

CyberSource Bill Me Later[®]

Implementation Guide

Simple Order API

January 2007



CyberSource[®]
the power of payment

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 650-965-6000 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>.

For Customer Support, contact the appropriate office:

- Within the United States, go to <http://www.cybersource.com/esupport>.
- Outside the United States, email eu_support@cybersource.com or call +44 0 1189 653 545.

Copyright

© 2007 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

CyberSource, the CyberSource logo, SmartCert, and PaylinX are registered trademarks of CyberSource Corporation in the U.S. and other countries. The Power of Payment, 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

Documentation Changes and Enhancements	v
Chapter 1	
Introduction to Bill Me Later	1
What is Bill Me Later?.....	1
Requirements	1
Processing a Bill Me Later Transaction	1
Getting Started	2
Learn How Bill Me Later Works	2
Set Up Your Banking Relationships	3
Get Your CyberSource Test Account.....	3
Get Your Client SDK Working.....	3
Learn How the Simple Order API Works	4
Implement Your System.....	4
Test Your System.....	4
Go Live.....	4
Chapter 2	
Introduction to the Simple Order API	5
Overview of the API	5
Simple Order API Clients	5
Which Version of the API to Use	6
Using Name-Value Pairs.....	6
Using XML	6
Constructing Requests	6
Parsing Replies	7
Correlating Schema and Name-Value Pair Fields Names	7
Requests	9
Using Items or a Grand Total in the Request.....	9
Using Items	10
Using a Grand Total	11
Using Coupons	11
How Coupons are Processed	12
Coupon Constraints.....	12
Including a Coupon in the Request.....	12
Replies.....	13
Decisions.....	13
Reason Codes.....	14
Missing or Invalid Fields.....	14

Tracking Orders.....	15
Merchant Reference Code	15
Request ID	16
Reconciliation ID	16
Chapter 3	
Simple Order API for Bill Me Later	17
Which Version of the API to Use	17
Requesting a Bill Me Later Authorization	17
Signing Up a Customer for Bill Me Later	18
Sending the Shipping Address and Shipping Amount.....	18
Requesting a Bill Me Later Capture.....	19
Requesting a Bill Me Later Credit.....	20
Processing Partial Captures and Credits	21
FDMS	21
Partial Captures.....	21
Partial Credits.....	21
Paymentech.....	21
Voiding Bill Me Later Transactions	22
API Fields	22
Data Type Definitions.....	22
Request Fields.....	23
Basic Fields.....	23
Additional Authorization Fields for Paymentech.....	37
Reply Fields.....	40
Product Codes.....	45
Example Requests and Replies.....	46
Using Name-Value Pairs.....	46
Using XML	47
Testing.....	49
Going Live	50
Appendix A	
Reason Codes	52
Index	55

Documentation Changes and Enhancements

The following table lists changes made in the last six releases of this document:

Month of Release	Changes
January 2007	<ul style="list-style-type: none">• The ship-to email field is now required. See shipTo_email in Table 3 on page 23.
October 2006	<ul style="list-style-type: none">• Updated the information about when to start using the request token. See the descriptions for the request token fields in Table 3 on page 23 and Table 5 on page 40.
September 2006	<ul style="list-style-type: none">• Updated the information about how long authorizations stay in the CyberSource database and thus how long you have to perform follow-on credits. See “Requesting a Bill Me Later Credit” on page 20.• Updated the information about credits for FDMS to say that it does not support stand-alone credits. See “Requesting a Bill Me Later Credit” on page 20.• Updated the information about missing and invalid fields. See “Missing or Invalid Fields” on page 14.• Added entries for the request token to the tables of API fields. See Table 3 on page 23 and Table 5 on page 40.
January 2006	<ul style="list-style-type: none">• Added a new section called “Getting Started” on page 2.• Added a new section called “Testing” on page 49.• Added a new section called “Going Live” on page 50.
November 2005	<ul style="list-style-type: none">• Updated the information about how long authorizations stay in the CyberSource database and thus how long you have to perform follow-on credits. See “Requesting a Bill Me Later Credit” on page 20.
October 2005	<ul style="list-style-type: none">• Added information about an additional account number (6219930000000000) that you can use to sign up a new customer. See “Signing Up a Customer for Bill Me Later” on page 18.• Added information about voiding Bill Me Later captures and credits. See “Voiding Bill Me Later Transactions” on page 22.• Added information about using coupons. See “Using Coupons” on page 11.• Updated the information about how long authorizations stay in the CyberSource database and thus how long you have to perform follow-on credits. See “Requesting a Bill Me Later Credit” on page 20.

Chapter 1

Introduction to Bill Me Later

This chapter gives an introduction to Bill Me Later® and describes how to get started with your implementation.

What is Bill Me Later?

Bill Me Later is a payment method that allows your customer to make purchases by using an instant line of credit. To process Bill Me Later transactions, you use the CyberSource Credit Card Services, which are part of the CyberSource Internet Commerce SuiteSM (ICS) of services. The implementation has been designed so that processing a Bill Me Later transaction is very similar to processing a credit card, even though no credit card is involved.

Requirements

To use Bill Me Later, you must first sign a contract with i4Commerce®. As part of establishing that relationship, you determine who will underwrite your Bill Me Later transactions. This will be either:

- First Data Merchant Services (FDMS), or
- Paymentech

Processing a Bill Me Later Transaction

CyberSource has implemented the Bill Me Later payment method by using the existing ICS Credit Card Services.

When you want to request a Bill Me Later authorization, you send a request for credit card authorization. Instead of sending a credit card number, you send the customer's Bill Me Later account number. You must also supply additional information specific to the Bill Me Later transaction.

When you are ready to bill the customer, you send a request for a credit card capture. If you fulfill the order immediately when processing the authorization (for example, if you sell electronic software that does not require shipping, you can request authorization and capture at the same time.

Later, if you need to credit a customer who paid using Bill Me Later, you send a request for a credit card credit.

Important If you are using FDMS, currently you may not request the capture for a Bill Me Later transaction and then on the same day request a credit for that transaction. If you need to process a credit, request it at least one day after requesting the capture. Currently if you try requesting both on the same day, FDMS may try to process the credit before the capture, causing your credit to be rejected.

If necessary, you can cancel a Bill Me Later capture or credit. This is called a void and is only possible if you make the request to cancel the capture or credit before CyberSource has sent the information to i4Commerce.

Getting Started

Here are the things you should do to begin your Bill Me Later implementation:

Learn How Bill Me Later Works

Read this chapter to understand how Bill Me Later processing works. Because Bill Me Later transactions are handled like credit card transactions, read the second chapter in the [Credit Card Services Implementation Guide](#) if you would like to know more about credit card processing.

Set Up Your Banking Relationships

- Set up your underwriting relationship with FDMS or Paymentech.
- Consider implementing fraud prevention features. Your fraud rate may affect the fees you pay to your merchant bank or payment processor. If your fraud rate is extremely high, you may lose your merchant bank account. Contact your CyberSource account representative for information about preventing fraud.
- Consider any legal requirements related to accepting Bill Me Later payments. Contact your legal counsel to discuss these requirements.

Get Your CyberSource Test Account

- If you have not done so already, contact CyberSource to obtain your test account and CyberSource merchant ID and to subscribe you to CyberSource's reports. You will receive a login and password for the [Enterprise Business Center](#), a Web portal that lets you manage your orders with CyberSource. You can also download CyberSource's reports there.
- Log in to the test version of the Enterprise Business Center at <https://ebctest.cybersource.com> and look around. Make sure to look at the **Transaction Search** screens. Once you process test transactions, you can search for and view the transactions here.

Get Your Client SDK Working

If you have not integrated with CyberSource before, you will be using CyberSource's Simple Order API, which is the newer of CyberSource's two APIs. Make sure you do not select a client for the SCMP API, which is the older legacy API.

- Choose a Simple Order API CyberSource client (SDK), available on the [client download page](#).
- Read the documentation that accompanies the client, and install and test the client.

Learn How the Simple Order API Works

- Read Chapter 2, “[Introduction to the Simple Order API](#),” on page 5, to learn the basics of how the Simple Order API works and how to use the specific API fields for Bill Me Later transactions.

Implement Your System

- Write code that creates your requests and uses the client SDK to send the requests to and receive replies from CyberSource. See the documentation for your client for information about how to do this. You will be using the request and reply API fields listed in Chapter 3, “[Simple Order API for Bill Me Later](#),” on page 17.
- Read the [Reporting Developer’s Guide](#) to learn about downloading the CyberSource Payment Submission Detail Report, which you can use to determine when to fulfill your orders.

Test Your System

- Send test transactions to CyberSource’s test server. See “[Testing](#)” on page 49 for information.
- To help you get familiar with the Enterprise Business Center, search for and view your test transactions in the test version of the Enterprise Business Center at <https://ebctest.cybersource.com>.

Go Live

- Once you have thoroughly tested your system and are ready to accept production transactions, you can go live. As part of the process, you will need to provide your merchant bank information to CyberSource so that your processor can deposit funds to your account. See “[Going Live](#)” on page 50.
- Your production transactions will be available for you to view in the production version of the Enterprise Business Center at <https://ebc.cybersource.com>.

Chapter 2

Introduction to the Simple Order API

This chapter describes basic information about to use the Simple Order API:

[Overview of the API](#)

[Requests](#)

[Replies](#)

[Tracking Orders](#)

Overview of the API

The CyberSource Simple Order API enables you to access ICS services using either name-value pairs or XML. Depending on the platform you are using and your experience, you might be able to use the CyberSource SOAP interface instead. For more information about the SOAP interface, contact CyberSource Customer Support.

The name-value pair interface is based on the XML schema. This guide uses the name-value pairs when discussing the interface. If you plan to use XML, you can easily translate the name-value pairs in this guide to the corresponding XML elements that you will use (see "[Correlating Schema and Name-Value Pair Fields Names](#)" on page 7).

To ensure the integrity and privacy of the messages that you use to access the ICS services, CyberSource requires that all messages be digitally signed according to the WS-Security standard. CyberSource's clients handle this requirement for you.

Simple Order API Clients

The Simple Order API clients contain the following:

- Client libraries used to communicate with CyberSource and access the ICS services
- Security libraries used to digitally sign the messages
- For SOAP users, the SOAP proxy classes

- Sample code for digitally signing the messages and using the client libraries

For the latest list of available Simple Order API clients and related documentation, see the [Downloads page](#) on the Support Center.

Which Version of the API to Use

CyberSource updates the Simple Order API on a regular basis to introduce new API fields and functionality. With each update, the API receives a new version number (for example, 1.18). To determine the latest version of the API, go to

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

You should use the latest version to take advantage of the full functionality of the ICS services. See the [Simple Order API Release Notes](#) for information about the changes to the API.

When configuring your Simple Order API client, you indicate which version of the API you want to use (see the documentation for your client for instructions).

Using Name-Value Pairs

If you choose name-value pairs, you create a request that includes the required name-value pairs for the service(s) that you want to use. The client digitally signs and sends your request. You only need to create the request message and parse the reply message, which also contains name-value pairs. See “[Example Requests and Replies](#)” on page 46 for an example name-value pair request and reply.

Using XML

If using XML, you create an XML message that contains the information for calling the ICS services you want to use. The client digitally signs and sends your request. You only need to create the request XML message and parse the XML reply message.

Constructing Requests

The XML schema is located at <https://ics2ws.ic3.com/commerce/1.x/transactionProcessor>.

This is also where you send your request (which is discussed in the documentation for your client).

When you are testing and have not gone live with your site, send your requests to <https://ics2wstest.ic3.com/commerce/1.x/transactionProcessor>.

In general, the schema's structure is separated into a request message and a reply message. Inside the request message are elements for the basic invoice information, tender information, and then service-specific information. For example, the information specific to a credit card authorization (which you use for a Bill Me Later authorization) is in the `<ccAuthService>` element.

To indicate in the request that you want to run a service, set the `run` attribute for the service's element to `"true"`. For example, to request credit card authorization, set the `run` attribute for the `<ccAuthService>` element to `"true"`.

Parsing Replies

The reply message includes general information for the entire request, and then information relevant to the results of each service you requested. For example, the reply information relevant to the credit card authorization is in the `<ccAuthReply>` element. For more information on interpreting the reply, see [“Replies”](#) on page 13. Note that the service reply elements (for example, the `<ccAuthReply>` element) contain additional child elements that are not relevant to Bill Me Later but are relevant to credit cards. See the schema for the full list of elements in the reply.

XML replies from CyberSource always contain the namespace prefix `c:`. You need to use an XML parser that supports namespaces.

See [“Example Requests and Replies”](#) on page 46 for an example XML request and reply.

Correlating Schema and Name-Value Pair Fields Names

This guide discusses the API using name-value pairs and not the element names in the XML schema. You can easily translate between the name-value pair field name and the corresponding XML element name.

The relationship between the XML element names and the name-value pair field names is as follows:

- Each name-value pair field name matches the corresponding XML element name.
- The XML schema shows hierarchy with an underscore (`_`) separating the name of the parent element from the name of the child element.

For example, the XML schema has a `<card>` element with several child elements. [Table 1](#) shows the `<card>` child element names in the XML schema, and the corresponding name-value pair field names.

Table 1 Example of Schema Names and Name-Value Pair Field Names

Schema Names	Corresponding Name-Value Pair Field Names
<code><card></code>	
<code><accountNumber></code>	card_accountNumber
<code><expirationMonth></code>	card_expirationMonth
<code><expirationYear></code>	card_expirationYear
<code></card></code>	

The same convention is used for reply fields.

Note If you are using SOAP, the complex types in the XML schema translate to classes of the same name. For example, the `<Card>` complex type in the schema translates to a `Card` class in the SOAP client.

The XML schema also includes several numbered types and elements. These are complex types or elements that you might include more than once in a request. For example, if a customer's order includes more than one item, you include multiple `<item>` elements in your request. Each item is numbered, starting with 0.

The XML schema uses an `id` attribute in the `item`'s opening tag to indicate the number. For example:

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

For the name-value field names, this is represented as **item_0**. Note that in this situation, 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.

[Table 2](#) shows an example of the numbered `<item>` element and the corresponding name-value pair field names. If you are using SOAP, the client contains a corresponding `Item` class.

Table 2 Example of Numbered Schema Names and Name-Value Pair Field Names

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

Requests

A request for an ICS service contains general information and information specific to the services that you request. General information, for example, includes information about you, the merchant; about the customer and their form of payment; about the items the customer is buying; and so on.

Using Items or a Grand Total in the Request

For some services, you must specify the amount of the transaction. You can provide either the individual products the consumer is purchasing or a grand total for the transaction.

Note If you are using CyberSource's Decision Manager, it is recommended that you provide individual item information instead of a grand total for the order. Decision Manager can be configured to use the individual item information to assess the risk of the order and determine if the purchaser is following your business rules. For more information about Decision Manager, see the [Decision Manager Developer's Guide](#).

Using Items

Items are the products that the consumer purchases from you. When you send a request for a service that requires an amount, you can send item-specific information, such as the quantity of each item ordered and the unit price for each item. The items are referred to as **item_0**, **item_1**, **item_2**, and so on. CyberSource uses the information you provide for each item to calculate the grand total for the order.

The values for the **item_#** fields must not contain carets (^) or colons (:) because these characters are reserved for use by the ICS services.

Required Item-Level Fields. If you are using the item-level fields, which fields are required depends on the value you use for **item_#_productCode**. If **item_#_productCode** is one of the following values:

- `default`
- `stored_value`
- a value related to shipping and handling

or if you omit **item_#_productCode**, which causes it to default to the value `default`, then the only required field is **item_#_unitPrice**. If you do not set **item_#_quantity**, it defaults to 1.

If you do not set **item_#_productCode** to one of the values in the previous list, the following fields are required:

- **item_#_unitPrice**
- **item_#_quantity**
- **item_#_productName**
- **item_#_productSKU**

The **item_#_taxAmount** field is always optional.

Specifying Tax. To include tax for an item, use the **item_#_taxAmount** field. This value is the total tax for the entire quantity of that item. In other words, the value is not multiplied by **item_#_quantity**. For example:

```
item_0_unitPrice=10.00
item_0_quantity=5
item_0_taxAmount=4.00
```

The grand total for this transaction is $(10.00 * 5) + 4.00 = 54.00$.

Specifying Freight Charges. To include a shipping and handling charge for the order, you must include an additional item with **item_#_productCode** set to one of the following values:

- shipping_only
- handling_only
- shipping_and_handling

For example:

```
item_0_unitPrice=10.00
item_0_quantity=5
item_0_taxAmount=4.00
item_1_unitPrice=4.95
item_1_quantity=1
item_1_productCode=shipping_only
```

The grand total for this transaction is $(10.00 * 5) + 4.00 + (4.95 * 1) = 58.95$.

Using a Grand Total

Instead of using items, you may send a grand total for the order in the **purchaseTotals_grandTotalAmount** field. This field is useful if you do not have information for each item or do not care about the information for each item and simply want to use a transaction total. If you include **purchaseTotals_grandTotalAmount** in your request, CyberSource uses this value and does not use item-level information to calculate the transaction's grand total. If you provide item-level information, it still appears in the Transaction Detail page in the [Business Center](#).

Important If you include **purchaseTotals_grandTotalAmount** in your request, you may not include the Tax Calculation service as part of the request. For more information about the Tax Calculation service, see the [Tax Calculation Implementation Guide](#).

Using Coupons

You can offer your customers virtual coupons at your Web store. CyberSource defines a coupon as a non-taxable, fixed amount deducted from an order total.

Coupon examples you might implement include:

- Register now and get \$100 off your purchase!
- Spring clearance! Get \$10 off any order!
- Thank you for ordering again within 30 days! We're taking \$5 off your order!

How Coupons are Processed

The following steps outline how CyberSource processes a request with a coupon. Note how tax is calculated if your request also includes CyberSource's Tax Calculation service.

- 1 All the line items are totaled and then the coupon amounts are deducted, resulting in an order subtotal.
- 2 If you request Tax Calculation, tax is calculated for all taxable line items to get an order tax total. The Tax Calculation service ignores coupon line items because they are not taxable.
- 3 The order subtotal and order tax total are added to get an order grand total.
- 4 The order grand total is used by other CyberSource services you requested.

For example, if you requested credit card authorization in the request with the coupon, the authorization service uses the grand total as the amount to authorize.

Coupon Constraints

You cannot use coupons to do the following:

- Apply a discount to a specific item in a multi-item order
- Apply a discount to a specific item before tax is calculated (if taxable)
- Apply a percentage discount

Also, the total coupon amount cannot be greater than the order grand total. Precalculate your order totals before you send your requests to CyberSource so that you do not send orders with negative subtotals. CyberSource returns an error for orders with negative subtotals.

Including a Coupon in the Request

To request a coupon with an order, include in the request an item with the product code set to `coupon`. For example, if your request contains two items, `item_0` and `item_1`, request

a coupon by adding **item_2**. The example below show how to specify a \$10 coupon. The **quantity**, **productName**, and **productSKU** fields are required.

```
item_2_unitPrice=10.00
item_2_quantity=1
item_2_productCode=coupon
item_2_productName=Spring Clearance
item_2_productSKU=349209
```

Replies

The reply gives you the results of your request. To use the reply information, you must integrate it into your system and any other system that uses that data. This includes storing the data and passing it to any other systems that need the information.

Write an error handler to interpret the information that you receive. Do not show the reply information directly to customers. Instead, present an appropriate response that tells customers the result.

Important Because CyberSource may add reply fields and reason codes at any time, proceed as follows:

- You should parse the reply data according to the names of the fields instead of their order in the reply. For more information on parsing reply fields, see the documentation for your client.
 - Your error handler should use the **decision** field to determine the result if it receives a reason code that it does not recognize.
-

Decisions

In the reply, you receive the **decision** field, which summarizes the overall result of your request. Look at this field first to determine your course of action. The **decision** can be one of the following:

- **ACCEPT**: The request succeeded
- **ERROR**: There was a system error
- **REJECT**: One or more of the services was declined

Errors are due to system issues usually unrelated to the content of the request itself. You must design your transaction management system to include a way to correctly handle CyberSource system errors. Depending on which payment processor is handling the transaction, the error may indicate a valid CyberSource system error, or it may indicate a processor rejection because of some type of invalid data. In either case, CyberSource recommends that you do not design your system to endlessly retry sending a transaction in the case of a system error. See the documentation for the CyberSource client (SDK) you are using for more information about how to handle system errors and retries.

Requests can be rejected by CyberSource or the payment processor. To determine the reason for the reject decision, use the **reasonCode** field.

You are charged for all accepted and rejected requests. You are not charged for requests that result in errors.

Reason Codes

After looking at the decision field, use the **reasonCode** field to determine the reason for the decision and decide if you want to take further action.

If the decision was `ERROR`, the **reasonCode** tells you what type of error occurred.

If the decision was `REJECT`, the **reasonCode** tells you the reason for the reject and whether you can take action that might result in a successful order. For descriptions of the reason codes for Bill Me Later, see Appendix A, "[Reason Codes](#)," on page 52.

The `<service>_reasonCode` fields tell you the result of each individual service that you requested. For example, if you request a credit card authorization, you receive **ccAuthReply_reasonCode** in the reply. These fields are useful for debugging your system.

Note CyberSource reserves the right to add new reason codes at any time. If your error handler receives a reason code that it does not recognize, it should use the **decision** field to determine the result.

Missing or Invalid Fields

You are responsible for ensuring that the data that you send to CyberSource is complete (no missing fields) and correct (no invalid data). To do this, verify the data entered on your Web sites and point-of-sale applications before sending the information to CyberSource.

If you send a request with missing or invalid information, you will receive the appropriate reason code(s) and one or more reply fields, **invalidField_0...N** or **missingField_0...N**, which list the fields that you need to correct. The service(s) that you requested and the nature of the missing or invalid information will be used to determine the number and the content of the reply fields. For example, if three required fields are missing from your request, you will receive at least one and up to three reply fields named **missingField_0**, **missingField_1**, and **missingField_2**. You should correct these fields and resubmit the request.

Because the API behavior pertaining to these reply fields is always subject to change, do not use these fields to communicate with consumers.

Note For XML, the `<missingField>` and `<invalidField>` elements are not numbered. Instead, you receive multiple `<missingField>` or `<invalidField>` elements in the reply.

For SOAP, you receive an array of the missing fields and an array of the invalid fields.

Tracking Orders

The API provides different fields that act as identifiers for tracking orders.

Note Use the **merchantReferenceCode** to track your orders, and use the **requestID** to link a follow-on request, such as a capture or credit, to the original request.

Merchant Reference Code

The **merchantReferenceCode** is your own tracking number for an order that you send in the request. Record this number so that you can keep track of the order as it moves through different phases of processing with CyberSource.

For example, you might do the following with a single order:

- Send a request for credit card authorization to authorize a card and check for fraud.
- The next day, ship the item and send a request to capture the authorization.

- A week later, if the customer returns one of the products in the order, request a credit.

For this single order, you send three separate requests to perform different services. If you send the same **merchantReferenceCode** for each of these requests, you can efficiently track the order in the CyberSource reports and transaction search screens.

Note CyberSource recommends that you use a unique merchant reference code for each order.

Request ID

The **requestID** is returned to you in the reply and is a unique identifier that CyberSource assigns to the request. In the example above, you would receive a different **requestID** in each reply.

The **requestID** is useful if you need to discuss a specific request with Customer Support.

Also, you use the **requestID** to link a follow-on request to the original request. For example, when you send a request for a follow-on credit for a credit card, you include the **requestID** from the capture. CyberSource then uses the value to locate the capture information, reducing the amount of information you must provide in the credit request.

Reconciliation ID

For each of the services that you use with Bill Me Later, you receive a unique reconciliation ID in the reply.

Note The reconciliation ID is sometimes also referred to as the transaction reference number.

The ID is assigned by CyberSource, and you use it to reconcile the transactions in your CyberSource reports with orders in your system. This number appears in one of the following fields, depending on the service you requested:

- **ccAuthReply_reconciliationID**
- **ccCaptureReply_reconciliationID**
- **ccCreditReply_reconciliationID**

Chapter 3

Simple Order API for Bill Me Later

This chapter describes the Simple Order API fields for Bill Me Later and includes these sections:

- [Which Version of the API to Use](#)
- [Requesting a Bill Me Later Authorization](#)
- [Requesting a Bill Me Later Capture](#)
- [Requesting a Bill Me Later Credit](#)
- [Processing Partial Captures and Credits](#)
- [Voiding Bill Me Later Transactions](#)
- [API Fields](#)
- [Product Codes](#)
- [Example Requests and Replies](#)
- [Testing](#)
- [Going Live](#)

Which Version of the API to Use

To process Bill Me Later with the Simple Order API, you must use version 1.5 or later of the API. To determine the latest version of the API, go to

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

You specify which version to use when you configure your CyberSource client. See the documentation for your client for information about to do this.

Requesting a Bill Me Later Authorization

To request a Bill Me Later authorization, you use the credit card authorization service, **ccAuthService**. The fields to use with the authorization are in [Table 3](#) on page 23 and [Table 4](#) on page 37. Although the tables indicate that some of the fields are optional, if you have the information available to you when the order is placed, you must provide all of the optional fields in the authorization.

Use the **card_accountNumber** to hold the Bill Me Later account number instead of the standard customer credit card number. See the field descriptions for **card_cardType**, **card_expirationMonth**, and **card_expirationYear** for specific values to use for Bill Me Later. If you plan to include a shipping charge or a shipping address, see “[Sending the Shipping Address and Shipping Amount](#)” on page 18.

Signing Up a Customer for Bill Me Later

When a customer wants to use Bill Me Later for the first time and sign up for an account, you send a credit card authorization request with a special value for **card_accountNumber** that indicates this is a new account. CyberSource accepts either of the numbers below. Use whichever one is appropriate based on your contract with i4Commerce.

- 5049900000000000
- 6219930000000000

In the response you receive the customer’s new account number in the **ccAuthReply_bmlAccountNumber** reply field. Make sure to return that information to the customer for their future use.

CyberSource returns the **ccAuthReply_bmlAccountNumber** field in the reply only if the value being returned differs from the **card_accountNumber** that you supplied in the request.

Sending the Shipping Address and Shipping Amount

If you choose to send the shipping address information, you must include the complete address, or none of the shipping address information. If you send only part of the address, your request will be declined. A complete shipping address includes:

- **shipTo_street1**
- **shipTo_city**
- **shipTo_country**
- **shipTo_state** (if country is U.S. or Canada)
- **shipTo_postalCode** (if country is U.S. or Canada)

If you choose not to send the shipping address information, the billing address information will be used as the shipping address information.

To include a shipping amount in the request, create a separate item especially for the shipping. Set the **item#_unitPrice** field to the shipping amount, and set the **item#_productCode** field to one of the following product codes, depending on which is applicable:

- shipping_only
- handling_only
- shipping_and_handling

See “[Product Codes](#)” on page 45 for a complete list of the possible product code values you can use.

The following example shows the items from a request that includes a shipping charge. Note that **item#_quantity** defaults to 1 if you do not include the field.

```
item_0_unitPrice=12.95
item_0_quantity=4
item_0_productName=DVD
item_1_unitPrice=24.95
item_1_quantity=2
item_1_productName=book
item_2_unitPrice=4.95
item_2_productCode=shipping_only
```

Requesting a Bill Me Later Capture

Once you have shipped the goods, request a capture for the Bill Me Later transaction. To do this, send a request and set **ccCaptureService_run=true**. Also include all of the required fields for a capture request listed in [Table 3](#) on page 23. If you fulfill the order immediately when processing the authorization (for example, if you sell electronic software that does not require shipping), you can request authorization and capture at the same time. Simply set both **ccAuthService_run** and **ccCaptureService_run=true** in the same request, and include the fields required for both services. If both services require the same field, you only need to include the field once in the request.

Requesting a Bill Me Later Credit

If you need to process a refund for a Bill Me Later transaction, request a credit. To do this, send a request and set `ccCreditService_run=true`, and include the other required fields for a credit (see [Table 3](#) on page 23). You may not request the credit service simultaneously with a request for authorization or capture.

Important If you are using FDMS, you may not request the capture for a Bill Me Later transaction and then on the same day request a credit for that transaction. If you need to process a credit, request it at least one day after requesting the capture. If you try requesting both on the same day, FDMS may try to process the credit before the capture, causing your credit to be rejected.

When processing a Bill Me Later credit through Paymentech, you will perform either a follow-on credit or a stand-alone credit. When processing a Bill Me Later credit through FDMS, you will perform a follow-on credit.

A follow-on credit uses stored authorization information from CyberSource's database. Use the `ccCreditService_captureRequestID` field to send the `requestID` value from the associated capture. CyberSource uses the request ID to look up the customer's billing and account information from the original authorization, so you do not have to supply those fields in the credit request. Follow-on credits work only within 60 days of the authorization. After 60 days, the authorization information is no longer in the database. After 60 days, you can perform a stand-alone credit if you are using Paymentech. After 60 days, you cannot perform a credit if you are using FDMS.

For FDMS, if the request includes `ccCreditService_reconciliationID`, CyberSource will not use the value for the reconciliation ID that is stored in the database. Instead, CyberSource will use the value that you provided in the `ccCreditService_reconciliationID` field. Make sure that you send the correct value for this field: It should be the same value that you received in the associated capture's `ccCaptureReply_reconciliationID` field. If you do not use the correct value, FDMS might reject your credit, or you might credit the wrong payment.

A stand-alone credit does not use any stored authorization information from the database. You do not send the `ccCreditService_captureRequestID` field. Instead, you must supply all the customer's billing and account information (see [Table 3](#) on page 23).

If you do not have the logic in your system to determine if 60 days have passed, you can do one of the following:

- Try a follow-on request, and if it is declined because it is too late and the request ID is not recognized, do a stand-alone credit
- Always do a stand-alone credit no matter when you request the credit

Processing Partial Captures and Credits

If you are using either FDMS or Paymentech, you can use partial captures and partial credits. For example, you can do a Bill Me Later authorization for \$100, and then later do two separate captures, each for \$50. You might want to do this if a customer purchases multiple items and you cannot ship both at the same time.

You might want to do partial credits if a customer returns one item out of a multiple-item order.

FDMS

When processing partial captures or credits with FDMS, you must assign a unique identifier to each partial capture or credit associated with the authorization.

Partial Captures

To process multiple partial captures, with each capture request, include the **ccCaptureService_partialPaymentID** field. Set it to a value of your choice that is unique within the scope of the order. For example, set the field to 1 for the first capture, then set the field to 2 for the second capture, and so on. Or use a and b instead of 1 and 2. The values are used to identify the different payments related to the order.

Partial Credits

To process multiple partial credits, with each credit request, include the **ccCreditService_partialPaymentID** field. Set it to a value of your choice that is unique within the scope of the order. Note that if you performed partial payments for this order, you had to specify a unique value for **ccCaptureService_partialPaymentID** for each payment. You must not reuse any of those values when you specify **ccCreditService_partialPaymentID**. For example, if you used 1 and 2 for the partial payments, you should use new values such as 3, 4, and so on for any partial credits. The values are used to identify the different payments and credits related to the order.

Paymentech

To process partial captures or credits, you do not need to include any additional fields in your requests. Simply request each capture or credit for the partial amount. For each partial credit, you can send the request ID from a previous capture to reduce the number of customer information fields that you must supply in the credit request. If you processed multiple captures for the order, then use the request ID of the first capture.

Voiding Bill Me Later Transactions

You request a void when you want to cancel a Bill Me Later capture or credit request that you have submitted to CyberSource. A transaction can be voided only if CyberSource has not already submitted the capture or credit information to i4Commerce. Usually CyberSource submits that type of information once a day, so your window for successfully performing a void is relatively small. CyberSource will decline your void request if the capture or credit information has already been sent to i4Commerce.

When you void a transaction, the transaction is at the end of its life and cannot be the source of another follow-on capture or credit. For example, if you authorize and capture a Bill Me Later transaction, and then you void the capture, you cannot submit another capture request that uses the authorization code or CyberSource request ID from the original authorization. If you still want to capture that transaction, you must re-authorize the transaction and capture the new authorization.

You cannot undo a void. Also, you are not allowed to perform a follow-on credit for a transaction that has been voided.

API Fields

Unless otherwise noted, all of the field names are case sensitive, and the fields accept special characters (for example, @, #, and %).

Note The values of the **item_#_** fields must not contain carets (^) or colons (:), as these characters are reserved for use by the ICS services. The values of all request fields must not contain newlines or carriage returns. However, they can contain embedded spaces and any other printable characters. All leading and trailing spaces will be removed.

Data Type Definitions

For more information about these data types, see the [World Wide Web Consortium \(W3C\) XML Schema Part 2: Datatypes specification](#). The supported encoding type is UTF-8.

- **Integer** — A whole number. {..., -3, -2, -1, 0, 1, 2, 3, ...}
- **String** — Contains letters, numbers, spaces, and special characters (such as @ and #).

Request Fields

To see example requests and replies, see “[Example Requests and Replies](#)” on page 46.

Basic Fields

[Table 3](#) lists the fields that you use with your authorization, capture, credit, or void request. If you are using Paymentech for Bill Me Later transactions, you also use the fields listed in [Table 4](#) when you are requesting an authorization. The tables indicate whether the field is required or optional for each service. Although the tables show some of the fields as optional for the authorization service, if you have the information available to you when the order is placed, you must provide all of the optional fields in your authorization request.

Table 3 Request Fields for Bill Me Later

Request Field	Description	Auth	Capture	Credit	Void	Data Type & Length
billTo_city	City of the billing address.	R	N/A	R (1)	N/A	String (50)
billTo_country	Country of the billing address. Use the two-character country codes .	R	N/A	R (1)	N/A	String (2)
billTo_dateOfBirth	The customer's date of birth. Use format <i>YYYY-MM-DD</i> or <i>YYYYMMDD</i> .	R (2)	N/A	N/A	N/A	String (10)
billTo_email	Customer's email address, including the full domain name (for example, <i>jdoe@example.com</i>).	R	N/A	R (1)	N/A	String (255)
billTo_firstName	Customer's first name. The value should be the same as the one that appears on the card.	R	N/A	R (1)	N/A	String (60)
billTo_hostname	DNS resolved hostname from billTo_ipAddress .	O	N/A	N/A	N/A	String (60)

O = Optional; R = Required; N/A = Not applicable

(1) Optional if **ccCreditService_captureRequestID** is included. See “[Requesting a Bill Me Later Credit](#)” on page 20.

(2) If using Paymentech, the field is required only for the first Bill Me Later transaction for that customer.

(3) Required when country of the address is U.S. or Canada

Table 3 Request Fields for Bill Me Later (Continued)

Request Field	Description	Auth	Capture	Credit	Void	Data Type & Length
billTo_httpBrowserType	Customer's browser as identified from the HTTP header data. For example, Mozilla is the value that identifies the Netscape browser.	O	N/A	N/A	N/A	String (40)
billTo_ipAddress	IP address of the customer (for example, 10.1.27.63).	O	N/A	N/A	N/A	String (15)
billTo_lastName	Customer's last name.	R	N/A	R (1)	N/A	String (60)
billTo_phoneNumber	Customer's phone number.	O	N/A	O	N/A	String (15)
billTo_postalCode	Postal code of the billing address. The field must contain between five and nine digits. If the value of billTo_country is CA, the number of characters in billTo_postalCode must follow these rules: <ul style="list-style-type: none"> • If the number of characters is greater than three, the first three characters must be of the format [alpha][numeric][alpha]. • If the number of characters is seven, the last three characters must be of the format [numeric][alpha][numeric]. 	See footnote 3	N/A	See footnotes 3 and 1	N/A	String (10)

O = Optional; R = Required; N/A = Not applicable

(1) Optional if **ccCreditService_captureRequestID** is included. See ["Requesting a Bill Me Later Credit"](#) on page 20.

(2) If using Paymentech, the field is required only for the first Bill Me Later transaction for that customer.

(3) Required when country of the address is U.S. or Canada

Table 3 Request Fields for Bill Me Later (Continued)

Request Field	Description	Auth	Capture	Credit	Void	Data Type & Length
billTo_ssn	The customer's Social Security number. Do not include dashes. If using Paymentech, use format NNNNNNNNNN. If using FDMS, include only the last four digits of the number.	R ⁽²⁾	N/A	N/A	N/A	String (9)
billTo_state	State or province of the billing address. Use the two-character codes .	See footnote 3	N/A	See footnotes 3 and 1	N/A	String (2)
billTo_street1	First line of the billing street address.	R	N/A	R ⁽¹⁾	N/A	String (60)
billTo_street2	Second line of the billing street address. Used for additional address information, for example: Attention: Accounts Payable	O	N/A	N/A	N/A	String (60)
bml_customerRegistrationDate	Date the customer first registered at the merchant's site, whether or not they used Bill Me Later. Use format YYYYMMDD.	R ⁽²⁾	N/A	N/A	N/A	String (8)
bml_customerTypeFlag	Whether the customer is a new or existing customer with the merchant. This field can contain one of the following values: <ul style="list-style-type: none"> E: Existing customer N: New customer 	R	N/A	N/A	N/A	String (2)
bml_itemCategory	Product description code assigned by i4Commerce when you signed up for the service.	R	N/A	N/A	N/A	String with numbers only (4)

O = Optional; R = Required; N/A = Not applicable

(1) Optional if **ccCreditService_captureRequestID** is included. See ["Requesting a Bill Me Later Credit"](#) on page 20.

(2) If using Paymentech, the field is required only for the first Bill Me Later transaction for that customer.

(3) Required when country of the address is U.S. or Canada

Table 3 Request Fields for Bill Me Later (Continued)

Request Field	Description	Auth	Capture	Credit	Void	Data Type & Length
bml_merchantPromotionCode	Merchant's promotion code assigned by i4Commerce.	O	N/A	N/A	N/A	String (4)
bml_productDeliveryTypeIndicator	<p>Delivery type for the product. This field can contain one of the following values:</p> <ul style="list-style-type: none"> • <code>shipping_and_handling</code> (for physically delivered products) • <code>electronic_software</code> (for digitally downloaded products) • <code>service</code> (for products that are services) • <code>default</code> (for any other type of delivery) <p>If sending multiple items in the request with different product delivery types, send the value for the item with the largest dollar amount. If all items have equal amounts, send the value that represents the largest number of items.</p>	R	N/A	N/A	N/A	String (30)
bml_tcVersion	The Bill Me Later Terms and Conditions version number the customer agreed to. The field cannot contain any alphabetic characters; do not include a period.	R ⁽²⁾	N/A	N/A	N/A	String with numbers only (5)

O = Optional; R = Required; N/A = Not applicable

(1) Optional if `ccCreditService_captureRequestID` is included. See "[Requesting a Bill Me Later Credit](#)" on page 20.

(2) If using Paymentech, the field is required only for the first Bill Me Later transaction for that customer.

(3) Required when country of the address is U.S. or Canada

Table 3 Request Fields for Bill Me Later (Continued)

Request Field	Description	Auth	Capture	Credit	Void	Data Type & Length
card_accountNumber	Bill Me Later account number. Use either 5049900000000000 or 6219930000000000 to activate a new Bill Me Later account. Use whichever one is appropriate based on your contract with i4Commerce.	R	N/A	R (1)	N/A	String with numbers only (20)
card_cardType	Type of card to authorize. Use 028 for Bill Me Later.	R	N/A	R	N/A	String (3)
card_expirationMonth	Expiration month (<i>MM</i>) of the credit card. Use 12 for Bill Me Later.	R	N/A	R (1)	N/A	String (2)
card_expirationYear	Expiration year (<i>YYYY</i>) of the credit card. Use 2021 for Bill Me later.	R	N/A	R (1)	N/A	String (4)
ccAuthService_run	Whether to include ccAuthService in your request. This field can contain one of the following values: <ul style="list-style-type: none"> <code>true</code>: Include the service in your request <code>false</code> (default): Do not include the service in your request 	R	N/A	N/A	N/A	String (5)
ccCaptureService_authRequestID	Value of the request ID returned in the requestID field of a previous authorization reply.	N/A	R unless auth is called with capture	N/A	N/A	String (26)

O = Optional; R = Required; N/A = Not applicable

(1) Optional if **ccCreditService_captureRequestID** is included. See [“Requesting a Bill Me Later Credit”](#) on page 20.

(2) If using Paymentech, the field is required only for the first Bill Me Later transaction for that customer.

(3) Required when country of the address is U.S. or Canada

Table 3 Request Fields for Bill Me Later (Continued)

Request Field	Description	Auth	Capture	Credit	Void	Data Type & Length
ccCaptureService_authRequestToken	<p>The requestToken value returned from a previous request for ccAuthService. If you request the authorization and capture services together, the capture request does not require a request token. The field is an encoded string that contains no confidential information, such as an account or card verification number. The string may contain up to 256 characters.</p> <p>You can start using the request token immediately, or you can do so at any time during the transition period:</p> <ul style="list-style-type: none"> • After you start using the request token, you must do so for all future follow-on requests. • If you cannot use the request token immediately, CyberSource will store the request token data for you and retrieve it as necessary when you send follow-on requests. However, you will need to start using the request token before the end of the transition period. • You can read the migration guide that describes the request token. 	N/A	R	N/A	N/A	String (256)

O = Optional; R = Required; N/A = Not applicable

(1) Optional if **ccCreditService_captureRequestID** is included. See “[Requesting a Bill Me Later Credit](#)” on page 20.

(2) If using Paymentech, the field is required only for the first Bill Me Later transaction for that customer.

(3) Required when country of the address is U.S. or Canada

Table 3 Request Fields for Bill Me Later (Continued)

Request Field	Description	Auth	Capture	Credit	Void	Data Type & Length
ccCaptureService_partialPaymentID	Merchant-generated value that identifies a specific partial capture associated with an order. Make sure the field value you use for each capture request is unique within the scope of the order. Used only for FDMS. See “Processing Partial Captures and Credits” on page 21.	N/A	R for partial captures	N/A	N/A	String (25)
ccCaptureService_run	Whether to include ccCaptureService in your request. This field can contain one of the following values: <ul style="list-style-type: none"> <code>true</code>: Include the service in your request <code>false</code> (default): Do not include the service in your request 	N/A	R	N/A	N/A	String (5)

O = Optional; R = Required; N/A = Not applicable

(1) Optional if **ccCreditService_captureRequestID** is included. See [“Requesting a Bill Me Later Credit”](#) on page 20.

(2) If using Paymentech, the field is required only for the first Bill Me Later transaction for that customer.

(3) Required when country of the address is U.S. or Canada

Table 3 Request Fields for Bill Me Later (Continued)

Request Field	Description	Auth	Capture	Credit	Void	Data Type & Length
ccCreditService_captureRequestToken	<p>The requestToken value returned from a previous request for ccCaptureService. This field is required only for follow-on credit requests; it is not required for stand-alone credit requests. The field is an encoded string that contains no confidential information, such as an account or card verification number. The string may contain up to 256 characters.</p> <p>You can start using the request token immediately, or you can do so at any time during the transition period:</p> <ul style="list-style-type: none"> • After you start using the request token, you must do so for all future follow-on requests. • If you cannot use the request token immediately, CyberSource will store the request token data for you and retrieve it as necessary when you send follow-on requests. However, you will need to start using the request token before the end of the transition period. • You can read the migration guide that describes the request token. 	N/A	N/A	R	N/A	String (256)

O = Optional; R = Required; N/A = Not applicable

(1) Optional if **ccCreditService_captureRequestID** is included. See “[Requesting a Bill Me Later Credit](#)” on page 20.

(2) If using Paymentech, the field is required only for the first Bill Me Later transaction for that customer.

(3) Required when country of the address is U.S. or Canada

Table 3 Request Fields for Bill Me Later (Continued)

Request Field	Description	Auth	Capture	Credit	Void	Data Type & Length
ccCreditService_reconciliationID	Set this field to the value you received in the ccCaptureReply_reconciliationID field in the capture reply. See “Requesting a Bill Me Later Credit” on page 20.	N/A	N/A	O for FDMS	N/A	String (25)
ccCreditService_run	Whether to include ccCreditService in your request. This field can contain one of the following values: <ul style="list-style-type: none"> <code>true</code>: Include the service in your request <code>false</code> (default): Do not include the service in your request 	N/A	N/A	R	N/A	String (5)

O = Optional; R = Required; N/A = Not applicable

(1) Optional if **ccCreditService_captureRequestID** is included. See [“Requesting a Bill Me Later Credit”](#) on page 20.

(2) If using Paymentech, the field is required only for the first Bill Me Later transaction for that customer.

(3) Required when country of the address is U.S. or Canada

Table 3 Request Fields for Bill Me Later (Continued)

Request Field	Description	Auth	Capture	Credit	Void	Data Type & Length
item_#_productCode	Type of product, which is also used to determine the category that the product falls under (electronic, handling, physical, service, or shipping). The default value is default. See “Product Codes” on page 45 for a list of valid values. See “Sending the Shipping Address and Shipping Amount” on page 18 for information about how to include a shipping charge for the order. For ccAuthService , if you set this to a value other than default, stored_value, or any of the values related to shipping and/or handling, the item_#_quantity , item_#_productName , and item_#_productSKU fields are required.	See field description	O	O	N/A	String (30)
item_#_productName	Name of the product. For ccAuthService , required if item_#_productCode is NOT default, stored_value, or one of the values related to shipping and/or handling.	See field description	O	O	N/A	String (30)
item_#_productSKU	Product’s identifier code. For ccAuthService , required if item_#_productCode is NOT default, stored_value, or one of the values related to shipping and/or handling.	See field description	O	O	N/A	String (15)

O = Optional; R = Required; N/A = Not applicable

(1) Optional if **ccCreditService_captureRequestID** is included. See [“Requesting a Bill Me Later Credit”](#) on page 20.

(2) If using Paymentech, the field is required only for the first Bill Me Later transaction for that customer.

(3) Required when country of the address is U.S. or Canada

Table 3 Request Fields for Bill Me Later (Continued)

Request Field	Description	Auth	Capture	Credit	Void	Data Type & Length
item_#_quantity	Quantity of the product being purchased. The default value is 1. For ccAuthService , required if item_#_productCode is NOT default, stored_value, or one of the values related to shipping and/or handling.	See field description	O	O	N/A	Integer (10)
item_#_taxAmount	Total tax to apply to the product. The item_#_taxAmount field is additive. For example, if you send one item with unitPrice of \$10.00 and taxAmount of \$0.80, and you send another item with unitPrice of \$20.00 and taxAmount of \$1.60, the total amount authorized will be for \$32.40, not \$30.00 with \$2.40 of tax included. The item_#_taxAmount and the item_#_unitPrice must be in the same currency.	O	O	O	N/A	String (15)
item_#_unitPrice	Per-item price of the product. You must include either this field or purchaseTotals_grandTotalAmount in your request. You can include a decimal point (.) in this field, but you cannot include any other special characters. The amount will be truncated at the request level to the correct number of decimal places.	See description			N/A	String (15)

O = Optional; R = Required; N/A = Not applicable

(1) Optional if **ccCreditService_captureRequestID** is included. See "[Requesting a Bill Me Later Credit](#)" on page 20.

(2) If using Paymentech, the field is required only for the first Bill Me Later transaction for that customer.

(3) Required when country of the address is U.S. or Canada

Table 3 Request Fields for Bill Me Later (Continued)

Request Field	Description	Auth	Capture	Credit	Void	Data Type & Length
merchantID	Your CyberSource merchant ID. Use the same merchantID for evaluation, testing, and production.	R	R	R	R	String (30)
merchantReference Code	Merchant-generated order reference or tracking number. See "Tracking Orders" on page 15 for more information.	R	R	R	R	String (50)
purchaseTotals_currency	Currency used for the order. Use the ISO codes . For the ccCaptureService , you must use the same currency that you used in your request for ccAuthService .	R	R	R	N/A	String (5)
purchaseTotals_grandTotalAmount	Grand total for the order. You must include either this field or item_#_unitPrice in your request. See "Using Items or a Grand Total in the Request" on page 9 for more information.		See description		N/A	String (15)
shipTo_city	City to which to ship the product.	O	N/A	N/A	N/A	String (50)
shipTo_country	Country to which to ship the product. For Bill Me Later, this field can contain US, CA, or GB.	O	N/A	N/A	N/A	String (2)
shipTo_email	Email address of the person receiving the shipment (for example, <code>jdoe@example.com</code>).	R	N/A	N/A	N/A	String (50)
shipTo_firstName	First name of the person receiving the shipment.	O	N/A	N/A	N/A	String (60)
shipTo_lastName	First name of the person receiving the shipment.	O	N/A	N/A	N/A	String (60)

O = Optional; R = Required; N/A = Not applicable

(1) Optional if **ccCreditService_captureRequestID** is included. See ["Requesting a Bill Me Later Credit"](#) on page 20.

(2) If using Paymentech, the field is required only for the first Bill Me Later transaction for that customer.

(3) Required when country of the address is U.S. or Canada

Table 3 Request Fields for Bill Me Later (Continued)

Request Field	Description	Auth	Capture	Credit	Void	Data Type & Length
shipTo_phoneNumber	Phone number for the person receiving the shipment. Use format <code>AAEEEEENNNXXXX</code> where <code>A</code> = Area code, <code>E</code> = Exchange, <code>N</code> = Number, and <code>X</code> = Extension.	O	N/A	N/A	N/A	String (14)
shipTo_postalCode	Postal code to which to ship the product.	O ⁽³⁾	N/A	N/A	N/A	String (10)
shipTo_shippingMethod	Shipping method for the product. For example, <code>FEDEX</code> .	O	N/A	N/A	N/A	String (10)
shipTo_state	State or province to which to ship the product. Use the two-character codes .	O ⁽³⁾	N/A	N/A	N/A	String (2)
shipTo_street1	First line of the address to which to ship the product.	O	N/A	N/A	N/A	String (60)
shipTo_street2	Second line of the address to which to ship the product.	O	N/A	N/A	N/A	String (60)
voidService_run	Whether to include voidService in your request. This field can contain one of the following values: <ul style="list-style-type: none"> <code>true</code>: Include the service in your request. <code>false</code> (default): Do not include the service in your request. 	N/A	N/A	N/A	R	String (5)
voidService_voidRequestID	The requestID of the capture or credit you want to void.	N/A	N/A	N/A	R	String (26)

O = Optional; R = Required; N/A = Not applicable

(1) Optional if **ccCreditService_captureRequestID** is included. See [“Requesting a Bill Me Later Credit”](#) on page 20.

(2) If using Paymentech, the field is required only for the first Bill Me Later transaction for that customer.

(3) Required when country of the address is U.S. or Canada

Table 3 Request Fields for Bill Me Later (Continued)

Request Field	Description	Auth	Capture	Credit	Void	Data Type & Length
voidService_ voidRequestToken	<p>The requestToken value returned from a previous request for a service that you want to void. You can void Bill Me Later (BML) transactions that have not been batched or voided before. For example, if you want to void a capture, set this field to the requestToken value returned in the reply for the capture request. The field is an encoded string that contains no confidential information, such as an account or card verification number. The string may contain up to 256 characters.</p> <p>You can start using the request token immediately, or you can do so at any time during the transition period:</p> <ul style="list-style-type: none"> • After you start using the request token, you must do so for all future follow-on requests. • If you cannot use the request token immediately, CyberSource will store the request token data for you and retrieve it as necessary when you send follow-on requests. However, you will need to start using the request token before the end of the transition period. • You can read the migration guide that describes the request token. 	N/A	N/A	N/A	R	String (256)

O = Optional; R = Required; N/A = Not applicable

(1) Optional if **ccCreditService_captureRequestID** is included. See [“Requesting a Bill Me Later Credit”](#) on page 20.

(2) If using Paymentech, the field is required only for the first Bill Me Later transaction for that customer.

(3) Required when country of the address is U.S. or Canada

Additional Authorization Fields for Paymentech

If you are using Paymentech, you can use these additional request fields with an authorization. Although the table shows the fields as optional for the authorization service, if you have the information available to you when the order is placed, you must provide all of the optional fields in your authorization request.

Table 4 Additional Authorization Request Fields for Paymentech

Request Field	Description	Required/ Optional for the Auth	Data Type & Length
bml_billToPhoneType	Type of the phone number for the customer. This field can contain one of the following values: <ul style="list-style-type: none"> • D: Day • H: Home • N: Night • W: Work 	Optional	String (1)
bml_customerBillingAddressChange	Whether the customer has updated their billing address at the merchant site. This field can contain one of the following values: <ul style="list-style-type: none"> • true: Billing address has been updated • false: Billing address has not been updated 	Optional	String (5)
bml_customerEmailChange	Whether the customer has updated their email address at the merchant site. This field can contain one of the following values: <ul style="list-style-type: none"> • true: Email address has been updated • false: Email address has not been updated 	Optional	String (5)
bml_customerHasCheckingAccount	Whether the customer has a checking account. This field can contain one of the following values: <ul style="list-style-type: none"> • true: Customer has checking account • false: Customer does not have checking account 	Optional	String (5)
bml_customerHasSavingsAccount	Whether the customer has a savings account. This field can contain one of the following values: <ul style="list-style-type: none"> • true: Customer has savings account • false: Customer does not have savings account 	Optional	String (5)

Table 4 Additional Authorization Request Fields for Paymentech (Continued)

Request Field	Description	Required/ Optional for the Auth	Data Type & Length
bml_customer PasswordChange	Whether the customer has changed passwords at the merchant site. This field can contain one of the following values: <ul style="list-style-type: none"> • <code>true</code>: Password has been changed • <code>false</code>: Password has not been changed 	Optional	String (5)
bml_customer PhoneChange	Whether the customer has updated their phone number at the merchant site. This field can contain one of the following values: <ul style="list-style-type: none"> • <code>true</code>: Phone number has been updated • <code>false</code>: Phone number has not been updated 	Optional	String (5)
bml_employerCity	City of the address of the customer's employer.	Optional	String (50)
bml_employer CompanyName	Name of the customer's employer.	Optional	String (30)
bml_employer Country	Country of the address of the customer's employer. This field can contain <code>US</code> , <code>CA</code> , or <code>GB</code> .	Optional	String (2)
bml_employer PhoneNumber	Telephone number of the customer's employer. Use format <code>AAAEENNNNNXXXX</code> where <code>A</code> = Area code, <code>E</code> = Exchange, <code>N</code> = Number, and <code>X</code> = Extension.	Optional	String (14)
bml_employer PhoneType	Type of the phone number of the customer's employer. This field can contain one of the following values: <ul style="list-style-type: none"> • <code>D</code>: Day • <code>H</code>: Home • <code>N</code>: Night • <code>W</code>: Work 	Optional	String (1)
bml_employer PostalCode	Postal code of the address of the customer's employer.	Optional	String (10)
bml_employerState	State or province of the address of the customer's employer. Use the two-character codes .	Optional	String (2)
bml_employerStreet1	First line of the address of the customer's employer.	Optional	String (60)
bml_employerStreet2	Second line of the address of the customer's employer.	Optional	String (60)

Table 4 Additional Authorization Request Fields for Paymentech (Continued)

Request Field	Description	Required/ Optional for the Auth	Data Type & Length
bml_gross HouseholdIncome	The customer's gross household income.	Optional	String (10)
bml_household IncomeCurrency	Currency type of the customer's household income. Use the ISO codes .	Optional	String (3)
bml_preapproval Number	The 16-digit preapproval number for the customer. Use if you are participating in the Bill Me Later preapproval program.	Optional	String (16)
bml_residenceStatus	Status of the customer's residence (for example, rent, own, and so on). This field can contain one of the following values: <ul style="list-style-type: none"> • O: Own • R: Rent • X: Other 	Optional	String (1)
bml_ shipToPhoneType	Type (for example, work, home) of the phone number of the person receiving the shipment. This field can contain one of the following values: <ul style="list-style-type: none"> • D: Day • H: Home • N: Night • W: Work 	Optional	String (1)
bml_yearsAt CurrentResidence	Number of years the customer has lived at their current residence.	Optional	Integer (2)
bml_yearsWith CurrentEmployer	Number of years the customer has worked with their current employer.	Optional	Integer (2)

Reply Fields

[Table 5](#) lists the reply fields that you receive for a Bill Me Later authorization, capture, credit, or void. Note that you may receive additional fields not listed here that are relevant to credit card processing but are not relevant to Bill Me Later transactions.

To see example requests and replies, see [“Example Requests and Replies”](#) on page 46.

Table 5 Reply Fields for Bill Me Later

Reply Field	Description	Returned By:				Data Type & Length
		Auth	Capture	Credit	Void	
ccAuthReply_amount	Total amount of the authorization.	Yes	No	No	No	String (15)
ccAuthReply_authorizationCode	Authorization code for the transaction.	Yes	No	No	No	String (6)
ccAuthReply_authorizedDateTime	Time of authorization. The format is <i>YYYY-MM-DDThh:mm:ssZ</i> . For example, 2005-08-11T22:47:57Z is equal to August 11, 2005, at 10:47:57 P.M. The <i>T</i> separates the date and the time. The <i>Z</i> indicates UTC.	Yes	No	No	No	String (20)
ccAuthReply_bmlAccountNumber	Bill Me Later account number. Returned when creating a new Bill Me Later account.	Yes	No	No	No	String with numbers only (20)
ccAuthReply_processorResponse	Processor response code. Note Do not use this field to interpret the result of the authorization.	Yes	No	No	No	String (10)
ccAuthReply_reasonCode	Numeric value corresponding to the result of the credit card authorization request. See Appendix A, “Reason Codes,” on page 52 for a list of possible values.	Yes	No	No	No	Integer (5)
ccAuthReply_reconciliationID	Reference number for the transaction.	Yes	No	No	No	String (60)
ccCaptureReply_amount	Total amount of the capture.	No	Yes	No	No	String (15)

Table 5 Reply Fields for Bill Me Later (Continued)

Reply Field	Description	Returned By:				Data Type & Length
		Auth	Capture	Credit	Void	
ccCaptureReply_reasonCode	Numeric value corresponding to the result of the capture request. See Appendix A, " Reason Codes ," on page 52 for a list of possible values.	No	Yes	No	No	Integer (5)
ccCaptureReply_reconciliationID	Reference number that you use to reconcile your CyberSource reports with your processor reports. For FDMS, see " Requesting a Bill Me Later Credit " on page 20.	No	Yes	Yes	No	String (60)
ccCaptureReply_requestDateTime	Time when capture is requested. The format is <i>YYYY-MM-DDThh:mm:ssZ</i> . For example, 2005-08-11T22:47:57Z is equal to August 11, 2005, at 10:47:57 P.M. The T separates the date and the time. The Z indicates UTC.	No	Yes	No	No	String (20)
ccCreditReply_amount	Total amount of the credit.	No	No	Yes	No	String (15)
ccCreditReply_reasonCode	Numeric value corresponding to the result of the credit request. See Appendix A, " Reason Codes ," on page 52 for a list of possible values.	No	No	Yes	No	Integer (5)
ccCreditReply_reconciliationID	Reference number that you use to reconcile your CyberSource reports with your processor reports.	No	No	Yes	No	String (60)
ccCreditReply_requestDateTime	Time when credit is requested. The format is <i>YYYY-MM-DDThh:mm:ssZ</i> . For example, 2005-08-11T22:47:57Z is equal to August 11, 2005, at 10:47:57 P.M. The T separates the date and the time. The Z indicates UTC.	No	No	Yes	No	String (20)

Table 5 Reply Fields for Bill Me Later (Continued)

Reply Field	Description	Returned By:				Data Type & Length
		Auth	Capture	Credit	Void	
decision	Summarizes the result of the overall request. The field can contain one of the following values: <ul style="list-style-type: none"> • ACCEPT • ERROR • REJECT 	Yes	Yes	Yes	Yes	String (6)
invalidField_0...N	Fields in the request that contained invalid data. These reply fields are included as an aid to software developers only. No attempt should be made to use these fields for end user interaction. See “Missing or Invalid Fields” on page 14.	Yes	Yes	Yes	Yes	String (100)
merchantReferenceCode	Order reference or tracking number that you provided in the request.	Yes	Yes	Yes	Yes	String (50)
missingField_0...N	Required fields that were missing from the request. These reply fields are included as an aid to software developers only. No attempt should be made to use these fields for end user interaction. See “Missing or Invalid Fields” on page 14.	Yes	Yes	Yes	Yes	String (100)
purchaseTotals_currency	Currency used for the order.	Yes	Yes	Yes	No	String (5)
reasonCode	Numeric value corresponding to the result of the overall request. See Appendix A, “Reason Codes,” on page 52 for a list of possible values.	Yes	Yes	Yes	Yes	Integer (5)
requestID	Identifier for the request.	Yes	Yes	Yes	Yes	String (26)

Table 5 Reply Fields for Bill Me Later (Continued)

Reply Field	Description	Returned By:				Data Type & Length
		Auth	Capture	Credit	Void	
requestToken	<p>Request token data created by CyberSource for each reply. You need to store the contents of this field so that you can retrieve and send it in follow-on requests. The field is an encoded string that contains no confidential information, such as an account or card verification number. The string can contain up to 256 characters.</p> <p>If you request the authorization and capture services together, the request token is for the capture reply only.</p> <p>You can start using the request token immediately, or you can do so at any time during the transition period:</p> <ul style="list-style-type: none"> • After you start using the request token, you must do so for all future follow-on requests. • If you cannot use the request token immediately, CyberSource will store the request token data for you and retrieve it as necessary when you send follow-on requests. However, you will need to start using the request token before the end of the transition period. • You can read the migration guide that describes the request token. 	Yes	Yes	Yes	Yes	String (256)
voidReply_currency	<p>Currency used for the order. Formatted using the ISO codes.</p>	No	No	No	Yes	String (5)

Table 5 Reply Fields for Bill Me Later (Continued)

Reply Field	Description	Returned By:				Data Type & Length
		Auth	Capture	Credit	Void	
voidReply_reasonCode	Numeric value corresponding to the result of the void request. See Appendix A, " Reason Codes ," on page 52 for a list of possible values.	No	No	No	Yes	Integer (5)
voidReply_requestDateTime	Time when void is requested. The format is <i>YYYY-MM-DDThh:mm:ssZ</i> . For example, 2005-08-11T22:47:57Z is equal to August 11, 2005, at 10:47:57 P.M. The T separates the date and the time. The Z indicates UTC.	No	No	No	Yes	String (20)

Product Codes

The following table lists the values you can use for the product code, which you set using the `item_#_productCode` request field.

Table 6 Product Code Values

Product Code	Definition
adult_content	Adult content.
coupon	Coupon applied to the entire order.
default	Default value for the product code. CyberSource uses <code>default</code> when a request provides no value for the product code.
electronic_good	Electronic product other than software.
electronic_software	Software distributed electronically rather than on tapes, disks, or other media.
gift_certificate	Gift certificate not issued with CyberSource Stored Value Services.
handling_only	Separate charge that is generally a fee imposed by the seller on the customer. The fee pays for the seller's administrative selling costs.
service	Service that you perform for the customer.
shipping_and_handling	Shipping is a separate charge for shipping the product to the purchaser. Handling is generally a fee imposed by the seller to pay for administrative selling costs.
shipping_only	Charge for transporting tangible personal property from the seller to the purchaser. Documentation must be maintained that clearly establishes where title to the tangible personal property passed from the seller to the purchaser.
stored_value	Stored Value certificate.
subscription	Subscription to a Web site or other content.

Example Requests and Replies

Using Name-Value Pairs

Example Bill Me Later Authorization Request

```
ccAuthService_run=true
merchantID=infodev
merchantReferenceCode=482046C3A7E94F5
purchaseTotals_currency=USD
billTo_firstName=John
billTo_lastName=Doe
billTo_street1=1295 Charleston Rd.
billTo_city=Mountain View
billTo_state=CA
billTo_postalCode=94043
billTo_country=US
billTo_phoneNumber=650-965-6000
billTo_email=jdoe@example.com
billTo_dateOfBirth=19541217
billTo_ssn=1234
billTo_hostname=Phx.qw.aol.com
billTo_ipAddress=128.456.789.000
billTo_httpBrowserType=Mozilla/4.0
item_0_unitPrice=49.95
item_0_quantity=1
item_1_unitPrice=12.95
item_1_quantity=2
card_cardType=028
card_expirationMonth=12
card_expirationYear=2021
card_accountNumber=5049900000000000
bml_customerRegistrationDate=20030305
bml_customerTypeFlag=E
bml_itemCategory=1234
bml_productDeliveryTypeIndicator=default
bml_tcVersion=4
```

Example Bill Me Later Authorization Reply

Note that in the reply you may receive additional fields not listed here that are relevant to credit card processing but are not relevant to Bill Me Later transactions.

```
requestID=0305782650000167905080
merchantReferenceCode=482046C3A7E94F5
decision=ACCEPT
reasonCode=100
purchaseTotals_currency=USD
ccAuthReply_reasonCode=100
ccAuthReply_amount=75.85
ccAuthReply_authorizationCode=784932
ccAuthReply_authorizedDateTime=2004-04-28T23:44:27Z
ccAuthReply_processorResponse=A
ccAuthReply_bmlAccountNumber=1234567891011121314
ccAuthReply_reconciliationID=00162649MVH2W459
```

Using XML

Example Bill Me Later Authorization Request

```
<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.18">
  <merchantID>infodev</merchantID>
  <merchantReferenceCode>482046C3A7E94F5</merchantReferenceCode>
  <billTo>
    <firstName>John</firstName>
    <lastName>Doe</lastName>
    <street1>1295 Charleston Rd.</street1>
    <city>Mountain View</city>
    <state>CA</state>
    <postalCode>94043</postalCode>
    <country>US</country>
    <phoneNumber>650-965-6000</phoneNumber>
    <email>jdoe@example.com</email>
    <ipAddress>123.456.789.000</ipAddress>
    <hostname>Phx.qw.aol.com</hostname>
    <dateOfBirth>19541217</dateOfBirth>
    <ssn>1234</ssn>
  </billTo>
</requestMessage>
```

```
<httpBrowserType>Mozilla/4.0</httpBrowserType>
</billTo>
<item id="0">
  <unitPrice>49.95</unitPrice>
  <quantity>1</quantity>
</item>
<item id="1">
  <unitPrice>12.95</unitPrice>
  <quantity>2</quantity>
</item>
<purchaseTotals>
  <currency>USD</currency>
</purchaseTotals>
<card>
  <accountNumber>5049900000000000</accountNumber>
  <expirationMonth>12</expirationMonth>
  <expirationYear>2021</expirationYear>
  <cardType>028</cardType>
</card>
<bml>
  <customerRegistrationDate>20030305</customerRegistrationDate>
  <customerTypeFlag>E</customerTypeFlag>
  <itemCategory>1234</itemCategory>
  <productDeliveryTypeIndicator>default
</productDeliveryTypeIndicator>
  <tcVersion>4</tcVersion>
</bml>
<ccAuthService run="true"/>
</requestMessage>
```

Example Bill Me Later Authorization Reply

Note that the <ccAuthReply> element will contain additional child elements that are relevant to credit cards but not relevant to Bill Me Later. See the schema for the full list of child elements included in <ccAuthReply>.

For information about why each element in the reply contains c: at the beginning (for example, <c:requestID>), see the information about namespaces in [“Parsing Replies”](#) on page 7.

```
<c:replyMessage xmlns:c="urn:schemas-cybersource-com:transaction-data-1.18">
  <c:merchantReferenceCode>482046C3A7E94F5</c:merchantReferenceCode>
  <c:requestID>0305782650000167905080</c:requestID>
  <c:decision>ACCEPT</c:decision>
  <c:reasonCode>100</c:reasonCode>
  <c:purchaseTotals>
    <c:currency>USD</c:currency>
  </c:purchaseTotals>
  <c:ccAuthReply>
    <c:reasonCode>100</c:reasonCode>
    <c:amount>75.85</c:amount>
    <c:authorizationCode>123456</c:authorizationCode>
    <c:authorizedDateTime>2004-04-28T23:44:27Z
    </c:authorizedDateTime>
    <c:processorResponse>A</c:processorResponse>
    <c:bmlAccountNumber>1234567891011121314</c:bmlAccountNumber>
    <c:reconciliationID>00162649MVH2W459</c:reconciliationID>
  </c:ccAuthReply>
</c:replyMessage>
```

Testing

Important You must have a test account with CyberSource. Contact CyberSource Customer Support if you do not already have your test account.

You can send a test authorization to authorize a new Bill Me Later payment and open a new Bill Me Later account. You can then capture that test authorization.

- Use your regular CyberSource merchant ID to perform testing

- Use a real combination for the city, state, and postal code
- Use a real combination for the area code and telephone number
- Use a non-existent account and domain name for the customer's email address (for example, random@example.com)
- When testing the Simple Order API, use the test URL
`https://ics2wstest.ic3.com/commerce/1.x/transactionProcessor`

Going Live

You must go live with CyberSource before you begin to accept real payments customers. When you go live, your CyberSource account is updated so that you can send transactions to CyberSource's production server. If you have not already done so, you will need to provide your banking information to CyberSource so that your processor can deposit funds to your merchant bank account.

To go live:

- 1 Go to the CyberSource Knowledgebase at <http://www.cybersource.com/esupport>.
- 2 Submit a question (click the **Submit a Question** link).
- 3 Log in with your CyberSource merchant ID and password (the same ones you use to log in to the Enterprise Business Center).
- 4 On the **Submit a Question** page, fill in the contact information.
- 5 For the **General Topic**, select **Getting Set Up on CyberSource**.
- 6 For the **Specific Topic**, select **Go Live**.
- 7 For the **Subject**, enter "Go Live".
- 8 In the question field, indicate that you would like to go live.
- 9 If you have not already submitted your banking information to CyberSource, include the information in the question field and select the check box for **This question contains sensitive banking information**.
- 10 Click **Submit question**.

You will receive an email with your support ticket number, and a CyberSource representative will contact you to complete the process.

- 11** Once CyberSource has confirmed that you are live, make sure to update your system so that you send requests to the production server and not the test server. See the documentation for your client SDK for instructions.

After you go live, use real account numbers and other data to test every CyberSource service that your integration supports. Because these are real transactions, use small amounts, such as one dollar, to do the tests. If you have more than one CyberSource merchant ID, test each one separately. Process an authorization, capture, and credit for each configuration, and use your bank statements to verify that money is deposited into and withdrawn from your merchant bank account.

Appendix A

Reason Codes

The following table lists the reason codes returned for Bill Me Later. See [“Replies”](#) on page 13 for a discussion of replies, decisions, and reason codes.

Note CyberSource reserves the right to add new reason codes at any time. If your error handler receives a reason code that it does not recognize, it should use the **decision** field to determine the result.

Table 7 Reason Codes

Reason Code	Description
100	Successful transaction.
101	The request is missing one or more required fields. Possible action: See the reply fields missingField_0...N for which fields are missing. Resend the request with the complete information. See “Missing or Invalid Fields” on page 14.
102	One or more fields in the request contains invalid data. Possible action: See the reply fields invalidField_0...N for which fields are invalid. Resend the request with the correct information. See “Missing or Invalid Fields” on page 14.
150	Error: General system failure. See the documentation for your CyberSource client (SDK) for information about how to handle retries in the case of system errors.
151	Error: The request was received but there was a server timeout. This error does not include timeouts between the client and the server. Possible action: To avoid duplicating the transaction, do not resend the request until you have reviewed the transaction status in the Enterprise Business Center. See the documentation for your CyberSource client (SDK) for information about how to handle retries in the case of system errors.

Table 7 Reason Codes (Continued)

Reason Code	Description
152	Error: The request was received, but a service did not finish running in time. Possible action: To avoid duplicating the transaction, do not resend the request until you have reviewed the transaction status in the Enterprise Business Center. See the documentation for your CyberSource client (SDK) for information about how to handle retries in the case of system errors.
221	The customer matched an entry on the processor's negative file. Possible action: Review the order and contact the payment processor.
230	The authorization request was approved by the issuing bank but declined by CyberSource because it did not pass the card verification (CV) check. Possible action: You can capture the authorization, but consider reviewing the order for the possibility of fraud.
231	Invalid account number. Possible action: Request a different card or other form of payment.
233	General decline by the processor. Possible action: Request a different card or other form of payment.
234	There is a problem with your CyberSource merchant configuration. Possible action: Do not resend the request. Contact Customer Support to correct the configuration problem.
235	The requested amount exceeds the originally authorized amount. Occurs, for example, if you try to capture an amount larger than the original authorization amount. Possible action: Issue a new authorization and capture request for the new amount.
236	Processor failure. Possible action: Wait a few minutes and resend the request.
238	The authorization has already been captured. Possible action: No action required.
239	The requested transaction amount must match the previous transaction amount. Possible action: Correct the amount and resend the request.
240	The card type sent is invalid or does not correlate with the credit card number. Possible action: Confirm that the card type correlates with the credit card number specified in the request, then resend the request.
241	The request ID is invalid. Possible action: Request a new authorization, and if successful, proceed with the capture.

Table 7 Reason Codes (Continued)

Reason Code	Description
242	<p>You requested a capture, but there is no corresponding, unused authorization record. Occurs if there was not a previously successful authorization request or if the previously successful authorization has already been used by another capture request.</p> <p>Possible action: Request a new authorization, and if successful, proceed with the capture.</p>
246	<p>The capture or credit is not voidable because the capture or credit information has already been submitted to your processor. Or, you requested a void for a type of transaction that cannot be voided.</p> <p>Possible action: No action required.</p>
247	<p>You requested a credit for a capture that was previously voided.</p> <p>Possible action: No action required.</p>
250	<p>Error: The request was received, but there was a timeout at the payment processor.</p> <p>Possible action: To avoid duplicating the transaction, do not resend the request until you have reviewed the transaction status in the Enterprise Business Center.</p>

Index

A

ACCEPT decision [13](#)
authorization [17](#)

C

canceling a transaction [2](#), [22](#)
capture [19](#)
changes to document [v](#)
clients [3](#), [5](#)
coupons [11](#)
credits [20](#)

D

data types, definitions of [22](#)
decisions [13](#)

E

encoding type [22](#)
Enterprise Business Center [3](#)
ERROR decision [13](#)
errors, described [13](#)
example requests and replies [46](#)

F

FDMS [1](#)
follow-on credits [20](#)
formatting requirements [22](#)
freight charges [9](#), [18](#)

G

going live [50](#)
grand total amounts [9](#)

I

i4Commerce [1](#)
ICS [1](#)
Internet Commerce Suite [1](#)
invalid fields [14](#)
items [8](#), [22](#)

M

merchant reference code [15](#)
missing fields [14](#)

N

name-value pairs [6](#), [46](#)
new Bill Me Later account [18](#)

O

orders, tracking [15](#)

P

partial captures and credits [21](#)
Paymentech [1](#), [37](#)
product codes [9](#), [45](#)

R

reason codes
 described [14](#)
 listed [52](#)
reconciliation IDs [16](#), [20](#)
reference number, for tracking orders [15](#)
REJECT decision [13](#)
replies [13](#)
reply fields [40](#)
reports [3](#), [4](#)

request fields [23](#)
request IDs [20](#)
requests [9](#)

S

shipping address [18](#)
shipping and handling [9](#), [18](#)
SOAP [5](#)
software, updating [3](#)
special characters [22](#)
stand-alone credits [20](#)

T

tax [9](#)
testing your system [4](#), [51](#)
tracking orders [15](#)

U

updating software [3](#)
UTF-8 [22](#)

V

version of Simple Order API [6](#)
void [2](#), [22](#)

W

WS Security [5](#)

X

XML [6](#), [47](#)