Visa Click to Pay

Simple Order API

Developer Guide

cybersource
A Visa Solution

Version: 22.02

Contents

Recent Revisions to This Document

22.02

Added a link to the test card numbers to Testing 3D Secure 2.x with Visa Click to Pay (on page 10).

22.01

Added examples for Authorization, Capture, Credit, and Authorization Reversal.

21.02

Updated Using 3D Secure with Visa Click to Pay (on page 9).

Added Testing 3D Secure 2.x with Visa Click to Pay (on page 10).

21.01

Changed the name of *Visa Checkout* to *Visa Click to Pay*.

Updated Supported Countries, Regions, and Payment Currencies (on page 32).

19.04

Updated the enrollment URL. See Visa Click to Pay Documents (on page 6).

Updated countries, regions, and currencies. See Supported Countries, Regions, and Payment Currencies (on page 32).

19.03

Updated the XML request example. See Visa Click to Pay Data Request (on page 28).

Added the vcReply_newUser response field. See vcReply_newUser (on page 23).

About This Guide

Audience and Purpose

This guide is written for application developers who want to use the Cybersource Simple Order API to integrate Visa Click to Pay into their order management system.

Implementing Cybersource services requires software development skills. You must write code that uses the API request and response fields to integrate the Cybersource services into your existing order management system.

Text and Command Conventions

Convention	Usage
bold	Field and service names in text; for example: Include the **getVisaCheckoutDataService_run** field.
`screen text`	• XML elements • Code examples • Values for API fields; for example: Set the **ccAuthService_run** field to `true`.

Related Documentation

Cybersource Documents

- *Getting Started with Cybersource Advanced for the Simple Order API* (PDF | HTML)
- *Credit Card Services Using the Simple Order API* (PDF | HTML)
- *Payer Authentication Using the Simple Order API* (PDF | HTML)

Refer to the Support Center for complete Cybersource technical documentation:

http://www.cybersource.com/support_center/support_documentation

Visa Click to Pay Documents

- *Getting Started with Visa Click to Pay* (published by Visa)
- *Visa Click to Pay JavaScript Integration Guide*

To obtain these documents, contact your local Cybersource sales representative:

http://www.cybersource.com/locations

You can also obtain these documents by signing up for a Visa Click to Pay developer account:

https://developer.visa.com/#enroll

Customer Support

For support information about any Cybersource service, visit the Support Center:

http://www.cybersource.com/support

Integrating Visa Click to Pay into Your System

Visa Click to Pay is Visa's solution for e-commerce payments based upon the EMV® Secure Remote Commerce (EMV SRC) standards and specifications. With EMV SRC, a single payment profile can be used with a variety of consumer devices and participating online merchants. The standards include a common payment icon and user experience for card-based digital transactions, support for cardholder verification methods, and a common data payload built on primary account numbers (PANs) and the ability to support network tokens.

Requirements

- You must have a Visa Click to Pay merchant account. If you do not already have a Visa Click to Pay merchant account, contact your local Cybersource sales representative: http://www.cybersource.com/locations
- You must have a Cybersource account. If you do not already have a Cybersource account, contact your local Cybersource sales representative.
- You must contact Cybersource Customer Support to have your account configured for Visa Click to Pay.
- When you use the Simple Order API in XML format, you must use version 1.105 or later of the XML schema.
- You must be familiar with the Cybersource credit card services as described in Credit Card Services Using the Simple Order API. .
- If you are including payer authentication in your Visa Click to Pay implementation, you must be familiar with the Cybersource payer authentication services as described in Payer Authentication Using the Simple Order API. .

Supported Countries

For a list of the countries and associated currencies from which you can accept Visa Click to Pay payments, refer to Supported Countries, Regions, and Payment Currencies (on page 32).

Visa Click to Pay Process

Visa Click to Pay uses Visa Checkout services and API fields.

1. You send data to Visa Click to Pay to display the Visa Click to Pay button on your checkout page. For details about this step, contact your Cybersource sales representative and consult *Getting Started with Visa Click to Pay* (published by Visa). To obtain this document, see Visa Click to Pay Documents (on page 6).

2. You retrieve the Visa Click to Pay payment data so that you can display it to your customer. However, you cannot retrieve the PAN unless your account is configured for it. See Getting Visa Click to Pay Data (on page 8). The primary account number (PAN) is not required in order to process a Visa Click to Pay transaction.

3. Include the following required fields:

    - **ccAuthService_run**
    - **merchantID**
    - **merchantReferenceCode**
    - **paymentSolution**
    - **purchaseTotals_currency**
    - **purchaseTotals_grandTotalAmount** or at least one **item_#_unitPrice** field
    - **vc_orderID**

    For descriptions of these fields, see Credit Card Services Using the Simple Order API.

4. Cybersource obtains payment data from Visa Click to Pay and includes it in the authorization request that is sent to the processor.

5. For follow-on transactions such as full authorization reversal, capture, and credit, you must include the following fields in your request in addition to the required fields documented in Credit Card Services Using the Simple Order API.

    - paymentSolution
    - vc_orderID

Getting Visa Click to Pay Data

Visa Click to Pay uses Visa Checkout services and API fields.

The Visa Checkout data service enables you to receive the decrypted Visa Click to Pay data in the response message. However, you cannot retrieve the PAN unless your account is configured for it. You can use the retrieved data to help the customer confirm the purchase.

See Simple Order API Fields (on page 13) for:

- Descriptions of these required request fields
- Descriptions of all response fields

Create a Visa Click to Pay Data Request

1. Set the **getVisaCheckoutDataService_run** field to true.
2. Do not include any other Cybersource services in the request.
3. Include the following required fields in the request:
    - **getVisaCheckoutDataService_run**
    - **merchantID**
    - **merchantReferenceCode**
    - **paymentSolution**
    - **vc_orderID**

Using 3D Secure with Visa Click to Pay

Payer authentication is the Cybersource implementation of 3D Secure.

For Visa Click to Pay, Cybersource supports the following kinds of payer authentication:

- American Express SafeKey
- Mastercard Identity Check
- Visa Secure

To integrate payer authentication see:

- Credit Card Services Using the Simple Order API
- Payer Authentication Using the Simple Order API

When you implement 3D Secure 2.x with Visa Click to Pay, you must integrate the Cardinal Cruise Direct API version of Payer Authentication as described in the Payer Authentication Using the Simple Order API and include the following fields:

- **paymentSolution** –set to visacheckout
- **vc_orderID** –set to **callID** field in the visacheckout reply payload

Contact customer support to configure your account to support this integration to ensure the correct StepUpURL fields are returned by payer authentication. If you have previously on-boarded with 3D Secure 1 or 3D Secure 2.x Hybrid or Standard Payer Authentication methods you will still need to contact customer support.

Important: With Visa Click to Pay, you must include the payer authentication enrollment service **payerAuthEnrollService** and the credit card authorization service **ccAuthService** in the same request message in order to decrypt the primary account number (PAN) and complete the rest of the payer authentication flow. When you submit a separate request message for each service, the payer authentication enrollment service **payerAuthEnrollService** request fails.

Visa Secure

For Visa Click to Pay transactions, do not map the Visa Secure data from the decrypt Visa Click to Pay data service response message to the payer authentication fields in the authorization request. The data is mapped for you. The transaction information that is sent to the processor includes the Visa Secure data.

Testing 3D Secure 2.x with Visa Click to Pay

Get test card numbers from the Payer Authentication developer guide in the Testing Payer Authentication section. See the Test Cases for 3-D Secure 2.x section in the Payer Authentication developer guide here.

Important: Only the Visa test card numbers listed for the 3-D Secure 2.x test cases in the Payer Authentication developer guide are configured for Visa Click to Pay. Other test card types will not work.

Use the Visa card number specified in the test with the card's expiration date set to the month of January and the current year plus three. For example, for 2022, use 2025. You also need the minimum required fields for an order. Be sure to remove spaces in card numbers when testing.

The XID values are included in 3D Secure 2.x test cases for legacy reasons.

While the 3D Secure version and directory server transaction ID fields are returned for the Check Enrollment and Validate Authentication services, this data is not included in the 3D Secure 2.x test cases.

Using Decision Manager with Visa Click to Pay

While the Visa Click to Pay response contains many of the fields necessary to run Decision Manager it does not include these essential Decision Manager fields:

- Device fingerprint
- True IP address

You must capture these fields independently.

API Fields

Formatting Restrictions

Do not use the following characters: < > $ % ^ * _ = [ ] \ { } | ; ~ ` Using these characters may result in data validation errors.

Data Type Definitions

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

Data Type	Description
Date and time	Format is yyyy-MM-DDThh:mm:ssZ where: • T separates the date and the time. • Z indicates Coordinated Universal Time (UTC), also known as Greenwich Mean Time (GMT). **Example:** 2021-01-11T22:47:57Z is January 11, 2021, at 22:47:57 (10:47:57 p.m.).
Integer	Whole number {..., -3, -2, -1, 0, 1, 2, 3, ...}
String	Sequence of letters, numbers, spaces, and special characters

Simple Order API Fields

Request Fields

Request Fields

Field	Description	Used By: Required (R) or Optional (O)	Data Type & Length
getVisaCheckoutDataService_run	Whether to include **getVisaCheckoutDataService** in your request. Possible values: • `true`: Include the service in your request. • `false` (default): Do not include the service in your request.	getVisaCheckoutDataService (R)	String (5)
merchantID	Your Cybersource merchant ID. Use the same merchant ID for evaluation, testing, and production.	getVisaCheckoutDataService (R)	String (30)
merchantReferenceCode	Order reference number or tracking number generated by you. Cybersource recommends that you send a unique value for each transaction so that you can perform meaningful searches for the transaction. For information about tracking orders, see Getting Started with Cybersource Advanced for the Simple Order API.	getVisaCheckoutDataService (R)	String (50)
paymentSolution	Type of payment solution that is being used for the transaction. The value for Visa Click to Pay is `visacheckout`.	getVisaCheckoutDataService (R)	String (12)
vc_orderID	Identifier for the Visa Click to Pay order. Visa Click to Pay provides a unique order ID for every transaction in the Visa Click to Pay callID field.	getVisaCheckoutDataService (R)	String (48)

Request Fields (continued)

Field	Description	Used By: Required (R) or Optional (O)	Data Type & Length
wallet_discountAmount	Total discount amount. The discount amount must be a positive value. Includes a decimal point and a maximum of four decimal places.	ics_auth (O)	String (14)
wallet_eventType	Type of transaction event. Possible values: • `Create`: Card-on-file saved (outside of a purchase flow). • `Confirm`: Order placed. • `Confirm_COF`: Order placed using a card-on-file. • `Cancel`: Order canceled. • `Fraud`: Order rejected by risk or fraud review. • `Other`: None of the events above, or a payment event after a `Confirm` or `Confirm_COF` order event. The default value is `Confirm`.	ics_auth (O)	String (15)
wallet_giftWrapAmount	Gift-wrapping total that is sent after a successful authorization. Includes a decimal point and a maximum of four decimal places.	ics_auth (O)	String (14)
wallet_promotionCode	Promotion code that is sent after a successful authorization. The valid characters for the wallet promotion code are: • Numbers • Letters • The following special characters:	ics_auth (O)	String (100)

Request Fields (continued)

Field	Description	Used By: Required (R) or Optional (O)	Data Type & Length
	asterisk (*), at (@), dash (-), dollar sign ($), exclamation point (!), hash (#), parentheses (( )), percent (%), plus (+), underscore (_), comma (,), and space. Use a period to separate multiple promotion codes.		
wallet_subtotalAmount	Subtotal amount that contains purchase details. Cybersource does not validate this field. Includes a decimal point and a maximum of two decimal places.	ics_auth (O)	String (10)
wallet_totalPurchaseAmount	Total purchase amount. By default, Cybersource uses the grand total amount of the authorization. Includes a decimal point and a maximum of two decimal places.	ics_auth (O)	String (10)

Response Fields

Visa Click to Pay returns all decrypted data to you, except the PAN, unless your account is configured to receive it. The purpose of the fields in the Visa Click to Pay encrypted payment data is to pass information from Visa Click to Pay to the processor. Consequently, many decrypted fields and values might not be useful to you.

Response Fields

Field	Description	Data Type & Length
billTo_city	Decrypted city in the billing address.	String (100)
billTo_county	Name of the municipality. This value is common for addresses in Mexico.	String (80)
billTo_country	Decrypted country in the billing address. For the possible values, see ISO Standard Country Codes.	String (2)

Response Fields (continued)

Field	Description	Data Type & Length
billTo_defaultIndicator	Shipping address is flagged as the default shipping address by the customer. Possible values: • `true`: This billing address is the customer's default address. • `false`: This billing address is not the customer's default billing address.	String (5)
billTo_name	Decrypted customer name.	String (256)
billTo_phoneNumber	Decrypted customer phone number.	String (30)
billTo_pointOfReference	Decrypted location information. In some countries, such as Mexico and India, it is common to ask for a point of reference or landmark for the billing or shipping address. For example, "Across the street from the grocery store."	String (140)
billTo_postalCode	Decrypted postal code in the billing address.	String (100)
billTo_state	Decrypted state or province in the billing address. For possible values, see State, Province, and Territory Codes for the United States and Canada.	String (3)
billTo_street1	Decrypted first line of the street address in the billing address as it appears on the credit card issuer's records.	String (100)
billTo_street2	Decrypted additional address information in the billing address.	String (100)
billTo_street3	Decrypted additional address information in the billing address.	String (100)
billTo_street4	Decrypted additional address information in the billing address.	String (100)
card_accountNumber	Decrypted customer's credit card number. Returned only when your account is configured to receive it. For more information about receiving the PAN, see *Getting Started with Visa Click to Pay* (PDF | HTML)*Getting Started with Visa Secure Remote Commerce* (PDF | HTML).	String (20)

Response Fields (continued)

Field	Description	Data Type & Length
card_expirationMonth	Decrypted two-digit month in which the credit card expires. Format: MM. Possible values: `01` through `12`.	String with numbers only (2)
card_expirationYear	Decrypted four-digit year in which the credit card expires. Format: YYYY.	String with numbers only (4)
card_prefix	Decrypted credit card prefix. This value is the first six digits of the cardholder's account number.	String with numbers only (6)
card_suffix	Decrypted credit card suffix. This value is the last four digits of the cardholder's account number. You can use this value on the receipt that you give to the cardholder.	String with numbers only (4)
decision	Summarizes the result of the overall request. Possible values: • `ACCEPT` • `ERROR` • `REJECT` For details about these values, see the information about handling replies in Getting Started with Cybersource Advanced for the Simple Order API.	String (6)
getVisaCheckoutDataReply_reasonCode	Numeric value corresponding to the result of the Visa Click to Pay decryption request. See Reason Codes (on page 31).	Integer (5)
invalidField_0 through invalidField_N	Fields in the request that have invalid data. For information about missing or invalid fields, see Getting Started with Cybersource Advanced for the Simple Order API. **Note:** These fields are included as an aid to software developers only. Do not use these fields to interact with your customers.	String (100)

Response Fields (continued)

Field	Description	Data Type & Length
merchantReferenceCode	Order reference number or tracking number that you provided in the request. If you included multibyte characters in this field in the request, the returned value might include corrupted characters.	String (50)
missingField_0 through missingField_N	Required fields that were missing from the request. For information about missing or invalid fields, see Getting Started with Cybersource Advanced for the Simple Order API. **Note:** These fields are included as an aid to software developers only. Do not use these fields to interact with your customers.	String (100)
personalID_number	Personal ID number. Only returned if your account is configured to receive personally identifiable information (PII) such as a primary account number (PAN).	String (18)
purchaseTotals_currency	Decrypted currency used for the order. For the possible values, see ISO Standard Currency Codes.	String (3)
reasonCode	Numeric value corresponding to the result of the overall request. See Reason Codes (on page 31).	Integer (5)
requestID	Identifier for the request. This value is provided by Cybersource.	String (26)
shipTo_addressVerificationStatus	Decrypted verification status for the shipping address. The verification status is determined by Visa Click to Pay. Possible values: • `FAILED` • `NOT_VERIFIED` • `VERIFIED`	String (12)
shipTo_city	Decrypted city of the shipping address.	String (100)
shipTo_country	Decrypted country of the shipping address. For the possible values, see ISO Standard Country Codes.	String (2)
shipTo_default	Status of the default shipping address. Determines whether it is flagged as the default shipping address by the customer. Possible values:	String (5)

Response Fields (continued)

Field	Description	Data Type & Length
	• `true`: This shipping address is the customer's default shipping address. • `false`: This shipping address is not the customer's default shipping address.	
shipTo_id	Decrypted identifier for the shipping address. This value is generated by Visa Click to Pay.	String (36)
shipTo_name	Decrypted name of the recipient.	String (256)
shipTo_phoneNumber	Decrypted phone number for the shipping address.	String (30)
shipTo_pointOfReference	In some countries, such as Mexico and India, it is common to ask for a point of reference or landmark for the billing or shipping address. For example, "Across the street from the grocery store."	String (140)
shipTo_postalCode	Decrypted postal code of the shipping address. Consists of 5 to 9 digits.	String (100)
shipTo_state	Decrypted state or province of the shipping address. For possible values, see State, Province, and Territory Codes for the United States and Canada.	String (3)
shipTo_street1	Decrypted first line of the shipping address.	String (100)
shipTo_street2	Decrypted second line of the shipping address.	String (100)
shipTo_street3	Decrypted third line of the shipping address.	String (100)
shipTo_street4	Decrypted fourth line of the shipping address.	String (100)
vcReply_ageOfAccount	Number of days since the Visa Click to Pay account was created.	Numeric (9)
vcReply_alternateShippingAddressCountryCode	Decrypted country code for the alternate shipping address.	String (2)
vcReply_alternateShippingAddressPostalCode	Decrypted postal code for the alternate shipping address.	String (10)
vcReply_avsCodeRaw	Decrypted raw (unmapped) AVS code provided by Visa Click to Pay.	String (10)

Response Fields (continued)

Field	Description	Data Type & Length
vcReply_billingAddressAdditionalLocation	Extracts and provides the additional location from the first line of the billing address. In some countries, such as Mexico and India, Visa Click to Pay obtains street information as a separate line item from the customer.	String (100)
vcReply_billingAddressStreetName	Extracts and provides the street name from the first line of the billing address. In countries such as Mexico and India, Visa Click to Pay obtains street information as a separate line item from the customer.	String (116)
vcReply_cardArt_#_fileName	Decrypted URL, including file name, for the card art. Visa Click to Pay provides the card art values. Any number of **vcReply_cardArt_#_fileName** fields that can be included in the encrypted payment data.	String (100)
vcReply_cardArt_#_height	Decrypted height for the card art in pixels. Possible values: `1` through `4096`. Visa Click to Pay provides the card art values. Any number of **vcReply_cardArt_#_height** fields that can be included in the encrypted payment data.	Positive Integer (4)
vcReply_cardArt_#_width	Decrypted width for the card art in pixels. Possible values: `1` through `4096`. Visa Click to Pay provides the card art values. Any number of **vcReply_cardArt_#_width** fields that can be included in the encrypted payment data.	Positive Integer (4)
vcReply_cardFirstName	Customer's first name as printed on the card.	String (256)
vcReply_cardGroup	Decrypted card group. Possible values: • `CREDIT` • `DEBIT` • `DEBIT/CREDIT`	String (12)
vcReply_cardLastName	Customer's last name as printed on the card.	String (256)
vcReply_cardType	Decrypted card type. Possible values:	String (10)

Response Fields (continued)

Field	Description	Data Type & Length
	• `AMEX` • `DISCOVER` • `MASTERCARD` • `VISA`	
vcReply_cardVerificationStatus	Decrypted verification status for the card. Possible values: • `FAILED` • `NOT_VERIFIED` • `VERIFIED`	String (12)
vcReply_creationTimeStamp	Decrypted time stamp for the creation of the Visa Click to Pay order. Format: Unix time, which is also called *epoch time*.	String (20)

vcReply_customData_#_name	Name for the name-value pair of custom data values that you define. You can define up to 100 Visa Click to Pay custom data name-value pairs to include in encrypted payment data. Use this field to specify the name for the name-value pair.
	
	Format: vcReply_customData_#_name=*name of field*
	where # equals the number of the name-value pair (0 through 99) and *name* equals the name for the name-value pair.
	
	Examples:
	
	`vcReply_customData_0_name=lastname`
	
	`vcReply_customData_1_name=firstname`
	
	`vcReply_customData_2_name=company`
		String (1024 for the name-value pair; the combined name and value fields)

vcReply_customData_#_value	Value for the name-value pair of custom data values that you define. You can define up to 100 Visa Click to Pay custom data name-value pairs to include in encrypted payment data. Use this field to specify the value for the name-value pair.	String (1024 for the name-value pair; the

Response Fields (continued)

Field	Description	Data Type & Length

	Format: vcReply_customData_#_value=*value of field*
	where # equals the number of the name-value pair (0 through 99) and *value* equals the value for the name-value pair.
	
	Examples:
	
	`vcReply_customData_0_value=Smith`
	
	`vcReply_customData_1_value=Jane`
	
	`vcReply_customData_2_value=Foster City Flowers`
		combined name and value fields)

vcReply_cvnCodeRaw	Decrypted raw (unmapped) CVN code provided by Visa Click to Pay.	String (10)
vcReply_discountAmount	Decrypted discount amount that you provided to Visa Click to Pay.	String (7)
vcReply_eci	Decrypted e-commerce indicator. Visa Click to Pay generates this value. Possible values for Visa, American Express, and JCB: • `05`: Card issuer is liable. • `06`: Card issuer is liable. • `07`: Merchant is liable. Possible values for Mastercard: • `01`: Merchant is liable. • `02`: Card issuer is liable.	String (20)
vcReply_eciRaw	Decrypted raw (unmapped) e-commerce indicator. Visa Click to Pay generates this value. Possible values for Visa, American Express, and JCB:	String (no maximum length)

Response Fields (continued)

Field	Description	Data Type & Length
	• `05`: Card issuer is liable. • `06`: Card issuer is liable. • `07`: Merchant is liable. Possible values for Mastercard: • `01`: Merchant is liable. • `02`: Card issuer is liable.	
vcReply_expiredCard	Card used for Visa Click to Pay payment is an expired card. Possible values are: • `true`: This card is an expired card. • `false`: This card is not an expired card.	String (5)
vcReply_giftWrapAmount	Decrypted gift wrap amount that you provided to Visa Click to Pay.	String (7)
vcReply_issuerID	Decrypted issuer ID.	String (100)
vcReply_merchantReferenceID	Decrypted tracking number for the Visa Click to Pay order. You provide this value to Visa Click to Pay.	String (100)
vcReply_nameOnCard	Decrypted name that is on the credit card.	String (256)
vcReply_newUser	Status of the user at the time of checkout. Possible values are: • `true`: This card is a new user. • `false`: This card is not a new user.	String (5)
vcReply_paresStatus	Decrypted payer authentication result enrollment status. Visa Click to Pay generates this value. Possible values:	String (1)

Response Fields (continued)

Field	Description	Data Type & Length
	• `A`: Proof of authentication attempt was generated. • `N`: Customer failed or canceled authentication. Transaction denied. • `U`: Authentication not completed regardless of the reason. • `Y`: Customer was successfully authenticated.	
vcReply_paresTimeStamp	Decrypted time stamp for the payer authentication result. Visa Click to Pay generates this value. Format: Unix time, which is also called *epoch time*.	String (no maximum length)
vcReply_paymentInstrumentID	Decrypted unique internal ID associated with the payment instrument. Visa Click to Pay generates this value.	String
vcReply_paymentInstrumentNickName	Decrypted name that the customer assigned to the payment instrument.	String (100)
vcReply_promotionCode	Decrypted promotion code that you provided to Visa Click to Pay.	String (100)
vcReply_riskAdvice	Decrypted risk advice to use with your fraud model. Visa Click to Pay provides the risk advice. Possible values: • `HIGH`: Higher than medium level of risk anticipated. • `LOW`: Lower than medium level of risk anticipated. • `MEDIUM`: Medium level of risk anticipated. • `UNAVAILABLE`: No information available.	String (11)
vcReply_riskScore	Decrypted risk score to use with your fraud model. Visa Click to Pay provides the risk score. Possible values: `0` through `99`. A value of `0` indicates that a risk score is not available. For values `1` through `99`, a higher score indicates a higher perceived risk.	Positive Integer (2)

Response Fields (continued)

Field	Description	Data Type & Length
vcReply_shippingAddressStreetName	Extracts and provides the street name from the first line of the billing address. In some countries, such as Mexico and India, Visa Click to Pay obtains street name information as a separate line item from the customer.	String (116)
vcReply_shippingHandlingAmount	Decrypted shipping and handling amount that you provided to Visa Click to Pay.	String (7)
vcReply_subtotalAmount	Decrypted subtotal amount that you provided to Visa Click to Pay.	String (7)
vcReply_taxAmount	Decrypted tax amount that you provided to Visa Click to Pay.	String (20)
vcReply_totalPurchaseAmount	Decrypted total purchase amount that you provided to Visa Click to Pay.	String (7)
vcReply_uncategorizedAmount	Decrypted amount of uncategorized charges that you provided to Visa Click to Pay.	String (20)
vcReply_vcAccountEmail	Decrypted email associated with customer's Visa Click to Pay account.	String (265)
vcReply_vcAccountEncryptedID	Encrypted login ID for customer's Visa Click to Pay account.	String (100)
vcReply_vcAccountFirstName	Decrypted first name from the login information for customer's Visa Click to Pay account.	String (265)
vcReply_vcAccountFullName	Visa Click to Pay customer's full name.	String (256)
vcReply_vcAccountLastName	Decrypted last name from the login information for customer's Visa Click to Pay account.	String (265)
vcReply_vcAccountLoginName	Decrypted login name for customer's Visa Click to Pay account.	String (128)
vcReply_veresEnrolled	Decrypted verification response enrollment status. Visa Click to Pay generates this value. Possible values: • `N`: Card not enrolled. • `U`: Unable to authenticate regardless of the reason. • `Y`: Card enrolled. Authentication available.	String (1)

Response Fields (continued)

Field	Description	Data Type & Length
vcReply_veresTimeStamp	Decrypted time stamp for the verification response. Visa Click to Pay generates this value. Format: Unix time, which is also called *epoch time*.	String (no maximum length)
vcReply_walletReferenceID	Decrypted order identifier. This value is generated by Visa Click to Pay.	String (100)
vcReply_xid	Decrypted transaction identifier. Visa Click to Pay generates this value.	String (40)

Simple Order API Examples

Name-Value Pair Examples

Visa Click to Pay Data Request

```
getVisaCheckoutDataService_run=true
merchantID=Foster_City_Flowers
merchantReferenceCode=123456
paymentSolution=visacheckout
vc_orderID=335161017227386762
```

Visa Click to Pay Data Response

```
billTo_street1=100 Main Street
billTo_street2=Suite 1234
billTo_city=Foster City
billTo_country=US
billTo_state=CA
billTo_postalCode=94404
card_prefix=987654
card_suffix=1111
purchaseTotals_currency=USD
card_accountNumber=4111111111111111
card_expirationMonth=09
card_expirationYear=2018
billTo_name=Jane Smith
billTo_phoneNumber=6501234567
getVisaCheckoutDataReply_reasonCode=100
decision=ACCEPT
reasonCode=100
merchantReferenceCode=123456
requestID=4067382331040172491847
shipTo_addressVerificationStatus=VERIFIED
shipTo_street1=100 Main Street
shipTo_street2=Suite 1234
shipTo_city=Foster City
shipTo_country=US
shipTo_id=jz0l2LMWLobl8IEcNuSBj0J9uO2zSsNx1ETZGjPI
shipTo_name=Jane Smith
shipTo_phoneNumber=6501234567
shipTo_state=CA
shipTo_postalCode=94404
```

```
vcReply_vcAccountEmail=jsmith@example.com
vcReply_vcAccountEncryptedID=nIPl7vnm6EZj+n10rjEK5LiPMqn1DKX48B8GzXDY
vcReply_vcAccountFirstName=Jane
vcReply_vcAccountLastName=Smith
vcReply_vcAccountLoginName=jsmith@example.com
vcReply_alternateShippingAddressCountryCode=US
vcReply_alternateShippingAddressPostalCode=94404
vcReply_avsCodeRaw=Y
vcReply_cardArt_0_fileName=https://secure.checkout.visa.com/CardArt/uWO
   vgFoQISxPh.png
vcReply_cardArt_0_width=164
vcReply_cardArt_0_height=105
vcReply_cardGroup=CREDIT
vcReply_cardType=VISA
vcReply_cardVerificationStatus=VERIFIED
vcReply_creationTimeStamp=1406568920102
vcReply_cvnCodeRaw=M
vcReply_discountAmount=1
vcReply_giftWrapAmount=2
vcReply_issuerID=null
vcReply_merchantReferenceID=Order12345
vcReply_nameOnCard=Jane Smith
vcReply_paymentInstrumentID=XNLbQ16j8hxholOVMq5skxNn6GUDPYDTqRgdWpb3kbk
vcReply_paymentInstrumentNickName=Business Credit Card
vcReply_promotionCode=SUMMER SALE 123
vcReply_riskAdvice=LOW
vcReply_riskScore=0
vcReply_shippingHandlingAmount=2
vcReply_subtotalAmount=10
vcReply_taxAmount=1
vcReply_totalPurchaseAmount=16
vcReply_uncategorizedAmount=2
vcReply_walletReferenceID=2kd94lcjksf04vcoqasdpde90trk
```

XML Examples

Visa Click to Pay Data Request

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.105">
   <merchantID>Foster_City_Flowers</merchantID>
   <merchantReferenceCode>123456</merchantReferenceCode>
   <paymentSolution>visacheckout</paymentSolution>
   <vc>
      <orderID>335161017227386762</orderID>
```

```
   </vc>
   <getVisaCheckoutDataService run="true"/>
</requestMessage>
```

Visa Click to Pay Data Response

```
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.105">
   <c:merchantReferenceCode>123456</c:merchantReferenceCode>
   <c:requestID>4067382331040172491847</c:requestID>
   <c:decision>ACCEPT</c:decision>
   <c:reasonCode>100</c:reasonCode>
   <c:purchaseTotals><c:currency>USD</c:currency></c:purchaseTotals>
   <c:shipTo>
      <c:street1>100 Main Street</c:street1>
      <c:street2>Suite 1234</c:street2>
      <c:city>Foster City</c:city>
      <c:state>CA</c:state>
      <c:postalCode>94404</c:postalCode>
      <c:country>US</c:country>
      <c:phoneNumber>6501234567</c:phoneNumber>
      <c:name>Jane Smith</c:name>
      <c:id>jz0l2LMWLobl8IEcNuSBj0J9uO2zSsNx1ETZGjPI</c:id>
      <c:addressVerificationStatus>VERIFIED</c:addressVerificationStatus>
   </c:shipTo>
   <c:billTo>
      <c:street1>100 Main Street</c:street1>
      <c:street2>Suite 1234</c:street2>
      <c:city>Foster City</c:city>
      <c:state>CA</c:state>
      <c:postalCode>94404</c:postalCode>
      <c:country>US</c:country>
      <c:phoneNumber>6501234567</c:phoneNumber>
      <c:name>Jane Smith</c:name>
   <c:/billTo>
   <c:card>
      <c:accountNumber>4111111111111111</c:accountNumber>
      <c:expirationMonth>09</c:expirationMonth>
      <c:expirationYear>2018</c:expirationYear>
      <c:suffix>1111</c:suffix>
      <c:prefix>987654</c:prefix>
   </c:card>
   <c:vcReply>
      <c:creationTimeStamp>1406568920102<c:creationTimeStamp>

 <c:alternateShippingAddressCountryCode>US</c:alternateShippingAddressCountryCode>

 <c:alternateShippingAddressPostalCode>94404</c:alternateShippingAddressPostalCo
de>
```

```
      <c:vcAccountLoginName>jsmith@example.com</c:vcAccountLoginName>
      <c:vcAccountFirstName>Jane</c:vcAccountFirstName>
      <c:vcAccountLastName>Smith</c:vcAccountLastName>

 <c:vcAccountEncryptedID>nIPl7vnm6EZLiPMqn1DKX48B8GzXDY</c:vcAccountEncryptedID>
      <c:vcAccountEmail>jsmith@example.com</c:vcAccountEmail>
      <c:merchantReferenceID>Order12345</c:merchantReferenceID>
      <c:subtotalAmount>10</c:subtotalAmount>
      <c:shippingHandlingAmount>2</c:shippingHandlingAmount>
      <c:taxAmount>1</c:taxAmount>
      <c:discountAmount>1</c:discountAmount>
      <c:giftWrapAmount>2</c:giftWrapAmount>
      <c:uncategorizedAmount>2</c:uncategorizedAmount>
      <c:totalPurchaseAmount>16</c:totalPurchaseAmount>
      <c:walletReferenceID>2kd94lcjksf04vcoqasdpde90trk</c:walletReferenceID>
      <c:promotionCode>SUMMER SALE 123</c:promotionCode>

 <c:paymentInstrumentID>XNLbQ16j8hxholOGUDPYDTqRgdWpb3kbk</c:paymentInstrumentID>
      <c:cardVerificationStatus>VERIFIED</c:cardVerificationStatus>
      <c:issuerID>null</c:issuerID>
      <c:paymentInstrumentNickName>Business Credit
 Card</c:paymentInstrumentNickName>
      <c:nameOnCard>Jane Smith</c:nameOnCard>
      <c:cardType>VISA</c:cardType>
      <c:cardGroup>CREDIT</c:cardGroup>
      <c:cardArt id="0">

 <c:fileName>https://secure.checkout.com/CardArt/uWOvgFoQISxPh.png</c:fileName>
         <c:height>105</c:height>
         <c:width>164</c:width>
      </c:cardArt>
      <c:riskAdvice>LOW</c:riskAdvice>
      <c:riskScore>0</c:riskScore>
      <c:avsCodeRaw>Y</c:avsCodeRaw>
      <c:cvnCodeRaw>M</c:cvnCodeRaw>
   </c:vcReply>
   <c:getVisaCheckoutDataReply>
      <c:reasonCode>100</c:reasonCode>
   </c:getVisaCheckoutDataReply>
</c:replyMessage>
```

Reason Codes

Table 8 (on page 31) lists the reason codes returned by the Simple Order API for Visa Click to Pay. See Getting Started with Cybersource Advanced for the Simple Order API for a discussion of replies, decisions, and reason codes.

Important: Because Cybersource can add response fields and reason codes at any time:

- You must parse the response data according to the names of the fields instead of the field order in the response. For more information about parsing response fields, see the documentation for your client.
- Your error handler should be able to process new reason codes without problems.
- Your error handler should use the **decision** field to determine the result if it receives a reason code that it does not recognize.

Reason Codes

Reason Code	Description
100	Successful transaction.
150	General system failure. See the documentation for your Cybersource client for information about handling retries in the case of system errors.

Supported Countries, Regions, and Payment Currencies

The following table identifies the countries, regions, and associated currencies from which payments are accepted:

Country or Region	Currency	Code
Argentina	Argentine peso	ARS
Australia	Australian dollar	AUD
Canada	Canadian dollar	CAD
Chile	Chilean peso	CLP
China Mainland	Chinese yuan renminbi	CNY
Colombia	Columbian peso	COP
France	Euro	EUR
Hong Kong	Hong Kong dollar	HKD
India	Indian rupee	INR
Ireland	Euro	EUR
Kuwait	Kuwaiti dinar	KWD
Malaysia	Malaysian ringgit	MYR
Mexico	Mexican peso	MXN
New Zealand	New Zealand dollar	NZD
Peru	Peruvian nuevo sol	PEN
Poland	Euro	EUR
Qatar	Qatari rial	QAR
Saudi Arabia	Saudi Arabian riyal	SAR
Singapore	Singapore dollar	SGD
South Africa	South African rand	ZAR
Spain	Euro	EUR
Ukraine	Ukrainian hryvnia	UAH
United Arab Emirates	United Arab Emirates dirham	AED
United Kingdom	British pound sterling	GBP

Country or Region	Currency	Code
United States of America	United States dollar	USD