

# Pay by Bank Integration

Simple Order API



Developer Guide  
Pilot



## Cybersource Contact Information

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

For sales questions about any Cybersource service, email [sales@cybersource.com](mailto:sales@cybersource.com) or call 650-432-7350 or 888-330-2300 (toll free in the United States).

For support information about any Cybersource service, visit the Support Center: <https://www.cybersource.com/support>

## Copyright

© 2020. 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 in 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 Unpublished rights reserved under the copyright laws of the United States.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. Visa, Visa International, Cybersource, the Visa logo, and the Cybersource logo are the registered trademarks of Visa International in the United States and other countries. All other trademarks, service marks, registered marks, or registered service marks are the property of their respective owners.

## Confidentiality Notice

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

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

## Revision

Version: 24.02

# Contents

<b>Pay by Bank Integration Developer Guide.....</b>	<b>4</b>
Recent Revisions to This Document.....	5
<b>Introduction to Pay by Bank.....</b>	<b>6</b>
Sale Workflow.....	7
Refund Workflow.....	9
Transaction Statuses Workflow.....	11
<b>Pay by Bank Services.....</b>	<b>14</b>
Sale.....	14
Line Items.....	15
Processing a Sale.....	16
Simple Order Example: Processing a Sale.....	18
Refund.....	19
Required Fields for Processing a Refund.....	20
Simple Order Example: Issuing a Refund.....	20
Check Status.....	21
Required Fields for a Check Status Request.....	22
Simple Order Example: Requesting a Check Status.....	22
Webhook Subscriptions.....	23
Required Fields for Subscribing to a Webhook.....	24
REST Example: Subscribing to a Webhook.....	25
<b>Reason Codes and Pay by Bank Response Codes.....</b>	<b>27</b>
<b>Generating Reports Using the Business Center.....</b>	<b>29</b>

# Pay by Bank Integration Developer Guide

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

## Audience and Purpose

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

## Conventions

This statement appears in this document:



### Important

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

## Related Documentation

Visit the [Cybersource documentation hub](#) to find additional technical documentation.

## Customer Support

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

# Recent Revisions to This Document

## 24.02

- Updated the possible webhook notification response statuses. See [Webhook Subscriptions](#) on page 23.
- Updated the **organizationID** field and currency in the webhook notification example. See [REST Example: Subscribing to a Webhook](#) on page 25.
- Added example for a check status of a sale. See [Simple Order Example: Requesting a Check Status](#) on page 22.
- Updated response status in the refund example to **pending**. See [Simple Order Example: Issuing a Refund](#) on page 20.

## 24.01

Initial release.

# Introduction to Pay by Bank

Pay by Bank is a Bank of America solution that enables you to offer your e-commerce customers the ability to pay directly from their bank account in real time without providing payment card information. When a payment is processed, your customer's bank transfers the funds directly to your merchant bank account.

In the UK, Pay by Bank follows the open banking regulations for bank-to-bank transfers.

## Benefits of Using Pay by Bank

Pay by Bank makes processing e-commerce transactions simpler, faster, and more secure by:

- Reducing your amount of customer data storage because card data is no longer needed.
- Streamlining reconciliation data into an easy-to-read report.
- Using a customer's bank authentication network to securely process payments.
- Avoiding card processing fees.
- Providing an easy connection to Pay by Bank through an API integration.

## Supported Services

Sale, refund, and check status services are available with Pay by Bank. For more information, see these sections:

- [Sale](#)
- [Refund](#)
- [Check Status](#)

You can also subscribe to webhook notifications. With webhook subscriptions, Cybersource will automatically notify you when a sale or a refund is completed. See [Webhook Subscriptions](#) on page 23.

## Requirements

You must have a merchant ID for each country in which you process transactions. Contact your Cybersource account manager for more information.

## Supported Currencies

This table lists the currencies supported by the Pay by Bank services.

### Supported Currencies

Currency	Currency Code
British pound sterling	GBP
Euro	EUR*

\*Refunds are not currently supported for EUR; this capability will be added in the future.

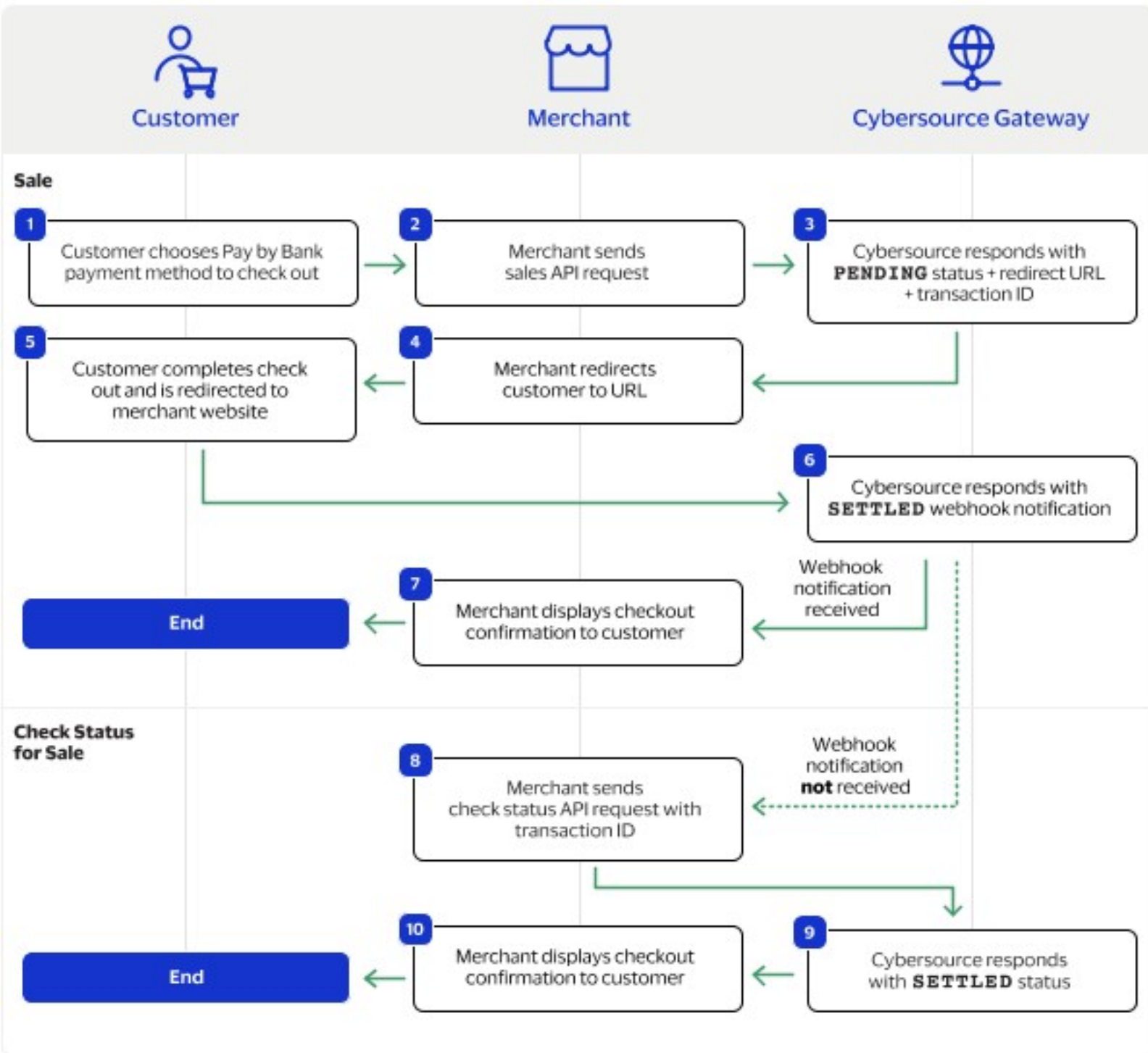
## Endpoints

Send API service requests to one of these endpoints:

- Production: <https://ics2ws.ic3.com/commerce/1.x/transactionProcessor>
- Test: <https://ics2wstest.ic3.com/commerce/1.x/transactionProcessor>

## Sale Workflow

This workflow describes the sequence of events that comprises a successful Pay by Bank sale.



### Processing Sale Workflow

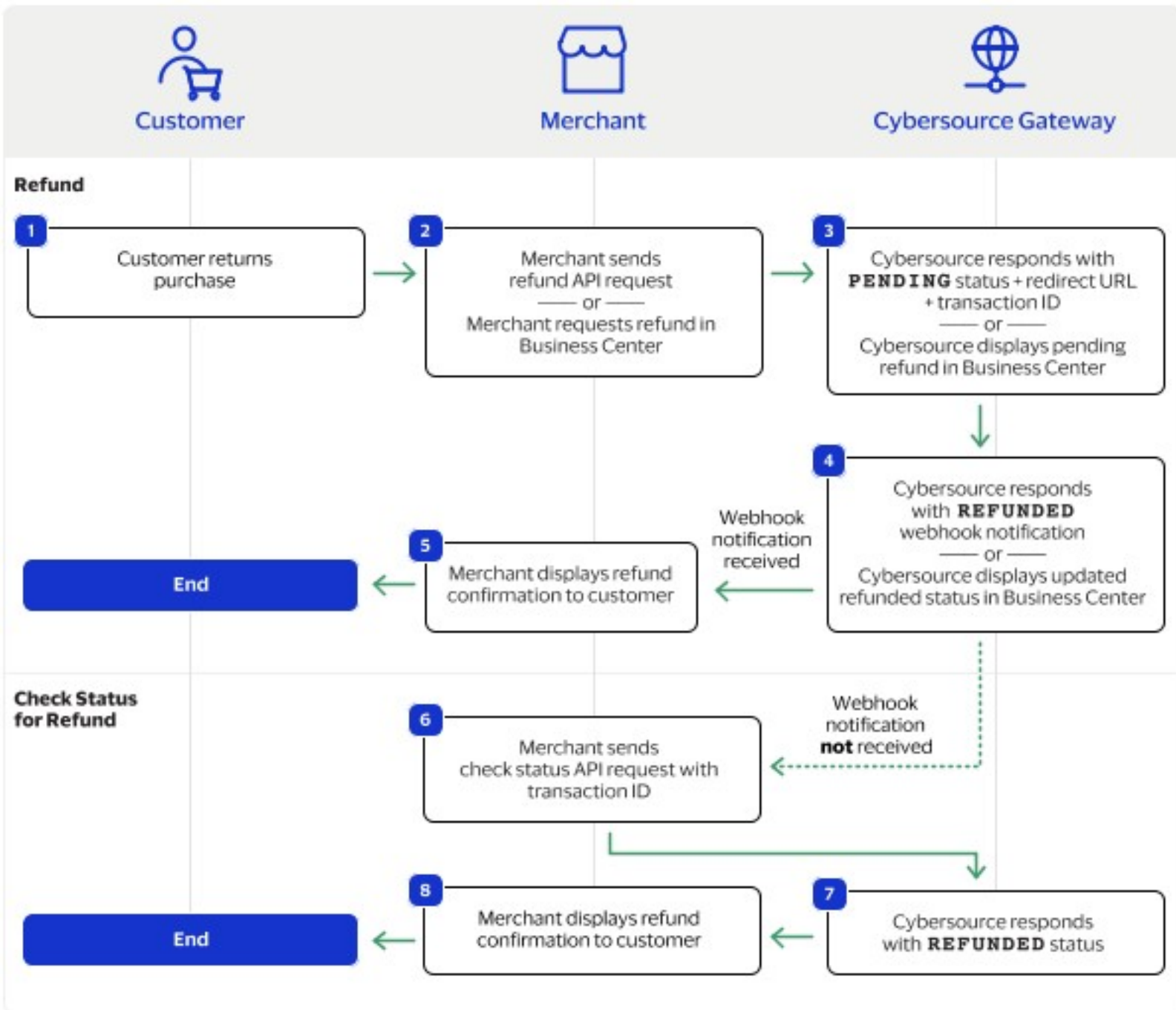
1. The customer chooses the Pay by Bank payment option and their bank while checking out on the merchant's website.
2. The merchant sends Cybersource a sale API request. For more information, see [Sale](#) on page 14.
3. Cybersource responds to the merchant with a **PENDING** status and a redirect URL.



4. The merchant redirects the customer to the URL.
5. The customer completes checkout using their account credentials and is then redirected to the merchant website.
6. Cybersource sends a **SETTLED** sale status webhook notification to the merchant.
7. The merchant displays a payment completion confirmation to the customer.
8. To check the status of a sale, the merchant sends a check status API request if they did not receive a webhook notification. For more information, see [Check Status](#) on page 21.
9. Cybersource sends the merchant one of these statuses:
  - **ABANDONED**: The customer did not complete checkout within the allotted time, and the redirect URL has expired.
  - **PENDING**: The customer did not complete checkout.
  - **SETTLED**: The sale request is settled for the requested amount.
10. The merchant confirms the completed checkout to the customer when the sale status updates to **SETTLED**.

## Refund Workflow

This workflow describes the sequence of events that comprises a successful Pay by Bank refund.



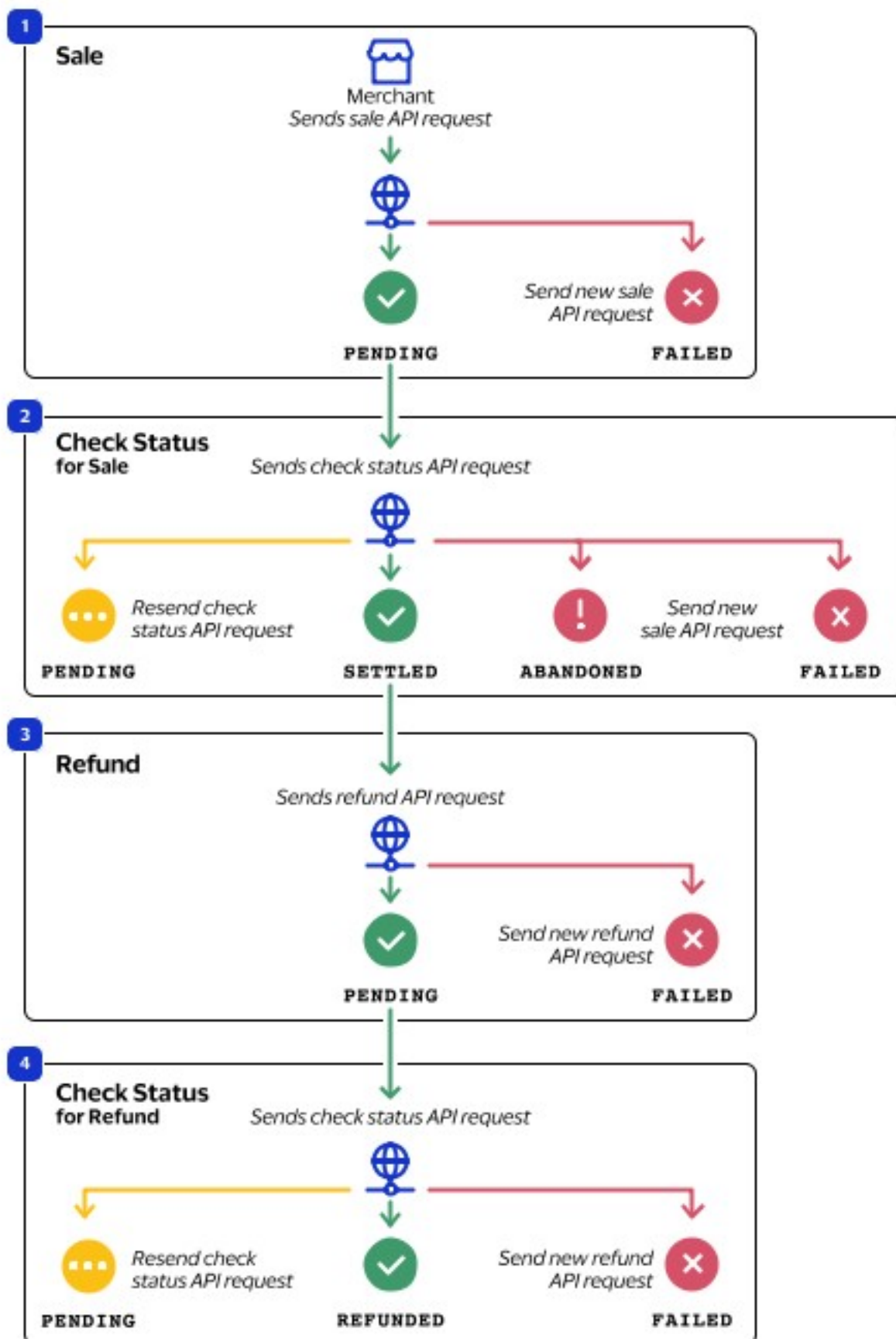
### Issuing a Refund Workflow

1. The customer returns a purchased item to the merchant.
2. The merchant issues a refund in one of two ways:
  - By sending a refund API request to Cybersource. For more information, see [Refund](#) on page 19.
  - By using the Business Center to issue a refund.
3. Cybersource sends a response status in one of two ways:

- By sending a **PENDING** response status if the merchant sent a refund API request.
  - By displaying a pending refund in the Business Center if the merchant used the Business Center to issue the refund.
4. Cybersource sends an updated status when the refund has completed processing in one of two ways:
    - Sending a **REFUNDED** webhook notification if the merchant sent a refund API request.
    - Displaying a completed refund if the merchant used the Business Center to issue the refund.
  5. The merchant displays a confirmation of the refund to the customer.
  6. To check the status of a refund, the merchant sends a check status API request if they did not receive either a webhook notification or an updated status in the Business Center. For more information, see [Check Status](#) on page 21.
  7. Cybersource responds with one of these statuses:
    - **FAILED**: The refund request failed. A failed request can be due to Pay by Bank rejecting the transaction or due to a technical error.
    - **PENDING**: The refund request is accepted but is not completed. Request the check status service to retrieve status updates.
    - **REFUNDED**: The refund amount is successfully refunded.
  8. The merchant confirms the completed refund to the customer when the status updates to **REFUNDED**.

## Transaction Statuses Workflow

This workflow describes the sequence of possible statuses you can receive when processing a Pay by Bank transaction.



### Transaction Statuses Workflow

1. The merchant sends a sale API request to Cybersource and receives one of these possible statuses:

- **PENDING**: The sale request is accepted but the customer has not completed checkout. A webhook notification informs you when the customer completes checking out and the sale settles. See [Webhook Subscriptions](#) on page 23. If you do not receive a webhook notification, send a check status request. See [Check Status](#) on page 21.
  - **REJECT**: The sale is not successful. Send a new sale request.
2. The merchant sends a check status API request to Cybersource to verify the status of the sale and receives one of these statuses:
- **ABANDONED**: The customer did not complete checkout before the redirect URL expired.
  - **COMPLETED**: The customer completed the checkout process and the sale is currently under review. You can send the check status request in 60-minute intervals to retrieve the current status.
  - **PENDING**: The sale request is accepted but is not complete. You can send the check status request in 60-minute intervals until the status updates.
  - **REJECT**: The sale request is not successful. An unsuccessful request can be due to either Pay by Bank rejecting the transaction or a technical error.
  - **SETTLED**: The sale is complete for the requested amount.
3. If the customer decides to return the purchase, the merchant sends a refund API request to Cybersource and receives one of these statuses:
- **PENDING**: The refund request is accepted and is processing. A webhook notification informs you when the refund is complete. See [Webhook Subscriptions](#) on page 23. If you do not receive a webhook notification, send a check status request. See [Check Status](#) on page 21.
  - **REJECT**: The refund request is not successful. Send a new refund request.
4. The merchant sends a check status API request to Cybersource to verify the refund status and receives one of these statuses:
- **PENDING**: The refund request is accepted but is not complete. Continue to request the check status service in 60-minute intervals until the status updates.
  - **REFUNDED**: The settled amount is successfully refunded back to the customer's account.
  - **REJECT**: The refund request is not successful. An unsuccessful request can be due to either Pay by Bank rejecting the transaction or a technical error.



# Pay by Bank Services

This section describes how to successfully request the sale, refund, and check status services. For more information, see these sections:

- [Sale](#)
- [Refund](#)
- [Check Status](#)

You can also subscribe to receive webhook notifications to know when a sale or refund status updates. For more information, see [Webhook Subscriptions](#) on page 23.

## Sale

Send a sale request to begin processing a payment. A response to a successful request includes a pending status and a URL in the **apSaleReply\_merchantURL** field. Redirect the customer to the URL to allow the customer to complete the payment on their bank's website.

After sending a successful request, you can send a follow-on check status request or a refund request. These follow-on services require the request ID from the sale response, which is the **requestID** field value.

Pay by Bank requires that you include line items in your sale request. *Line items* are used to include information about the goods and services that your customers purchase, such as product name, quantity, and price.

## Endpoints

Set the **apSaleService\_run** field to `true`, and send the request to one of these endpoints:

Production: `https://ics2ws.ic3.com/commerce/1.x/transactionProcessor`

Test: `https://ics2wstest.ic3.com/commerce/1.x/transactionProcessor`

## Response Statuses

Cybersource responds to your sale request with one of these statuses in the **apSaleReply\_status** field:

- **PENDING:** The sale request is accepted but the customer has not completed checkout. A webhook notification informs you when the customer completes checking out and the sale settles. See [Webhook Subscriptions](#) on page 23. If you do not receive a webhook notification, send a check status request. See [Check Status](#) on page 21.
- **REJECT:** The sale is not successful. Send a new sale request.

Cybersource also responds with a reason code in the **apSaleReply\_reasonCode** field.

For more information about reason codes, see [Reason Codes and Pay by Bank Response Codes](#) on page 27.

If you send a check status request to retrieve the current sale status, these are the possible responses:

- **ABANDONED:** The customer did not complete checkout before the redirect URL expired.
- **COMPLETED:** The customer completed the checkout process and the sale is currently under review. You can send the check status request in 60-minute intervals to retrieve the current status.
- **PENDING:** The sale request is accepted but is not complete. You can send the check status request in 60-minute intervals until the status updates.
- **REJECT:** The sale request is not successful. An unsuccessful request can be due to either Pay by Bank rejecting the transaction or a technical error.
- **SETTLED:** The sale is complete for the requested amount.

## Line Items

Line items are represented as the **item\_#\_** fields, starting with **item\_0\_** and increasing in numerical order.

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

<b>item_#_productName</b>	Replace the <b>#</b> character with the number <b>0</b> for the first item and consecutive numbers for any additional items.
<b>item_#_quantity</b>	Replace the <b>#</b> character with the number <b>0</b> for the first item and consecutive numbers for any additional items.
<b>item_#_totalAmount</b>	Replace the <b>#</b> character with the number <b>0</b> for the first item and consecutive numbers for any additional items.
<b>item_#_unitPrice</b>	Replace the <b>#</b> character with the number <b>0</b> for the first item and consecutive numbers for any additional items.

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

**item\_#\_productCode**  
**item\_#\_productDescription**  
**item\_#\_productSKU**

## item\_#\_taxAmount

Including Line Items in a Service Request

This example shows three valid line items with optional fields.

```
<item id="0">
  <unitPrice>10.00</unitPrice>
  <quantity>1</quantity>
  <productName>Pants</productName>
  <productSKU>123456</productSKU>
  <totalAmount>10.00</totalAmount>
</item>
<item id="1">
  <unitPrice>15.00</unitPrice>
  <quantity>3</quantity>
  <productName>Pants</productName>
  <productSKU>7654321</productSKU>
  <totalAmount>45.00</totalAmount>
</item>
<item id="2">
  <unitPrice>30.00</unitPrice>
  <quantity>2</quantity>
  <productName>Dress</productName>
  <productSKU>987654</productSKU>
  <totalAmount>60.00</totalAmount>
</item>
```

## Processing a Sale

**Processing a sale requires you to use information from the API response message to redirect the customer to their bank's website to complete the checkout. Follow these steps to process a sale.**

1. Send a **POST** request to the <https://ics2ws.ic3.com/commerce/1.x/transactionProcessor> endpoint and include these required fields:

**apPaymentType**

Set to **PBB**.

**apSaleService\_cancelURL**

Set to the URL to which the customer is redirected after cancelling the Pay by Bank payment.

**apSaleService\_failureURL**

Set to the URL to which the customer is redirected after the Pay by Bank payment fails.

**apSaleService\_run**

Set to **true**.

**apSaleService\_successURL**

Set to the URL to which the customer is redirected after successfully completing the Pay by Bank payment.

**billTo\_email**

**billTo\_firstName****billTo\_lastName****invoiceHeader\_merchantDescriptor****item\_#\_productName**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

**item\_#\_quantity**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

**item\_#\_totalAmount**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

**item\_#\_unitPrice**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

**merchantID****merchantReferenceCode****purchaseTotals\_currency**

Set to either GBP or EUR.

**purchaseTotals\_grandTotalAmount**

2. Choose from these optional fields to include additional information in your request:

**apSaleService\_dateCollect****billTo\_city****billTo\_country****billTo\_street1****fundTransfer\_accountNumber****item\_#\_productCode**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

**item\_#\_productDescription**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

**item\_#\_productSKU**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

**item\_#\_taxAmount**

Replace the # character with the number 0 for the first item and consecutive numbers for any additional items.

3. Redirect the customer to the URL in the **apSaleReply\_merchantURL** response field.

```
<merchantURL>https://checkout.b.banked.com/75ca2d53-58bd-434f-a889-bc92d2902dd4</merchantURL>
```

4. When the customer validates the purchase on their bank's website, the customer is redirected to the merchant's website that you provided in the **apSaleService\_successURL** request field.

```
<successURL>https://www.merchant.redirect.url.from.request.html?actionsuccess</successURL>
```

**When the sale is successfully processed, Cybersource sends a **SETTLED** webhook notification. For information about how to set up webhook subscriptions, see [Webhook Subscriptions](#) on page 23. If you do not receive a webhook notification, send a check status request. For information about how to send a check status request, see [Check Status](#) on page 21.**

## Simple Order Example: Processing a Sale

### Request

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.224">
  <merchantID>test_merchant</merchantID>
  <merchantReferenceCode>refnum-1234</merchantReferenceCode>
  <invoiceHeader>
    <merchantDescriptor>Test Store</merchantDescriptor>
  </invoiceHeader>
  <billTo>
    <firstName>John</firstName>
    <lastName>Smith</lastName>
    <email>test@cybs.com</email>
  </billTo>
  <item id="0">
    <unitPrice>25.00</unitPrice>
    <quantity>2</quantity>
    <productName>Shirt</productName>
    <totalAmount>50.00</totalAmount>
  </item>
  <purchaseTotals>
    <currency>GBP</currency>
    <grandTotalAmount>50.00</grandTotalAmount>
  </purchaseTotals>
  <apPaymentType>PBB</apPaymentType>
  <apSaleService run="true">
    <cancelURL>https://www.merchant.redirect.url.from.request.html?actioncancel</cancelURL>
    <successURL>https://www.merchant.redirect.url.from.request.html?actionsuccess</successURL>
    <failureURL>https://www.merchant.redirect.url.from.request.html?actionfail</failureURL>
  </apSaleService>
</requestMessage>
```

### Response to a Successful Request

```
<replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.224">
```



```

<merchantReferenceCode>refnum-1234</merchantReferenceCode>
<requestID>7320611088736718403012</requestID>
<decision>ACCEPT</decision>
<reasonCode>100</reasonCode>
<requestToken>AxjnrwSTjNnjJvI5t7XEA0oZYjWmdLnDlR2MyenvxnP5VudCpnQ5IZNJMrjQ3YIScZs8ZN5HNva4gAA0Qwd</requestToken>
<purchaseTotals>
  <currency>GBP</currency>
</purchaseTotals>
<apSaleReply>
  <reasonCode>100</reasonCode>
  <paymentStatus>pending</paymentStatus>
  <responseCode>000001</responseCode>
  <merchantURL>https://checkout.b.banked.com/eu/1a6a306d-2582-4fe8-9f52-a8a5c44725f3?token=eyJhbGciOi...</merchantURL>
  <processorTransactionID>1a6a306d-2582-4fe8-9f52-a8a5c44725f3</processorTransactionID>
  <reconciliationID>XFZ3Y3CJG1LO</reconciliationID>
  <amount>50.00</amount>
  <processorResponse>000001</processorResponse>
  <dateTime>2024-11-20T00:05:12Z</dateTime>
</apSaleReply>
</replyMessage>

```

## Refund

Send a refund request to issue a full or partial refund to the customer's bank account. Refunds can be requested only when the sale is settled. A refund request requires the request ID from the sale response.



### Important

You can only refund payments processed with GBP currency.

## Endpoints

Set the **apRefundService\_run** field to **true**, and send the request to one of these endpoints:

Production: <https://ics2ws.ic3.com/commerce/1.x/transactionProcessor>

Test: <https://ics2wstest.ic3.com/commerce/1.x/transactionProcessor>

## Response Statuses

Cybersource responds to your refund request with one of these statuses in the **apRefundReply\_status** field:

- **PENDING**: The refund request is accepted and is processing. A webhook notification informs you when the refund is complete. See [Webhook Subscriptions](#) on page 23. If you do not receive a webhook notification, send a check status request. See [Check Status](#) on page 21.
- **REJECT**: The refund request is not successful. Send a new refund request.

Cybersource also responds with a reason code in the **apRefundReply\_reasonCode** field. For more information about reason codes, see [Reason Codes and Pay by Bank Response Codes](#) on page 27.

If you send a check status request to retrieve the current refund status, these are the possible responses:

- **PENDING**: The refund request is accepted but is not complete. Continue to request the check status service in 60-minute intervals until the status updates.
- **REFUNDED**: The settled amount is successfully refunded back to the customer's account.
- **REJECT**: The refund request is not successful. An unsuccessful request can be due to either Pay by Bank rejecting the transaction or a technical error.

## Required Fields for Processing a Refund

<b>apPaymentType</b>	Set to <b>PBB</b> .
<b>apRefundService_refundRequestID</b>	Set to the requestID field value from the sale response.
<b>apRefundService_run</b>	Set to <b>true</b> .
<b>merchantID</b>	
<b>merchantReferenceCode</b>	
<b>purchaseTotals_currency</b>	Set to either <b>GBP</b> or <b>EUR</b> .
<b>purchaseTotals_grandTotalAmount</b>	

## Simple Order Example: Issuing a Refund

Request

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.224">
  <merchantID>test_merchant</merchantID>
  <merchantReferenceCode>refnum-1234</merchantReferenceCode>
  <purchaseTotals>
    <currency>GBP</currency>
    <grandTotalAmount>50.00</grandTotalAmount>
  </purchaseTotals>
  <apPaymentType>PBB</apPaymentType>
  <apRefundService run="true">
    <refundRequestID>7320611088736718403012</refundRequestID>
  </apRefundService>
</requestMessage>
```

Response to a Successful Request

```
<replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.224">
  <merchantReferenceCode>refnum-1234</merchantReferenceCode>
  <requestID>7320623649276944803009</requestID>
  <decision>ACCEPT</decision>
  <reasonCode>100</reasonCode>
</replyMessage>
```

```

<requestToken>AxjnrwSTjNoPxrDkGWTBAOoZYjWmdpnD1WJL1snvxnQUrGdB8yHJDJpJ16MaG7BCTjNoMA4uTuwaFAAAbwGm</requestToken>
<purchaseTotals>
  <currency>GBP</currency>
</purchaseTotals>
<apRefundReply>
  <reasonCode>100</reasonCode>
  <transactionID>d88ba72d-0493-4f7e-a6a1-33cc9536f8bf</transactionID>
  <status>PENDING</status>
  <processorResponse>000001</processorResponse>
  <amount>50.00</amount>
  <dateTime>2024-11-20T00:26:06Z</dateTime>
  <reconciliationID>XFZ3Z3CJXI96</reconciliationID>
  <returnRef>21XZV304L8GA</returnRef>
  <paymentStatus>pending</paymentStatus>
  <responseCode>000001</responseCode>
</apRefundReply>
</replyMessage>

```

## Check Status

Send a check status request to retrieve status updates for either a pending sale or refund. A check status request requires the request ID from the sale response or the refund response, which is the **requestID** field value. Cybersource recommends using the check status service only when you do not receive a webhook notification. To set up a webhook subscription, see [Webhook Subscriptions](#) on page 23.

## Endpoints

Set the **apCheckStatusService\_run** field to **true**, and send the request to one of these endpoints:

Production: <https://ics2ws.ic3.com/commerce/1.x/transactionProcessor>

Test: <https://ics2wstest.ic3.com/commerce/1.x/transactionProcessor>

## Response Statuses

Cybersource responds to your check sale status request with one of these statuses in the **apCheckStatusReply\_status** field:

### Check Status for a Sale

- **ABANDONED**: The customer did not complete checkout within the allotted time, and the redirect URL has expired.
- **COMPLETED**: The customer completed the checkout process.
- **PENDING**: The sale request is accepted but is not complete. Request the check status service to retrieve status updates.

- **REJECT**: The sale request failed. A failed request can be due to either Pay by Bank rejecting the transaction or due to a technical error.
- **SETTLED**: The sale request is settled for the requested amount.

### Check Status for a Refund

- **PENDING**: The refund request is accepted but is not complete. Request the check status service to retrieve status updates.
- **REFUNDED**: The settled amount is successfully refunded.
- **REJECT**: The refund request failed. A failed request can be due to either Pay by Bank rejecting the transaction or due to a technical error.

The check status service also responds with a reason code in the **apCheckStatusReply\_reasonCode** field. For more information about reason codes, see [Reason Codes and Pay by Bank Response Codes](#) on page 27.

## Required Fields for a Check Status Request

<b>apCheckStatusService_checkStatusRequestID</b>	Set to the requestID field value from the sale response.
<b>apCheckStatusService_run</b>	Set to <b>true</b> .
<b>apPaymentType</b>	Set to <b>PBB</b> .
<b>merchantID</b>	
<b>merchantReferenceCode</b>	

## Simple Order Example: Requesting a Check Status

Request a Check Status for a Sale

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.224">
  <merchantID>test_merchant</merchantID>
  <merchantReferenceCode>refnum-1234</merchantReferenceCode>
  <apPaymentType>PBB</apPaymentType>
  <apCheckStatusService run="true">
    <checkStatusRequestID>7320611088736718403012</checkStatusRequestID>
  </apCheckStatusService>
</requestMessage>
```

Response to a Successful Check Status Request for a Sale

```
<replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.224">
  <merchantReferenceCode>refnum-1234</merchantReferenceCode>
```

```

<requestID>7320543959006044803009</requestID>
<decision>ACCEPT</decision>
<reasonCode>100</reasonCode>
<requestToken>AxjnrwSTjNj0qM/TDM/
BAOoZYjWmdpnd1WJLVunvxnOenedB8iHJDJpJl6MaG7BCTjNjg48YhEDyDAAAowJl</requestToken>
<apCheckStatusReply>
  <reasonCode>100</reasonCode>
  <reconciliationID>XFZ3Z3CJXI57</reconciliationID>
  <paymentStatus>settled</paymentStatus>
  <processorResponse>00004</processorResponse>
</apCheckStatusReply>
</replyMessage>

```

#### Request a Check Status for a Refunded Payment

```

<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.224">
  <merchantID>test_merchant</merchantID>
  <merchantReferenceCode>refnum-1234</merchantReferenceCode>
  <apPaymentType>PBB</apPaymentType>
  <apCheckStatusService run="true">
    <checkStatusRequestID>7320623649276944803009</checkStatusRequestID>
  </apCheckStatusService>
</requestMessage>

```

#### Response to a Successful Check Status Request for a Refunded Payment

```

<replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.218">
  <merchantReferenceCode>refnum-1234</merchantReferenceCode>
  <requestID>7309750023736565003011</requestID>
  <decision>ACCEPT</decision>
  <reasonCode>100</reasonCode>
  <requestToken>AxjnrwSTjEMo5d95JXMDAOoZYjWmjBlOc2GzaqnvxgP4UudB8yGxDJpJl6MZA97iTjDn1zIjYEDeAAAA4AIy</
requestToken>
  <apCheckStatusReply>
    <reasonCode>100</reasonCode>
    <reconciliationID>XFZ402N9X66U</reconciliationID>
    <paymentStatus>refunded</paymentStatus>
    <processorResponse>00004</processorResponse>
  </apCheckStatusReply>
</replyMessage>

```

## Webhook Subscriptions

Webhooks are automated notifications generated by system events that occur in your organizations. You can enroll in a sale or refund event and designate a URL to receive notifications when the event updates. You set up your webhook subscriptions only once.

### Important

Webhooks are supported only with the REST API. If you are integrating with the Pay by Bank Simple Order API, consider the APIs when integrating webhooks.



Notifications that contain sensitive, personally identifiable information such as account numbers are sent using message-level encryption.  
Transport Layer Security is required in order to ensure data integrity.

## Endpoints

- Production: <https://api.cybersource.com/notification-subscriptions/v2/webhooks>
- Test: <https://apitest.cybersource.com/notification-subscriptions/v1/webhooks>

## Notification Statuses

Webhook subscriptions send these status notifications:

- **ABANDONED**: The customer did not complete the transaction.
- **CANCELLED**: The customer did not complete the checkout using the redirect URL.
- **COMPLETED**: The transaction is complete.
- **FAILED**: The payment failed.
- **PENDING**: The payment was initiated.
- **REFUNDED**: The settled amount is successfully refunded.
- **SETTLED**: The sale request is settled for the requested amount.

## Additional Requirements

Webhooks API Developer Guide

There are additional requirements for implementing webhooks that are not discussed in this guide. For more information, see the [Webhooks Implementation Guide](#) for the REST API.

Create New Webhooks

To create an alternative payment notification webhook, see the [Create a Webhook](#) section in the Cybersource REST API Reference.

## Required Fields for Subscribing to a Webhook

**healthCheckUrl**

**organizationId**

**products.eventTypes**

**Set to** `payments.payments.updated`.

**products.productId**

**Set to** `alternativePaymentMethods`.

**webhookUrl**

## REST Example: Subscribing to a Webhook



### Important

There are additional requirements for implementing webhooks that are not discussed in this guide. For more information, see the [Webhooks Implementation Guide](#) for the REST API.

### Request

```
{
  "name": "My Custom Webhook",
  "description": "Sample Webhook from Developer Center",
  "organizationId": "organizationId",
  "productId": "alternativePaymentMethods",
  "eventTypes": [
    "payments.payments.updated"
  ],
  "webhookUrl": "https://MyWebhookServer.com:8443/simulateClient",
  "healthCheckUrl": "https://MyWebhookServer.com:8443/simulateClientHealthCheck",
  "notificationScope": "SELF",
  "retryPolicy": {
    "algorithm": "ARITHMETIC",
    "firstRetry": 1,
    "interval": 1,
    "numberOfRetries": 3,
    "deactivateFlag": "false",
    "repeatSequenceCount": 0,
    "repeatSequenceWaitTime": 0
  },
  "securityPolicy": {
    "securityType": "KEY",
    "proxyType": "external"
  }
}
```

### Encrypted Response to Successful Request

```
{
  "eventType": "payments.payments.updated",
  "productId": "alternativePaymentMethods",
  "organizationId": "organizationId",
  "transactionTraceId": "6584251287300178228949",
  "payload": "{CBC}nL1LK67IH1LQKad3qyhi1PoeDGrHo/6C1IZ17HSC5YX9TQFIgCOhVPi139kmB/xiVCALGdU8YjqmOEwo3iUXKy6oq517eDiTHv/Crq0pOiEzs0dtuTvYhriXrpsVWMwyRK9yOLwihWj31C1yPD/1xA7xpTGp9kX3ABKzzZxKj+Hs+cU/mYzLYxfN7GSDSbqZwhbJgXaKapT81MeXUT1PejyqGyCPagWNqZJFHx6q10BJKB8XZ0pZIPT+v+bMe5EnbQyv9Nj6BJX1BBgqhvKmXrSqD16",
  "metadata": {
    "notificationTypeID": "5",
    "notificationReason": "payment"
  }
}
```

## Decrypted Response to Successful Request

```
{
  "id": "6584251287300178228949",
  "reconciliationID": "85100194sale",
  "status": "settled",
  "submitTimeUTC": "2022-07-21T17:38:48Z",
  "clientReferenceInformation.code": "TC85100-1",
  "orderInformation.amountDetails.totalAmount": "1999.99",
  "orderInformation.amountDetails.currency": "GBP"
}
```

# Reason Codes and Pay by Bank Response Codes

This table describes some of the reason codes and Pay by Bank response codes that you receive when you request a Pay by Bank service. Use the values listed in the table to determine the transaction result.

For additional reason codes and descriptions, see the [Reason Codes for the Simple Order API Developer Guide](#) in the Cybersource Developer Center.

## Reason Codes and Pay by Bank Response Codes

Reason Code	Pay by Bank Response Code	Description
100	00000: Completed status 00001: Pending status 00002: Abandoned status 00003: Authorized status 00004: Settled status 00006: Refunded status	Successful transaction.
102	10000: Failed status	One or more fields in the request contain invalid data.
150	20000: Failed status 20001: Failed status 20002: Failed status	Possible reasons: The signature was not included in the HTTP header. The signature in the HTTP header has expired, or it is not a valid signature.

Reason Code	Pay by Bank Response Code	Description
203	30000: Failed status 30100: Failed status 30200: Failed status 30400: Failed status 30500: s status Failed status	Pay by Bank declined the transaction because of funding source problems, or the transaction was flagged as high risk.
204	30350: Failed status	Payment declined because of insufficient funds in the account.
233	30600: Failed status 30700: Failed status	Pay by Bank declined the transaction because of tax errors or government compliance errors.



# Generating Reports Using the Business Center

You can generate various types of reports for your financial and reconciliation data. For more information about how to automate your reports, see the [Reporting Developer Guide](#). For more information about how to use your Business Center account to generate reports, see the [Reporting User Guide](#).

The [Reporting User Guide](#) contains these relevant topics:

- How and When Reports Are Generated
- Downloading Available Reports
- Subscribing to Standard Reports

## Additional Resources

For additional information about how to use the Business Center and work with reports, see these helpful resources.

### Business Center Navigation

For an overview of the various resources available in the Business Center, see this YouTube video:

<https://www.youtube.com/watch?v=UDmAWGHPbWs>

### Getting Started with the Business Center

For a step-by-step demonstration of how to navigate in the Business Center, see this YouTube video:

[https://www.youtube.com/watch?v=2qi\\_g2DParl](https://www.youtube.com/watch?v=2qi_g2DParl)

### Managing Report Subscriptions

For an overview of how to manage report subscriptions in the Downloadable Reports section in the Business Center, see this YouTube video:

<https://www.youtube.com/watch?v=tFlmkXtvxWE>

## Downloading Reports

For an overview of how to download available reports in the Reports section in the Business Center, see this YouTube video:

<https://www.youtube.com/watch?v=EOsIUYjJvmw>