Online Bank Transfers
Using the Simple Order API

August 2019
CyberSource Contact Information

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

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

For support information about any CyberSource Service, visit the Support Center at http://www.cybersource.com/support.

Copyright

© 2019 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 Unpublished rights reserved under the copyright laws of the United States.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
Merchant Account Types 9
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 17
Refunding a Payment 17
Testing 18
Name-Value Pair Examples 19
   Sale Service 19
   Check Status Service 20
   Refund Service 21
XML Examples 22
   Sale Service 22
   Check Status Service 24
   Refund Service 25
Recent Revisions to This Document

<table>
<thead>
<tr>
<th>Release</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>August 2019</td>
<td>Clarified information about required fields for merchants who use a CyberSource settlement and services account. Added new information about multibyte character strings. See &quot;Working with Multibyte Character Strings,&quot; page 77.</td>
</tr>
</tbody>
</table>
| March 2018 | Bancontact: updated the transaction flow. See "Transaction Flow," page 14. iDEAL:  
  - Updated the transaction flow. See "Transaction Flow," page 47.  
  - Updated the XML sale request example. See "XML Examples," page 57. Sofort:  
  - Updated the possible values of the `billTo_language` field. See "Request Fields," page 78. |
| August 2017 | Updated all the Sale service examples. |
| July 2017 | Updated the `apRefundReply_paymentStatus` reply field. See "Reply Fields," page 87. |
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 you must complete in order to make a payment, request the status of a payment, or refund a payment. It is intended to help you provide a seamless payment experience for the customer.

Conventions

Note and Important Statements

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

An Important statement contains information essential to successfully completing a task or learning a concept.
Text and Command Conventions

<table>
<thead>
<tr>
<th>Convention</th>
<th>Usage</th>
</tr>
</thead>
</table>
| **bold**   | ■ Field and service names in text; for example: Include the ics_applications field.  
■ Items that you are instructed to act upon; for example: Click **Save**. |
| *italic*   | ■ Filenames and pathnames. For example: Add the filter definition and mapping to your web.xml file.  
■ Placeholder variables for which you supply particular values. |
| **Screen text** | ■ XML elements.  
■ Code examples and samples.  
■ Text that you enter in an API environment; for example: Set the apSaleService_run field to **true**. |

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](http://www.cybersource.com/support)
Online banking services enable customers to pay for goods using direct online bank transfers from their bank account to your account.

**Merchant Account Types**

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.

  !important
  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.

  !important
  The CyberSource system performs compliance checks in a production environment on customer data for merchants using a CyberSource settlement and services account. This compliance check is skipped in the CyberSource test environment. To facilitate this check in the production environment, new fields must be sent in the sale API request. If these fields are not sent in the request, the sale service might not show an error in the test environment but will show an error in the production environment. Therefore, merchants must send these fields in their requests: first name, last name, street1, city, country.
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 are supported by CyberSource.

<table>
<thead>
<tr>
<th>Payment Method</th>
<th>Country</th>
<th>Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bancontact (see Chapter 2, &quot;Processing Bancontact Transactions,&quot; on page 13).</td>
<td>Belgium</td>
<td>Sale</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Check Status</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Refund</td>
</tr>
<tr>
<td>EPS (see Chapter 3, &quot;Processing EPS Transactions,&quot; on page 26).</td>
<td>Austria</td>
<td>Sale</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Check Status</td>
</tr>
<tr>
<td>giropay (see Chapter 4, &quot;Processing giropay Transactions,&quot; on page 36).</td>
<td>Germany</td>
<td>Sale</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Check Status</td>
</tr>
<tr>
<td>iDEAL (see Chapter 5, &quot;Processing iDEAL Transactions,&quot; on page 46.</td>
<td>Netherlands</td>
<td>Options</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sale</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Check Status</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Refund</td>
</tr>
<tr>
<td>Sofort (see Chapter 6, &quot;Processing Sofort Transactions,&quot; on page 61).</td>
<td>Austria</td>
<td>Sale</td>
</tr>
<tr>
<td></td>
<td>Belgium</td>
<td>Check Status</td>
</tr>
<tr>
<td></td>
<td>Germany</td>
<td>Refund</td>
</tr>
<tr>
<td></td>
<td>Italy</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Netherlands</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Spain</td>
<td></td>
</tr>
</tbody>
</table>
Options Service

The options service is valid only for iDEAL transactions (see Chapter 5, "Processing iDEAL Transactions," on page 46).

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 link to a static web page. The text should inform the customer that the payment is being processed and that 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 configuration settings to include the URLs in all CyberSource service requests.
Chapter 1  Introduction

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 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:


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

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

Important

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.

Important

Note

If the status returned is pending, wait 5 minutes before making the request. If there is no status change, wait 20 minutes before making another request. After 20 minutes, the status will be settled, abandoned, or failed.
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

When you use a CyberSource settlement and services account, CyberSource automatically requests the export compliance service for every transaction. Export compliance verifies customer information by comparing it to lists maintained by government agencies. If a customer’s billing information appears on any of these lists, the transaction is declined.

The export compliance service can cause customers to experience transaction errors. If no customer billing information is sent in the sale service request in the test environment, no export compliance errors occur. However, if no customer billing information is sent in the sale service request in the production environment, export compliance errors occur.

---

**Important**

If you have a CyberSource settlement and services account, you must send the following customer information in every sale service request: first name, last name, city, street1, country.

---

If you do not have a CyberSource settlement and services account, you do not need to send customer billing data. For more information on CyberSource merchant accounts, see Chapter 1, "Merchant Account Types," on page 9.

---

**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 these **required** fields in the request:
- `apPaymentType`—set to MCH.
- `apSaleService_cancelURL`—required when your CyberSource account does not include this value.
- `apSaleService_successURL`—required when your CyberSource account does not include this value.
- `apSaleService_failureURL`—required when your CyberSource account does not include this value.
- `billTo_city`—required when you have a CyberSource settlement and services account.
- `billTo_country`—required when you have a CyberSource settlement and services account.
- `billTo_firstName`—required when you have a CyberSource settlement and services account.
- `billTo_lastName`—required when you have a CyberSource settlement and services account.
- `billTo_street1`—required when you have a CyberSource settlement and services account.
- `invoiceHeader_merchantDescriptor`
- `merchantID`
- `merchantReferenceCode`
- `purchaseTotals_currency`
- `purchaseTotals_grandTotalAmount`

For examples of a sale request and reply, see "Name-Value Pair Examples," page 19. For detailed descriptions of all request and reply fields, see Appendix A, "API Fields," on page 76.
Checking a Payment Status

To check the status of a payment:

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

**Step 2** Include these 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 examples of a check status request and reply, see "Name-Value Pair Examples," page 19. For detailed descriptions of all request and reply fields, see Appendix A, "API Fields," on page 76.

Refunding a Payment

To refund a payment:

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

**Step 2** Include these 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`
Chapter 2  Processing Bancontact Transactions

- purchaseTotals_currency
- purchaseTotals_grandTotalAmount

For examples of a refund request and reply, see “Name-Value Pair Examples,” page 19. For detailed descriptions of all request and reply fields, see Appendix A, "API Fields," on page 76.

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`
- `purchaseTotals_grandTotalAmount=100.99`
- `purchaseTotals_currency=EUR`
- `apPaymentType=MCH`
- `invoiceHeader_merchantDescriptor=World largest online store`

Example 2  Sale Reply

- `merchantReferenceCode=refnum1234`
- `requestID=4703329437356002101200`
- `decision=ACCEPT`
- `reasonCode=100`
- `requestToken=Ahjn7wSR/e+dp++Ut4+hHjueHZ1ctWbBgxctGjNP95AQ40tIJYafsJMMm3/2Lid9AAA0B1c`
- `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%20%E6%B8%AC%E8%A6%86%E6%B8%AC&param_url=https%3A%2F%2Fwww.abc.com&param_special=+@#%~&sign=fda1df42b6250a10e2e3f1c0fc87862`
- `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+rHR63PjIhHiKkNkwYMGPTRP95A4NIYagMJMMm3/2LidVC6CR/
e+qqrSe6xTgAAAFFRi
apCheckStatusReply_reasonCode=100
apCheckStatusReply_reconciliationID=TC200024
apCheckStatusReply_paymentStatus=settled
apCheckStatusReply_processorResponse=000004
Refund Service

Example 5  Refund Request

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

Example 6  Refund Reply

```plaintext
merchantReferenceCode=refnum1234
requestID=4703332594746002601200
decision=ACCEPT
reasonCode=100
requestToken=AhjnwrSR/e+0FzHfgunhHkuWXN132HLVmwYMXLVgzasU/3kBdjX0gfKN2EmGTb/
7FxOqhBf/vfpxR+aL5QAAAjAE5
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
```
Chapter 2  Processing Bancontact Transactions

XML Examples

Sale Service

Example 7  Sale Request

```xml
<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>
  </billTo>
  <purchaseTotals>
    <currency>EUR</currency>
    <grandTotalAmount>20.00</grandTotalAmount>
  </purchaseTotals>
  <apPaymentType>MCH</apPaymentType>
  <apSaleService run="true">
  </apSaleService>
</requestMessage>
```
Example 8   Sale Reply

```xml
<rs:replyMessage xmlns:rs="urn:schemas-cybersource-com:transaction-data-1.126">
  <rs:merchantReferenceCode>refnum1234</rs:merchantReferenceCode>
  <rs:requestID>459806160576069801200</rs:requestID>
  <rs:decision>ACCEPT</rs:decision>
  <rs:reasonCode>100</rs:reasonCode>
  <rs:requestToken>Ahjn7wSR8octWbBxctGjNP93fNRq9IYyfr58Mn3/2lidVC5Bmr</rs:requestToken>
  <rs:purchaseTotals>
    <c:currency>EUR</c:currency>
  </rs:purchaseTotals>
  <rs:apSaleReply>
    <rs:reasonCode>100</rs:reasonCode>
    <rs:paymentStatus>pending</rs:paymentStatus>
    <rs:responseCode>00001</rs:responseCode>
    <rs:merchantURL>https://merchant.redirect.com/url.do?param_utf=%27%22%3C%20%E6%B8%AC%E8%9A%A6%E6%8B%AC&amp;param_url=https%3A%2F%2Fwww.abc.com&amp;param_special=+@#~-&amp;sign=fdaa1df426260a10e2e3f1c0fc87862</rs:merchantURL>
    <rs:reconciliationID>sale9530019443</rs:reconciliationID>
  </rs:apSaleReply>
</rs:replyMessage>
```
Check Status Service

Example 9  Check Status Request

```xml
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.126">
  <merchantID>mid12345</merchantID>
  <merchantReferenceCode>refnum1234</merchantReferenceCode>
  <apPaymentType>MCH</apPaymentType>
  <apCheckStatusService run="true">
    <checkStatusRequestID>459806154776606901200</checkStatusRequestID>
  </apCheckStatusService>
</requestMessage>
```

Example 10  Check Status Reply

```xml
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.126">
  <c:requestID>4598061583426069501200</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:requestToken>AhjnrwSR5sHLhu2bJ/u75qNRpB+xcTqoXQSP1C7Cgee2EWcAnW0l</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

```xml
<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>459806176634607001200</refundRequestID>
  </apRefundService>
</requestMessage>
```

Example 12  Refund Reply

```xml
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.126">
  <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>
```
Electronic Payment Standard (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 account.

---

**Important**

Refunds are currently not supported for EPS transactions.

---

## Transaction Flow

For a transaction flow diagram see:


1. You display the EPS payment method on your checkout page.

---

**Note**

For logo guidelines see:

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

---

2. This step is optional. You add the EPS bank selection widget to your web page:

   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 chooses the BIC using the widget (optional) or enters their BIC and clicks Pay.
You request the sale service (**apSaleService**) and include the corresponding BIC (**bankInfo_swiftCode**) that the customer chose 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.

CyberSource responds with a merchant URL to direct the customer to their online banking page to confirm their payment details.

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.

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.

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.

If the status returned is **pending**, wait 5 minutes before making the request. If there is no status change, wait 20 minutes before making another request. After 20 minutes the status will be **settled**, **abandoned**, or **failed**.

### Performing a Sale

When you use a CyberSource settlement and services account, CyberSource automatically requests the export compliance service for every transaction. Export compliance verifies customer information by comparing it to lists maintained by government agencies. If a customer’s billing information appears on any of these lists, the transaction is declined.

The export compliance service can cause customers to experience transaction errors. If no customer billing information is sent in the sale service request in the test environment,
no export compliance errors occur. However, if no customer billing information is sent in the sale service request in the production environment, export compliance errors occur.

---

**Important**

If you have a CyberSource settlement and services account, you must send the following customer information in every sale service request: first name, last name, city, street1, country.

---

If you do not have a CyberSource settlement and services account, you do not need to send customer billing data. For more information on CyberSource merchant accounts, see Chapter 1, “Merchant Account Types,” on page 9.

---

**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 these *required* fields in the request:

- *apPaymentType*—set to *EPS*.
- *apSaleService_cancelURL*—required when your CyberSource account does not include this value.
- *apSaleService_failureURL*—required when your CyberSource account does not include this value.
- *apSaleService_successURL*—required when your CyberSource account does not include this value.
- *bankInfo_swiftCode*
- *billTo_city*—required when you have a CyberSource settlement and services account.
- *billTo_country*—required when you have a CyberSource settlement and services account.
- *billTo_firstName*—required when you have a CyberSource settlement and services account.
- *billTo_lastName*—required when you have a CyberSource settlement and services account.
- *billTo_street1*—required when you have a CyberSource settlement and services account.
Chapter 3  Processing EPS Transactions

- invoiceHeader_merchantDescriptor
- merchantID
- merchantReferenceCode
- purchaseTotals_currency
- purchaseTotals_grandTotalAmount

For examples of a sale request and reply, see "Name-Value Pair Examples," page 31. For detailed descriptions of all request and reply fields, see Appendix A, "API Fields," on page 76.

Checking a Payment Status

To check the status of a payment:

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

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

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

  ! Important
  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 examples of a check status request and reply, see "Name-Value Pair Examples," page 31. For detailed descriptions of all request and reply fields, see Appendix A, "API Fields," on page 76.
Testing

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

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

To test an EPS transaction:

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 your web site.

The payment status updates to settled and you are directed to the success URL that you included in the Sale service request.
Chapter 3  Processing EPS Transactions

Name-Value Pair Examples

Sale Service

Example 13  Sale Request

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>apSaleService_run</td>
<td>true</td>
</tr>
<tr>
<td>merchantID</td>
<td>mid12345</td>
</tr>
<tr>
<td>merchantReferenceCode</td>
<td>refnum1234</td>
</tr>
<tr>
<td>billTo_firstName</td>
<td>John</td>
</tr>
<tr>
<td>billTo_lastName</td>
<td>Smith</td>
</tr>
<tr>
<td>billTo_street1</td>
<td>1 The Street</td>
</tr>
<tr>
<td>billTo_city</td>
<td>Reading</td>
</tr>
<tr>
<td>billTo_state</td>
<td>Berkshire</td>
</tr>
<tr>
<td>billTo_postalCode</td>
<td>RG26DH</td>
</tr>
<tr>
<td>billTo_country</td>
<td>GB</td>
</tr>
<tr>
<td>billTo_email</td>
<td><a href="mailto:null@example.com">null@example.com</a></td>
</tr>
<tr>
<td>purchaseTotals_grandTotalAmount</td>
<td>1999.99</td>
</tr>
<tr>
<td>purchaseTotals_currency</td>
<td>EUR</td>
</tr>
<tr>
<td>apPaymentType</td>
<td>EPS</td>
</tr>
<tr>
<td>invoiceHeader_merchantDescriptor</td>
<td>World largest online store</td>
</tr>
</tbody>
</table>

Example 14  Sale Reply

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>merchantReferenceCode</td>
<td>refnum1234</td>
</tr>
<tr>
<td>requestID</td>
<td>4703329437356002101200</td>
</tr>
<tr>
<td>decision</td>
<td>ACCEPT</td>
</tr>
<tr>
<td>reasonCode</td>
<td>100</td>
</tr>
<tr>
<td>requestToken</td>
<td>Ahjn?wSR/e+dp++Ut4+hHjueHZlctWbGxctGjNP95AQ40tIJYafsJMMm3/2LidVC6A</td>
</tr>
<tr>
<td>purchaseTotals_currency</td>
<td>EUR</td>
</tr>
<tr>
<td>apSaleReply_reasonCode</td>
<td>100</td>
</tr>
<tr>
<td>apSaleReply_paymentStatus</td>
<td>pending</td>
</tr>
<tr>
<td>apSaleReply_responseCode</td>
<td>00001</td>
</tr>
<tr>
<td>apSaleReply_merchantURL</td>
<td><a href="https://merchant.redirect.com/url.do?param_utf=%27%22%3C%3E%20%E6%B8%AC%E8%A9%A6%E6%B8%AC&amp;param">https://merchant.redirect.com/url.do?param_utf=%27%22%3C%3E%20%E6%B8%AC%E8%A9%A6%E6%B8%AC&amp;param</a>_ url=https%3A%2F%2Fwww.abc.com&amp;param_special=+@#%~_ &amp;sign=fdaa1df42b6260a10e2e3f1c0fc87862</td>
</tr>
<tr>
<td>apSaleReply_reconciliationID</td>
<td>sale9530019443</td>
</tr>
</tbody>
</table>
Check Status Service

Example 15  Check Status Request

apCheckStatusService_run=true
merchantID=mid12345
merchantReferenceCode=refnum1234
apPaymentType=EPS
apCheckStatusService_checkStatusRequestID=470333126846602201200

Example 16  Check Status Reply

merchantReferenceCode=refnum1234
requestID=470333133136602301200
decision=ACCEPT
reasonCode=100
requestToken=AhjnrwSR/e+HR63PjIhHiKKwYM4TRP95AO411JYagMJMMm3/27idVC6CR/
e+qrSe6xTgAAA1FhRi
apCheckStatusReply_reasonCode=100
apCheckStatusReply_reconciliationID=TC200024
apCheckStatusReply_paymentStatus=approved
apCheckStatusReply_processorResponse=000004
XML Examples

Sale Service

Example 17  Sale Request

```xml
<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>
  </billTo>
  <purchaseTotals>
    <currency>EUR</currency>
    <grandTotalAmount>20.00</grandTotalAmount>
  </purchaseTotals>
  <apPaymentType>EPS</apPaymentType>
  <apSaleService run="true">
  </apSaleService>
</requestMessage>
```
Example 18  Sale Reply

```xml
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.126">
  <c:requestID>4598061680576069801200</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:requestToken>Abjn7wSR8octWbGxctGjNP93fNRq91IJYafr58Mm3/2lidVC5Bmr</c:requestToken>
  <c:purchaseTotals>
    <c:currency>EUR</c:currency>
  </c:purchaseTotals>
  <c:apSaleReply>
    <c:reasonCode>100</c:reasonCode>
    <c:paymentStatus>pending</c:paymentStatus>
    <c:merchantURL>https://merchant.redirect.com/url.do?param_utm%22%3C%20%E6%B8%AC%E8%A9%A6%E6%B8%AC%3E%20%E6%B8%AC%3E%20%E6%B8%AC&param_url=https%3A%2F%2Fwww.abc.com&param_special=%3F~&sign=fdaa1df42b6260a10e2e3f1c0fc87862</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

```
<replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.126">
  <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>
</replyMessage>
```
giropay 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 account.

---

Important

Refunds are currently not supported for giropay transactions.

---

**Transaction Flow**

For a transaction flow diagram see:


1. Display the giropay payment method on your checkout page.

---

Note

Logo standards:

https://www.giropay.de/haendler/tool-center/werbemittel-giropay/

---

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

   http://api.girocheckout.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 Bank Identification Code (BIC) to the browser. You can also configure the widget to display all banks or only the banks that support giropay.

3. The customer chooses the giropay 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.

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.

If the status returned is pending, wait 5 minutes before making the request. If the status did not change, wait 15 minutes before making another request. After 15 minutes, the status will be settled, abandoned, or failed.

Performing a Sale

When you use a CyberSource settlement and services account, CyberSource automatically requests the export compliance service for every transaction. Export compliance verifies customer information by comparing it to lists maintained by government agencies. If a customer’s billing information appears on any of these lists, the transaction is declined.

The export compliance service can cause customers to experience transaction errors. If no customer billing information is sent in the sale service request in the test environment,
no export compliance errors occur. However, if no customer billing information is sent in the sale service request in the production environment, export compliance errors occur.

---

**Important**

If you have a CyberSource settlement and services account, you must send the following customer information in every sale service request: first name, last name, city, street1, country.

---

If you do not have a CyberSource settlement and services account, you do not need to send customer billing data. For more information on CyberSource merchant accounts, see Chapter 1, “Merchant Account Types,” on page 9.

---

**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 these fields in the request:

- apPaymentType—set to GPY.
- apSaleService_cancelURL—required when your CyberSource account does not include this value.
- apSaleService_failureURL—required when your CyberSource account does not include this value.
- apSaleService_successURL—required when your CyberSource account does not include this value.
- bankInfo_swiftCode
- billTo_city—required when you have a CyberSource settlement and services account.
- billTo_country—required when you have a CyberSource settlement and services account.
- billTo_firstName—required when you have a CyberSource settlement and services account.
- billTo_lastName—required when you have a CyberSource settlement and services account.
- billTo_street1—required when you have a CyberSource settlement and services account.
Checking a Payment Status

To check the status of a payment:

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

**Step 2** Include these 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 verify the status of a refund, include the value of the `requestID` field that was returned in the refund service response.

For examples of a check status request and reply, see "Name-Value Pair Examples," page 41. For detailed descriptions of all request and reply fields, see Appendix A, "API Fields," on page 76.
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:

---

Step 1 Request the Sale service (see "Performing a Sale," page 37).

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 Transaction Authentication Number (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
purchaseTotals_grandTotalAmount=1999.99
purchaseTotals_currency=EUR
apPaymentType=GPY
invoiceHeader_merchantDescriptor=World largest online store
```

Example 22  Sale Reply

```
merchantReferenceCode=refnum1234
requestID=4703329437356002101200
decision=ACCEPT
reasonCode=100
requestToken=Ahjn7wSR/e+dp++Ut4+hHjueH21ctWbGxctGjNP95AQ40tIYafsJMMm3/2LidVC6A
purchaseTotals_currency=EUR
apSaleReply_reasonCode=100
apSaleReply_paymentStatus=pending
apSaleReply_responseCode=00001
apSaleReply_merchantURL=https://merchant.redirect.com/url.do?param_seq=%27%22%3C%20%E6%B8%AC%E8%A9%A6%E6%B8%AC%27%20%E5%8F%8B%E8%80%80%20%E6%B8%AC%27%20%E6%B8%AC&param_seq=url=https%3A%2F%2Fwww.abc.com&param_special=+@%~&sign=fdaa1df42b6260a10e2e3f1c0fc87862
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+rHR63PjIhHiKkNkwYMGRTRP95AQ41NIJJYagMJMMm3/2LidVCbCR/
e+qqrSe6xTgAAAAFH/Ri
apCheckStatusReply_reasonCode=100
apCheckStatusReply_reconciliationID=TC200024
apCheckStatusReply_paymentStatus=authorized
apCheckStatusReply_processorResponse=000004
XML Examples

Sale Service

Example 25 Sale Request

```xml
<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>
  </billTo>
  <purchaseTotals>
    <currency>EUR</currency>
    <grandTotalAmount>20.00</grandTotalAmount>
  </purchaseTotals>
  <apPaymentType>GPY</apPaymentType>
  <apSaleService run="true">
  </apSaleService>
</requestMessage>
```
Example 26  Sale Reply

```xml
<apSaleReply>
  <paymentStatus>pending</paymentStatus>
  <responseCode>00001</responseCode>
  <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=+%23&amp;sign=fdaa1df42b6260a10e2e3f1c0fc87862</merchantURL>
  <reconciliationID>sale9530019443</reconciliationID>
</apSaleReply>
```
Check Status Service

Example 27  Check Status Request

```xml
<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

```xml
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.126">
  <c:requestID>4598061583426069501200</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:requestToken>AhjnrwSR5sHLhu2bJ/u75qNRpB+xcTqoXQP1C7Cgee2EWCAnW01</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>
```
iDEAL transactions involve processing payments using direct online bank transfers. These banks support the iDEAL payment method:

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

Note: 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

An iDEAL transaction flow can be viewed here:
https://www.ideal.nl/demo/

For a complete transaction flow diagram see:

1 Display the iDEAL payment method on your checkout page.

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.

   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 chooses 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.

---

**Important**

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 inline frame (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.

---

**Note**

If the status returned is pending, wait 5 minutes before making the request. If the status does not change, wait 15 minutes before making another request. 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.
Chapter 5  Processing iDEAL Transactions

Requesting Options

To request options from the processor:

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

**Step 2**  Include these 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 76.

Performing a Sale

When you use a CyberSource settlement and services account, CyberSource automatically requests the export compliance service for every transaction. Export compliance verifies customer information by comparing it to lists maintained by government agencies. If a customer’s billing information appears on any of these lists, the transaction is declined.

The export compliance service can cause customers to experience transaction errors. If no customer billing information is sent in the sale service request in the test environment, no export compliance errors occur. However, if no customer billing information is sent in the sale service request in the production environment, export compliance errors occur.

If you have a CyberSource settlement and services account, you must send the following customer information in every sale service request: first name, last name, city, street1, country.

If you do not have a CyberSource settlement and services account, you do not need to send customer billing data. For more information on CyberSource merchant accounts, see Chapter 1, "Merchant Account Types," on page 9.

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 these fields in the request:

- **apPaymentType**—set to `IDL`.
- **apSaleService_cancelURL**—required when your CyberSource account does not include this value.
- **apSaleService_failureURL**—required when your CyberSource account does not include this value.
- **apSaleService_paymentOptionID**—include the bank ID value returned in the Options service response.
- **apSaleService_successURL**—required when your CyberSource account does not include this value.
- **billTo_city**—required when you have a CyberSource settlement and services account.
- **billTo_country**—required when you have a CyberSource settlement and services account.
- **billTo_email**
- **billTo_firstName**—required when you have a CyberSource settlement and services account.
- **billTo_lastName**—required when you have a CyberSource settlement and services account.
- **billTo_street1**—required when you have a CyberSource settlement and services account.
- **invoiceHeader_merchantDescriptor**
- **merchantID**
- **merchantReferenceCode**
- **purchaseTotals_currency**
- **purchaseTotals_grandTotalAmount**

For examples of a sale request and reply, see "Name-Value Pair Examples," page 54. For detailed descriptions of all request and reply fields, see Appendix A, "API Fields," on page 76.
Checking a Payment Status

To check the status of a payment:

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

**Step 2** Include these 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 examples of a check status request and reply, see "Name-Value Pair Examples," page 54. For detailed descriptions of all request and reply fields, see Appendix A, "API Fields," on page 76.
Refunding a Payment

To refund a payment:

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

**Step 2** Include these 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 examples of a refund request and reply, see "Name-Value Pair Examples," page 54. For detailed descriptions of all request and reply fields, see Appendix A, "API Fields," on page 76.
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 and include `ideal-FVLBNL22` in the `apSaleService_paymentOptionID` field. See "Requesting Options," page 49.

<table>
<thead>
<tr>
<th>Important</th>
</tr>
</thead>
<tbody>
<tr>
<td>If you include an empty value in the <code>apSaleService_paymentOptionID</code> field, choose from the banks available on the bank selection page that is displayed.</td>
</tr>
</tbody>
</table>

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

**Step 3** Choose 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 chose Paid in **Step 3**, you are directed to the success URL and can request the check status service or the refund service.

If you chose 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 chose 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 chose 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

```plaintext
apOptionsService.run=true
merchantID=mid12345
merchantReferenceCode=refnum1234
apPaymentType=IDL
```

Example 30  Options Reply

```plaintext
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

```plaintext
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
purchaseTotals_grandTotalAmount=100.99
purchaseTotals_currency=EUR
apPaymentType=IDL
invoiceHeader_merchantDescriptor=World largest online store
apSaleService_paymentOptionID=00001
```

Example 32  Sale Reply

```plaintext
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_special=+@#%~&sign=fdaa1df42b6260a10e2e3f1c0fc87862
apSaleReply_reconciliationID=sale9530019443
requestID=4703331268466002201200
```

Check Status Service
Example  Check Status Request

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

Example  Check Status Reply

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

Refund Service

Example 33  Refund Request

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

Example 34  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 35  Options Request

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

Example 36  Options Reply

```xml
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.126">
  <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 37  Sale Request

```xml
<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>
  </billTo>
  <purchaseTotals>
    <currency>EUR</currency>
    <grandTotalAmount>20.00</grandTotalAmount>
  </purchaseTotals>
  <apPaymentType>IDL</apPaymentType>
  <apSaleService run="true">
  </apSaleService>
</requestMessage>
```
Example 38  Sale Reply

```xml
<replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.126">
  <merchantReferenceCode/refnum1234</merchantReferenceCode>
  <decision>ACCEPT</decision>
  <reasonCode>100</reasonCode>
  <requestID>4598061547766069401200</requestID>
  <purchaseTotals>
    <currency>EUR</currency>
  </purchaseTotals>
  <apSaleReply>
    <reasonCode>100</reasonCode>
    <paymentStatus>pending</paymentStatus>
    <responseCode>00001</responseCode>
    <merchantURL>https://merchant.redirect.com/url.do?param_utf=%27%22%3C%3E%20%E6%B8%AC%E8%A9%A6%E6%B8%AC&amp;amp;param_ url=https%3A%2F%2Fwww.abc.com&amp;amp;param_special=+@#%~_
&amp;amp;sign=fdaa1df42b6260a10e2e3f1c0fc87862</merchantURL>
    <reconciliationID>sale9530019443</reconciliationID>
  </apSaleReply>
</replyMessage>
```

Example 39  Check Status Request

```xml
<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 40  Check Status Reply

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

Example 41  Refund Request

```xml
:requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.126">
  <merchantID>mid12345</merchantID>
  <merchantReferenceCode>reftnum1234</merchantReferenceCode>
  <purchaseTotals>
    <currency>EUR</currency>
    <grandTotalAmount>20.00</grandTotalAmount>
  </purchaseTotals>
  <apPaymentType>IDL</apPaymentType>
  <apRefundService run="true">
    <refundRequestID>4598061766346070001200</refundRequestID>
  </apRefundService>
</requestMessage>
```

Example 42  Refund Reply

```xml
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.126">
  <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 account.

Transaction Flow

A Sofort transaction flow can be viewed here:


For a complete transaction flow diagram see:


1 Display the Sofort payment option on your checkout page.

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.
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 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 65.

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

When you use a CyberSource settlement and services account, CyberSource automatically requests the export compliance service for every transaction. Export compliance verifies customer information by comparing it to lists maintained by government agencies. If a customer’s billing information appears on any of these lists, the transaction is declined.

The export compliance service can cause customers to experience transaction errors. If no customer billing information is sent in the sale service request in the test environment, no export compliance errors occur. However, if no customer billing information is sent in the sale service request in the production environment, export compliance errors occur.

If you have a CyberSource settlement and services account, you must send the following customer information in every sale service request: first name, last name, city, street1, country.

If you do not have a CyberSource settlement and services account, you do not need to send customer billing data. For more information on CyberSource merchant accounts, see Chapter 1, “Merchant Account Types,” on page 9.

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 these required fields in the request:

- apPaymentType—set to SOF.
- apSaleService_cancelURL—required when your CyberSource account does not include this value.
- apSaleService_successURL—required when your CyberSource account does not include this value.

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
Online Bank Transfers Using the Simple Order API

Chapter 6  Processing Sofort Transactions

- billTo_city—required when you have a CyberSource settlement and services account.
- billTo_country—required when you have a CyberSource settlement and services account.

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.

- billTo_firstName—required when you have a CyberSource settlement and services account.
- billTo_lastName—required when you have a CyberSource settlement and services account.
- billTo_street1—required when you have a CyberSource settlement and services account.
- invoiceHeader_merchantDescriptor

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 examples of a sale request and reply, see "Name-Value Pair Examples," page 69. For detailed descriptions of all request and reply fields, see Appendix A, "API Fields," on page 76.
Chapter 6  Processing Sofort Transactions

Checking a Payment Status

When to Check a Payment Status

Pending Status

Step 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.

Step 2 When the status is pending, wait 15 minutes and request the check status service again.

Step 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

Step 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.

Step 2 When the status is authorized, request the check status service every four hours until the status updates to settled.

Step 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.

Important 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 these fields in the request:

- `apCheckStatusService_checkStatusRequestID`

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 examples of a check status request and reply, see "Name-Value Pair Examples," page 69 For detailed descriptions of all request and reply fields, see Appendix A, "API Fields," on page 76.
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 that you 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 these 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 examples of a refund request and reply, see "Name-Value Pair Examples," page 69. For detailed descriptions of all request and reply fields, see Appendix A, "API Fields," on page 76.
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 63).

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>
<thead>
<tr>
<th>Country</th>
<th>Bank Test Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Austria</td>
<td>00000</td>
</tr>
<tr>
<td>Belgium</td>
<td>Demo Bank</td>
</tr>
<tr>
<td>Germany</td>
<td>888888888</td>
</tr>
<tr>
<td>Italy</td>
<td>00000</td>
</tr>
<tr>
<td>Netherlands</td>
<td>Demo Bank</td>
</tr>
<tr>
<td>Spain</td>
<td>Demo Bank</td>
</tr>
</tbody>
</table>

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 68, (the payment status will be *settled*) and do one of these:

- 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 43  Sale Request**

```plaintext
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_paymentOptionID=00001
```
Example 44  Sale Reply

```
merchantReferenceCode=refnum1234
requestID=4703329437356002101200
decision=ACCEPT
reasonCode=100
requestToken=Ahjn7wSR/e+dp++Ut4+hHjueH21ctWbBgxctGjNP95AQ40tIJYafsJMMm3/2LidVCAA0B1c
purchaseTotals_currency=EUR
apSaleReply_reasonCode=100
apSaleReply_paymentStatus=pending
apSaleReply_responseCode=00001
apSaleReply_merchantURL=https://merchant.redirect.com/url.do?param.utf=%27%22%3E%20%E6%B8%AC%E8%A9%A6%E6%B8%AC&param_url=https%3A%2F%2Fwww.abc.com&param_special=+@#%~&sign=fdaa1df42b6260a10e2e3f1c0fc87862
apSaleReply_reconciliationID=sale9530019443
```

Check Status Service

Example 45  Check Status Request

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

Example 46  Check Status Reply

```
merchantReferenceCode=refnum1234
requestID=4703331331366002301200
decision=ACCEPT
reasonCode=100
requestToken=AhjnwrwSR/e+rHR63PjIhHiKkNywYMGRP95AQ41NIJYagMJMm3/2LidVC6CR/e+qqrSe6xTgAAAPhRi
apCheckStatusReply_reasonCode=100
apCheckStatusReply_reconciliationID=TC200024
apCheckStatusReply_paymentStatus=settled
apCheckStatusReply_processorResponse=000004
```
Refund Service

Example 47  Refund Request

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

Example 48  Refund Reply

```java
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 49  Sale Request

```xml
<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">
  </apSaleService>
</requestMessage>
```
Example 50  Sale Reply

```xml
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.126">
    <c:requestID>459806168057609801200</c:requestID>
    <c:decision>ACCEPT</c:decision>
    <c:reasonCode>100</c:reasonCode>
    <c:requestToken>Ahjn7wSR8octWbGxctGjNP93fNRq9IJJYaf58Mm3/2LidVC5Bmr</c:requestToken>
    <c:purchaseTotals>
        <c:currency>EUR</c:currency>
    </c:purchaseTotals>
    <c:apSaleReply>
        <c:reasonCode>100</c:reasonCode>
        <c:paymentStatus>pending</c:paymentStatus>
        <c:merchantURL>https://merchant.redirect.com/url.do?param_utf=%27%22%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=fdaa1df42b6260a10e2e3f1c0fc87862</c:merchantURL>
        <c:reconciliationID>sale9530019443</c:reconciliationID>
    </c:apSaleReply>
</c:replyMessage>
```
Chapter 6  Processing Sofort Transactions

Check Status Service

Example 51  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 52  Check Status Reply

```
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.126">
  <c:requestID>4598061583426069501200</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:requestToken>AhjnrwSR5sHLhu2bJ/u75qNRpB+xcTqoXQSP1C7Cgee2EWcAnW0l</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 53  Refund Request

```xml
<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 54  Refund Reply

```xml
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.126">
    <c:requestID>4598061583426069501200</c:requestID>
    <c:decision>ACCEPT</c:decision>
    <c:requestToken>AhjnwrSR8oXaIP1sUnmh38hoFxOqhdBI+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>000006</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

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date and time</td>
<td>Format is YYYY-MM-DDTh:mm:ssZ, where:</td>
</tr>
<tr>
<td></td>
<td>- T separates the date and the time</td>
</tr>
<tr>
<td></td>
<td>- Z indicates Coordinated Universal Time (UTC), also known as Greenwich Mean Time (GMT)</td>
</tr>
<tr>
<td>Integer</td>
<td>Whole number {..., -3, -2, -1, 0, 1, 2, 3, ...}</td>
</tr>
<tr>
<td>String</td>
<td>Sequence of letters, numbers, spaces, and special characters</td>
</tr>
</tbody>
</table>
Numbered Elements

The CyberSource XML schema includes several numbered elements. You can include these complex elements more than once in a request. For example, when 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:

```xml
<Item id="0">
  <unitPrice>
    <quantity>
  </item>
</Item>
```

As a name-value pair field name, this tag is called `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 the Simple Object Access Protocol (SOAP), the client contains a corresponding `Item` class.

**Example 55** Numbered XML Schema Element Names and Name-Value Pair Field Names

<table>
<thead>
<tr>
<th>XML Schema Element Names</th>
<th>Corresponding Name-Value Pair Field Names</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;item id=&quot;0&quot;&gt;</code></td>
<td><code>item_0_unitPrice</code></td>
</tr>
<tr>
<td></td>
<td><code>item_0_quantity</code></td>
</tr>
<tr>
<td><code>&lt;item id=&quot;1&quot;&gt;</code></td>
<td><code>item_1_unitPrice</code></td>
</tr>
<tr>
<td></td>
<td><code>item_1_quantity</code></td>
</tr>
</tbody>
</table>

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

---

Working with Multibyte Character Strings

Many international languages – such as Korean, Chinese, and Russian – require more than one ASCII byte per character. Therefore, a string of multibyte characters can exceed the allotted string length of some API fields.

In many international languages, a short name that visually has only 3 or 4 characters may be equivalent to 10 or 15 bytes in UTF-8. Therefore, it is important to take into consideration the overall length of a multibyte string to be sure it fits into the allotted ASCII string length.
For best success when working with international languages, make sure the total size of a multibyte string does not exceed the ASCII string length for a specific API field.

### 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

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
<th>Used By: Required (R) or Optional (O)</th>
<th>Data Type &amp; Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>apCheckStatusService_checkStatusRequestID</td>
<td>Request ID of the sale transaction for which you want to check the status. This value is returned in the Sale service response.</td>
<td>Check Status (R)</td>
<td>String (26)</td>
</tr>
<tr>
<td>apCheckStatusService_run</td>
<td>Whether to include <code>apCheckStatusService</code> in your request. Possible values:</td>
<td>Check Status (R)</td>
<td>String (5)</td>
</tr>
</tbody>
</table>
|                                     | - true: Include the service in your request.  
|                                     | - false (default): Do not include the service in your request.              |                                        |                    |
| apOptionsService_limit              | The maximum number of options to be retrieved from the processor and displayed to the consumer. Possible values: 1–250  
|                                     | **Note** This field is valid only for iDEAL transactions.                   | Options (O)                          | String (3)         |

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*.
### Table 4  Request Fields  (Continued)

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
<th>Used By: Required (R) or Optional (O)</th>
<th>Data Type &amp; Length</th>
</tr>
</thead>
</table>
| apOptionsService_offset | 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. Possible values: 0–9999  
For example:  
&lt;apOptionsService&gt;  
 &lt;limit&gt;10&lt;/limit&gt;  
 &lt;offset&gt;0&lt;/offset&gt;  
&lt;/apOptionsService&gt;  
The above values display 10 options to the customer, starting with the first item on the list received from the processor.  
&lt;apOptionsService&gt;  
 &lt;limit&gt;10&lt;/limit&gt;  
 &lt;offset&gt;10&lt;/offset&gt;  
&lt;/apOptionsService&gt;  
The above values display 10 options to the customer, starting with the 11th item on the list received from the processor.  
**Note** This field is valid only for iDEAL transactions. | Options (O) | String (4) |
| apOptionsService_run    | Whether to include apOptionService in your request. Possible values:  
- **true**: Include the service in your request.  
- **false** *(default)*: Do not include the service in your request.  
**Note** This field is valid only for iDEAL transactions. | Options (R) | String (5) |

---

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*. 

---
### Table 4  Request Fields  (Continued)

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
<th>Used By:</th>
<th>Data Type &amp; Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>apPaymentType</td>
<td>Identifier for the payment method:</td>
<td>Options (R)</td>
<td>String (3)</td>
</tr>
<tr>
<td>EPS: EPS</td>
<td></td>
<td>Sale (R)</td>
<td></td>
</tr>
<tr>
<td>GPY: giropay</td>
<td></td>
<td>Check Status (R)</td>
<td></td>
</tr>
<tr>
<td>IDL: iDEAL</td>
<td></td>
<td>Refund (R)</td>
<td></td>
</tr>
<tr>
<td>MCH: Bancontact</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SOF: Sofort</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>apRefundService_refundRequestID</td>
<td>Request ID of the sale transaction for which you are requesting a refund. Include the value of the requestID field that was returned in the sale service response.</td>
<td>Refund (R)</td>
<td>String (26)</td>
</tr>
<tr>
<td>apRefundService_run</td>
<td>Whether to include apRefundService in your request. Possible values:</td>
<td>Refund (R)</td>
<td>String (5)</td>
</tr>
<tr>
<td>true: Include the service in your request.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>false (default): Do not include the service in your request.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>apSaleService_cancelURL</td>
<td>URL the consumer is directed to after canceling the payment.</td>
<td>Sale (R when your CyberSource account does not include this value; otherwise, O)</td>
<td>String (255)</td>
</tr>
<tr>
<td>apSaleService_failureURL</td>
<td>URL the consumer is directed to after a payment fails.</td>
<td>Sale (R when your CyberSource account does not include this value; otherwise, O)</td>
<td>String (255)</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> This field is valid only for iDEAL and Bancontact transactions.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>apSaleService_paymentOptionID</td>
<td>Payment option ID name.</td>
<td>Sale (O)</td>
<td>String (60)</td>
</tr>
<tr>
<td></td>
<td>This is the bank's swift code. Include the option ID name returned in the Options service response.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Example</strong> ideal-BUNQNL2A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> This field is valid only for iDEAL transactions.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

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*.  

---
### Table 4  Request Fields (Continued)

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
<th>Used By: Required (R) or Optional (O)</th>
<th>Data Type &amp; Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>apSaleService_run</td>
<td>Whether to include <code>apSaleService</code> in your request. Possible values:</td>
<td>Sale (R)</td>
<td>String (5)</td>
</tr>
<tr>
<td></td>
<td>- <strong>true</strong>: Include the service in your request.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- <strong>false</strong> (default): Do not include the service in your request.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>apSaleService_successURL</td>
<td>URL the consumer is directed to after completing the payment.</td>
<td>Sale (R when your CyberSource account does not include this value; otherwise, O)</td>
<td>String (255)</td>
</tr>
<tr>
<td>apSaleService_transactionTimeout</td>
<td>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 <code>abandoned</code>. <strong>Note</strong> 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). When the transaction times out, Sofort changes the transaction status to <code>abandoned</code>.</td>
<td>Sale (O)</td>
<td>Integer (5)</td>
</tr>
<tr>
<td>bankInfo_swiftCode</td>
<td>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. <strong>Note</strong> This field is valid only for giropay and EPS transactions.</td>
<td>Sale (O)</td>
<td>String (20)</td>
</tr>
<tr>
<td>billTo_city</td>
<td>Credit card billing city.</td>
<td>Sale (R when you have a CyberSource settlement and services account; otherwise, O.)</td>
<td>String (50)</td>
</tr>
<tr>
<td>billTo_company</td>
<td>Name of the customer’s company.</td>
<td>Sale (O)</td>
<td>String (60)</td>
</tr>
</tbody>
</table>

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](#).
Table 4  Request Fields (Continued)

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
<th>Used By: Required (R) or Optional (O)</th>
<th>Data Type &amp; Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>billTo_country</td>
<td>Credit card billing country. Use the two-digit character ISO Standard Country Codes.</td>
<td>Sale (R when you have a CyberSource settlement and services account; otherwise, O.)</td>
<td>String (2)</td>
</tr>
<tr>
<td>billTo_email</td>
<td>Customer’s email address, including the full domain name.</td>
<td>Sale (O)</td>
<td>String (255)</td>
</tr>
<tr>
<td>billTo_firstName</td>
<td>Customer’s first name. This name must be the same as the name on the card.</td>
<td>Sale (R when you have a CyberSource settlement and services account; otherwise, O.)</td>
<td>String (60)</td>
</tr>
<tr>
<td>billTo_language</td>
<td>Language code and country code used for the transaction.</td>
<td>Sale (O)</td>
<td>String (5)</td>
</tr>
<tr>
<td></td>
<td>Format: [language code][dash][country code]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Possible values:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- DE-BE: German-Belgium</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- DE-DE (default): German-German</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- DE-ES: German-Spain</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- DE-IT: German-Italy</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- EN-AT: English-Austria</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- EN-BE: English-Belgium</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- EN-DE: English-German</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- EN-ES: English-Spain</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- EN-IT: English-Italy</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- EN-NL: English-Netherlands</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- ES-ES: Spanish-Spain</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- FR-BE: French-Belgium</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- IT-IT: Italian-Italy</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- NL-BE: Dutch-Belgium</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- NL-NL: Dutch-Netherlands</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note**  This field is valid only for Sofort transactions.

---

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.
### Table 4 Request Fields (Continued)

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
<th>Used By: Required (R) or Optional (O)</th>
<th>Data Type &amp; Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>billTo_lastName</td>
<td>Customer’s last name. This name must be the same as the name on the card.</td>
<td>Sale (R when you have a CyberSource settlement and services account; otherwise, O.)</td>
<td>String (60)</td>
</tr>
<tr>
<td>billTo_state</td>
<td>Credit card billing state or province. Use the State, Province, and Territory Codes for the United States and Canada.</td>
<td>Sale (O)</td>
<td>String (2)</td>
</tr>
<tr>
<td>billTo_street1</td>
<td>Credit card billing street address as it appears in the credit card issuer’s records.</td>
<td>Sale (R when you have a CyberSource settlement and services account; otherwise, O.)</td>
<td>String (60)</td>
</tr>
<tr>
<td>invoiceHeader_merchantDescriptor</td>
<td>Description displayed on the customer’s statement. When you include more than one consecutive space, extra spaces are removed.</td>
<td>Sale (R)</td>
<td>For Bancontact and iDEAL: String (35) For Sofort, EPS and giropay: String (27)</td>
</tr>
<tr>
<td>invoiceHeader_merchantDescriptorCity</td>
<td>City for your business location. This value is displayed on the customer’s statement.</td>
<td>Sale (O)</td>
<td>String (50)</td>
</tr>
<tr>
<td>invoiceHeader_merchantDescriptorContact</td>
<td>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.</td>
<td>Sale (O)</td>
<td>String (150)</td>
</tr>
<tr>
<td>invoiceHeader_merchantDescriptorCountry</td>
<td>Country code for your business location. Use the standard ISO Standard Country Codes. This value is displayed on the customer's statement.</td>
<td>Sale (O)</td>
<td>String (2)</td>
</tr>
<tr>
<td>invoiceHeader_merchantDescriptorPostalCode</td>
<td>Postal code for your business location. This value is displayed on the customer's statement.</td>
<td>Sale (O)</td>
<td>String (10)</td>
</tr>
</tbody>
</table>

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.
### Table 4  Request Fields  (Continued)

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
<th>Used By: Required (R) or Optional (O)</th>
<th>Data Type &amp; Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>invoiceHeader_merchantDescriptorState</td>
<td>State code or region code for your business location. This value is displayed on the customer's statement.</td>
<td>Sale (O)</td>
<td>String (50)</td>
</tr>
<tr>
<td>invoiceHeader_merchantDescriptorStreet</td>
<td>Street address for your business location. If the street address is more than 38 characters, use meaningful abbreviations.</td>
<td>Sale (O)</td>
<td>String (60)</td>
</tr>
<tr>
<td>item_#_productCode</td>
<td>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 default.</td>
<td>Sale (O)</td>
<td>String (255)</td>
</tr>
<tr>
<td>item_#_productName</td>
<td>This field is required when item_#_productCode is not default or one of the values related to shipping and/or handling.</td>
<td>Sale (O)</td>
<td>String (255)</td>
</tr>
<tr>
<td>item_#_productSKU</td>
<td>Identification code for the product.</td>
<td>Sale (O)</td>
<td>String (255)</td>
</tr>
<tr>
<td>item_#_quantity</td>
<td>The default is 1. This field is required when item_#_productCode is not default or one of the values related to shipping and/or handling.</td>
<td>Sale (O)</td>
<td>Integer (10)</td>
</tr>
</tbody>
</table>

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*. 

---

*Online Bank Transfers Using the Simple Order API | August 2019*
### Table 4  Request Fields (Continued)

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
<th>Used By: Required (R) or Optional (O)</th>
<th>Data Type &amp; Length</th>
</tr>
</thead>
</table>
| 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:
   item_0_unitPrice=10.00
   item_0_quantity=1
   item_0_taxAmount=0.80
   item_1_unitPrice=20.00
   item_1_quantity=1
   item_1_taxAmount=1.60
2 The total amount authorized will be 32.40, not 30.00 with 2.40 of tax included. | 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. | Sale (See description) | String (15) |
| merchantID               | Your CyberSource merchant ID. Use the same merchant ID for evaluation, testing, and production. | Required for all service requests. | String (30) |

---

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*. 

---
Table 4  Request Fields  (Continued)

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
<th>Used By: Required (R) or Optional (O)</th>
<th>Data Type &amp; Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>merchantReferenceCode</td>
<td>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. To ensure success when working with the sale service in this field, use only hyphens (-) and underscores (_) as your special characters.</td>
<td>Required for all service requests.</td>
<td>String (50)</td>
</tr>
<tr>
<td>purchaseTotals_currency</td>
<td>Currency used for the order. Use the three character ISO Standard Currency Codes.</td>
<td>Sale (R)</td>
<td>String (5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Refund (R)</td>
<td></td>
</tr>
<tr>
<td>purchaseTotals_grandTotalAmount</td>
<td>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. You must include either purchaseTotals_grandTotalAmount or item_#<em>unitPrice and item</em>#_taxAmount in your request.</td>
<td>Sale (R)</td>
<td>Decimal (15)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Refund (R)</td>
<td></td>
</tr>
</tbody>
</table>

Note

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

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
<th>Returned By:</th>
<th>Data Type &amp; Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>apCheckStatusReply_</td>
<td>Description of the payment status. Possible values:</td>
<td>Check Status</td>
<td>String (15)</td>
</tr>
<tr>
<td>paymentStatus</td>
<td><strong>PENDING:</strong> Payment was initiated. Do not ship the goods.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>SETTLED:</strong> Payment is complete and the funds have been settled in your account. Ship the goods.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>FAILED:</strong> Payment failed.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>ABANDONED:</strong> 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.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>AUTHORIZED:</strong> Customer authorized the payment. This status is valid only for Sofort transactions.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>apCheckStatusReply_</td>
<td>Numeric value corresponding to the result of the check status service request. See &quot;Reason Codes and Processor Response Codes,&quot; page 91.</td>
<td>Check Status</td>
<td>Integer (5)</td>
</tr>
<tr>
<td>reasonCode</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>apCheckStatusReply_</td>
<td>Reconciliation ID that was used for the transaction.</td>
<td>Check Status</td>
<td>String (60)</td>
</tr>
<tr>
<td>reconciliationID</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>apOptionsReply_count</td>
<td>The total number of options requested from the processor. Possible values: 1–250</td>
<td>Options</td>
<td>Numeric (3)</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> This field is valid only for iDEAL transactions.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>apOptionsReply_offset</td>
<td>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. Possible values: 0–9999</td>
<td>Options</td>
<td>Numeric (4)</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> This field is valid only for iDEAL transactions.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 5  Reply Fields (Continued)

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
<th>Returned By:</th>
<th>Data Type &amp; Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>apOptionsReply_option_##_id</td>
<td>Unique identifier associated with the option name.</td>
<td>Options</td>
<td>String (34)</td>
</tr>
<tr>
<td><strong>Example</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>apOptionsReply_option_0_id=ideal-ABNANL2A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>apOptionsReply_option_1_id=ideal-ASNBNL21</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>apOptionsReply_option_2_id=ideal-BUNQNL2A</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>This field is valid only for iDEAL transactions.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>apOptionsReply_option_##_name</td>
<td>Name associated with the option ID.</td>
<td>Options</td>
<td>String (60)</td>
</tr>
<tr>
<td><strong>Example</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>apOptionsReply_option_0_name=ABN AMRO</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>apOptionsReply_option_1_name=ASN Bank</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>apOptionsReply_option_2_name=Bunq</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Important</strong></td>
<td>CyberSource recommends caching the information and displaying the bank name to the customer in a drop-down menu on your checkout page.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>This field is valid only for iDEAL transactions.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>apOptionsReply_reasonCode</td>
<td>Numeric value corresponding to the result of the options service request. See &quot;Reason Codes and Processor Response Codes,&quot; page 91.</td>
<td>Options</td>
<td>Integer (5)</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>This field is valid only for iDEAL transactions.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>This field is valid only for iDEAL transactions.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
<td>Returned By:</td>
<td>Data Type &amp; Length</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>apOptionsReply_totalCount</td>
<td>Total number of options available from the processor.</td>
<td>Options</td>
<td>Numeric (6)</td>
</tr>
<tr>
<td></td>
<td>Possible values: 0–100000</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> This field is valid only for iDEAL transactions.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>apRefundReply_amount</td>
<td>Total amount for the refund.</td>
<td>Refund</td>
<td>String (15)</td>
</tr>
<tr>
<td>apRefundReply_dateTime</td>
<td>Date and time of the refund.</td>
<td>Refund</td>
<td>String (14)</td>
</tr>
<tr>
<td>apRefundReply_paymentStatus</td>
<td>Refund status from the processor.</td>
<td>Refund</td>
<td>String (15)</td>
</tr>
<tr>
<td></td>
<td>Possible values:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- <strong>REFUNDED</strong>: Refund was processed successfully.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- <strong>PENDING</strong>: Refund is pending.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- <strong>FAILED</strong>: Refund failed.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>apRefundReply_reasonCode</td>
<td>Numeric value corresponding to the result of the refund service request.</td>
<td>Refund</td>
<td>Integer (5)</td>
</tr>
<tr>
<td></td>
<td>See &quot;Reason Codes and Processor Response Codes,&quot; page 91.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>apRefundReply_reconciliationID</td>
<td>Reference number that you use to reconcile your CyberSource reports with your reports.</td>
<td>Refund</td>
<td>String (60)</td>
</tr>
<tr>
<td>apSaleReply_merchantURL</td>
<td>URL for redirecting the customer for capturing the payment. Make a POST request to this URL.</td>
<td>Sale</td>
<td>String (2048)</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> For iDEAL payments, the URL is valid for 15 minutes before the session timeouts.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>apSaleReply_paymentStatus</td>
<td>Description of the payment status. Possible values:</td>
<td>Sale</td>
<td>String (15)</td>
</tr>
<tr>
<td></td>
<td>- <strong>PENDING</strong>: Payment has not yet been processed.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CyberSource recommends requesting the Check Status service (see &quot;Checking a Payment Status,&quot; page 51) to retrieve the latest status of the payment. The status can be <strong>settled, failed, or abandoned</strong>.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- <strong>FAILED</strong>: Payment failed.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>apSaleReply_processorTransactionID</td>
<td>Processor-generated payment reference code that identifies the transaction.</td>
<td>Sale</td>
<td>String (50)</td>
</tr>
</tbody>
</table>
## Table 5  Reply Fields (Continued)

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
<th>Returned By:</th>
<th>Data Type &amp; Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>apSaleReply_reasonCode</td>
<td>Numeric value corresponding to the result of the sale service request. See &quot;Reason Codes and Processor Response Codes,&quot; page 91.</td>
<td>Sale</td>
<td>Integer (5)</td>
</tr>
<tr>
<td>apSaleReply_reconciliationID</td>
<td>Reconciliation ID that was used for the transaction.</td>
<td>Sale</td>
<td>String (60)</td>
</tr>
<tr>
<td>decision</td>
<td>Summarizes the result of the overall request. Possible values:</td>
<td>All services</td>
<td>String (6)</td>
</tr>
<tr>
<td></td>
<td>• ACCEPT</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• ERROR</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• REJECT</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>See &quot;Reason Codes and Processor Response Codes,&quot; page 91.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>merchantReferenceCode</td>
<td>Order reference or tracking number that you provided in the request. If you included multi-byte characters in this field in the request, the returned value might include corrupted characters.</td>
<td>All services</td>
<td>String (50)</td>
</tr>
<tr>
<td>purchaseTotals_currency</td>
<td>Currency used for the order. Value: EUR</td>
<td>Sale</td>
<td>String (5)</td>
</tr>
<tr>
<td>requestID</td>
<td>Identifier for the request generated by the client.</td>
<td>All services</td>
<td>String (26)</td>
</tr>
<tr>
<td>requestToken</td>
<td>Request token data created by CyberSource for each reply. The field is an encoded string that contains no confidential information such as an account or card verification number. The string can contain a maximum of 256 characters.</td>
<td>All services</td>
<td>String (256)</td>
</tr>
</tbody>
</table>
## Reason Codes and Processor Response Codes

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

<table>
<thead>
<tr>
<th>Reason Code</th>
<th>Processor Response Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>00001—status: pending.</td>
<td>Successful transaction.</td>
</tr>
<tr>
<td></td>
<td>00002—status: abandoned.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>00003—status: authorized</td>
<td></td>
</tr>
<tr>
<td></td>
<td>00004—status: settled.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>00006—status: refunded.</td>
<td></td>
</tr>
<tr>
<td>102</td>
<td>10000—status: failed.</td>
<td>One or more fields in the request contain invalid data.</td>
</tr>
<tr>
<td>150</td>
<td>20000—status: failed.</td>
<td>Possible reasons:</td>
</tr>
<tr>
<td></td>
<td>20001—status: failed.</td>
<td>The signature was not included in the HTTP header.</td>
</tr>
<tr>
<td></td>
<td>20002—status: failed.</td>
<td>The signature in the HTTP header has expired or it's not a valid signature.</td>
</tr>
<tr>
<td></td>
<td>30000—status: failed.</td>
<td>The API version in the HTTP header was missing or is not supported.</td>
</tr>
<tr>
<td></td>
<td>30100—status: failed.</td>
<td></td>
</tr>
<tr>
<td>203</td>
<td>30200—status: failed.</td>
<td>Processor declined the transaction due to funding source problems or the transaction was flagged as high risk.</td>
</tr>
<tr>
<td></td>
<td>30400—status: failed.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30500—status: failed.</td>
<td></td>
</tr>
<tr>
<td>204</td>
<td>30350—status: failed.</td>
<td>Payment declined due to insufficient funds in the account.</td>
</tr>
<tr>
<td>233</td>
<td>30600—status: failed.</td>
<td>Processor declined the transaction due to tax errors or government compliance errors.</td>
</tr>
<tr>
<td></td>
<td>30700—status: failed.</td>
<td></td>
</tr>
</tbody>
</table>