_leg_#_fee	**Replace the # character with the number 0 for the first leg and consecutive numbers for any additional legs.**
airlineData_leg_#_stopoverCode	**Set the value to O (capital letter "O") to indicate the leg of a trip is allowed to stopover. Default value.** **Set the value to Xto indicate the leg of a trip is not allowed to stopover.** **Replace the # character with the number 0 for the first leg and consecutive numbers for any additional legs.**
airlineData_leg_#_tax	**Replace the # character with the number 0 for the first leg and consecutive numbers for any additional legs.**

airlineData_planNumber

airlineData_totalClearingAmount

airlineData_totalFee

otherTax_localTaxAmount

purchaseTotals_taxAmount

Related Information:

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

Simple Order Example: Capturing an American Express Airline Payment

This example shows a successful capture using an American Express card.

Request

```
<merchantID>fdiglobal_cert_no_cbps</merchantID>
<merchantReferenceCode>TLPANI6-28-1</merchantReferenceCode>
<purchaseTotals>
 <currency>USD</currency>
 <taxAmount>3</taxAmount>
 <grandTotalAmount>50.00</grandTotalAmount>
</purchaseTotals>
<otherTax>
 <localTaxAmount>20</localTaxAmount>
</otherTax>
<ccCaptureService run="true">
 <authRequestID>6841863354646003603681</authRequestID>
 <industryDatatype>airline</industryDatatype>
</ccCaptureService>
<airlineData>
 <agentName>Thomascook</agentName>
 <ticketNumber>asd23</ticketNumber>
 <passengerName>John</passengerName>
 <customerCode>LDTYOU789</customerCode>
 <totalFee>200</totalFee>
 <leg id="0">
  <carrierCode>AA</carrierCode>
  <flightNumber>UA780</flightNumber>
  <originatingAirportCode>PDX</originatingAirportCode>
  <class>1</class>
  <stopoverCode>O</stopoverCode>
  <departureDate>20231010</departureDate>
  <destination>SJ</destination>
  <fareBasis>AB1234</fareBasis>
  <exchangeTicket>tickeT12</exchangeTicket>
  <departureTime>1720</departureTime>
  <endorsementsRestrictions>Endorsement123456789</endorsementsRestrictions>
  <fee>999.99</fee>
  <tax>99</tax>
 </leg>
 <leg id="1">
  <carrierCode>AA</carrierCode>
  <flightNumber>UA780</flightNumber>
  <originatingAirportCode>PDX</originatingAirportCode>
  <class>1</class>
  <stopoverCode>O</stopoverCode>
  <departureDate>20231010</departureDate>
  <destination>SJ</destination>
  <fareBasis>AB1234</fareBasis>
  <exchangeTicket>tickeT12</exchangeTicket>
  <departureTime>1720</departureTime>
  <endorsementsRestrictions>Endorsement123456789</endorsementsRestrictions>
  <fee>999.99</fee>
  <tax>99</tax>
 </leg>
 <leg id="2">
  <carrierCode>AA</carrierCode>
  <flightNumber>UA780</flightNumber>
  <originatingAirportCode>PDX</originatingAirportCode>
  <class>1</class>
  <stopoverCode>O</stopoverCode>
  <departureDate>20231010</departureDate>
  <destination>SJ</destination>
  <fareBasis>AB1234</fareBasis>
  <exchangeTicket>tickeT12</exchangeTicket>
  <departureTime>1720</departureTime>
  <endorsementsRestrictions>Endorsement123456789</endorsementsRestrictions>
  <fee>999.99</fee>
  <tax>99</tax>
 </leg>
 <leg id="3">
  <carrierCode>AA</carrierCode>
  <flightNumber>UA780</flightNumber>
  <originatingAirportCode>PDX</originatingAirportCode>
  <class>1</class>
  <stopoverCode>O</stopoverCode>
  <departureDate>20231010</departureDate>
  <destination>SJ</destination>
  <fareBasis>AB1234</fareBasis>
  <exchangeTicket>tickeT12</exchangeTicket>
  <departureTime>1720</departureTime>
  <endorsementsRestrictions>Endorsement123456789</endorsementsRestrictions>
  <fee>999.99</fee>
  <tax>99</tax>
 </leg>
<planNumber>2B</planNumber>
<exchangeTicketAmount>87654</exchangeTicketAmount>
<exchangeTicketFee>32345</exchangeTicketFee>
</airlineData>
```

Response to a Successful Request

```
<merchantReferenceCode>gZZE9w03mnzQef4ePI1CnJeu7e+d0lTRmBZOSpG6gheF3JyRat</
merchantReferenceCode>
<requestID>6859997862716741203010</requestID>
```

```
<decision>ACCEPT</decision>
<reasonCode>100</reasonCode>
<requestToken>Axj//wSTc
+GYPlCCQzRCABEg3YMmThg0cMpMmxUlOJaiOYF36MwNRHMC9btmkDZxBipDJpJl6MWWh/4YN5Nz4RQvZVvubf8AEW8g</
requestToken>
<purchaseTotals>
  <currency>USD</currency>
</purchaseTotals>
<ccCaptureReply>
  <reasonCode>100</reasonCode>
  <requestDateTime>2023-06-05T21:16:26Z</requestDateTime>
  <amount>50.00</amount>
  <reconciliationID>702280482IIXTJ8K</reconciliationID>
</ccCaptureReply>
```

Simple Order Example: Capturing a Discover Airline Payment

This example shows a successful capture using a Discover card.

Request

```
<merchantID>fdiglobal_cert_no_cbps</merchantID>
<merchantReferenceCode>TLPANI6-28-1</merchantReferenceCode>
<purchaseTotals>
  <currency>USD</currency>
  <taxAmount>3</taxAmount>
  <grandTotalAmount>50.00</grandTotalAmount>
</purchaseTotals>
<otherTax>
  <localTaxAmount>20</localTaxAmount>
</otherTax>
<ccCaptureService run="true">
  <authRequestID>6917686812546643503007</authRequestID>
  <industryDatatype>airline</industryDatatype>
</ccCaptureService>
<airlineData>
  <agentCode>AGC12345</agentCode>
  <agentName>Thomascook</agentName>
  <ticketIssuerAddress>123 Happy St</ticketIssuerAddress>
  <ticketNumber>asd23</ticketNumber>
  <restrictedTicketIndicator>1</restrictedTicketIndicator>
  <passengerName>John</passengerName>
  <customerCode>LDTYOU789</customerCode>
  <totalFee>200</totalFee>
  <leg id="0">
    <carrierCode>AA</carrierCode>
    <flightNumber>123456789</flightNumber>
    <originatingAirportCode>PDX</originatingAirportCode>
    <class>1</class>
    <stopoverCode>O</stopoverCode>
    <departureDate>20231010</departureDate>
    <destination>SJ</destination>
    <fareBasis>AB1234</fareBasis>
    <conjunctionTicket>ConjunctTicketNum00012340</conjunctionTicket>
    <exchangeTicket>tickeT12</exchangeTicket>
    <departureTime>1720</departureTime>
    <departureTimeSegment>P</departureTimeSegment>
    <arrivalTime>2220</arrivalTime>
    <arrivalTimeSegment>P</arrivalTimeSegment>
    <endorsementsRestrictions>Endorsement123456789</endorsementsRestrictions>
    <fare>100000000</fare>
    <fee>999.99</fee>
    <tax>99</tax>
  </leg>
  <leg id="1">
    <carrierCode>AA</carrierCode>
    <flightNumber>456789123</flightNumber>
    <originatingAirportCode>PDX</originatingAirportCode>
    <class>1</class>
    <stopoverCode>O</stopoverCode>
    <departureDate>20231010</departureDate>
    <destination>SJ</destination>
    <fareBasis>AB1234</fareBasis>
    <conjunctionTicket>ConjunctTicketNum00012340</conjunctionTicket>
    <exchangeTicket>tickeT12</exchangeTicket>
    <departureTime>1720</departureTime>
    <departureTimeSegment>P</departureTimeSegment>
    <arrivalTime>2220</arrivalTime>
    <arrivalTimeSegment>P</arrivalTimeSegment>
    <endorsementsRestrictions>Endorsement123456789</endorsementsRestrictions>
    <fare>100000000</fare>
    <fee>999.99</fee>
    <tax>99</tax>
  </leg>
  <leg id="2">
    <carrierCode>AA</carrierCode>
    <flightNumber>987654321</flightNumber>
    <originatingAirportCode>PDX</originatingAirportCode>
    <class>1</class>
    <stopoverCode>O</stopoverCode>
    <departureDate>20231010</departureDate>
    <destination>SJ</destination>
    <fareBasis>AB1234</fareBasis>
    <conjunctionTicket>ConjunctTicketNum00012340</conjunctionTicket>
    <exchangeTicket>tickeT12</exchangeTicket>
    <departureTime>1720</departureTime>
    <departureTimeSegment>P</departureTimeSegment>
    <arrivalTime>2220</arrivalTime>
    <arrivalTimeSegment>P</arrivalTimeSegment>
    <endorsementsRestrictions>Endorsement123456789</endorsementsRestrictions>
    <fare>100000000</fare>
    <fee>999.99</fee>
    <tax>99</tax>
  </leg>
  <leg id="3">
    <carrierCode>AA</carrierCode>
    <originatingAirportCode>PDX</originatingAirportCode>
    <class>1</class>
    <stopoverCode>O</stopoverCode>
    <departureDate>20231010</departureDate>
    <destination>SJ</destination>
    <fareBasis>AB1234</fareBasis>
    <conjunctionTicket>ConjunctTicketNum00012340</conjunctionTicket>
    <exchangeTicket>tickeT12</exchangeTicket>
    <departureTime>1720</departureTime>
    <departureTimeSegment>P</departureTimeSegment>
    <arrivalTime>2220</arrivalTime>
    <arrivalTimeSegment>P</arrivalTimeSegment>
    <endorsementsRestrictions>Endorsement123456789</endorsementsRestrictions>
    <fare>100000000</fare>
    <fee>999.99</fee>
    <tax>99</tax>
  </leg>
  <ticketIssueDate>20231010</ticketIssueDate>
  <planNumber>2B</planNumber>
  <arrivalDate>09152023</arrivalDate>
  <exchangeTicketAmount>87654</exchangeTicketAmount>
  <exchangeTicketFee>32345</exchangeTicketFee>
</airlineData>
```

Response to a Successful Request

```
<merchantReferenceCode>Postman-1691769389</merchantReferenceCode>
<requestID>6917693897226880503009</requestID>
<decision>ACCEPT</decision>
<reasonCode>100</reasonCode>
<requestToken>Axj//
wSTdwJJvAPHy0zhABEg3cuWrVi3bxJjZlCbsWKiOZZgEcQRRHMswJWWkDZxBkvDJpJl6MWWh/4YGpN3AjCQiR7rp58AI3Qm</
requestToken>
<purchaseTotals>
  <currency>USD</currency>
</purchaseTotals>
<ccCaptureReply>
  <reasonCode>100</reasonCode>
  <requestDateTime>2023-08-11T15:56:29Z</requestDateTime>
  <amount>50.00</amount>
  <reconciliationID>79955177DL62B711</reconciliationID>
</ccCaptureReply>
```

Simple Order Example: Capturing a Mastercard Airline Payment

This example shows a successful capture using a Mastercard.

Request

```
<purchaseTotals>
  <currency>USD</currency>
  <taxAmount>3</taxAmount>
  <grandTotalAmount>50.00</grandTotalAmount>
</purchaseTotals>
<otherTax>
  <localTaxAmount>20</localTaxAmount>
</otherTax>
<ccCaptureService run="true">
  <authRequestID>6917735883576902203011</authRequestID>
  <industryDatatype>airline</industryDatatype>
</ccCaptureService>
<airlineData>
```

```
  <agentName>Thomascook</agentName>
  <ticketNumber>asd23</ticketNumber>
  <passengerName>John</passengerName>
  <customerCode>LDTYOU789</customerCode>
  <documentType>01</documentType>
  <totalFee>200</totalFee>
  <leg id="0">
    <carrierCode>AA</carrierCode>
    <flightNumber>123456789</flightNumber>
    <originatingAirportCode>PDX</originatingAirportCode>
    <class>1</class>
    <stopoverCode>O</stopoverCode>
    <departureDate>20231010</departureDate>
    <destination>SJ</destination>
    <fareBasis>1</fareBasis>
    <exchangeTicket>tickeT12</exchangeTicket>
    <departureTime>1235</departureTime>
    <endorsementsRestrictions>Endorsement123456789</endorsementsRestrictions>
    <fee>999.99</fee>
    <tax>99</tax>
  </leg>
  <leg id="1">
    <carrierCode>AA</carrierCode>
    <flightNumber>987654321</flightNumber>
    <originatingAirportCode>PDX</originatingAirportCode>
    <class>1</class>
    <stopoverCode>O</stopoverCode>
    <departureDate>20231010</departureDate>
    <destination>SJ</destination>
    <fareBasis>1</fareBasis>
    <exchangeTicket>tickeT12</exchangeTicket>
    <departureTime>1640</departureTime>
    <endorsementsRestrictions>Endorsement123456789</endorsementsRestrictions>
    <fee>999.99</fee>
    <tax>99</tax>
  </leg>
  <leg id="2">
    <carrierCode>AA</carrierCode>
    <flightNumber>78945612</flightNumber>
    <originatingAirportCode>PDX</originatingAirportCode>
    <class>1</class>
    <stopoverCode>O</stopoverCode>
    <departureDate>20231010</departureDate>
    <destination>SJ</destination>
    <fareBasis>1</fareBasis>
    <exchangeTicket>tickeT12</exchangeTicket>
    <departureTime>2245</departureTime>
    <endorsementsRestrictions>Endorsement123456789</endorsementsRestrictions>
    <fee>999.99</fee>
    <tax>99</tax>
  </leg>
  <leg id="3">
    <carrierCode>AA</carrierCode>
    <originatingAirportCode>PDX</originatingAirportCode>
    <class>1</class>
    <stopoverCode>O</stopoverCode>
    <departureDate>20231010</departureDate>
    <destination>SJ</destination>
    <fareBasis>1</fareBasis>
    <exchangeTicket>tickeT12</exchangeTicket>
    <departureTime>2345</departureTime>
    <endorsementsRestrictions>Endorsement123456789</endorsementsRestrictions>
    <fee>999.99</fee>
    <tax>99</tax>
  </leg>
  <planNumber>2B</planNumber>
  <exchangeTicketAmount>87654</exchangeTicketAmount>
  <exchangeTicketFee>32345</exchangeTicketFee>
</airlineData>
```

Response to a Successful Request

```
<merchantReferenceCode>Postman-1691773605</merchantReferenceCode>
<requestID>6917736057356909103011</requestID>
<decision>ACCEPT</decision>
<reasonCode>100</reasonCode>
<requestToken>Axj//
wSTdwLfhG391UejABEg3cuWrlqwaxJjZlCb2JiiOZZjYawJRHMsxs7ukDZxBkvDJpJl6MWWh/4YTJN3At7mYK5AtoMAkEAT</
requestToken>
<purchaseTotals>
  <currency>USD</currency>
</purchaseTotals>
<ccCaptureReply>
  <reasonCode>100</reasonCode>
  <requestDateTime>2023-08-11T17:06:45Z</requestDateTime>
  <amount>50.00</amount>
  <reconciliationID>79959505DL62B7XL</reconciliationID>
</ccCaptureReply>
```

Simple Order Example: Capturing a Visa Airline Payment

This example shows a successful capture using a Visa card.

Request

```
<merchantID>fdiglobal_cert_no_cbps</merchantID>
<merchantReferenceCode>TLPANI6-28-1</merchantReferenceCode>
<purchaseTotals>
 <currency>USD</currency>
 <taxAmount>3</taxAmount>
 <grandTotalAmount>50.00</grandTotalAmount>
</purchaseTotals>
<otherTax>
 <localTaxAmount>20</localTaxAmount>
</otherTax>
<ccCaptureService run="true">
 <authRequestID>6841863354646003603681</authRequestID>
 <industryDatatype>airline</industryDatatype>
</ccCaptureService>
<airlineData>
 <agentName>Thomascook</agentName>
 <ticketNumber>asd23</ticketNumber>
 <restrictedTicketIndicator>1</restrictedTicketIndicator>
 <passengerName>John</passengerName>
 <customerCode>LDTYOU789</customerCode>
 <totalFee>200</totalFee>
 <leg id="0">
  <carrierCode>AA</carrierCode>
  <originatingAirportCode>PDX</originatingAirportCode>
  <class>1</class>
  <stopoverCode>O</stopoverCode>
  <departureDate>20231010</departureDate>
  <destination>SJ</destination>
  <exchangeTicket>tickeT12</exchangeTicket>
  <endorsementsRestrictions>Endorsement123456789</endorsementsRestrictions>
  <fee>999.99</fee>
  <tax>99</tax>
 </leg>
 <leg id="1">
  <carrierCode>AA</carrierCode>
  <originatingAirportCode>PDX</originatingAirportCode>
  <class>1</class>
  <stopoverCode>O</stopoverCode>
  <departureDate>20231010</departureDate>
  <destination>SJ</destination>
  <exchangeTicket>tickeT12</exchangeTicket>
  <endorsementsRestrictions>Endorsement123456789</endorsementsRestrictions>
  <fee>999.99</fee>
  <tax>99</tax>
 </leg>
 <leg id="2">
  <carrierCode>AA</carrierCode>
  <originatingAirportCode>PDX</originatingAirportCode>
  <class>1</class>
  <stopoverCode>O</stopoverCode>
  <departureDate>20231010</departureDate>
  <destination>SJ</destination>
  <exchangeTicket>tickeT12</exchangeTicket>
  <endorsementsRestrictions>Endorsement123456789</endorsementsRestrictions>
  <fee>999.99</fee>
  <tax>99</tax>
 </leg>
 <leg id="3">
  <carrierCode>AA</carrierCode>
  <originatingAirportCode>PDX</originatingAirportCode>
  <class>1</class>
  <stopoverCode>O</stopoverCode>
  <departureDate>20231010</departureDate>
  <destination>SJ</destination>
  <exchangeTicket>tickeT12</exchangeTicket>
  <endorsementsRestrictions>Endorsement123456789</endorsementsRestrictions>
  <fee>999.99</fee>
  <tax>99</tax>
 </leg>
 <planNumber>2B</planNumber>
<exchangeTicketAmount>87654</exchangeTicketAmount>
<exchangeTicketFee>32345</exchangeTicketFee>
</airlineData>
```

Response to a Successful Request

```
<merchantReferenceCode>gZZE9w03mnzQef4ePI1CnJeu7e+d01TRmBZOSpG6gheF3JyRat</
merchantReferenceCode>
<requestID>6859975187656465603010</requestID>
<decision>ACCEPT</decision>
<reasonCode>100</reasonCode>
<requestToken>Axj//wSTc
+FHr3ehRlnCABEg3YMmjFkxaNpMmxQoN3CiOYF36MwNRHMC8fqekDZxBipDJpJl6MWWh/4YdZNz4RQvZVvubf8AHG8a</
requestToken>
<purchaseTotals>
  <currency>USD</currency>
</purchaseTotals>
<ccCaptureReply>
  <reasonCode>100</reasonCode>
  <requestDateTime>2023-06-05T20:38:39Z</requestDateTime>
  <amount>50.00</amount>
  <reconciliationID>702412146IIXPP78</reconciliationID>
</ccCaptureReply>
```

Captures for Ancillary Purchases

This section shows the fields necessary to capture an airline payment for ancillary purchases.

Ancillary purchases are any additional services, such as baggage, meals, and paid seats, your customers can purchase. Captures for ancillary purchases must be separate for captures for ticket purchases. FDC Nashville Global supports up to four ancillary purchases for each capture.

Capture Restrictions

When ancillary purchases are made at the same time as the ticket purchase, Cybersource recommends sending separate capture requests for the ticket and ancillary purchases. When ancillary purchases are not made at the same time as the ticket purchase, Cybersource recommends sending separate authorization and capture requests for the ancillary and ticket purchases.

Endpoint

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

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

Required Fields for an Airline Capture

Include these fields to request an airline capture for an ancillary purchase. For ancillary-specific fields, see *Optional Fields for an Ancillary Airline Capture* on page 106.

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

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

Related Information:

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

Optional Fields for an Ancillary Airline Capture

Choose from these optional fields to include additional information when capturing an ancillary airline payment.

> **Important**
>
> FDC Nashville Global supports up to four ancillary purchases for each capture.

airlineData_ticketUpdateIndicator	**Set the value to `C` for changing an existing ticket.** **Set the value to `N` for a new ticket.**
ancillaryData_connectedTicketNumber	
ancillaryData_passengerName	
ancillaryData_service_#_categoryCode	**Replace the `#` character with a sequential number from `0` to `3`.** **For a list of possible values, see** *Ancillary Service Category Codes* **on page 31.**
ancillaryData_service_#_subcategoryCode	**Replace the `#` character with a sequential number from `0` to `3`.** **For a list of possible values, see** *Ancillary Service Category Codes* **on page 31.**
ancillaryData_ticketNumber	

Related Information:

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

Simple Order Example: Capturing an Airline Payment for Ancillary Purchases

Request

```
<merchantID>fdiglobal_cert_no_cbps</merchantID>
<merchantReferenceCode>TLPANI6-28-1</merchantReferenceCode>
<billTo>
  <firstName>John</firstName>
  <lastName>Smith</lastName>
  <street1>201 S. Division St.</street1>
  <city>Ann Arbor</city>
  <state>MI</state>
  <postalCode>48104-2201</postalCode>
  <country>US</country>
  <email>null@cybersource.com</email>
</billTo>
<purchaseTotals>
  <currency>USD</currency>
  <taxAmount>3</taxAmount>
  <grandTotalAmount>50.00</grandTotalAmount>
</purchaseTotals>
<card>
  <accountNumber>40120000333300-26</accountNumber>
  <expirationMonth>12</expirationMonth>
  <expirationYear>2025</expirationYear>
</card>
<otherTax>
  <localTaxAmount>20</localTaxAmount>
</otherTax>
<ccCaptureService run="true">
  <authRequestID>6841863354646003603681</authRequestID>
  <industryDatatype>airline</industryDatatype>
</ccCaptureService>
<ancillaryData>
  <ticketNumber>Ticket15CharMax</ticketNumber>
  <passengerName>AncillaryNameMax20ch</passengerName>
  <connectedTicketNumber>CONNTICKET12345</connectedTicketNumber>
  <service id="0">
  <categoryCode>S1</categoryCode>
  <subcategoryCode>SSC1</subcategoryCode>
  </service>
  <service id="1">
  <categoryCode>S2</categoryCode>
  <subcategoryCode>SSC2</subcategoryCode>
  </service>
  <service id="2">
  <categoryCode>S3</categoryCode>
  <subcategoryCode>SSC3</subcategoryCode>
  </service>
</ancillaryData>
```

Response to a Successful Request

```
<replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.142">
```

```
  <merchantReferenceCode>gZZE9w03mnzQef4ePI1CnJeu7e+d0lTRmBZOSpG6gheF3JyRat</
merchantReferenceCode>
  <requestID>6860000963746172003008</requestID>
  <decision>ACCEPT</decision>
  <reasonCode>100</reasonCode>
  <requestToken>Axj//wSTc
+GjQq8v7GLAABEg3YMmThg1YMpMmxUlQW6iOYF3w4gFRHMC9ivukDZxBipDJpJl6MWWh/4YG5Nz4Q0FQ1aj5UEA1Xej</
requestToken>
  <purchaseTotals>
 <currency>USD</currency>
  </purchaseTotals>
  <ccCaptureReply>
 <reasonCode>100</reasonCode>
 <requestDateTime>2023-06-05T21:21:36Z</requestDateTime>
 <amount>50.00</amount>
 <reconciliationID>702280502IIXTJA7</reconciliationID>
  </ccCaptureReply>
</replyMessage>
```

Follow-On Credits

This section describes how to process an airline follow-on credit, which requires you to have the transaction ID from the original payment.

Endpoint

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

Required Fields for an Airline Follow-On Credit

Include these fields to process an airline follow-on credit.

ccCreditService_captureRequestID	
ccCreditService_run	**Set the value to `true`.**
merchantID	
merchantReferenceCode	**Set to the `merchant_ref_number` value used in corresponding capture request.**
purchaseTotals_currency	
purchaseTotals_grandTotalAmount	

Related Information:

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

Optional Fields for an Airline Follow-On Credit

Choose from these optional fields to include additional information when processing an airline follow-on credit.

airlineData_clearingCount

airlineData_clearingSequence

airlineData_totalClearingAmount

Related Information:

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

Simple Order API Example: Processing an Airline Follow-On Credit

This example shows a successful airline follow-on credit.

Request

```
<merchantID>fdiglobal_cert_no_cbps</merchantID>
<merchantReferenceCode>TLPANI6-28-1</merchantReferenceCode>
<purchaseTotals>
  <currency>USD</currency>
  <taxAmount>3</taxAmount>
  <grandTotalAmount>50.00</grandTotalAmount>
</purchaseTotals>
<otherTax>
  <localTaxAmount>20</localTaxAmount>
</otherTax>
<ccCreditService run="true">
  <captureRequestID>6897061702136732503007</captureRequestID>
  <industryDatatype>airline</industryDatatype>
</ccCreditService>
<airlineData>
  <agentName>Thomascook</agentName>
  <ticketNumber>asd23</ticketNumber>
  <restrictedTicketIndicator>1</restrictedTicketIndicator>
  <passengerName>John</passengerName>
  <customerCode>LDTYOU789</customerCode>
  <totalFee>100</totalFee>
  <leg id="0">
    <carrierCode>AA</carrierCode>
    <originatingAirportCode>PDX</originatingAirportCode>
    <class>1</class>
    <stopoverCode>O</stopoverCode>
    <departureDate>20231010</departureDate>
    <destination>SJ</destination>
    <exchangeTicket>ticketT12</exchangeTicket>
    <endorsementsRestrictions>Endorsement123456789</endorsementsRestrictions>
    <fee>20.00</fee>
    <tax>5.00</tax>
  </leg>
  <leg id="1">
```

```
    <carrierCode>AA</carrierCode>
    <originatingAirportCode>PDX</originatingAirportCode>
    <class>1</class>
    <stopoverCode>O</stopoverCode>
    <departureDate>20231010</departureDate>
    <destination>SJ</destination>
    <exchangeTicket>tickeT12</exchangeTicket>
    <endorsementsRestrictions>Endorsement123456789</endorsementsRestrictions>
    <fee>20.00</fee>
    <tax>5.00</tax>
  </leg>
  <leg id="2">
    <carrierCode>AA</carrierCode>
    <originatingAirportCode>PDX</originatingAirportCode>
    <class>1</class>
    <stopoverCode>O</stopoverCode>
    <departureDate>20231010</departureDate>
    <destination>SJ</destination>
    <exchangeTicket>tickeT12</exchangeTicket>
    <endorsementsRestrictions>Endorsement123456789</endorsementsRestrictions>
    <fee>20.00</fee>
    <tax>5.00</tax>
  </leg>
  <leg id="3">
    <carrierCode>AA</carrierCode>
    <originatingAirportCode>PDX</originatingAirportCode>
    <class>1</class>
    <stopoverCode>O</stopoverCode>
    <departureDate>20231010</departureDate>
    <destination>SJ</destination>
    <exchangeTicket>tickeT12</exchangeTicket>
    <endorsementsRestrictions>Endorsement123456789</endorsementsRestrictions>
    <fee>20.00</fee>
    <tax>5.00</tax>
  </leg>
  <planNumber>2B</planNumber>
  <exchangeTicketAmount>87654</exchangeTicketAmount>
  <exchangeTicketFee>32345</exchangeTicketFee>
</airlineData>
```

Response to a Successful Request

```
<merchantReferenceCode>Postman-1689706432</merchantReferenceCode>
<requestID>6897064319436878203011</requestID>
<decision>ACCEPT</decision>
<reasonCode>100</reasonCode>
<requestToken>Axj/7wSTdeP+wFCga0iDABEg3atWzZg0axpbaU3bN4aiOY4OFKwFRHMcHVE2kDZxBj/
DJpJl6MWWh/8Dcm68fRtGD+gYpAAAFQXQ</requestToken>
<purchaseTotals>
  <currency>USD</currency>
</purchaseTotals>
<ccCreditReply>
  <reasonCode>100</reasonCode>
  <requestDateTime>2023-07-18T18:53:52Z</requestDateTime>
  <amount>50.00</amount>
  <reconciliationID>75566045FK6J767C</reconciliationID>
```

```
</ccCreditReply>
```

Payer Authentication Processing

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

- American Express: SafeKey
- 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?q=rest-sample&type=all&language=&sort=*

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 America 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**

American Express SafeKey

American Express SafeKey is the authentication service in the American Express card network that uses the 3-D Secure protocol to validate customers at checkout. When you request an authorization using a supported card type and a supported processor, you can include payer authentication data in the request.

> **Important**
>
> FDC Nashville Global supports 3-D Secure 2.0 transactions using an American Express payment card.

> **Important**
>
> FDC Nashville Global supports 3-D Secure 2.0 with America Express transactions.

Before implementing payer authentication for American Express SafeKey, contact customer support to have your account configured for this feature.

Fields Specific to the American Express SafeKey Use Case

These API fields are required specifically for this use case.

ccAuthService_cavv	Required when payer authentication is successful.
ccAuthService_commerceIndicator	Set this field to one of these values: • `aesk`: Successful authentication (3#D Secure value of `05`). • `aesk_attempted`: Authentication was attempted (3#D Secure value of `06`). • `internet`: Authentication failed or was not attempted (3#D Secure value of `07`).

Endpoint

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

Related Information

- *Payer Authentication Developer Guide | Simple Order API*

Required Fields for Processing an Authorization Using American Express SafeKey

These fields must be included in a request for an authorization with American SafeKey. The values for these fields are in the response from the payer authentication validate service. When you request the payer authentication validate and authorization services together, the data is automatically passed from one service to the other.

> **Important**
>
> When using relaxed requirements for address data and the expiration date, 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 124.

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: • `aesk`: Successful authentication (3#D Secure value of `05`). • `aesk_attempted`: Authentication was attempted (3#D Secure value of `06`). • `internet`: Authentication failed or was not attempted (3#D Secure value of `07`).

ccAuthService_eciRaw	Required when the payer authentication validation service returns a raw ECI value.
merchantID	
merchantReferenceCode	
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*

Optional Field for Processing an Authorization Using American Express SafeKey

This field is optional in a request for an authorization with American Express SafeKey. The value for this field is in the response from the payer authentication validate service. When you request the payer authentication validate and authorization services together, the data is automatically passed from one service to the other.

ccAuthService_xid

Related Information

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

Simple Order Example: Processing an Authorization Using American Express SafeKey

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=aesk
merchant_id=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
requestToken=Ahj/7wSR5C/4Icd2fdAKakGLadfg5535r/ghx3Z90AoBj3u
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
```

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.

Important

FDC Compass supports 3-D Secure 2.0 with Mastercard transactions. FDC Compass also supports the use of digital payments in 3-D Secure 2.0 transactions when used with Mastercard transactions.

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 using relaxed requirements for address data and the expiration date, 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 124.

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
requestToken=Ahj/7wSR5C/4Icd2fdAKakGLadfg5535r/ghx3Z90AoBj3u
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.

> **Important**
>
> FDC Nashville Global supports 3-D Secure 2.0 with Visa transactions. FDC Nashville Global also supports the use of digital payments in 3-D Secure 2.0 transactions when used with Visa transactions.

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 using relaxed requirements for address data and the expiration date, 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 124.

billTo_city	
billTo_country	
billTo_email	
billTo_firstName	
billTo_lastName	
billTo_postalCode	
billTo_state	
billTo_street1	
card_accountNumber	
card_expirationMonth	
card_expirationYear	
ccAuthService_cavv	Required when payer authentication is successful. Otherwise, this field is optional.

ccAuthService_commerceIndicatorv	Set 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 to `true`.
ccAuthService_xid	
merchant_referenceCode	
purchaseTotals_currency	
purchaseTotals_grandTotalAmount	

Related Information

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

Related information

- API field reference guide for the Simple Order API

Simple Order Example: Processing an Authorization Using Visa Secure

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
requestToken=Ahj/7wSR5C/4Icd2fdAKakGLadfg5535r/ghx3Z90AoBj3u
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	
billTo_state	
billTo_street1	
card_expirationMonth	**You can submit an expiration date that has expired. This exception does not apply when you combine any of the services listed above with any other service.** **This field is required for payment network token transactions and subscription creation requests.**
card_expirationYear	**You can submit an expiration date that has expired. This exception does not apply when you combine any of the services listed above with any other service.** **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:

- *Credentialed Transactions Developer Guide*
- *Token Management Service Developer Guide*
- *API field reference guide for the Simple Order API*
- Github repositories: *https://github.com/Cybersource?q=rest-sample&type=all&language=&sort=*

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.

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 using relaxed requirements for address data and the expiration date, 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 124.

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`.**

Related Information

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

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
```

Using Stored Customer Credentials During a CIT

After customers store their credentials on file, they can recall 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 using relaxed requirements for address data and the expiration date, 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 124.

billTo_city

billTo_country

billTo_email

billTo_firstName

billTo_lastName

billTo_postalCode

billTo_state

billTo_street1

card_accountNumber

card_expirationMonth

card_expirationYear

ccAuthService_run

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

Related Information

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

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

Some card companies require additional information when making authorizations with stored credentials.

Discover

Discover requires the authorization amount from the original transaction when sending a request:

subsequentAuthOriginalAmount

Related Information

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

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
subsequentAuth=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
```

Merchant-Initiated Reauthorization Transactions with PAN

A reauthorization occurs when the completion or fulfillment of the original order or service extends beyond the authorized amount time limit. There are two common reauthorization scenarios:

- Split or delayed shipments by a retailer
- Extended car rentals, hotel stays, or cruise line bookings

Endpoint

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

Required Fields for Processing Merchant-Initiated Reauthorized Transactions

Use these required fields to process a merchant-initiated reauthorization transaction.

> **Important**
>
> When using relaxed requirements for address data and the expiration date, 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 124.

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	
subsequentAuthReason	**Set the value to `3`.**
subsequentAuth	**Set the value to `true`.**

Related Information

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

Card-Specific Required Field for Processing a Merchant-Initiated Transactions

Discover

The listed card requires an additional field:

subsequentAuthOriginalAmount	**Provide the original transaction amount.**

Related Information

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

Simple Order Example: Processing a Merchant-Initiated Reauthorized Transaction

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
ccAuthService_commerceIndicator=install
merchantId=pa_ctv_sg101
merchantReferenceCode=33557799
purchaseTotals_currency=USD
purchaseTotals_grandTotalAmount=100.00
subsequentAuth=true
subsequentAuthReason=3
```

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
```

Installment Payments

An installment payment is a single purchase of goods or services billed to a customer in multiple transactions over a period of time agreed to by you and the customer. The agreement enables you to charge a specific amount at specified intervals.

Installments Service for Installment Payments

Important

Do not use this document if you are using the Installments service. When using the Installments service, Cybersource saves and stores payment credentials for installment transactions, ensuring compliance with COF best practices.

Customer-Initiated Installment Payments with PAN

An installment payment is a single purchase of goods or services billed to a customer in multiple transactions over a period of time agreed to by you and the customer. The agreement enables you to charge a specific amount at specified intervals.

> **Important**
>
> Do not use this document if you are using the Installments service. When using the Installments service, Cybersource saves and stores payment credentials for installment transactions, ensuring compliance with COF best practices.

Prerequisites

The first transaction in an installment payment is a customer-initiated transaction (CIT). Before you can perform a subsequent merchant-initiated transaction (MIT), you must store the customer's credentials for later use. Before you can store the user's credentials, you must get the customer's consent to store their private information. This process 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`.

Successful Response

Store the **ccAuthReply_paymentNetworkTransactionID** field value from the successful response message. You must include the network transaction ID in subsequent MIT authorization requests to associate the CIT to the MIT.

Required Fields for Initial Customer-Initiated Installment Payments with a PAN

Include these required fields to authorize an initial customer-initiated installment payment using a PAN.

billTo_city

billTo_country

billTo_email

billTo_firstName

billTo_lastName

billTo_postalCode

billTo_state

billTo_street1

card_accountNumber

card_expirationMonth

card_expirationYear	
ccAuthService_commerceIndicator	**Set the value to `internet` or a payer authentication value.**
ccAuthService_run	**Set the value to `true`.**
merchantID	
merchantReferenceCode	
purchaseTotals_currency	
purchaseTotals_grandTotalAmount	
subsequentAuthFirst	**Set the value to `true`.**
subsequentAuth	**Set the value to `false`.**

Card-Specific Fields for Authorizing Initial Installment Payments

Use this required field if you are authorizing an initial installment payment using the card type referenced below.

Mastercard	**subsequentAuthReason** **Set the value to `9`.**

Related Information

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

Simple Order Example: Authorizing Initial Customer-Initiated Installment Payments with a PAN

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
ccAuthService_commerceIndicator=internet
merchantId=pa_ctv_sg101
merchantReferenceCode=33557799
purchaseTotals_currency=USD
purchaseTotals_grandTotalAmount=100.00
subsequentAuth=false
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
```

Merchant-Initiated Installment Payments with PAN

After the initial CIT installment payment, subsequent installment payments are merchant-initiated transactions (MITs).

Endpoint

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

Required Fields for Authorizing Merchant-Initiated Subsequent Installment Payments

Use these required fields to authorize merchant-initiated subsequent installment payments.

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

merchantID	
merchantReferenceCode	
purchaseTotals_currency	
purchaseTotals_grandTotalAmount	
subsequentAuthReason	**Set the value to** `9`**.** **Important** Only required for Mastercard transactions.
subsequentAuth	**Set the value to** `true`**.**
subsequentAuthStoredCredential	**Set the value to** `true`**.**

Related Information

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

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

Some card companies require additional information when making authorizations with stored credentials.

Discover

Discover requires the authorization amount from the original transaction when sending a request:

subsequentAuthOriginalAmount

Related Information

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

Simple Order Example: Authorizing Merchant-Initiated Subsequent Installment Payments

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_commerceIndicator=install
merchantId=pa_ctv_sg101
merchantReferenceCode=33557799
purchaseTotals_currency=USD
purchaseTotals_grandTotalAmount=100.00
subsequentAuth=true
subsequentAuthOriginalAmount=100.00  //This field is for Discover only.
subsequentAuthReason=9
subsequentAuthStoredCredential=true
subsequentAuthTransactionID=23976974322
```

Response to a Successful Response

```
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
```

Recurring Payments

A recurring payment is a credentials-on-file (COF) transaction in a series of payments that you bill to a customer for a fixed amount at regular intervals that do not exceed one year between transactions. The series of recurring payments is the result of an agreement between you and the customer for the purchase of goods or services that are provided at regular intervals.

Mastercard uses standing order and subscription payments instead of recurring payments. See *Mastercard Standing Order Payments* on page 143 and *Mastercard Subscription Payments* on page 147.

Recurring Billing Service for Recurring Payments

> **Important**
>
> Do not use this document for the Recurring Billing service.

> Use the *Recurring Billing Developer Guide*. When you use the Recurring Billing service, Cybersource saves and stores payment credentials for recurring transactions, ensuring compliance with COF best practices.

Customer-Initiated Recurring Payment with PAN

A recurring payment is a credentials-on-file (COF) transaction in a series of payments that you bill to a customer at a fixed amount, at regular intervals that do not exceed one year between transactions. The series of recurring payments is the result of an agreement between you and the customer for the purchase of goods or services that are provided at regular intervals.

Mastercard uses standing order and subscription payments instead of recurring payments. See *Mastercard Standing Order Payments* on page 143 and *Mastercard Subscription Payments* on page 147.

Prerequisites

The first transaction in a recurring payment is a customer-initiated transaction (CIT). Before you can perform a subsequent merchant-initiated transaction (MIT), you must store the customer's credentials for later use. Before you can store the customer's credentials, you must get their consent to store their private information. This is also known as establishing a relationship with the customer.

Recurring Billing Service for Recurring Payments

> **Important**
>
> Do not use this document for the Recurring Billing service.
>
> Use the *Recurring Billing Developer Guide*. When you use the Recurring Billing service, Cybersource saves and stores payment credentials for recurring transactions, ensuring compliance with COF best practices.

Endpoint

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

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

Successful Response

Store the **ccAuthReply_paymentNetworkTransactionID** field value from the successful response message. You must include the network transaction ID in subsequent MIT authorization requests to associate the CIT to the MIT.

Required Fields for Authorizing a Customer-Initiated Recurring Payment with PAN Using Simple Order API

Use these required fields to request an initial customer-initiated recurring payment.

billTo_city

billTo_country	
billTo_email	
billTo_firstName	
billTo_lastName	
billTo_postalCode	
billTo_state	
billTo_street1	
card_accountNumber	
card_expirationMonth	
card_expirationYear	
ccAuthService_commerceIndicator	Set the value to `internet`, a payer authentication value, or `MOTO`.
ccAuthService_firstRecurringPayment	Set the value to `true`.
ccAuthService_run	Set the value to `true`.
merchantID	
merchantReferenceCode	
purchaseTotals_currency	
purchaseTotals_grandTotalAmount	
subsequentAuthFirst	Set the value to `true`.
subsequentAuth	Set the value to `false`.

Related Information

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

Simple Order Example: Authorizing a Customer-Initiated Recurring Payment with a PAN

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
```

```
ccAuthService_commerceIndicator=internet
merchantId=pa_ctv_sg101
merchantReferenceCode=33557799
purchaseTotals_currency=USD
purchaseTotals_grandTotalAmount=100.00
subsequentAuth=false
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
```

Merchant-Initiated Recurring Payments with PAN

After the initial recurring payment (CIT), subsequent recurring payments are merchant-initiated transactions (MITs).

Endpoint

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

Required Fields for Authorizing a Merchant-Initiated Recurring Payment

Use these required fields to authorize subsequent recurring payments.

billTo_city

billTo_country

billTo_email

billTo_firstName

billTo_lastName

billTo_postalCode

billTo_state

billTo_street1

card_accountNumber

card_expirationMonth

card_expirationYear	
ccAuthService_commerceIndicator	**Set the value to** `recurring`**.**
ccAuthService_run	**Set the value to** `true`**.**
merchantID	
merchantReferenceCode	
purchaseTotals_currency	
purchaseTotals_grandTotalAmount	
subsequentAuthReason	**Set the value to** `7`**.**
subsequentAuth	**Set the value to** `true`**.**
subsequentAuthStoredCredential	**Set the value to** `true`**.**
subsequentAuthTransactionID	**Set this field to the network transaction identifier that was returned in the ccAuthReply_paymentNetworkTransactionID field in the response message for the original authorization.**

Related Information

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

Card-Specific Required Fields for Authorizing Subsequent Recurring Payments

Some card companies require additional information when making authorizations with stored credentials.

Discover

Include the authorization amount from the original transaction in this field:

subsequentAuthOriginalAmount

Mastercard

Mastercard supports subscription and standing order payments instead of recurring payments. See *Mastercard Subscription Payments* on page 147 and *Mastercard Standing Order Payments* on page 143.

Related Information

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

Simple Order Example: Authorizing a Merchant-Initiated Recurring Payment

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_commerceIndicator=recurring
merchantId=pa_ctv_sg101
merchantReferenceCode=33557799
purchaseTotals_currency=USD
purchaseTotals_grandTotalAmount=100.00
subsequentAuth=true
subsequentAuthOriginalAmount=100.00  //This field is for Discover only.
subsequentAuthReason=7
subsequentAuthStoredCredential=true
subsequentAuthTransactionID=23976974322
```

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
```

Mastercard Standing Order Payments

A standing order payment is a recurring COF transaction that is a variable amount at a regular interval, such as a utility bill, not to exceed one year between transactions. The series of recurring payments is the result of an agreement between you and the customer for the purchase of goods or services that are provided at regular intervals.

Mastercard Initial CIT Standing Order Payment

The first transaction in a standing order payment is a customer-initiated transaction (CIT). Before you can perform a subsequent merchant-initiated transaction (MIT), you must store the customer's credentials for later use. Before you can store the user's credentials, you must get the customer's consent to store their private information. This process 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 Authorizing Initial CIT Standing Order Payments

Use these required fields to authorize initial customer-initated standing order payments.

billTo_city	
billTo_country	
billTo_email	
billTo_firstName	
billTo_lastName	
billTo_postalCode	
billTo_state	
billTo_street1	
card_accountNumber	
card_expirationMonth	
card_expirationYear	
ccAuthService_commerceIndicator	**Set the value to** `internet`**.**
ccAuthService_run	**Set the value to** `true`**.**
merchantID	
merchantReferenceCode	
purchaseTotals_currency	
purchaseTotals_grandTotalAmount	
subsequentAuthFirst	**Set the value to** `true`**.**
subsequentAuthReason	Set the value to `8`.

Related Information

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

Simple Order Example: Authorizing Initial CIT Standing Order Payments

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
ccAuthService_commerceIndicator=internet
merchantId=pa_ctv_sg101
merchantReferenceCode=33557799
purchaseTotals_currency=USD
purchaseTotals_grandTotalAmount=100.00
subsequentAuthFirst=True
subsequentAuthReason=8
```

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
```

Mastercard Initial CIT Standing Order Payment with TMS

The first transaction in a standing order payment is a customer-initiated transaction (CIT). Before you can perform a subsequent merchant-initiated transaction (MIT), you must store the customer's credentials for later use. Before you can store the user's credentials, you must get the customer's consent to store their private information. This process is also known as establishing a relationship with the customer.

Creating a TMS Token

When sending the initial CIT, you can create a TMS token to store the customer's credentials for the subsequent MITs. To create a TMS token, include the **processingInformation.actionTokenTypes** field in the authorization request. Set the field to one of these values based on the TMS token type you want to create:

Customer

Customer tokens store one or more customer payment instrument tokens and shipping address tokens.

Including a customer token in subsequent MITs eliminates the need to include billing

information, card information, and the previous transaction's ID.

```
"processingInformation": {
  "actionTokenTypes": [
   "customer"
]
```

For more information about this TMS token type, see *Customer Tokens* in the Token Management Service Developer Guide.

Payment Instrument

Payment instrument tokens store an instrument identifier token, card information, and billing information. Payment instruments are not linked to a customer token. Including a payment instrument in subsequent MITs eliminates the need to include billing information, card information, and the previous transaction's ID.

```
"processingInformation": {
  "actionTokenTypes": [
   "paymentInstrument"
]
```

For more information about this TMS token type, see *Payment Instrument Token* in the Token Management Service Developer Guide.

Instrument Identifier

Instrument identifier tokens store a PAN. Including an instrument identifier in subsequent MITs eliminates the need to include a PAN and the previous transaction's ID.

```
"processingInformation": {
  "actionTokenTypes": [
   "instrumentIdentifier"
]
```

For more information about this TMS token type, see *Instrument Identifier Token* in the Token Management Service Developer Guide.

Instrument Identifier, Payment Instrument, and Customer Identifier

You can also create multiple TMS token types in the same authorization. This example includes an instrument identifier, a

payment instrument, and a customer token in the same authorization:

```
"processingInformation": {
  "actionTokenTypes": [
     "instrumentIdentifier",
     "paymentInstrument",
     "customer"
]
```

Endpoint

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

Required Fields for Authorizing Initial CIT Standing Order Payments with TMS

Mastercard Subscription Payments

A subscription payment is a recurring COF transaction that is processed at a fixed amount at regular intervals not to exceed one year between transactions. The series of recurring payments is the result of an agreement between you and the customer for the purchase of goods or services that are provided at regular intervals.

Mastercard CIT Initial Subscription Payment

The first transaction in a subscription payment is a customer-initiated transaction (CIT). Before you can perform a subsequent merchant-initiated transaction (MIT), you must store the customer's credentials for later use. Before you can store the user's credentials, you must get the customer's consent to store their private information. This process 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`.

Successful Response

Store the **ccAuthReply_paymentNetworkTransactionID** field value from the successful response message. You must include the network transaction ID in subsequent MIT authorization requests to associate the CIT to the MIT.

Required Fields for Authorizing CIT Initial Subscription Payments

Use these required fields to authorize customer-initiated initial subscription payments.

billTo_city

billTo_country	
billTo_email	
billTo_firstName	
billTo_lastName	
billTo_postalCode	
billTo_state	
billTo_street1	
card_accountNumber	
card_expirationMonth	
card_expirationYear	
ccAuthService_commerceIndicator	**Set the value to** `recurring`**.**
ccAuthService_run	**Set the value to** `true`**.**
merchantID	
merchantReferenceCode	
purchaseTotals_currency	
purchaseTotals_grandTotalAmount	
subsequentAuthFirst	**Set the value to** `true`**.**
subsequentAuthReason	Set the value to `7`.

Related Information

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

Simple Order Example: Authorizing Initial CIT Subscription Payments

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
ccAuthService_commerceIndicator=internet
merchantId=pa_ctv_sg101
merchantReferenceCode=33557799
purchaseTotals_currency=USD
```

```
purchaseTotals_grandTotalAmount=100.00
subsequentAuthFirst=true
subsequentAuthReason=7
```

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
```

Unscheduled COF Payments

An unscheduled credentials-on-file (COF) transaction uses stored payment information for a fixed or variable amount that does not occur regularly. An account top-up is one kind of unscheduled COF.

Customer-Initiated Unscheduled COF Payment with PAN

An unscheduled credentials-on-file (COF) transaction uses stored payment information for a fixed or variable amount that does not occur regularly. An account top-up is one kind of unscheduled COF.

Prerequisites

The first transaction in an unscheduled COF payment is a customer-initiated transaction (CIT). Before you can perform a subsequent merchant-initiated transaction (MIT), you must store the customer's credentials for later use. Before you can store the user's credentials, you must get the customer's consent to store their private information. This process 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`.

Successful Response

Store the **ccAuthReply_paymentNetworkTransactionID** field value from the successful response message. You must include the network transaction ID in subsequent MIT authorization requests to associate the CIT to the MIT.

Required Fields for Authorizing Initial CIT Unscheduled COF Payments

These fields are required in a subsequent authorization request for an initial unscheduled COF payment:

billTo_city	
billTo_country	
billTo_email	
billTo_firstName	
billTo_lastName	
billTo_postalCode	
billTo_state	
billTo_street1	
card_accountNumber	
card_expirationMonth	
card_expirationYear	
ccAuthService_commerceIndicator	**Set the value to `internet` or a payer authentication value.**
ccAuthService_run	**Set the value to `true`.**
merchantID	
merchantReferenceCode	
purchaseTotals_currency	
purchaseTotals_grandTotalAmount	
subsequentAuthFirst	**Set the value to `true`.**
subsequentAuth	**Set the value to `false`.**

Related Information

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

Simple Order Example: Authorizing Initial CIT Unscheduled COF Payments

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
ccAuthService_commerceIndicator=internet
merchantId=pa_ctv_sg101
merchantReferenceCode=33557799
purchaseTotals_currency=USD
purchaseTotals_grandTotalAmount=100.00
subsequentAuth=false
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
```

Merchant-Initiated Unscheduled COF Payments with PAN

After the initial CIT unscheduled COF payment, subsequent unscheduled COF transactions are merchant-initiated transactions (MITs).

Endpoint

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

Required Fields for Authorizing Subsequent MIT Unscheduled COF Payments

These fields are required in a subsequent authorization request for a subsequent unscheduled COF payment:

billTo_city

billTo_country

billTo_email

billTo_firstName

billTo_lastName

billTo_postalCode

billTo_state

billTo_street1	
card_accountNumber	
card_expirationMonth	
card_expirationYear	
ccAuthService_commerceIndicator	**Set the value to `internet`.**
ccAuthService_run	**Set the value to `true`.**
merchantID	
merchantReferenceCode	
purchaseTotals_currency	
purchaseTotals_grandTotalAmount	
subsequentAuth	**Set the value to `true`.**
subsequentAuthStoredCredential	**Set the value to `true`.**
subsequentAuthTransactionID	**Set this field to the network transaction identifier that was returned in the ccAuthReply_paymentNetworkTransactionID field in the response message for the original authorization.**

Related Information

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

Card-Specific Required Field for Processing a Merchant-Initiated Transactions

Discover

The listed card requires an additional field:

subsequentAuthOriginalAmount	**Provide the original transaction amount.**

Related Information

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

Optional Field for Authorizing Subsequent MIT Unscheduled COF Transactions

You can use these optional fields to include additional information when authorizing request for an MIT unscheduled COF transaction:

subsequentAuthOriginalAmount	**Supported only for Discover for subsequent MIT unscheduled COF transactions.**

Related Information

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

Simple Order Example: Authorizing Subsequent MIT Unscheduled COF Payments

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_commerceIndicator=internet
merchantId=pa_ctv_sg101
merchantReferenceCode=33557799
purchaseTotals_currency=USD
purchaseTotals_grandTotalAmount=100.00
subsequentAuth=true
subsequentAuthOriginalAmount=100.00  //This field is for Discover only.
subsequentAuthStoredCredential=true
subsequentAuthTransactionID=23976974322
```

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
```

Visa Bill Payments Processing

This section shows you how to process a Visa Bill Payment for authorization and credit transactions.

Requirement

Before you process Visa Bill Payments, you must sign up with Visa to participate in the program.

Related Information

- See *Visa Bill Payments* on page 34 for a description of the Visa Bill Payment program.

Visa Bill Payments Authorizations

This section shows you how to process a Visa Bill Payment authorization.

Field Specific to this Use Case

Include this field in a standard authorization request for a Visa Bill Payment:

- **ccAuthService_billPayment**—Set the field 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 Processing a Visa Bill Payment Authorization

Important

> When using relaxed requirements for address data and the expiration date, 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 124.

billTo_city	
billTo_country	
billTo_email	
billTo_firstName	
billTo_lastName	
billTo_postalCode	
billTo_state	
billTo_street1	
card_accountNumber	
card_expirationMonth	
card_expirationYear	
ccAuthService_billPayment	Set the value to `true`.
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 a Visa Bill Payment 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
```

```
ccAuthService_billPayment=true
merchant_id=npr_paymentech
merchant_referenceCode=TC42703-1
purchaseTotals_currency=usd
purchaseTotals_grandTotalAmount=100
```

Response to a Successful Request

```
additional_processor_response=e1cdcafc-cdbb-4ef7-8788-a1234e844805
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
requestToken=Axj/7wSTZ2ltSmjDPebrABEg1buWrNi2awW9mNQpt26aVBa2DsAU0qC1sHZpA6cQIz4ZNuOAZxjceFoE5NnaW1KaMM
```

Response to a Declined Request

```
requestID=6629977932421985593067
requestToken=Axj77w5434574i9uAabDoL
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
```

Visa Bill Payments Captures

This section shows you how to process a Visa Bill Payment capture.

Field Specific to this Use Case

Include this field in a standard capture request for a Visa Bill Payment:

- **ccCaptureService_billPayment**—Set the field to `true`.

Endpoint

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

Required Fields for Capturing a Visa Bill Payment

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

Related Information

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

Simple Order Example: Capturing a Visa Bill Payment

Request

```
ccCaptureService_authRequestID=6629978499572480812782
ccCaptureService_run=true
ccCaptureService_billPayment=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
```

Visa Consumer Bill Payment Processing

This section shows you how to process a Visa Consumer Bill Payment.

Fields Specific to this Use Case

Include this field in a standard authorization when you request a Visa CBPS authorization:

invoiceHeader_businessApplicationID	**Set the value to** CB.

Endpoint

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

Requirements

> **Important**
>
> As a Visa CBPS provider you must comply with all Visa rules. For the Visa CBPS rules, see Rule ID# 0030635 of the Visa Rules "Acquirer Responsibilities Related to Consumer Bill Payment Service Providers" at *https://usa.visa.com/dam/VCOM/download/about-visa/visa-rules-public.pdf*.

Before you process Visa CBPS payments, you must:

- Contact your acquirer to register your account for the Visa Consumer Bill Pay Service.
- Contact Cybersource customer support to have your account configured for this feature.

Related Information

- See *Basic Authorizations* on page 36.

Required Fields for Processing a Visa Consumer Bill Payment Service Authorization

Important

When using relaxed requirements for address data and the expiration date, 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 124.

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.
invoiceHeader_businessApplicationID	**Set the value to** CB.
merchantID	
merchantReferenceCode	
purchaseTotals_grandTotalAmount	

Related Information

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

Simple Order Example: Processing a Visa Consumer Bill Payment Service 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=MRC987654321
purchaseTotals_currency=USD
purchaseTotals_grandTotalAmount=100
invoiceHeader_businessApplicationID=CB
```

Response to a Successful Request

```
requestID=6629977932421985593067
decision=ACCEPT
reasonCode=100
merchantReferenceCode=MRC987654321
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=MRC987654321
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
```