

# Online Bank Transfers

## Using the Simple Order API

July 2018



## CyberSource Contact Information

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

For sales questions about any CyberSource Service, email [sales@cybersource.com](mailto: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 at <http://www.cybersource.com/support>.

## Copyright

© 2018 CyberSource Corporation. All rights reserved. CyberSource Corporation ("CyberSource") furnishes this document and the software described in this document under the applicable agreement between the reader of this document ("You") and CyberSource ("Agreement"). You may use this document and/or software only in accordance with the terms of the Agreement. Except as expressly set forth in the Agreement, the information contained in this document is subject to change without notice and therefore should not be interpreted in any way as a guarantee or warranty by CyberSource. CyberSource assumes no responsibility or liability for any errors that may appear in this document. The copyrighted software that accompanies this document is licensed to You for use only in strict accordance with the Agreement. You should read the Agreement carefully before using the software. Except as permitted by the Agreement, You may not reproduce any part of this document, store this document in a retrieval system, or transmit this document, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written consent of CyberSource.

## Restricted Rights Legends

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

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 Unpublished rights reserved under the copyright laws of the United States.

## Trademarks

Authorize.Net, eCheck.Net, and The Power of Payment are registered trademarks of CyberSource Corporation, CyberSource, CyberSource Payment Manager, CyberSource Risk Manager, CyberSource Decision Manager, and CyberSource Connect are trademarks and/or service marks of CyberSource Corporation.

All other brands and product names are trademarks or registered trademarks of their respective owners.

# Contents

## **Recent Revisions to This Document** 6

### **About This Guide** 7

Audience and Purpose 7

Conventions 7

    Note and Important Statements 7

    Text and Command Conventions 8

Related Documents 8

Customer Support 8

---

## **Chapter 1 Introduction** 9

    Creating a Merchant Account 9

    Online Bank Transfer Services 10

        Supported Payment Methods 10

        Options Service 11

        Sale Service 11

        Check Status Service 12

        Refund Service 12

---

## **Chapter 2 Processing Bancontact Transactions** 13

    Transaction Flow 14

    Performing a Sale 15

    Checking a Payment Status 16

    Refunding a Payment 17

    Testing 17

    Name-Value Pair Examples 18

        Sale Service 18

        Check Status Service 19

        Refund Service 20

XML Examples	21
Sale Service	21
Check Status Service	23
Refund Service	24

---

**Chapter 3    Processing EPS Transactions    25**

Transaction Flow	25
Performing a Sale	27
Checking a Payment Status	28
Testing	28
Name-Value Pair Examples	29
Sale Service	29
Check Status Service	30
XML Examples	31
Sale Service	31
Check Status Service	33

---

**Chapter 4    Processing giropay Transactions    34**

Transaction Flow	34
Performing a Sale	36
Checking a Payment Status	37
Testing	37
Name-Value Pair Examples	38
Sale Service	38
Check Status Service	39
XML Examples	40
Sale Service	40
Check Status Service	42

---

**Chapter 5    Processing iDEAL Transactions    43**

Transaction Flow	44
Requesting Options	46
Performing a Sale	46
Checking a Payment Status	48
Refunding a Payment	49
Testing	50

Name-Value Pair Examples	51
Options Service	51
Sale Service	52
Check Status Service	53
Refund Service	53
XML Examples	54
Options Service	54
Sale Service	55
Check Status Service	56
Refund Service	57

---

<b>Chapter 6</b>	<b>Processing Sofort Transactions</b>	<b>58</b>
	Transaction Flow	58
	Performing a Sale	60
	Checking a Payment Status	61
	When to Check a Payment Status	61
	Pending Status	61
	Authorized Status	61
	How to Check a Payment Status	62
	Refunding a Payment	63
	How to Refund a Payment	63
	Testing	64
	Testing Refunds	65
	Name-Value Pair Examples	65
	Sale Service	65
	Check Status Service	66
	Refund Service	67
	XML Examples	68
	Sale Service	68
	Check Status Service	70
	Refund Service	71

---

<b>Appendix A</b>	<b>API Fields</b>	<b>72</b>
	Formatting Restrictions	72
	Data Type Definitions	72
	Numbered Elements	73
	Request Fields	74
	Reply Fields	82
	Reason Codes and Processor Response Codes	86

# Recent Revisions to This Document

Release	Changes
July 2018	<p>iDEAL: updated the test value for the <b>apSaleService_paymentOptionID</b> field. See <a href="#">"Testing," page 50</a>.</p> <p>Sofort: updated the description for the <b>apSaleService_transactionTimeout</b> field. See <a href="#">"Request Fields," page 74</a>.</p>
March 2018	<p>Bancontact: updated the transaction flow. See <a href="#">"Transaction Flow," page 14</a>.</p> <p>iDEAL:</p> <ul style="list-style-type: none"> <li>■ Updated the transaction flow. See <a href="#">"Transaction Flow," page 44</a>.</li> <li>■ Updated the XML sale request example. See <a href="#">"XML Examples," page 54</a>.</li> </ul> <p>Sofort:</p> <ul style="list-style-type: none"> <li>■ Updated the transaction Flow. See <a href="#">"Transaction Flow," page 58</a>.</li> <li>■ Updated the possible values of the <b>billTo_language</b> field. See <a href="#">"Request Fields," page 74</a>.</li> </ul>
December 2017	<p>Added the <b>apSaleReply_processorTransactionID</b> reply field. See <a href="#">"Reply Fields," page 82</a>.</p> <p>Added " ," <a href="#">page 72</a>.</p>
August 2017	Updated all the Sale service examples.
July 2017	Updated the <b>apRefundReply_paymentStatus</b> reply field. See <a href="#">"Reply Fields," page 82</a> .
June 2017	Added links to transaction flow diagrams for each payment method.

# About This Guide

## Audience and Purpose

---

This guide is written for merchants who want to offer Online Bank Transfer services to customers. This guide describes the tasks a merchant must complete in order to make a payment, request the status of a payment, or refund a payment. It is intended to help the merchant provide a seamless payment experience for the customer.

## Conventions

---

## Note and Important Statements



### Note

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

---



### Important

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

---

## Text and Command Conventions

Convention	Usage
<b>bold</b>	<ul style="list-style-type: none"> <li>Field and service names in text; for example: Include the <b>ics_applications</b> field.</li> <li>Items that you are instructed to act upon; for example: Click <b>Save</b>.</li> </ul>
<i>italic</i>	<ul style="list-style-type: none"> <li>Filenames and pathnames. For example: Add the filter definition and mapping to your <i>web.xml</i> file.</li> <li>Placeholder variables for which you supply particular values.</li> </ul>
Screen text	<ul style="list-style-type: none"> <li>XML elements.</li> <li>Code examples and samples.</li> <li>Text that you enter in an API environment; for example: Set the <b>apSaleService_run</b> field to <code>true</code>.</li> </ul>

## Related Documents

- [Getting Started with CyberSource Advanced for the Simple Order API \(PDF | HTML\)](#)—describes how to start using your CyberSource advanced account.
- [Simple Order API and SOAP Toolkit API Documentation and Downloads page](#).

Refer to the Support Center for complete CyberSource technical documentation:

[http://www.cybersource.com/support\\_center/support\\_documentation](http://www.cybersource.com/support_center/support_documentation)

## Customer Support

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

<http://www.cybersource.com/support>



# Introduction

## Creating a Merchant Account

---

Two types of merchant accounts are available:

- The CyberSource settlement and services account—this merchant account has no direct contract with a payment provider partner. The CyberSource Financial Settlement Partner (FSP) collects the funds on your behalf and settles the funds to your merchant account. Contact your sales representative for additional information.



CyberSource automatically requests the export compliance service for each transaction using this type of merchant account. Export Compliance verifies customer information by comparing it to many lists that are maintained by government agencies to support export controls. If a customer's first name and last name or company name appears on any of the lists, you receive information indicating that the transaction is declined.

---

- The processor direct contract account—this merchant account must use the payment provider selected by CyberSource and you must inform your sales representative of any existing direct contracts.

## Online Bank Transfer Services

Online banking services enable customers to pay for goods using direct online bank transfers from their bank account to your merchant account.

### Supported Payment Methods

Table 1 lists the supported payment methods currently available for processing online bank transfers. Additional payment methods that use the online bank transfer services will be supported by CyberSource.

**Table 1 Payment Methods**

Payment Method	Country	Services	Availability
Bancontact (see <a href="#">Chapter 2, "Processing Bancontact Transactions,"</a> on page 13).	Belgium	<ul style="list-style-type: none"> <li>■ Sale</li> <li>■ Check Status</li> <li>■ Refund</li> </ul>	Available for beta merchants.
EPS (see <a href="#">Chapter 3, "Processing EPS Transactions,"</a> on page 25).	Austria	<ul style="list-style-type: none"> <li>■ Sale</li> <li>■ Check Status</li> </ul>	Available for beta merchants.
giropay (see <a href="#">Chapter 4, "Processing giropay Transactions,"</a> on page 34).	Germany	<ul style="list-style-type: none"> <li>■ Sale</li> <li>■ Check Status</li> </ul>	Available for beta merchants.
iDEAL (see <a href="#">Chapter 5, "Processing iDEAL Transactions,"</a> on page 43).	Netherlands	<ul style="list-style-type: none"> <li>■ Options</li> <li>■ Sale</li> <li>■ Check Status</li> <li>■ Refund</li> </ul>	Available for beta merchants.
Sofort (see <a href="#">Chapter 6, "Processing Sofort Transactions,"</a> on page 58).	<ul style="list-style-type: none"> <li>■ Austria</li> <li>■ Belgium</li> <li>■ Germany</li> <li>■ Italy</li> <li>■ Netherlands</li> <li>■ Spain</li> </ul>	<ul style="list-style-type: none"> <li>■ Sale</li> <li>■ Check Status</li> <li>■ Refund</li> </ul>	Available for beta merchants.

## Options Service



**Important**

This service is valid only for iDEAL transactions (see [Chapter 5, "Processing iDEAL Transactions,"](#) on page 43).

The options service (**apOptionsService**) retrieves a list of bank option IDs and bank names which you can display to the customer on your web site. CyberSource recommends requesting the options service once a day and caching the information.

When a customer chooses a bank on your web site, you must include the associated option ID in the sale request (see ["Sale Service,"](#) page 11) which returns a bank redirect URL to direct the customer to. If the option ID is not included in the sale request, the bank redirect URL returned directs the customer to a bank selection page.

## Sale Service

The sale service (**apSaleService**) returns the redirect URL for customer's bank. The customer is directed to the URL to confirm their payment details.

As part of the sale service request you must specify URLs to direct the customer to after the payment is successful, or fails, or when the customer cancels the transaction:

- **Success URL**—the customer is directed to this URL after completing a payment. Do not use this URL to confirm a payment because the response is not signed and it could be tampered with.



**Important**

CyberSource recommends that the success URL links to a static web page. The text should inform the customer that the payment is being processed and they will be informed of the order status (via email or a dedicated web page), rather than stating the payment was successful.

- **Cancel URL**—the customer is directed to this URL after canceling a payment or closing the browser.
- **Failure URL**—the customer is directed to this URL when a payment fails due to insufficient funds or a decline from the issuer.



**Important**

You can include the URLs in the sale service request or CyberSource can update your merchant configuration settings to include the URLs in all CyberSource service requests.

## Check Status Service

The check status service returns the latest status of a transaction. It is a follow-on request that uses the request ID value returned from the sale service request. The request ID value links the check status request to the payment transaction.

The check status service can also be used to request the status of a refund payment. In the request include the request ID value returned from the refund service request. The request ID value links the check status request to the refund transaction.

When a payment is confirmed by the customer, they are directed to your success URL.



CyberSource recommends that the success URL links to a static web page. The text should inform the customer that the payment is being processed and they will be informed of the order status (via email or a dedicated web page), rather than stating the payment was successful.

---

CyberSource recommends waiting 30 seconds before requesting check status service (**apCheckStatusService**). The check status response confirms whether the payment was accepted between the customer and the processor.



CyberSource recommends shipping goods when the payment status is updated to *settled*.

---

## Refund Service

The refund service request (**apRefundService**) is a follow-on request that uses the request ID value returned from the sale request. The request ID value links the refund transaction to the original payment transaction.



CyberSource recommends that the payment status must be *settled* before refunding a payment and returning funds to the customer account.

---

# Processing Bancontact Transactions

*Bancontact* is an e-commerce payment system in Belgium that enables customers to pay for goods using direct online bank transfers from their bank accounts to your merchant account.

These banks support the Bancontact payment method:

- ABK Bank
- Argenta
- Axa
- Bank VanBreda
- CPH Banque
- Belfius
- Beobank
- BNP Paribas Fortis
- Bpost Bank
- CBC
- Crelan
- Deutsche Bank
- ING
- KBC
- Keytrade Bank
- Nagelmackers
- Record Bank
- VDK Spaarbank

## Transaction Flow

---

For a transaction flow diagram see:

[http://apps.cybersource.com/library/documentation/dev\\_guides/OnlineBankTransfers/Bancontact.pdf](http://apps.cybersource.com/library/documentation/dev_guides/OnlineBankTransfers/Bancontact.pdf)

- 1 Display the *Bancontact* payment method on your checkout page.



**Important**

Supported browsers are Google Chrome, Internet Explorer, Safari, and Firefox.

---

- 2 Request the sale service (**apSaleService**) and include the corresponding bank ID for the bank that the customer chooses. Also send the success, cancel, and failure return URLs so that the bank can return the customer to your page.
- 3 CyberSource responds with a URL to direct the customer to their online banking page to confirm their payment details.
- 4 When the payment is successful, the customer is directed to the success URL that you included in the sale service request.

When the payment is canceled, the customer is directed to the cancel URL that you included in the sale service request.

When the payment fails, the customer is directed to the failure URL that you included in the sale service request.

- 5 When a customer is directed to the success URL, you can request the check status service (**apCheckStatusService**) to retrieve the current status of the payment.

The transaction statuses are:

- Pending—the payment was initiated. Do not ship the goods.
- Abandoned—the customer did not complete the transaction within 15 minutes.
- Settled—the payment is complete. You can ship the goods.
- Failed—the payment failed. Look at the **reasonCode** field for a description.



**Note**

If the status returned is *pending*, wait 5 minutes before making the request. If there is no status change, wait 15 minutes before making another request. The payment status expires in 20 minutes.

---

- 6 If required, you can request the refund service (**apRefundService**) to refund a customer's payment only when the payment status is updated to *settled*.

Bancontact transactions support:

- Follow-on refunds of payments that have a status of *settled*.
- Partial refunds and multiple refunds.
- Refunds for the original payment amount and the addition of 25 EUR. If the original payment amount was 45 EUR, you can refund up to 70 EUR.

## Performing a Sale

---



**Important**

As part of the sale service request, CyberSource recommends including specific customer data fields for Export Compliance checks. If a customer's first name and last name or company name appears on any of the lists that are maintained by government agencies to support export controls, you receive information indicating that the transaction is declined.

---



**Note**

When **item-level** fields are included in the request, you must also include the **item\_#\_totalAmount** field.

---

### To perform a sale:

---

- Step 1** Set the **apSaleService\_run** field to `true`.
- Step 2** Include the following **required** fields in the request:
- `merchantID`
  - `merchantReferenceCode`
  - `invoiceHeader_merchantDescriptor`
  - `apPaymentType`—set to `MCH`.
  - `apSaleService_cancelURL`
  - `apSaleService_successURL`
  - `apSaleService_failureURL`
  - `purchaseTotals_currency`
  - `purchaseTotals_grandTotalAmount`

**Step 3** Include any of the following **optional** fields in the request:

- billTo\_city
- billTo\_company
- billTo\_country
- billTo\_email
- billTo\_firstName
- billTo\_lastName
- billTo\_state
- billTo\_street1

For detailed descriptions of all request and reply fields, see [Appendix A, "API Fields,"](#) on page 72. For a sale example, see [Example 1, "Sale Request,"](#) on page 18.

---

## Checking a Payment Status

---

**To check the status of a payment:**

---

**Step 1** Set the `apCheckStatusService_run` field to `true`.

**Step 2** Include the following required fields in the request:

- `apCheckStatusService_checkStatusRequestID`—include the value of the **requestID** field that was returned in the sale service response.
- `apPaymentType`—set to `MCH`.
- `merchantID`
- `merchantReferenceCode`



To verify the status of a refund, include the value of the **requestID** field that was returned in the refund service response.

---

For detailed descriptions of all request and reply fields, see [Appendix A, "API Fields,"](#) on page 72. For a check status example, see [Example 3, "Check Status Request,"](#) on page 19.

---



## Refunding a Payment

---

### To refund a payment:

---

**Step 1** Set the `apRefundService_run` field to `true`.

**Step 2** Include the following fields in the request:

- `apPaymentType`—set to `MCH`.
- `apRefundService_refundRequestID`—include the value of the **requestID** field that was returned in the sale service response.
- `merchantID`
- `merchantReferenceCode`
- `purchaseTotals_currency`
- `purchaseTotals_grandTotalAmount`

For detailed descriptions of all request and reply fields, see [Appendix A, "API Fields,"](#) on page 72. For a refund example, see [Example 5, "Refund Request,"](#) on page 20.

---

## Testing

---

For test transactions, send requests to the test server (CAS):

<https://ics2wstesta.ic3.com/commerce/1.x/transactionProcessor>

### To test a Bancontact transaction:

---

**Step 1** Request the Sale service (see ["Performing a Sale,"](#) page 15).

**Step 2** Redirect to the URL you receive in the **merchantURL** field returned in the Sale service response.

**Step 3** Select a payment status:

- a** Paid—the payment status updates to settled. You are directed to the success URL that you included in the Sale service request.
- b** Canceled—the payment status updates to abandoned. You are directed to the cancel URL that you included in the Sale service request.

**Step 4** Click **Further to your webshop**.

**Step 5** If you selected Paid in [Step 3](#), you are directed to the success URL and can request the check status service or the refund service.

If you selected Canceled in [Step 3](#), you are directed to the cancel URL and can request the check status service to view the latest status of the payment.

---

## Name-Value Pair Examples

---

### Sale Service

#### Example 1 Sale Request

---

```
apSaleService_run=true
merchantID=mid12345
merchantReferenceCode=refnum1234
billTo_firstName=John
billTo_lastName=Smith
billTo_street1=1 The Street
billTo_city=Reading
billTo_state=Berkshire
billTo_postalCode=RG26DH
billTo_country=GB
billTo_email=null@example.com
billTo_language=en-GB
purchaseTotals_grandTotalAmount=100.99
purchaseTotals_currency=EUR
apPaymentType=MCH
invoiceHeader_merchantDescriptor=World largest online store
apSaleService_cancelURL=https://www.redirect.url.html?action=cancel
apSaleService_successURL=https://www.redirect.url.html?action=success
apSaleService_failureURL=https://www.redirect.url.html?action=failure
apSaleService_paymentOptionID=00001
```

---

**Example 2 Sale Reply**


---

```

merchantReferenceCode=refnum1234
requestID=4703329437356002101200
decision=ACCEPT
reasonCode=100
requestToken=Ahjn7wSR/e+dp++Ut4+hHjueHZlctWbBgxctGjNP95AQ40tIJYafsJMMm3/2Lid9AAA0Bic
purchaseTotals_currency=EUR
apSaleReply_reasonCode=100
apSaleReply_paymentStatus=pending
apSaleReply_responseCode=00001
apSaleReply_merchantURL=https://merchant.redirect.com/url.do?param_
utf=%27%22%3C%3E%20%E6%B8%AC%E8%A9%A6%E6%B8%AC&param_
url=https%3A%2F%2Fwww.abc.com&param_special=+@#%~_
&sign=fdaaldf42b6260a10e2e3f1c0fc87862
apSaleReply_reconciliationID=sale9530019443

```

---

## Check Status Service

**Example 3 Check Status Request**


---

```

apCheckStatusService_run=true
merchantID=mid12345
merchantReferenceCode=refnum1234
apPaymentType=MCH
apCheckStatusService_checkStatusRequestID=4703331268466002201200

```

---

**Example 4 Check Status Reply**


---

```

merchantReferenceCode=refnum1234
requestID=4703331331366002301200
decision=ACCEPT
reasonCode=100
requestToken=AhjnrwSR/e+rHR63PjIhHiKkNkwYMGTRP95AQ41NIJYagMJMMm3/2LidVC6CR/
e+qqrSe6xTgAAAFhRi
apCheckStatusReply_reasonCode=100
apCheckStatusReply_reconciliationID=TC200024
apCheckStatusReply_paymentStatus=settled
apCheckStatusReply_processorResponse=000004

```

---

## Refund Service

### Example 5 Refund Request

---

```
apRefundService_run=true
merchantID=mid12345
merchantReferenceCode=refnum1234
purchaseTotals_currency=EUR
purchaseTotals_grandTotalAmount=100.99
apPaymentType=MCH
apRefundService_refundRequestID=4703332484226002501200
```

---

### Example 6 Refund Reply

---

```
merchantReferenceCode=refnum1234
requestID=4703332594746002601200
decision=ACCEPT
reasonCode=100
requestToken=AhjnrwSR/e+0FzHfgunhHkuWXN13ZHLVmwYMXLVgzasU/3kBDjX0gfKN2EmGTb/
7FxOqhdbI/vfZpxR+aL5QAAAAjAE5
purchaseTotals_currency=EUR
apRefundReply_reasonCode=100
apRefundReply_status=refunded
apRefundReply_processorResponse=00006
apRefundReply_amount=189.99
apRefundReply_dateTime=2016-08-04T17:54:32Z
apRefundReply_reconciliationID=refund953001950351
apRefundReply_paymentStatus=completed
```

---

## XML Examples

---

### Sale Service

#### Example 7 Sale Request

---

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.126">
  <merchantID>mid12345</merchantID>
  <merchantReferenceCode>refnum1234</merchantReferenceCode>
  <invoiceHeader>
    <merchantDescriptor>Online Store</merchantDescriptor>
  </invoiceHeader>
  <billTo>
    <firstName>John</firstName>
    <lastName>Smith</lastName>
    <street1>1 The Street</street1>
    <city>Reading</city>
    <state>Berkshire</state>
    <postalCode>RG26DH</postalCode>
    <country>GB</country>
    <email>>null@example.com</email>
    <language>en-GB</language>
  </billTo>
  <purchaseTotals>
    <currency>EUR</currency>
    <grandTotalAmount>20.00</grandTotalAmount>
  </purchaseTotals>
  <apPaymentType>MCH</apPaymentType>
  <apSaleService run="true">
    <cancelURL>https://www.redirect.url.html?action=cancel</cancelURL>
    <failureURL>https://www.redirect.url.html?action=failure</failureURL>
    <successURL>https://www.redirect.url.html?action=success</successURL>
  </apSaleService>
</requestMessage>
```

---

**Example 8 Sale Reply**


---

```

<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.126">
  <c:merchantReferenceCode>refnum1234</c:merchantReferenceCode>
  <c:requestID>4598061680576069801200</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:requestToken>Ahjn7wSR8octWbBgxctGjNP93fNRq9IJYafr58Mm3/2LidVC5Bmr</
  c:requestToken>
  <c:purchaseTotals>
    <c:currency>EUR</c:currency>
  </c:purchaseTotals>
  <c:apSaleReply>
    <c:reasonCode>100</c:reasonCode>
    <c:paymentStatus>pending</c:paymentStatus>
    <c:responseCode>00001</c:responseCode>
    <c:merchantURL>https://merchant.redirect.com/url.do?param_
    utf=%27%22%3C%3E%20%E6%B8%AC%E8%A9%A6%E6%B8%AC&amp;param_
    url=https%3A%2F%2Fwww.abc.com&amp;param_special=+@#%~_
    &amp;sign=fdaaldf42b6260a10e2e3f1c0fc87862</c:merchantURL>
    <c:reconciliationID>sale9530019443</c:reconciliationID>
  </c:apSaleReply>
</c:replyMessage>

```

---

## Check Status Service

### Example 9 Check Status Request

---

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.126">
  <merchantID>mid12345</merchantID>
  <merchantReferenceCode>refnum1234</merchantReferenceCode>
  <apPaymentType>MCH</apPaymentType>
  <apCheckStatusService run="true">
    <checkStatusRequestID>4598061547766069401200</checkStatusRequestID>
  </apCheckStatusService>
</requestMessage>
```

---

### Example 10 Check Status Reply

---

```
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.126">
  <c:merchantReferenceCode>refnum1234</c:merchantReferenceCode>
  <c:requestID>4598061583426069501200</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:requestToken>AhjnrwSR5sHLhu2bJ/u75qNRpB+xcTqoXQSP1C7Cgee2EWcAnW01</c:requestToken>
  <c:apCheckStatusReply>
    <c:reasonCode>100</c:reasonCode>
    <c:reconciliationID>checkstatus098766</c:reconciliationID>
    <c:paymentStatus>settled</c:paymentStatus>
    <c:processorResponse>000004</c:processorResponse>
  </c:apCheckStatusReply>
</c:replyMessage>
```

---

## Refund Service

### Example 11 Refund Request

---

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.126">
  <merchantID>mid12345</merchantID>
  <merchantReferenceCode>refnum1234</merchantReferenceCode>
  <purchaseTotals>
    <currency>EUR</currency>
    <grandTotalAmount>20.00</grandTotalAmount>
  </purchaseTotals>
  <apPaymentType>MCH</apPaymentType>
  <apRefundService run="true">
    <refundRequestID>4598061766346070001200</refundRequestID>
  </apRefundService>
</requestMessage>
```

---

### Example 12 Refund Reply

---

```
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.126">
  <c:merchantReferenceCode>refnum1234</c:merchantReferenceCode>
  <c:requestID>4598061803266070101200</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:requestToken>AhjnrwSR8oXaIP1sUnmh38hoFxoqhdBI+ULtKTzQAAAA+QQw</c:requestToken>
  <c:purchaseTotals>
    <c:currency>EUR</c:currency>
  </c:purchaseTotals>
  <c:apRefundReply>
    <c:reasonCode>100</c:reasonCode>
    <c:status>refunded</c:status>
    <c:processorResponse>0000006</c:processorResponse>
    <c:amount>20.00</c:amount>
    <c:dateTime>2016-04-04T21:43:01Z</c:dateTime>
    <c:reconciliationID>refund953001950351</c:reconciliationID>
    <c:paymentStatus>completed</c:paymentStatus>
  </c:apRefundReply>
</c:replyMessage>
```

---



# Processing EPS Transactions

EPS is an e-commerce payment system in Austria that enables customers to pay for goods using direct online bank transfers from their bank accounts to your merchant account.



Refunds are currently not supported for EPS transactions.

## Transaction Flow

---

For a transaction flow diagram see:

[http://apps.cybersource.com/library/documentation/dev\\_guides/OnlineBankTransfers/EPS\\_Giropay.pdf](http://apps.cybersource.com/library/documentation/dev_guides/OnlineBankTransfers/EPS_Giropay.pdf)

- 1 Display the EPS payment method on your checkout page.



For logo guidelines see:

<https://www.girosolution.de/tools-support/werbemittel/>

- 2 This is an optional step. Add the EPS bank selection widget to your web page:

[http://api.girocheckout.de/en:tools:eps\\_bankstatus\\_widget](http://api.girocheckout.de/en:tools:eps_bankstatus_widget)

The widget adds auto complete functionality to a form input field and can be configured to return the customer's BIC to the browser.

- 3 The customer chooses the EPS payment method.
- 4 The customer selects the BIC using the widget (optional) or enters their BIC and clicks Pay.

- 5 Request the sale service (**apSaleService**) and include the corresponding BIC (**bankInfo\_swiftCode**) that the customer selected in [Step 4](#) and the success, cancel, and failure return URLs for that bank so that the bank can return the consumer to your page.
- 6 CyberSource responds with a merchant URL to direct the customer to their online banking page to confirm their payment details.

**Note**

If no BIC value is included in the sale service request, the merchant URL returned in the sale service response directs the customer to a page that includes the EPS bank selection widget.

---

- 7 When the payment is successful, the customer is directed to the success URL that you included in the sale service request.

When the payment is canceled, the customer is directed to the cancel URL that you included in the sale service request.

When the payment fails, the customer is directed to the failure URL that you included in the sale service request.

- 8 When a customer is directed to the success URL, you can request the check status service (**apCheckStatusService**) to retrieve the current status of the payment.

The transaction statuses are:

- Pending—the payment was initiated. Do not ship the goods.
- Abandoned—the customer did not complete the transaction within 15 minutes.
- Settled—the payment is complete. You can ship the goods.
- Failed—the payment failed. See the **reasonCode** field for a description.

**Note**

If the status returned is *pending*, wait 5 minutes before making the request. If there is no status change, wait 15 minutes before making another request. After 15 minutes the status will be *settled*, *abandoned*, or *failed*.

---

## Performing a Sale

---



**Important**

As part of the sale service request, CyberSource recommends including specific customer data fields for Export Compliance checks. If a customer's first name and last name or company name appears on any of the lists that are maintained by government agencies to support export controls, you receive information indicating that the transaction is declined.

---



**Note**

When **item-level** fields are included in the request, you must also include the **item\_#\_totalAmount** field.

---

### To perform a sale:

---

- Step 1** Set the **apSaleService\_run** field to `true`.
- Step 2** Include the following **required** fields in the request:
- `apPaymentType`—set to `EPS`.
  - `apSaleService_cancelURL`
  - `apSaleService_failureURL`
  - `apSaleService_successURL`
  - `bankInfo_swiftCode`
  - `invoiceHeader_merchantDescriptor`
  - `merchantID`
  - `merchantReferenceCode`
  - `purchaseTotals_currency`
  - `purchaseTotals_grandTotalAmount`
- Step 3** Include any of the following **optional** fields in the request:
- `billTo_city`
  - `billTo_company`
  - `billTo_country`
  - `billTo_email`
  - `billTo_firstName`
  - `billTo_lastName`
  - `billTo_state`
  - `billTo_street1`
  - `item_#_unitPrice`

For detailed descriptions of all request and reply fields, see [Appendix A, "API Fields," on page 72](#).

---

## Checking a Payment Status

---

### To check the status of a payment:

---

**Step 1** Set the `apCheckStatusService_run` field to `true`.

**Step 2** Include the following fields in the request:

- `apCheckStatusService_checkStatusRequestID`—include the value of the **requestID** field that was returned in the sale service response.



To check the status of a refund, include the value of the **requestID** field that was returned in the refund service response.

---

- `apPaymentType`—set to `EPS`.
- `merchantID`
- `merchantReferenceCode`

For detailed descriptions of all request and reply fields, see [Appendix A, "API Fields,"](#) on [page 72](#). For a Check Status example, see [Example 15, "Check Status Request,"](#) on [page 30](#).

---

## Testing

---

For test transactions, send requests to the test server (CAS):

<https://ics2wstesta.ic3.com/commerce/1.x/transactionProcessor>

### To test an EPS transaction:

---



EPS test data:

<http://api.girocheckout.de/en:girocheckout:eps:start>

---

**Step 1** Request the Sale service (see ["Performing a Sale,"](#) [page 27](#)).

**Step 2** Redirect to the URL that you receive in the **merchantURL** field returned in the Sale service response.

- Step 3** Enter **VBOEATWWAPO** for the BIC.
- Step 4** Click **Continue to pay**.
- Step 5** Click **Login**.
- Step 6** Review transaction details and click **Contract drawing**.
- Step 7** Select **mobileTAN**.
- Step 8** Click **collect and record**.
- Step 9** Click **OK** to confirm the transaction.
- Step 10** Click **back** to return to the merchant web site.

The payment status updates to settled and you are directed to the success URL that you included in the Sale service request.

---

## Name-Value Pair Examples

---

### Sale Service

#### Example 13 Sale Request

---

```

apSaleService_run=true
merchantID=mid12345
merchantReferenceCode=refnum1234
billTo_firstName=John
billTo_lastName=Smith
billTo_street1=1 The Street
billTo_city=Reading
billTo_state=Berkshire
billTo_postalCode=RG26DH
billTo_country=GB
billTo_email=null@example.com
billTo_language=en-GB
purchaseTotals_grandTotalAmount=1999.99
purchaseTotals_currency=EUR
apPaymentType=EPS
invoiceHeader_merchantDescriptor=World largest online store
apSaleService_cancelURL=https://www.redirect.url.html?action=cancel
apSaleService_successURL=https://www.redirect.url.html?action=success
apSaleService_failureURL=https://www.redirect.url.html?action=failure

```

---

**Example 14 Sale Reply**


---

```

merchantReferenceCode=refnum1234
requestID=4703329437356002101200
decision=ACCEPT
reasonCode=100
requestToken=Ahjn7wSR/e+dp++Ut4+hHjueHZlctWbBgxctGjNP95AQ40tIJYafsJMMm3/2LidVC6A
purchaseTotals_currency=EUR
apSaleReply_reasonCode=100
apSaleReply_paymentStatus=pending
apSaleReply_responseCode=00001
apSaleReply_merchantURL=https://merchant.redirect.com/url.do?param_
utf=%27%22%3C%3E%20%E6%B8%AC%E8%A9%A6%E6%B8%AC&param_
url=https%3A%2F%2Fwww.abc.com&param_special=+@#%~_
&sign=fdaaldf42b6260a10e2e3flc0fc87862
apSaleReply_reconciliationID=sale9530019443

```

---

## Check Status Service

**Example 15 Check Status Request**


---

```

apCheckStatusService_run=true
merchantID=mid12345
merchantReferenceCode=refnum1234
apPaymentType=EPS
apCheckStatusService_checkStatusRequestID=4703331268466002201200

```

---

**Example 16 Check Status Reply**


---

```

merchantReferenceCode=refnum1234
requestID=4703331331366002301200
decision=ACCEPT
reasonCode=100
requestToken=AhjnrwSR/e+rHR63PjIhHiKkNkwYMGTRP95AQ41NIJYagMJMMm3/2LidVC6CR/
e+qqrSe6xTgAAAFhRi
apCheckStatusReply_reasonCode=100
apCheckStatusReply_reconciliationID=TC200024
apCheckStatusReply_paymentStatus=authorized
apCheckStatusReply_processorResponse=000004

```

---

## XML Examples

---

### Sale Service

#### Example 17 Sale Request

---

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.126">
  <merchantID>mid12345</merchantID>
  <merchantReferenceCode>refnum1234</merchantReferenceCode>
  <invoiceHeader>
    <merchantDescriptor>Online Store</merchantDescriptor>
  </invoiceHeader>
  <billTo>
    <firstName>John</firstName>
    <lastName>Smith</lastName>
    <street1>1 The Street</street1>
    <city>Reading</city>
    <state>Berkshire</state>
    <postalCode>RG26DH</postalCode>
    <country>GB</country>
    <email>>null@example.com</email>
    <language>en-GB</language>
  </billTo>
  <purchaseTotals>
    <currency>EUR</currency>
    <grandTotalAmount>20.00</grandTotalAmount>
  </purchaseTotals>
  <apPaymentType>EPS</apPaymentType>
  <apSaleService run="true">
    <cancelURL>https://www.redirect.url.html?action=cancel</cancelURL>
    <failureURL>https://www.redirect.url.html?action=failure</failureURL>
    <successURL>https://www.redirect.url.html?action=success</successURL>
  </apSaleService>
</requestMessage>
```

---

**Example 18 Sale Reply**


---

```

<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.126">
  <c:merchantReferenceCode>refnum1234</c:merchantReferenceCode>
  <c:requestID>4598061680576069801200</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:requestToken>Ahjn7wSR8octWbBgxctGjNP93fNRq9IJYafr58Mm3/2LidVC5Bmr</
  c:requestToken>
  <c:purchaseTotals>
    <c:currency>EUR</c:currency>
  </c:purchaseTotals>
  <c:apSaleReply>
    <c:reasonCode>100</c:reasonCode>
    <c:paymentStatus>pending</c:paymentStatus>
    <c:responseCode>00001</c:responseCode>
    <c:merchantURL>https://merchant.redirect.com/url.do?param_
    utf=%27%22%3C%3E%20%E6%B8%AC%E8%A9%A6%E6%B8%AC&param_
    url=https%3A%2F%2Fwww.abc.com&param_special=+@#%~_
    &sign=fdaaldf42b6260a10e2e3f1c0fc87862</c:merchantURL>
    <c:reconciliationID>sale9530019443</c:reconciliationID>
  </c:apSaleReply>
</c:replyMessage>

```

---



## Check Status Service

### Example 19 Check Status Request

---

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.126">
  <merchantID>mid12345</merchantID>
  <merchantReferenceCode>refnum1234</merchantReferenceCode>
  <apPaymentType>EPS</apPaymentType>
  <apCheckStatusService run="true">
    <checkStatusRequestID>4598061547766069401200</checkStatusRequestID>
  </apCheckStatusService>
</requestMessage>
```

---

### Example 20 Check Status Reply

---

```
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.126">
  <c:merchantReferenceCode>refnum1234</c:merchantReferenceCode>
  <c:requestID>4598061583426069501200</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:requestToken>AhjnrwSR5sHLhu2bJ/u75qNRpB+xcTqoXQSP1C7Cgee2EWcAnW01</c:requestToken>
  <c:apCheckStatusReply>
    <c:reasonCode>100</c:reasonCode>
    <c:reconciliationID>checkstatus098766</c:reconciliationID>
    <c:paymentStatus>settled</c:paymentStatus>
    <c:processorResponse>000004</c:processorResponse>
  </c:apCheckStatusReply>
</c:replyMessage>
```

---

# Processing giro pay Transactions

*giro pay* is an e-commerce payment system in Germany that enables customers to pay for goods using direct online bank transfers from their bank accounts to your merchant account.



Refunds are currently not supported for giro pay transactions.

## Transaction Flow

---

For a transaction flow diagram see:

[http://apps.cybersource.com/library/documentation/dev\\_guides/OnlineBankTransfers/EPS\\_Giro pay.pdf](http://apps.cybersource.com/library/documentation/dev_guides/OnlineBankTransfers/EPS_Giro pay.pdf)

- 1 Display the giro pay payment method on your checkout page.



Logo standards:

<https://www.giro pay.de/haendler/tool-center/werbemittel-giro pay/>

- 2 This is an optional step. Add the giro pay bank selection widget to your web site:

[http://api.giro checkout.de/en:tools:bankstatus\\_widget](http://api.giro checkout.de/en:tools:bankstatus_widget)

The widget adds auto complete functionality to a form input field and can be configured to return the customer's BIC to the browser. You can also configure the widget to display all banks or only the banks that support giro pay.

- 3 The customer chooses the giro pay payment method.
- 4 The customer selects the BIC using the widget (optional) or enters their BIC and clicks Pay.

- 5 Request the sale service (**apSaleService**) and include the corresponding BIC (**bankInfo\_swiftCode**) that the customer selected in [Step 4](#) and the success, cancel, and failure return URLs for that bank so that the bank can return the consumer to your page.
- 6 CyberSource responds with a merchant URL to direct the customer to their online banking page to confirm their payment details

**Note**

If no BIC value is included in the sale service request, the merchant URL returned in the sale service response directs the customer to a page that includes the giropay bank selection widget.

---

- 7 When the payment is successful, the customer is directed to the success URL that you included in the sale service request.

When the payment is canceled, the customer is directed to the cancel URL that you included in the sale service request.

When the payment fails, the customer is directed to the failure URL that you included in the sale service request.

- 8 When a customer is directed to the success URL, you can request the check status service (**apCheckStatusService**) to retrieve the current status of the payment.

The transaction statuses are:

- Pending—the payment was initiated. Do not ship the goods.
- Abandoned—the customer did not complete the transaction within the 15 minutes.
- Settled—the payment is complete. You can ship the goods.
- Failed—the payment failed. Look at the **reasonCode** field for a description.

**Note**

If the status returned is *pending*, wait 5 minutes before making the request. If there is no status change, wait 15 minutes before making another request. After 15 minutes, the status will be *settled*, *abandoned*, or *failed*.

---

## Performing a Sale

---



### Important

As part of the sale service request, CyberSource recommends including specific customer data fields for Export Compliance checks. If a customer's first name and last name or company name appears on any of the lists that are maintained by government agencies to support export controls, you receive information indicating that the transaction is declined.



### Note

When **item-level** fields are included in the request, you must also include the **item\_#\_totalAmount** field.

### To perform a sale:

---

- Step 1** Set the **apSaleService\_run** field to `true`.
- Step 2** Include the following fields in the request:
- `apPaymentType`—set to `GPY`.
  - `apSaleService_cancelURL`
  - `apSaleService_failureURL`
  - `apSaleService_successURL`
  - `bankInfo_swiftCode`
  - `merchantID`
  - `merchantReferenceCode`
  - `invoiceHeader_merchantDescriptor`
  - `purchaseTotals_currency`
  - `purchaseTotals_grandTotalAmount`
- Step 3** Include any of the following **optional** fields in the request:
- `billTo_city`
  - `billTo_company`
  - `billTo_country`
  - `billTo_email`
  - `billTo_firstName`
  - `billTo_lastName`
  - `billTo_state`
  - `billTo_street1`
  - `item_#_unitPrice`

For detailed descriptions of all request and reply fields, see [Appendix A, "API Fields," on page 72](#).

## Checking a Payment Status

---

### To check the status of a payment:

---

- Step 1** Set the `apCheckStatusService_run` field to `true`.
- Step 2** Include the following fields in the request:
- `merchantID`
  - `merchantReferenceCode`
  - `apPaymentType`—set to `GPY`.
  - `apCheckStatusService_checkStatusRequestID`—include the value of the **requestID** field that was returned in the sale service response.



To check the status of a refund, include the value of the **requestID** field that was returned in the refund service response.

---

For detailed descriptions of all request and reply fields, see [Appendix A, "API Fields,"](#) on page 72.

---

## Testing

---

For test transactions, send requests to the test server (CAS):

<https://ics2wstesta.ic3.com/commerce/1.x/transactionProcessor>

### To test a giropay transaction:

---



giropay test data:

<http://api.girocheckout.de/en:girocheckout:giropay:start>

---

- Step 1** Request the Sale service (see "Performing a Sale," page 36).
- Step 2** Redirect to the URL that you receive in the **merchantURL** field returned in the Sale service response.
- Step 3** Enter **TESTDETT421** for the **BIC**.
- Step 4** Click **Continue to pay**.

- Step 5** Enter `sepatest1` for the bank account/login.
- Step 6** Enter any five-digit PIN.
- Step 7** Click **Secure login**.
- Step 8** Review the details and click **Continue**.
- Step 9** Enter any six-digit TAN.
- Step 10** Click **Pay now**.

The payment status updates to settled and you are directed to the success URL that you included in the Sale service request.

---

## Name-Value Pair Examples

---

### Sale Service

#### Example 21 Sale Request

---

```

apSaleService_run=true
merchantID=mid12345
merchantReferenceCode=refnum1234
billTo_firstName=John
billTo_lastName=Smith
billTo_street1=1 The Street
billTo_city=Reading
billTo_state=Berkshire
billTo_postalCode=RG26DH
billTo_country=GB
billTo_email=null@example.com
billTo_language=en-GB
purchaseTotals_grandTotalAmount=1999.99
purchaseTotals_currency=EUR
apPaymentType=GPY
invoiceHeader_merchantDescriptor=World largest online store
apSaleService_cancelURL=https://www.redirect.url.html?action=cancel
apSaleService_successURL=https://www.redirect.url.html?action=success
apSaleService_failureURL=https://www.redirect.url.html?action=failure

```

---

**Example 22 Sale Reply**


---

```

merchantReferenceCode=refnum1234
requestID=4703329437356002101200
decision=ACCEPT
reasonCode=100
requestToken=Ahjn7wSR/e+dp++Ut4+hHjueHZlctWbBgxctGjNP95AQ40tIJYafsJMMm3/2LidVC6A
purchaseTotals_currency=EUR
apSaleReply_reasonCode=100
apSaleReply_paymentStatus=pending
apSaleReply_responseCode=00001
apSaleReply_merchantURL=https://merchant.redirect.com/url.do?param_
utf=%27%22%3C%3E%20%E6%B8%AC%E8%A9%A6%E6%B8%AC&param_
url=https%3A%2F%2Fwww.abc.com&param_special=+@#%~_
&sign=fdaaldf42b6260a10e2e3flc0fc87862
apSaleReply_reconciliationID=sale9530019443

```

---

## Check Status Service

**Example 23 Check Status Request**


---

```

apCheckStatusService_run=true
merchantID=mid12345
merchantReferenceCode=refnum1234
apPaymentType=GPY
apCheckStatusService_checkStatusRequestID=4703331268466002201200

```

---

**Example 24 Check Status Reply**


---

```

merchantReferenceCode=refnum1234
requestID=4703331331366002301200
decision=ACCEPT
reasonCode=100
requestToken=AhjnrwSR/e+rHR63PjIhHiKkNkwYMGTRP95AQ41NIJYagMJMMm3/2LidVC6CR/
e+qqrSe6xTgAAAFhRi
apCheckStatusReply_reasonCode=100
apCheckStatusReply_reconciliationID=TC200024
apCheckStatusReply_paymentStatus=authorized
apCheckStatusReply_processorResponse=000004

```

---

## XML Examples

---

### Sale Service

#### Example 25 Sale Request

---

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.126">
  <merchantID>mid12345</merchantID>
  <merchantReferenceCode>refnum1234</merchantReferenceCode>
  <invoiceHeader>
    <merchantDescriptor>Online Store</merchantDescriptor>
  </invoiceHeader>
  <billTo>
    <firstName>John</firstName>
    <lastName>Smith</lastName>
    <street1>1 The Street</street1>
    <city>Reading</city>
    <state>Berkshire</state>
    <postalCode>RG26DH</postalCode>
    <country>GB</country>
    <email>>null@example.com</email>
    <language>en-GB</language>
  </billTo>
  <purchaseTotals>
    <currency>EUR</currency>
    <grandTotalAmount>20.00</grandTotalAmount>
  </purchaseTotals>
  <apPaymentType>GPY</apPaymentType>
  <apSaleService run="true">
    <cancelURL>https://www.redirect.url.html?action=cancel</cancelURL>
    <failureURL>https://www.redirect.url.html?action=failure</failureURL>
    <successURL>https://www.redirect.url.html?action=success</successURL>
  </apSaleService>
</requestMessage>
```

---



**Example 26 Sale Reply**


---

```

<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.126">
  <c:merchantReferenceCode>refnum1234</c:merchantReferenceCode>
  <c:requestID>4598061680576069801200</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:requestToken>Ahjn7wSR8octWbBgxctGjNP93fNRq9IJYafr58Mm3/2LidVC5Bmr</
  c:requestToken>
  <c:purchaseTotals>
    <c:currency>EUR</c:currency>
  </c:purchaseTotals>
  <c:apSaleReply>
    <c:reasonCode>100</c:reasonCode>
    <c:paymentStatus>pending</c:paymentStatus>
    <c:responseCode>00001</c:responseCode>
    <c:merchantURL>https://merchant.redirect.com/url.do?param_
    utf=%27%22%3C%3E%20%E6%B8%AC%E8%A9%A6%E6%B8%AC&param_
    url=https%3A%2F%2Fwww.abc.com&param_special=+@#%~_
    &sign=fdaaldf42b6260a10e2e3f1c0fc87862</c:merchantURL>
    <c:reconciliationID>sale9530019443</c:reconciliationID>
  </c:apSaleReply>
</c:replyMessage>

```

---

## Check Status Service

### Example 27 Check Status Request

---

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.126">
  <merchantID>mid12345</merchantID>
  <merchantReferenceCode>refnum1234</merchantReferenceCode>
  <apPaymentType>GPY</apPaymentType>
  <apCheckStatusService run="true">
    <checkStatusRequestID>4598061547766069401200</checkStatusRequestID>
  </apCheckStatusService>
</requestMessage>
```

---

### Example 28 Check Status Reply

---

```
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.126">
  <c:merchantReferenceCode>refnum1234</c:merchantReferenceCode>
  <c:requestID>4598061583426069501200</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:requestToken>AhjnrwSR5sHLhu2bJ/u75qNRpB+xcTqoXQSP1C7Cgee2EWcAnW01</c:requestToken>
  <c:apCheckStatusReply>
    <c:reasonCode>100</c:reasonCode>
    <c:reconciliationID>checkstatus098766</c:reconciliationID>
    <c:paymentStatus>settled</c:paymentStatus>
    <c:processorResponse>000004</c:processorResponse>
  </c:apCheckStatusReply>
</c:replyMessage>
```

---

# Processing iDEAL Transactions

*iDEAL* is an e-commerce payment system in the Netherlands that enables customers to pay for goods using direct online bank transfers from their bank accounts to your merchant account.

These banks support the iDEAL payment method:

- ABN AMRO
- ASN Bank
- Bunq
- ING Bank
- Knab
- Rabobank
- RegioBank
- SNS Bank
- Triodos Bank
- Van Lanschot



CyberSource recommends that you make the iDEAL payment method available to Belgian customers. A large number of Dutch nationals or Dutch bank account holders resides in Belgium, and they prefer to use their Dutch accounts.

---

## Transaction Flow

---



**Note**

An iDEAL transaction flow can be viewed here:

<https://www.ideal.nl/demo/>

---

For a complete transaction flow diagram see:

[http://apps.cybersource.com/library/documentation/dev\\_guides/OnlineBankTransfers/iDEAL.pdf](http://apps.cybersource.com/library/documentation/dev_guides/OnlineBankTransfers/iDEAL.pdf)

- 1 Display the **iDEAL** payment method on your checkout page.



**Note**

You can display your logo (640 w x 150 h) and a wallpaper (1920 w x 1080 h) as part of the payment flow. If the bank option ID is not included in the sale service request, the logo and wallpaper are displayed.

---



**Important**

Supported browsers are Google Chrome, Internet Explorer, Safari, and Firefox.

---

- 2 Request the options service (**apOptionsService**) once a day, at any time, to retrieve the list of bank names. Each name represents the swift code associated with the bank and is prefixed with **ideal**, for example: `ideal_BUNQNL2A`.



**Important**

CyberSource recommends caching the information and displaying the bank name to the customer in a drop-down menu on your checkout page.

---

- 3 The customer selects the iDEAL payment method and chooses their bank from the list of options that you have cached.

A list of all available banks can be displayed in a drop-down menu in any order you choose. CyberSource recommends displaying the following six bank names and their logos at the top of the list:

- ING Bank
- Rabobank
- ABN AMRO
- SNS Bank
- ASN Bank
- Triodos Bank

- 4 Request the sale service (**apSaleService**) and include the corresponding bank ID for the bank that the customer chooses. Also send the success, cancel, and failure return URLs so that the bank can return the customer to your page.
- 5 CyberSource responds with a merchant URL to direct the customer to their online banking page to confirm payment details.



If the option ID is not included in the sale service request, the customer is directed to a URL where they must choose their bank from the list displayed. The redirect URLs are not supported with iFrame integrations.

---

- 6 When the payment is successful, the customer is directed to the success URL that you included in the sale service request.

When the payment is canceled, the customer is directed to the cancel URL that you included in the sale service request.

When the payment fails, the customer is directed to the failure URL that you included in the sale service request.

- 7 When a customer is directed to the success URL, you can request the check status service (**apCheckStatusService**) to retrieve the current status of the payment.

The transaction statuses are:

- Pending—the payment was initiated. Do not ship the goods.
- Abandoned—the customer did not complete the transaction within 15 minutes.
- Settled—the payment is complete. You can ship the goods.
- Failed—the payment failed. Look at the **reasonCode** field for a description.



If the status returned is *pending*, wait 5 minutes before making the request. If there is no status change, wait 15 minutes before making another request again. After 15 minutes the status will be *settled*, *abandoned*, or *failed*.

---

- 8 When the payment status is updated to *settled*, you can request the refund service (**apRefundService**) to refund a customer's payment. Multiple partial refunds of the total amount are also supported.

iDEAL transactions support:

- Follow-on refunds of payments that have a status of *settled*.
- Partial refunds and multiple refunds.
- Refunds for the original payment amount and the addition of 25 EUR. If the original payment amount was 45 EUR, you can refund up to 70 EUR.

## Requesting Options

---

### To request options from the processor:

---

**Step 1** Set the `apOptionsService_run` field to `true`.

**Step 2** Include the following fields in the request:

- `apPaymentType`—set to `IDL`.
- `merchantID`
- `merchantReferenceCode`

For detailed descriptions of all request and reply fields, see [Appendix A, "API Fields,"](#) on page 72.

---

## Performing a Sale

---



**Important**

As part of the sale service request, CyberSource recommends including specific customer data fields for Export Compliance checks. If a customer's first name and last name or company name appears on any of the lists that are maintained by government agencies to support export controls, you receive information indicating that the transaction is declined. The fields that are not required in order to perform a sale service request are highlighted with an asterisk (\*).

---



**Note**

When **item-level** fields are included in the request, you must also include the **item\_#\_totalAmount** field.

---

### To perform a sale:

---

**Step 1** Set the `apSaleService_run` field to `true`.

**Step 2** Include the following fields in the request:

- `apPaymentType`—set to `IDL`.
- `apSaleService_cancelURL`
- `apSaleService_failureURL`

- apSaleService\_paymentOptionID—include the bank ID value returned in the Options service response.
- apSaleService\_successURL
- billTo\_city\*
- billTo\_company\*
- billTo\_country\*
- billTo\_email\*
- billTo\_firstName\*
- billTo\_lastName\*
- billTo\_state\*
- billTo\_street1\*
- invoiceHeader\_merchantDescriptor
- item\_#\_unitPrice\*
- merchantID
- merchantReferenceCode
- purchaseTotals\_currency
- purchaseTotals\_grandTotalAmount

For detailed descriptions of all request and reply fields, see [Appendix A, "API Fields,"](#) on [page 72](#).

---

## Checking a Payment Status

---

### To check the status of a payment:

---

**Step 1** Set the `apCheckStatusService_run` field to `true`.

**Step 2** Include the following fields in the request:

- `apCheckStatusService_checkStatusRequestID`—include the value of the **requestID** field that was returned in the sale service response.
- `apPaymentType`—set to `IDL`.
- `merchantID`
- `merchantReferenceCode`



To check the status of a refund, include the value of the **requestID** field that was returned in the refund service response.

---

For detailed descriptions of all request and reply fields, see [Appendix A, "API Fields,"](#) on [page 72](#).

---



## Refunding a Payment

---

### To refund a payment:

---

**Step 1** Set the `apRefundService_run` field to `true`.

**Step 2** Include the following fields in the request:

- `apPaymentType`—set to `IDL`.
- `apRefundService_refundRequestID`—include the value of the **requestID** field that was returned in the sale service response.
- `merchantID`
- `merchantReferenceCode`
- `purchaseTotals_currency`
- `purchaseTotals_grandTotalAmount`

For detailed descriptions of all request and reply fields, see [Appendix A, "API Fields,"](#) on [page 72](#).

---

## Testing

---

For test transactions, send requests to the test server (CAS):

<https://ics2wstesta.ic3.com/commerce/1.x/transactionProcessor>

### To test a transaction:

---

- Step 1** Request the Sale service (see "Requesting Options," page 46) and include **ideal-FVLBNL22** in the **apSaleService\_paymentOptionID** field.



If you include an empty value in the **apSaleService\_paymentOptionID** field, choose from the banks available on the bank selection page that is displayed.

---

- Step 2** Redirect to the URL you receive in the **merchantURL** field returned in the Sale service response.
- Step 3** Select a payment status:
- a** Paid—the payment status updates to settled. You are directed to the success URL that you included in the Sale service request.
  - b** Canceled—the payment status updates to abandoned. You are directed to the cancel URL that you included in the Sale service request.
  - c** Failed—the payment status updates to abandoned. You are directed to the failure URL that you included in the Sale service request.
  - d** No Final status—the payment status updates to pending.
- Step 4** Click **Further to your webshop**.
- Step 5** If you selected Paid in **Step 3**, you are directed to the success URL and can request the check status service or the refund service.

If you selected Canceled in **Step 3**, you are directed to the cancel URL and can request the check status service to view the latest status of the payment.

If you selected Failed in **Step 3**, you are directed to the failure URL and can request the check status service to view the latest status of the payment.

If you selected No Final Status in **Step 3**, you can request the check status service to view the latest status of the payment.

---

## Name-Value Pair Examples

---

### Options Service

#### Example 29 Options Request

---

```
apOptionsService_run=true  
merchantID=mid12345  
merchantReferenceCode=refnum1234  
apPaymentType=IDL
```

---

#### Example 30 Options Reply

---

```
merchantReferenceCode=refnum1234  
decision=ACCEPT  
reasonCode=100  
apOptionsReply_reasonCode=100  
apOptionsReply_responseCode=00000  
apOptionsReply_count=9  
apOptionsReply_offset=0  
apOptionsReply_totalCount=100  
apOptionsReply_option_0_id=9  
apOptionsReply_option_0_name=VanLanschot  
apOptionsReply_option_1_id=8  
apOptionsReply_option_1_name=Tridos Bank  
apOptionsReply_option_2_id=7  
apOptionsReply_option_2_name=SNS Bank  
apOptionsReply_option_3_id=6  
apOptionsReply_option_3_name=RegioBank  
apOptionsReply_option_4_id=5  
apOptionsReply_option_4_name=Rabobank  
apOptionsReply_option_5_id=4  
apOptionsReply_option_5_name=Knab  
apOptionsReply_option_6_id=3  
apOptionsReply_option_6_name=ING  
apOptionsReply_option_7_id=2  
apOptionsReply_option_7_name=ASN  
apOptionsReply_option_8_id=1  
apOptionsReply_option_8_name=ABN AMRO
```

---

## Sale Service

### Example 31 Sale Request

---

```

apSaleService_run=true
merchantID=mid12345
merchantReferenceCode=refnum1234
billTo_firstName=John
billTo_lastName=Smith
billTo_street1=1 The Street
billTo_city=Reading
billTo_state=Berkshire
billTo_postalCode=RG26DH
billTo_country=GB
billTo_email=null@example.com
billTo_language=en-GB
purchaseTotals_grandTotalAmount=100.99
purchaseTotals_currency=EUR
apPaymentType=IDL
invoiceHeader_merchantDescriptor=World largest online store
apSaleService_cancelURL=https://www.redirect.url.html?action=cancel
apSaleService_successURL=https://www.redirect.url.html?action=success
apSaleService_failureURL=https://www.redirect.url.html?action=failure
apSaleService_paymentOptionID=00001

```

---

### Example 32 Sale Reply

---

```

merchantReferenceCode=refnum1234
decision=ACCEPT
reasonCode=100
purchaseTotals_currency=EUR
apSaleReply_reasonCode=100
apSaleReply_paymentStatus=pending
apSaleReply_responseCode=00001
apSaleReply_merchantURL=https://merchant.redirect.com/url.do?param_
utf=%27%22%3C%3E%20%E6%B8%AC%E8%A9%A6%E6%B8%AC&param_
url=https%3A%2F%2Fwww.abc.com&param_special=+@#%~_
&sign=fdaald42b6260a10e2e3f1c0fc87862
apSaleReply_reconciliationID=sale9530019443
requestID=4703331268466002201200

```

---

## Check Status Service

### Example 33 Check Status Request

---

```
apCheckStatusService_run=true
merchantID=mid12345
merchantReferenceCode=refnum1234
apPaymentType=IDL
apCheckStatusService_checkStatusRequestID=4703331268466002201200
```

---

### Example 34 Check Status Reply

---

```
merchantReferenceCode=refnum1234
decision=ACCEPT
reasonCode=100
apCheckStatusReply_reasonCode=100
apCheckStatusReply_reconciliationID=TC200024
apCheckStatusReply_paymentStatus=authorized
apCheckStatusReply_processorResponse=000004
```

---

## Refund Service

### Example 35 Refund Request

---

```
apRefundService_run=true
merchantID=mid12345
merchantReferenceCode=refnum1234
purchaseTotals_currency=EUR
purchaseTotals_grandTotalAmount=100.99
apPaymentType=IDL
apRefundService_refundRequestID=4703332484226002501200
```

---

### Example 36 Refund Reply

---

```
merchantReferenceCode=refnum1234
decision=ACCEPT
reasonCode=100
purchaseTotals_currency=EUR
apRefundReply_reasonCode=100
apRefundReply_status=refunded
apRefundReply_processorResponse=00006
apRefundReply_amount=100.99
apRefundReply_reconciliationID=refund953001950351
apRefundReply_paymentStatus=completed
```

---

## XML Examples

---

### Options Service

#### Example 37 Options Request

---

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.126">
  <merchantID>mid12345</merchantID>
  <merchantReferenceCode>refnum1234</merchantReferenceCode>
  <apPaymentType>IDL</apPaymentType>
  <apOptionsService run="true" />
</requestMessage>
```

---

#### Example 38 Options Reply

---

```
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.126">
  <c:merchantReferenceCode>refnum1234</c:merchantReferenceCode>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:apOptionsReply>
    <c:reasonCode>100</c:reasonCode>
    <c:responseCode>00003</c:responseCode>
    <c:count>5</c:count>
    <c:offset>0</c:offset>
    <c:totalCount>100</c:totalCount>
    <c:option data="0">
      <c:id>ideal-ABNANL2A</c:id>
      <c:name>ABN AMRO</c:name>
    </c:option>
    <c:option data="1">
      <c:id>ideal-ASNBNL21</c:id>
      <c:name>ASN Bank</c:name>
    </c:option>
    <c:option data="2">
      <c:id>ideal-BUNQNL2A</c:id>
      <c:name>Bunq</c:name>
    </c:option>
    <c:option data="3">
      <c:id>ideal-INGBNL2A</c:id>
      <c:name>ING</c:name>
    </c:option>
    <c:option data="4">
      <c:id>ideal-KNABNL2H</c:id>
      <c:name>Knab</c:name>
    </c:option>
  </c:apOptionsReply>
</c:replyMessage>
```

---

## Sale Service

### Example 39 Sale Request

---

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.126">
  <merchantID>mid12345</merchantID>
  <merchantReferenceCode>refnum1234</merchantReferenceCode>
  <invoiceHeader>
    <merchantDescriptor>Online Store</merchantDescriptor>
  </invoiceHeader>
  <billTo>
    <firstName>John</firstName>
    <lastName>Smith</lastName>
    <street1>1 The Street</street1>
    <city>Reading</city>
    <state>Berkshire</state>
    <postalCode>RG26DH</postalCode>
    <country>GB</country>
    <email>null@example.com</email>
    <language>en-GB</language>
  </billTo>
  <purchaseTotals>
    <currency>EUR</currency>
    <grandTotalAmount>20.00</grandTotalAmount>
  </purchaseTotals>
  <apPaymentType>IDL</apPaymentType>
  <apSaleService run="true">
    <cancelURL>https://www.redirect.url.html?action=cancel</cancelURL>
    <successURL>https://www.redirect.url.html?action=success</successURL>
    <failureURL>https://www.redirect.url.html?action=failure</failureURL>
  </apSaleService>
</requestMessage>
```

---

**Example 40 Sale Reply**


---

```
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.126">
  <c:merchantReferenceCode>refnum1234</c:merchantReferenceCode>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:requestID>4598061547766069401200</c:requestID>
  <c:purchaseTotals>
    <c:currency>EUR</c:currency>
  </c:purchaseTotals>
  <c:apSaleReply>
    <c:reasonCode>100</c:reasonCode>
    <c:paymentStatus>pending</c:paymentStatus>
    <c:responseCode>00001</c:responseCode>
    <c:merchantURL>https://merchant.redirect.com/url.do?param_
utf=%27%22%3C%3E%20%E6%B8%AC%E8%A9%A6%E6%B8%AC&param_
url=https%3A%2F%2Fwww.abc.com&param_special=+@#%~_
&sign=fdaald42b6260a10e2e3f1c0fc87862</c:merchantURL>
    <c:reconciliationID>sale9530019443</c:reconciliationID>
  </c:apSaleReply>
</c:replyMessage>
```

---

## Check Status Service

**Example 41 Check Status Request**


---

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.126">
  <merchantID>mid12345</merchantID>
  <merchantReferenceCode>refnum1234</merchantReferenceCode>
  <apPaymentType>IDL</apPaymentType>
  <apCheckStatusService run="true">
    <checkStatusRequestID>4598061547766069401200</checkStatusRequestID>
  </apCheckStatusService>
</requestMessage>
```

---

**Example 42 Check Status Reply**


---

```
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.126">
  <c:merchantReferenceCode>refnum1234</c:merchantReferenceCode>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:apCheckStatusReply>
    <c:reasonCode>100</c:reasonCode>
    <c:reconciliationID>checkstatus098766</c:reconciliationID>
    <c:paymentStatus>settled</c:paymentStatus>
    <c:processorResponse>000004</c:processorResponse>
  </c:apCheckStatusReply>
</c:replyMessage>
```

---



## Refund Service

### Example 43 Refund Request

---

```

<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.126">
  <merchantID>mid12345</merchantID>
  <merchantReferenceCode>refnum1234</merchantReferenceCode>
  <purchaseTotals>
    <currency>EUR</currency>
    <grandTotalAmount>20.00</grandTotalAmount>
  </purchaseTotals>
  <apPaymentType>IDL</apPaymentType>
  <apRefundService run="true">
    <refundRequestID>4598061766346070001200</refundRequestID>
  </apRefundService>
</requestMessage>

```

---

### Example 44 Refund Reply

---

```

<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.126">
  <c:merchantReferenceCode>refnum1234</c:merchantReferenceCode>
  <c:requestID>0305782650000167905080</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:purchaseTotals>
    <c:currency>EUR</c:currency>
  </c:purchaseTotals>
  <c:apRefundReply>
    <c:reasonCode>100</c:reasonCode>
    <c:status>refunded</c:status>
    <c:processorResponse>0000006</c:processorResponse>
    <c:amount>20.00</c:amount>
    <c:dateTime>2016-04-04T21:43:01Z</c:dateTime>
    <c:reconciliationID>refund953001950351</c:reconciliationID>
    <c:paymentStatus>completed</c:paymentStatus>
    <c:responseCode>00006</c:responseCode>
  </c:apRefundReply>
</c:replyMessage>

```

---

# Processing Sofort Transactions

*Sofort* is a payment processor and an e-commerce payment system in many European regions that enables customers to pay for goods using direct online bank transfers from their bank accounts to your merchant account.

## Transaction Flow

---



Note

A Sofort transaction flow can be viewed here:

<https://www.sofort.com/eng-DE/kaeufer/su/online-zahlen-mit-sofort-ueberweisung/>

---

For a complete transaction flow diagram see:

[http://apps.cybersource.com/library/documentation/dev\\_guides/OnlineBankTransfers/Sofort.pdf](http://apps.cybersource.com/library/documentation/dev_guides/OnlineBankTransfers/Sofort.pdf)

- 1 Display the *Sofort* payment option on your checkout page.



Important

Supported browsers are Google Chrome, Internet Explorer, Safari, and Firefox.

---

- 2 Request the CyberSource Sale service (**apSaleService**) and include the success and cancel return URLs so that the bank can return the consumer to your page.
- 3 CyberSource responds with a URL to direct the customer to their online banking page to confirm payment details.
- 4 When the payment is successful, the customer is directed to the success URL that you included in the sale service request.

When the payment is canceled or fails, the customer is directed to the cancel URL that you included in the sale service request.

- 5 After a customer is directed to the success URL, you can request the check status service (**apCheckStatusService**) to retrieve the current status of the payment.

The transaction statuses are:

- Pending—the payment was initiated. Do not ship the goods.
- Authorized—the customer has authorized the payment. Do not ship the goods.
- Settled—if you have a CyberSource partner bank account, you receive this payment status to indicate that it was received by CyberSource and will be funded to your bank account. It can take up to a few days for a transaction to settle. You can now ship the goods to the customer.
- Failed—the payment failed. View the **reasonCode** field for a description.



When the customer confirms the payment, the transaction status updates from *pending* to *authorized*. See ["When to Check a Payment Status," page 61](#).

---

If required, you can request that the refund service (**apRefundService**) refund a customer's payment only when the payment status is updated to *settled*.

Sofort transactions support:

- Follow-on refunds of payments that have a status of *settled*.
- Partial refunds and multiple refunds.



The refund transaction status updates from *pending* (at the time of the request) to *refunded* (when processed by the bank). To check the status of the refund, CyberSource recommends requesting the check status service once every four hours.

---



Refunds are only available if using the API, not via the Business Center. Automated refunds are available if you are using the CyberSource settlement services account.

---

## Performing a Sale

---



### Note

When **item-level** fields are included in the request, you must also include the **item\_#\_totalAmount** field.

---

### To perform a sale:

---

**Step 1** Set the **apSaleService\_run** field to `true`.

**Step 2** Include the following **required** fields in the request:

- **apPaymentType**—set to `SOF`.
- **apSaleService\_cancelURL**
- **apSaleService\_successURL**



### Note

CyberSource recommends using different URLs for the success URL and the cancel URL. Request the check status service when the customer is directed to the success URL.

---

- **apSaleService\_transactionTimeout**
- **billTo\_city**
- **billTo\_country**
- **billTo\_firstName**
- **billTo\_lastName**
- **billTo\_street1**
- **invoiceHeader\_merchantDescriptor**

You can configure the Sofort payment flow so that the customer cannot edit the country, and must use a bank account within the country from which you request the Sale service. For this configuration, contact customer support for the collector model. If the configuration is set to lock the customer country, you must send the **billTo\_country** field for the country from which you want the customer to pay from.

CyberSource recommends that you include a combination of static and dynamic numbers for this field value because some banks may expect a different value within a particular time frame. If this field is not included in the request, the merchant reference number is used instead.

- merchantID
- merchantReferenceCode
- purchaseTotals\_currency
- purchaseTotals\_grandTotalAmount

For detailed descriptions of all request and reply fields, see [Appendix A, "API Fields,"](#) on page 72. For a sale example, see [Example 45, "Sale Request,"](#) on page 65.

---

## Checking a Payment Status

---

### When to Check a Payment Status

#### Pending Status

- 1 When the customer is directed to the success URL or the failure URL, request the check status service to confirm the status has updated from *pending* to *authorized*.
- 2 When the status is *pending*, wait 15 minutes and request the check status service again.
- 3 If the status remains at *pending*, continue to request the check status service every 15 minutes. The status will update to *authorized* or *abandoned*.

#### Authorized Status

- 1 When the customer is directed to the success URL or the failure URL, request the check status service to confirm the status has updated from *pending* to *authorized*.
- 2 When the status is *authorized*, request the check status service every four hours until the status updates to *settled*.
- 3 If the payment status does not update from *authorized* to *settled* within ten days, request the check status service once a day until the status updates to *settled*.



For most payments, the status changes from *authorized* to *settled* within 10 days. However, a small number of payments may take more than 10 days to be processed, therefore the status does not update to *settled* until 10 days or more. When the status is *settled*, ship the goods based on availability or refund the funds to the customer.

---

## How to Check a Payment Status

### To check the status of a payment:

---

**Step 1** Set the `apCheckStatusService_run` field to `true`.

**Step 2** Include the following fields in the request:

- `apCheckStatusService_checkStatusRequestID`



**Important**

To check the status of a payment, include the request ID value returned from the sale request in the `apCheckStatusService_checkStatusRequestID` field. To check the status of a refund, include the request ID value returned from the refund request in the `apCheckStatusService_checkStatusRequestID` field.

---

- `apPaymentType`—set to `SOF`.
- `merchantID`
- `merchantReferenceCode`

For detailed descriptions of all request and reply fields, see [Appendix A, "API Fields,"](#) on [page 72](#). For a check status example, see [Example 47, "Check Status Request,"](#) on [page 66](#).

---

## Refunding a Payment

---

For Sofort payments, the refund payment status returned is always *pending*. CyberSource recommends requesting the check status service to retrieve the status *refunded*.

**Important**

If the refund payment status returned is *pending*, CyberSource recommends to request the check status service every 4 hours until the status updates to *refunded*. It can take up to 48 hours for the status to update from *pending* to *refunded*—this indicates that the customer's account has been funded.

---

## How to Refund a Payment

### To refund a payment:

---

**Step 1** Set the `apRefundService_run` field to `true`.

**Step 2** Include the following fields in the request:

- `apPaymentType`—set to `SOF`.
- `apRefundService_refundRequestID`—set to the request ID value returned from the sale request.
- `merchantID`
- `merchantReferenceCode`
- `purchaseTotals_currency`
- `purchaseTotals_grandTotalAmount`

For detailed descriptions of all request and reply fields, see [Appendix A, "API Fields,"](#) on [page 72](#). For a refund example, see [Example 49, "Refund Request,"](#) on [page 67](#).

---

## Testing

---

For test transactions, send requests to the test server (CAS):

<https://ics2wstesta.ic3.com/commerce/1.x/transactionProcessor>

### To test a Sofort transaction:

---

- Step 1** Request the Sale service (see "Performing a Sale," page 60).
- Step 2** Redirect to the URL that you receive in the **merchantURL** field returned in the Sale service response.
- Step 3** Enter the test details for the specific bank:

**Table 2 Bank Test Details**

Country	Bank Test Details
Austria	00000
Belgium	Demo Bank
Germany	88888888
Italy	00000
Netherlands	Demo Bank
Spain	Demo Bank

- Step 4** Click **Next**.
- Step 5** Enter any 4-digit account number and any 3-digit PIN, and click **Next**.
- Step 6** Select an account and click **Next**.
- Step 7** Enter 12345 and click **Next**.

The payment status updates to *authorized* and you are directed to the success URL that you included in the Sale service request.

- Step 8** Request the Check Status service to retrieve the payment status *settled*.
-



## Testing Refunds

### To test a Sofort refund:

---

Follow the steps listed in "Testing," page 64 (the payment status will be *settled*) and do one of the following:

- Request the Refund service and include the request ID value returned in the Sale service response, along with the amount 2000.00. The payment status returned is *pending*.
  - Request the Refund service and include the request ID value returned in the Sale service response, along with the amount 4000.00. The payment status returned is *failed*.
  - Request the Refund service and include the request ID value returned in the Sale service response, along with any amount less than 2000.00. The payment status returned is *refunded*.
- 

## Name-Value Pair Examples

---

### Sale Service

#### Example 45 Sale Request

---

```

apSaleService_run=true
merchantID=mid12345
merchantReferenceCode=refnum1234
billTo_firstName=John
billTo_lastName=Smith
billTo_street1=1 The Street
billTo_city=Reading
billTo_state=Berkshire
billTo_postalCode=RG26DH
billTo_country=GB
billTo_email=null@example.com
billTo_language=en-GB
purchaseTotals_grandTotalAmount=100.99
purchaseTotals_currency=EUR
apPaymentType=SOF
invoiceHeader_merchantDescriptor=World largest online store
apSaleService_cancelURL=https://www.redirect.url.html?action=cancel
apSaleService_successURL=https://www.redirect.url.html?action=success
apSaleService_paymentOptionID=00001

```

---

**Example 46 Sale Reply**


---

```

merchantReferenceCode=refnum1234
requestID=4703329437356002101200
decision=ACCEPT
reasonCode=100
requestToken=Ahjn7wSR/e+dp++Ut4+hHjueHZlctWbBgxctGjNP95AQ40tIJYafsJMMm3/2LidVCAA0Bic
purchaseTotals_currency=EUR
apSaleReply_reasonCode=100
apSaleReply_paymentStatus=pending
apSaleReply_responseCode=00001
apSaleReply_merchantURL=https://merchant.redirect.com/url.do?param_
utf=%27%22%3C%3E%20%E6%B8%AC%E8%A9%A6%E6%B8%AC&param_
url=https%3A%2F%2Fwww.abc.com&param_special=+@#%~_
&sign=fdaaldf42b6260a10e2e3f1c0fc87862
apSaleReply_reconciliationID=sale9530019443

```

---

## Check Status Service

**Example 47 Check Status Request**


---

```

apCheckStatusService_run=true
merchantID=mid12345
merchantReferenceCode=refnum1234
apPaymentType=SOF
apCheckStatusService_checkStatusRequestID=4703331268466002201200

```

---

**Example 48 Check Status Reply**


---

```

merchantReferenceCode=refnum1234
requestID=4703331331366002301200
decision=ACCEPT
reasonCode=100
requestToken=AhjnrwSR/e+rHR63PjIhHiKkNkwYMGTRP95AQ41NIJYagMJMMm3/2LidVC6CR/
e+qqrSe6xTgAAAFhRi
apCheckStatusReply_reasonCode=100
apCheckStatusReply_reconciliationID=TC200024
apCheckStatusReply_paymentStatus=settled
apCheckStatusReply_processorResponse=000004

```

---

## Refund Service

### Example 49 Refund Request

---

```
apRefundService_run=true  
merchantID=mid12345  
merchantReferenceCode=refnum1234  
purchaseTotals_currency=EUR  
purchaseTotals_grandTotalAmount=100.99  
apPaymentType=SOF  
apRefundService_refundRequestID=4703332484226002501200
```

---

### Example 50 Refund Reply

---

```
merchantReferenceCode=refnum1234  
requestID=4703332594746002601200  
decision=ACCEPT  
reasonCode=100  
requestToken=AhjnrwSR/e+0FzHfgunhHkuWXN13ZHLVmwYMXLVgzasU/3kBDjX0gfKN2EmGTb/  
purchaseTotals_currency=EUR  
apRefundReply_reasonCode=100  
apRefundReply_status=refunded  
apRefundReply_processorResponse=00006  
apRefundReply_amount=100.99  
apRefundReply_dateTime=2016-08-04T17:54:32Z  
apRefundReply_reconciliationID=refund953001950351  
apRefundReply_paymentStatus=completed
```

---

## XML Examples

---

### Sale Service

#### Example 51 Sale Request

---

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.126">
  <merchantID>mid12345</merchantID>
  <merchantReferenceCode>refnum1234</merchantReferenceCode>
  <invoiceHeader>
    <merchantDescriptor>201708031027373</merchantDescriptor>
  </invoiceHeader>
  <billTo>
    <firstName>John</firstName>
    <lastName>Smith</lastName>
    <street1>1 The Street</street1>
    <city>Reading</city>
    <state>Berkshire</state>
    <postalCode>RG26DH</postalCode>
    <country>GB</country>
    <email>>null@example.com</email>
    <language>en-GB</language>
  </billTo>
  <purchaseTotals>
    <currency>EUR</currency>
    <grandTotalAmount>100.00</grandTotalAmount>
  </purchaseTotals>
  <apPaymentType>SOF</apPaymentType>
  <apSaleService run="true">
    <cancelURL>https://www.redirect.url.html?action=cancel</cancelURL>
    <successURL>https://www.redirect.url.html?action=success</successURL>
  </apSaleService>
</requestMessage>
```

---

**Example 52 Sale Reply**


---

```

<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.126">
  <c:merchantReferenceCode>refnum1234</c:merchantReferenceCode>
  <c:requestID>4598061680576069801200</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:requestToken>Ahjn7wSR8octWbBgxctGjNP93fNRq9IJYafR58Mm3/2LidVC5Bmr</
  c:requestToken>
  <c:purchaseTotals>
    <c:currency>EUR</c:currency>
  </c:purchaseTotals>
  <c:apSaleReply>
    <c:reasonCode>100</c:reasonCode>
    <c:paymentStatus>pending</c:paymentStatus>
    <c:responseCode>00001</c:responseCode>
    <c:merchantURL>https://merchant.redirect.com/url.do?param_
    utf=%27%22%3C%3E%20%E6%B8%AC%E8%A9%A6%E6%B8%AC&param_
    url=https%3A%2F%2Fwww.abc.com&param_special=+@#%~_
    &sign=fdaaldf42b6260a10e2e3f1c0fc87862</c:merchantURL>
    <c:reconciliationID>sale9530019443</c:reconciliationID>
  </c:apSaleReply>
</c:replyMessage>

```

---

## Check Status Service

### Example 53 Check Status Request

---

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.126">
  <merchantID>mid12345</merchantID>
  <merchantReferenceCode>refnum1234</merchantReferenceCode>
  <apPaymentType>SOF</apPaymentType>
  <apCheckStatusService run="true">
    <checkStatusRequestID>4598061547766069401200</checkStatusRequestID>
  </apCheckStatusService>
</requestMessage>
```

---

### Example 54 Check Status Reply

---

```
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.126">
  <c:merchantReferenceCode>refnum1234</c:merchantReferenceCode>
  <c:requestID>4598061583426069501200</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:requestToken>AhjnrwSR5sHLhu2bJ/u75qNRpB+xcTqoXQSP1C7Cgee2EWcAnW01</c:requestToken>
  <c:apCheckStatusReply>
    <c:reasonCode>100</c:reasonCode>
    <c:reconciliationID>checkstatus098766</c:reconciliationID>
    <c:paymentStatus>settled</c:paymentStatus>
    <c:processorResponse>000004</c:processorResponse>
  </c:apCheckStatusReply>
</c:replyMessage>
```

---

## Refund Service

### Example 55 Refund Request

---

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.126">
  <merchantID>mid12345</merchantID>
  <merchantReferenceCode>refnum1234</merchantReferenceCode>
  <purchaseTotals>
    <currency>EUR</currency>
    <grandTotalAmount>100.00</grandTotalAmount>
  </purchaseTotals>
  <apPaymentType>SOF</apPaymentType>
  <apRefundService run="true">
    <refundRequestID>4598061583426069501200</refundRequestID>
  </apRefundService>
</requestMessage>
```

---

### Example 56 Refund Reply

---

```
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.126">
  <c:merchantReferenceCode>refnum1234</c:merchantReferenceCode>
  <c:requestID>4598061583426069501200</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:requestToken>AhjnrwSR8oXaIP1sUnmh38hoFxoqhdBI+ULtKTzQAAAA+QQw</c:requestToken>
  <c:purchaseTotals>
    <c:currency>EUR</c:currency>
  </c:purchaseTotals>
  <c:apRefundReply>
    <c:reasonCode>100</c:reasonCode>
    <c:status>refunded</c:status>
    <c:processorResponse>0000006</c:processorResponse>
    <c:amount>100.00</c:amount>
    <c:dateTime>2016-04-04T21:43:01Z</c:dateTime>
    <c:reconciliationID>refund953001950351</c:reconciliationID>
    <c:paymentStatus>completed</c:paymentStatus>
  </c:apRefundReply>
</c:replyMessage>
```

---

# API Fields

## Formatting Restrictions

---

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

**Note**

The values of the **item\_#\_** fields must not contain carets (^) or colons (:) because these characters are reserved for use by the CyberSource services.

Values for request-level and item-level fields must not contain new lines or carriage returns. However, they can contain embedded spaces and any other printable characters. CyberSource removes all leading and trailing spaces.

---

## Data Type Definitions

---

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

**Table 3** Data Type Definitions

Data Type	Description
Integer	Whole number {..., -3, -2, -1, 0, 1, 2, 3, ...}
String	Sequence of letters, numbers, spaces, and special characters

---



## Numbered Elements

---

The CyberSource XML schema includes several numbered elements. You can include these complex elements more than once in a request. For example, if a customer order includes more than one item, you must include multiple `<item>` elements in your request. Each item is numbered, starting with 0. The XML schema uses an `id` attribute in the item's opening tag to indicate the number. For example:

```
<item id="0">
```

For the name-value pair field names, this tag is represented as **item\_0**. In this portion of the field name, the underscore before the number does not indicate hierarchy in the XML schema. The item fields are generically referred to as **item\_#\_<element name>** in the documentation.

Below is an example of the numbered `<item>` element and the corresponding name-value pair field names. If you are using SOAP, the client contains a corresponding `Item` class.

### Example 57 Numbered XML Schema Element Names and Name-Value Pair Field Names

XML Schema Element Names	Corresponding Name-Value Pair Field Names
<pre>&lt;item id="0"&gt;   &lt;unitPrice&gt;   &lt;quantity&gt; &lt;/item&gt;</pre>	<pre>item_0_unitPrice item_0_quantity</pre>
<pre>&lt;item id="1"&gt;   &lt;unitPrice&gt;   &lt;quantity&gt; &lt;/item&gt;</pre>	<pre>item_1_unitPrice item_1_quantity</pre>



#### Important

When a request is in XML format and includes an `<item>` element, the element must include an `id` attribute. For example: `<item id="0">`.

## Request Fields

See [Getting Started with CyberSource Advanced for the Simple Order API](#) for a description of how name-value pair names relate to their corresponding XML element names.

**Table 4 Request Fields**

Field Name	Description	Used By: Required (R) or Optional (O)	Data Type & Length
apCheckStatusService_ checkStatusRequestID	Request ID of the sale transaction for which you want to check the status. This value is returned in the Sale service response.	Check Status (R)	String (26)
apCheckStatusService_run	Whether to include <b>apCheckStatusService</b> in your request. Possible values: <ul style="list-style-type: none"> <li>■ <code>true</code>: Include the service in your request.</li> <li>■ <code>false</code> (default): Do not include the service in your request.</li> </ul>	Check Status (R)	String (5)
apOptionsService_limit	The maximum number of options to be retrieved from the processor and displayed to the consumer.  Possible values: 1-250  <b>Note</b> This field is valid only for iDEAL transactions.	Options (O)	String (3)
<p>1 You must include either <b>item#_unitPrice</b> or <b>purchaseTotals_grandTotalAmount</b> in your request. For information about items and grand totals, see <a href="#">Getting Started with CyberSource Advanced for the Simple Order API</a>.</p>			

Table 4 Request Fields (Continued)

Field Name	Description	Used By: Required (R) or Optional (O)	Data Type & Length
apOptionsService_offset	<p>Offset from the first item in the list of options received from the processor. If you want to display the options in multiple lists, this number represents the first option displayed in each list.</p> <p>Possible values: 0-9999 For example:</p> <pre>&lt;apOptionsService&gt;   &lt;limit&gt;10&lt;/limit&gt;   &lt;offset&gt;0&lt;/offset&gt; &lt;/apOptionsService&gt;</pre> <p>The above values display 10 options to the customer, starting with the first item on the list received from the processor.</p> <pre>&lt;apOptionsService&gt;   &lt;limit&gt;10&lt;/limit&gt;   &lt;offset&gt;10&lt;/offset&gt; &lt;/apOptionsService&gt;</pre> <p>The above values display 10 options to the customer, starting with the 11th item on the list received from the processor.</p> <p><b>Note</b> This field is valid only for iDEAL transactions.</p>	Options (O)	String (4)
apOptionsService_run	<p>Whether to include <b>apOptionService</b> in your request. Possible values:</p> <ul style="list-style-type: none"> <li>■ <code>true</code>: Include the service in your request.</li> <li>■ <code>false</code> (default): Do not include the service in your request.</li> </ul> <p><b>Note</b> This field is valid only for iDEAL transactions.</p>	Options (R)	String (5)

<sup>1</sup> You must include either **item\_#\_unitPrice** or **purchaseTotals\_grandTotalAmount** in your request. For information about items and grand totals, see *Getting Started with CyberSource Advanced for the Simple Order API*.

Table 4 Request Fields (Continued)

Field Name	Description	Used By: Required (R) or Optional (O)	Data Type & Length
apPaymentType	Identifier for the payment method: <ul style="list-style-type: none"> <li>■ EPS: EPS</li> <li>■ GPY: giropay</li> <li>■ IDL: iDEAL</li> <li>■ MCH: Bancontact</li> <li>■ SOF: Sofort</li> </ul>	Options (R) Sale (R) Check Status (R) Refund (R)	String (3)
apRefundService_refundRequestID	Request ID of the sale transaction for which you are requesting a refund.  Include the value of the <b>requestID</b> field that was returned in the sale service response.	Refund (R)	String (26)
apRefundService_run	Whether to include <b>apRefundService</b> in your request. Possible values: <ul style="list-style-type: none"> <li>■ <code>true</code>: Include the service in your request.</li> <li>■ <code>false</code> (default): Do not include the service in your request.</li> </ul>	Refund (R)	String (5)
apSaleService_cancelURL	URL the consumer is directed to after cancelling the payment. Make a GET request to this URL.	Sale (O)	String (255)
apSaleService_failureURL	URL the consumer is directed to after a payment fails. Make a GET request to this URL.  <b>Note</b> This field is valid only for iDEAL and Bancontact transactions.	Sale (O)	String (255)
apSaleService_paymentOptionID	Payment option ID name.  This is the bank's swift code. Include the option ID name returned in the Options service response.  <b>Example</b> ideal-BUNQNL2A  <b>Note</b> This field is valid only for iDEAL transactions.	Sale (O)	String (60)
apSaleService_run	Whether to include <b>apSaleService</b> in your request. Possible values: <ul style="list-style-type: none"> <li>■ <code>true</code>: Include the service in your request.</li> <li>■ <code>false</code> (default): Do not include the service in your request.</li> </ul>	Sale (R)	String (5)

<sup>1</sup> You must include either **item\_#\_unitPrice** or **purchaseTotals\_grandTotalAmount** in your request. For information about items and grand totals, see *Getting Started with CyberSource Advanced for the Simple Order API*.

Table 4 Request Fields (Continued)

Field Name	Description	Used By: Required (R) or Optional (O)	Data Type & Length
apSaleService_successURL	URL the consumer is directed to after completing the payment. Make a GET request to this URL.	Sale (O)	String (255)
apSaleService_transactionTimeout	The time-out limit in seconds for the transaction. The time-out limit starts when the customer is directed to the merchant URL that is included in the sale service response. The maximum value is 99999 (about 27 hours). When the transaction times out, the payment system changes the status to <i>abandoned</i> .  <b>Note</b> This field is valid only for Sofort transactions.  If the field is not included in the request, the default time-out is 6 hours. The minimum value is 120 seconds (2 minutes).	Sale (O)	Integer (5)
bankInfo_swiftCode	Bank's SWIFT code. Unique address of the bank. Also known as the Bank Identification Code (BIC). This value is retrieved using the bank selection widget or it is entered directly by the customer.  <b>Note</b> This field is valid only for giro pay and EPS transactions.	Sale (O)	String (20)
billTo_city	Credit card billing city.	Sale (O)	String (50)
billTo_company	Name of the customer's company.	Sale (O)	String (60)
billTo_country	Credit card billing country. Use the two-digit character <i>ISO Standard Country Codes</i> .	Sale (R)	String (2)
billTo_email	Customer's email address, including the full domain name.	Sale (O)	String (255)
billTo_firstName	Customer's first name. This name must be the same as the name on the card.	Sale (O)	String (60)
1 You must include either <b>item_#_unitPrice</b> or <b>purchaseTotals_grandTotalAmount</b> in your request. For information about items and grand totals, see <i>Getting Started with CyberSource Advanced for the Simple Order API</i> .			

Table 4 Request Fields (Continued)

Field Name	Description	Used By: Required (R) or Optional (O)	Data Type & Length
billTo_language	<p>Language code and country code used for the transaction.</p> <p>Format: [language code][dash][country code]</p> <p>Possible values:</p> <ul style="list-style-type: none"> <li>■ DE-BE: German-Belgium</li> <li>■ DE-DE (default): German-Germany</li> <li>■ DE-ES: German-Spain</li> <li>■ DE-IT: German-Italy</li> <li>■ EN-AT: English-Austria</li> <li>■ EN-BE: English-Belgium</li> <li>■ EN-DE: English-Germany</li> <li>■ EN-ES: English-Spain</li> <li>■ EN-IT: English-Italy</li> <li>■ EN-NL: English-Netherlands</li> <li>■ ES-ES: Spanish-Spain</li> <li>■ FR-BE: French-Belgium</li> <li>■ IT-IT: Italian-Italy</li> <li>■ NL-BE: Dutch-Belgium</li> <li>■ NL-NL: Dutch-Netherlands</li> </ul> <p><b>Note</b> This field is valid only for Sofort transactions.</p>	Sale (O)	String (5)
billTo_lastName	Customer's last name. This name must be the same as the name on the card.	Sale (O)	String (60)
billTo_state	Credit card billing state or province. Use the <a href="#">State, Province, and Territory Codes for the United States and Canada</a> .	Sale (O)	String (2)
billTo_street1	Credit card billing street address as it appears in the credit card issuer's records.	Sale (O)	String (60)
<p><sup>1</sup> You must include either <b>item_#_unitPrice</b> or <b>purchaseTotals_grandTotalAmount</b> in your request. For information about items and grand totals, see <i>Getting Started with CyberSource Advanced for the Simple Order API</i>.</p>			

Table 4 Request Fields (Continued)

Field Name	Description	Used By: Required (R) or Optional (O)	Data Type & Length
invoiceHeader_ merchantDescriptor	Merchant description that is displayed on the customer's statement. When you include more than one consecutive space, extra spaces are removed.	Sale (R)	For Bancontact and iDEAL:  String (35)  For Sofort, EPS and giropay:  String (27)
invoiceHeader_ merchantDescriptorCity	City for your business location. This value is displayed on the customer's statement.	Sale (O)	String (50)
invoiceHeader_ merchantDescriptorContact	Merchant contact information, such as a phone number, that is displayed on the customer's statement. When you include more than one consecutive space, extra spaces are removed.	Sale (O)	String (150)
invoiceHeader_ merchantDescriptorCountry	Country code for your business location. Use the standard <a href="#">ISO Standard Country Codes</a> . This value is displayed on the customer's statement.	Sale (O)	String (2)
invoiceHeader_ merchantDescriptorPostalCode	Postal code for your business location. This value is displayed on the customer's statement.	Sale (O)	String (10)
invoiceHeader_ merchantDescriptorState	State code or region code for your business location. This value is displayed on the customer's statement.	Sale (O)	String (50)
invoiceHeader_ merchantDescriptorStreet	Street address for your business location. If the street address is more than 38 characters, use meaningful abbreviations.	Sale (O)	String (60)
item_#_productCode	Type of product. This value is used to determine the category that the product is in: electronic, handling, physical, service, or shipping. The default value is <code>default</code> .  See <a href="#">"Numbered Elements," page 73</a> .	Sale (O)	String (255)
item_#_productName	This field is required when <b>item_#_productCode</b> is not <code>default</code> or one of the values related to shipping and/or handling.  See <a href="#">"Numbered Elements," page 73</a> .	Sale (O)	String (255)
<p>1 You must include either <b>item_#_unitPrice</b> or <b>purchaseTotals_grandTotalAmount</b> in your request. For information about items and grand totals, see <i>Getting Started with CyberSource Advanced for the Simple Order API</i>.</p>			

Table 4 Request Fields (Continued)

Field Name	Description	Used By: Required (R) or Optional (O)	Data Type & Length
item_#_productSKU	Identification code for the product.  See <a href="#">"Numbered Elements," page 73.</a>	Sale (O)	String (255)
item_#_quantity	The default is 1. This field is required when <b>item_#_productCode</b> is not default or one of the values related to shipping and/or handling.  See <a href="#">"Numbered Elements," page 73.</a>	Sale (O)	Integer (10)
item_#_taxAmount	Total tax to apply to the product. This value cannot be negative. The tax amount and the unit price must be in the same currency.  The tax amount field is additive. The following example uses a two-exponent currency such as USD:  1 You include the following items in your request: <code>item_0_unitPrice=10.00</code> <code>item_0_quantity=1</code> <code>item_0_taxAmount=0.80</code> <code>item_1_unitPrice=20.00</code> <code>item_1_quantity=1</code> <code>item_1_taxAmount=1.60</code>  2 The total amount authorized will be 32.40, not 30.00 with 2.40 of tax included.  See <a href="#">"Numbered Elements," page 73.</a>	Sale (O)	String (15)
item_#_totalAmount	Total amount for all items selected.  This field is required when item-level fields are included in the request.  See <a href="#">"Numbered Elements," page 73.</a>	Sale (See description)	String (15)
item_#_unitPrice	Per-item price of the product. This value cannot be negative. You can include a decimal point (.), but you cannot include any other special characters. CyberSource truncates the amount to the correct number of decimal places.  See <a href="#">"Numbered Elements," page 73.</a>	Sale (O) <sup>1</sup>	String (15)
merchantID	Your CyberSource merchant ID. Use the same merchant ID for evaluation, testing, and production.	Required for all service requests.	String (30)

<sup>1</sup> You must include either **item\_#\_unitPrice** or **purchaseTotals\_grandTotalAmount** in your request. For information about items and grand totals, see *Getting Started with CyberSource Advanced for the Simple Order API*.



Table 4 Request Fields (Continued)

Field Name	Description	Used By: Required (R) or Optional (O)	Data Type & Length
merchantReferenceCode	Merchant-generated order reference or tracking number. CyberSource recommends that you send a unique value for each transaction so that you can perform meaningful searches for the transaction.	Required for all service requests.	String (50)
purchaseTotals_currency	Currency used for the order. Use the three character <i>ISO Standard Currency Codes</i> .	Sale (R) Refund (R)	String (5)
purchaseTotals_grandTotalAmount	Grand total for the order. This value cannot be negative. You can include a decimal point (.), but you cannot include any other special characters. CyberSource truncates the amount to the correct number of decimal places.  <b>Important</b> You must include either <b>purchaseTotals_grandTotalAmount</b> or <b>item#_unitPrice</b> and <b>item#_taxAmount</b> in your request.	Sale (R) Refund (R)	Decimal (15)

1 You must include either **item#\_unitPrice** or **purchaseTotals\_grandTotalAmount** in your request. For information about items and grand totals, see *Getting Started with CyberSource Advanced for the Simple Order API*.

## Reply Fields

Table 5 Reply Fields

Field Name	Description	Returned By:	Data Type & Length
apCheckStatusReply_ paymentStatus	<p>Description of the payment status.</p> <p>Possible values:</p> <ul style="list-style-type: none"> <li>■ <b>PENDING</b>: Payment was initiated. Do not ship the goods.</li> <li>■ <b>SETTLED</b>: Payment is complete and the funds have been settled in your merchant account. Ship the goods.</li> <li>■ <b>FAILED</b>: Payment failed.</li> <li>■ <b>ABANDONED</b>: Customer did not confirm the payment within 15 minutes or the customer canceled the payment. This status is valid only for iDEAL, Bancontact, EPS, and giropay transactions.</li> <li>■ <b>AUTHORIZED</b>: Customer authorized the payment. This status is valid only for Sofort transactions.</li> </ul>	Check Status	String (15)
apCheckStatusReply_ reasonCode	Numeric value corresponding to the result of the check status service request. See " <a href="#">Reason Codes and Processor Response Codes</a> ," page 86.	Check Status	Integer (5)
apCheckStatusReply_ reconciliationID	Reconciliation ID that was used for the transaction.	Check Status	String (60)
apOptionsReply_ count	<p>The total number of options requested from the processor.</p> <p>Possible values: 1–250</p> <p><b>Note</b> This field is valid only for iDEAL transactions.</p>	Options	Numeric (3)
apOptionsReply_ offset	<p>Offset from the first item in the list of options received from the processor. If you want to display the options in multiple lists, this number represents the first option displayed in each list.</p> <p>Possible values: 0–9999</p> <p><b>Note</b> This field is valid only for iDEAL transactions.</p>	Options	Numeric (4)

Table 5 Reply Fields (Continued)

Field Name	Description	Returned By:	Data Type & Length
apOptionsReply_option_#_id	<p>Unique identifier associated with the option name.</p> <p><b>Example</b></p> <pre>apOptionsReply_option_0_ id=ideal-ABNANL2A  apOptionsReply_option_1_ id=ideal-ASNBNL21  apOptionsReply_option_2_ id=ideal-BUNQNL2A</pre> <p><b>Note</b> This field is valid only for iDEAL transactions.</p>	Options	String (34)
apOptionsReply_option_#_name	<p>Name associated with the option ID.</p> <p><b>Example</b></p> <pre>apOptionsReply_option_0_ name=ABN AMRO  apOptionsReply_option_1_ name=ASN Bank  apOptionsReply_option_2_ name=Bunq</pre> <p><b>Important</b> CyberSource recommends caching the information and displaying the bank name to the customer in a drop-down menu on your checkout page.</p> <p><b>Note</b> This field is valid only for iDEAL transactions.</p>	Options	String (60)
apOptionsReply_reasonCode	<p>Numeric value corresponding to the result of the options service request. See "<a href="#">Reason Codes and Processor Response Codes</a>," page 86.</p> <p><b>Note</b> This field is valid only for iDEAL transactions.</p>	Options	Integer (5)
apOptionsReply_responseCode	<p>Response code from the processor. Range of values: 00000 – 99999. See "<a href="#">Reason Codes and Processor Response Codes</a>," page 86.</p> <p><b>Note</b> This field is valid only for iDEAL transactions.</p>	Options	Integer (5)

Table 5 Reply Fields (Continued)

Field Name	Description	Returned By:	Data Type & Length
apOptionsReply_totalCount	Total number of options available from the processor.  Possible values: 0-100000  <b>Note</b> This field is valid only for iDEAL transactions.	Options	Numeric (6)
apRefundReply_amount	Total amount for the refund.	Refund	String (15)
apRefundReply_dateTime	Date and time of the refund.	Refund	String (14)
apRefundReply_paymentStatus	Refund status from the processor.  Possible values: <ul style="list-style-type: none"><li>■ REFUNDED: Refund was processed successfully.</li><li>■ PENDING: Refund is pending.</li><li>■ FAILED: Refund failed.</li></ul>	Refund	String (15)
apRefundReply_reasonCode	Numeric value corresponding to the result of the refund service request. See " <a href="#">Reason Codes and Processor Response Codes</a> ," page 86.	Refund	Integer (5)
apRefundReply_reconciliationID	Reference number that you use to reconcile your CyberSource reports with your reports.	Refund	String (60)
apRefundReply_responseCode	Response code from the processor. Range of values: 00000 – 99999. See " <a href="#">Reason Codes and Processor Response Codes</a> ," page 86.	Refund	Integer (5)
apSaleReply_merchantURL	URL for redirecting the customer for capturing the payment. Make a POST request to this URL.  <b>Note</b> For iDEAL payments, the URL is valid for 15 minutes before the session timeouts.	Sale	String (2048)
apSaleReply_paymentStatus	Description of the payment status. Possible values: <ul style="list-style-type: none"><li>■ PENDING: Payment has not yet been processed. CyberSource recommends requesting the Check Status service (see "<a href="#">Checking a Payment Status</a>," page 48) to retrieve the latest status of the payment. The status can be <i>settled</i>, <i>failed</i>, or <i>abandoned</i>.</li><li>■ FAILED: Payment failed.</li></ul>	Sale	String (15)
apSaleReply_processorTransactionID	Processor-generated payment reference code that identifies the transaction.	Sale	String (50)

Table 5 Reply Fields (Continued)

Field Name	Description	Returned By:	Data Type & Length
apSaleReply_reasonCode	Numeric value corresponding to the result of the sale service request. See <a href="#">"Reason Codes and Processor Response Codes,"</a> page 86.	Sale	Integer (5)
apSaleReply_reconciliationID	Reconciliation ID that was used for the transaction.	Sale	String (60)
apSaleReply_responseCode	Response code from the processor.  Range of values: 00000–99999. See <a href="#">"Reason Codes and Processor Response Codes,"</a> page 86.	Sale	Integer (5)
decision	Summarizes the result of the overall request.  Possible values: <ul style="list-style-type: none"> <li>■ ACCEPT</li> <li>■ ERROR</li> <li>■ REJECT</li> </ul> See <a href="#">"Reason Codes and Processor Response Codes,"</a> page 86.	All services.	String (6)
merchantReferenceCode	Order reference or tracking number that you provided in the request. If you included multi-byte characters in this field in the request, the returned value might include corrupted characters.	All services.	String (50)
purchaseTotals_currency	Currency used for the order. Value: EUR	Sale Refund	String (5)
reasonCode	Numeric value corresponding to the result of the overall request. See <a href="#">"Reason Codes and Processor Response Codes,"</a> page 86.	All services	Integer (5)
requestID	Identifier for the request generated by the client.	All services.	String (26)
requestToken	Request token data created by CyberSource for each reply. The field is an encoded string that contains no confidential information such as an account or card verification number. The string can contain a maximum of 256 characters.	All services	String (256)

## Reason Codes and Processor Response Codes



**Important**

CyberSource recommends using the reason code value and the processor response code to determine the transaction result.

The following table describes the reason codes and processor response codes that the Simple Order API can return for the online bank transfer services.

**Table 6 Reason Codes and Processor Response Codes**

Reason Code	Processor Response Code	Description
100	<ul style="list-style-type: none"> <li>■ 00001—status: pending.</li> <li>■ 00002—status: abandoned.</li> <li>■ 00004—status: settled.</li> <li>■ 00006—status: refunded.</li> </ul>	Successful transaction.
102	10000—status: failed.	One or more fields in the request contain invalid data.
150	<ul style="list-style-type: none"> <li>■ 20000—status: failed.</li> <li>■ 20001—status: failed.</li> <li>■ 20002—status: failed.</li> <li>■ 30000—status: failed.</li> <li>■ 30100—status: failed.</li> </ul>	Possible reasons: <ul style="list-style-type: none"> <li>■ The signature was not included in the HTTP header.</li> <li>■ The signature in the HTTP header has expired or its not a valid signature.</li> <li>■ The API version in the HTTP header was missing or is not supported.</li> </ul>
203	<ul style="list-style-type: none"> <li>■ 30200—status: failed.</li> <li>■ 30400—status: failed.</li> <li>■ 30500—status: failed.</li> </ul>	Processor declined the transaction due to funding source problems or the transaction was flagged as high risk.
204	30350—status: failed.	Payment declined due to insufficient funds in the account.
233	<ul style="list-style-type: none"> <li>■ 30600—status: failed.</li> <li>■ 30700—status: failed.</li> </ul>	Processor declined the transaction due to tax errors or government compliance errors.