

Offline Transaction File Submission

Implementation Guide

September 2017



CyberSource Contact Information

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

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

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

Copyright

© 2017 CyberSource Corporation. All rights reserved. CyberSource Corporation ("CyberSource") furnishes this document and the software described in this document under the applicable agreement between the reader of this document ("You") and CyberSource ("Agreement"). You may use this document and/or software only in accordance with the terms of the Agreement. Except as expressly set forth in the Agreement, the information contained in this document is subject to change without notice and therefore should not be interpreted in any way as a guarantee or warranty by CyberSource. CyberSource assumes no responsibility or liability for any errors that may appear in this document. The copyrighted software that accompanies this document is licensed to You for use only in strict accordance with the Agreement. You should read the Agreement carefully before using the software. Except as permitted by the Agreement, You may not reproduce any part of this document, store this document in a retrieval system, or transmit this document, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written consent of CyberSource.

Restricted Rights Legends

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

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 Unpublished rights reserved under the copyright laws of the United States.

Trademarks

CyberSource, The Power of Payment, CyberSource Payment Manager, CyberSource Risk Manager, CyberSource Decision Manager, CyberSource Connect, Authorize.Net, and eCheck.net are trademarks and/or service marks of CyberSource Corporation. All other brands and product names are trademarks or registered trademarks of their respective owners.

Contents

Recent Revisions to this Document 5

About This Guide 6

- Note, Important, and Warning Statements 6
- Text and Command Conventions 7

Chapter 1	Offline Transaction File Submission	8
	API Fields	8
	Downloading a Template	9
	Creating a Custom Template	9
	Creating a Batch File	10
	Uploading a Batch File	11
	Creating and Uploading Batch Files Manually	13
	Security Keys	13
	File Format	14
	File Header	15
	Data Header	16
	Data Records	16
	Trailer Record	16
	File Examples	17
	File Submission Protocol	18
	File Receipt Response	19
	Transaction Processing	20
	Testing	20
	Viewing the Status of Your Batch File Submissions	20
	Elements	21
	<Batch>	21
	<ValidationErrors>	22
	<Error>	23
	Error Codes	23
	DTD	26
	Examples	26
	Viewing the Results of Your Batch File Requests	28

Reports	28
Response Files	29
File Header	29
Data Records	30

Chapter 2	Batch File Reports	31
	Batch Files Detail Report	31
	XML Format	32
	Elements	32
	DTD	39
	Example	40
	CSV Format	42
	Records	42
	Example	44
	Batch Files Daily Summary Report	45
	Batch Submission Detail Report: Overview	47
	Requesting Report with Query API	47
	Formulating the Query	47
	Viewing and Saving the Report	48
	DTD	49
	Example	51
	Downloading a Report from the Business Center	52

Chapter 3	Using Java Sample Code to Upload Files	53
	Using Sample Code for Basic Authentication	53
	Sample Code for Basic Authentication	57

Appendix A	API Field Mapping	62
	Reply-Level Fields	63
	Request-Level Fields	63
	Offer-Level Fields	73
	Reply Fields	75

Index	88
--------------	-----------

Recent Revisions to this Document

Release Date	Changes
September 2017	■ Updated information about header requirements in batch files.
August 2016	■ Updated information about creating custom batch templates.
December 2015	■ Updated Java sample code.
October 2014	■ Corrected batch upload validate XML file name. See "Viewing the Status of Your Batch File Submissions," page 20.
September 2014	■ Added note about requesting batch reports. See "Batch File Reports," page 31.
May 2014	■ Added information about custom batch templates. See "Creating a Custom Template," page 9.

About This Guide

Audience

This guide is written for users of the Business Center who are using batch files to process order requests.

Scope

The Offline Transaction File Submission system uses fields from the CyberSource Simple Order API, not from the SCMP API. If you use the SCMP API to process individual transactions, you will need to know which Simple Order API fields correspond to the SCMP API fields with which you are familiar.

Note, Important, and Warning Statements



Note

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



Important

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



Warning

A *Warning* contains information or instructions, which, if not heeded, can result in a security risk, irreversible loss of data, or significant cost in time or revenue or both.

Text and Command Conventions

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

Related Documents

To use a batch file to process credit card transactions, see [Credit Card Services with the Simple Order API](#) for information about the Simple Order API fields available with the CyberSource Credit Card Services.

To use a batch file to process electronic checks, see the Simple Order API chapter in the [Electronic Check Services for the Simple Order API](#) for information about the Simple Order API fields available with the CyberSource Electronic Check Services.

To use a batch file to process transactions that use any other CyberSource service, see the chapter about the Simple Order API in the implementation guide that describes that service.

All the implementation guides are available on the [Support Center](#).

Refer to the Support Center for complete CyberSource technical documentation:

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

Customer Support

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

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

Offline Transaction File Submission

Offline Transaction File Submission allows you to send a single file, called a batch file or batch transaction file, to CyberSource that contains a set (batch) of transaction requests instead of sending individual requests. The information you provide for each request in the batch file is the same information you provide for an individual service request.

In one file, you can include transactions that use different services, currencies, countries, merchant IDs, and card types, thus eliminating the burden of handling these values separately. However, each template provided through the Business Center covers only one type of transaction.

Most CyberSource services can be accessed in a batch file. However, the following services cannot be requested in a batch file:

- Payer authentication enrollment check (**payerAuthEnrollService**)
- Bank transfer (**bankTransferService**)
- PayPal button creation (**paypalButtonCreateService**)
- PayPal credit (**paypalCreditService**)

To use batch files in the Business Center, you must:

- Download the appropriate template.
- Create the batch file.
- Upload the batch file.



Important

CyberSource does not recommend this feature for TID-based processors or APACS-based processors because batch files for these kinds of processors can cause time outs and errors. For more information, contact your CyberSource Technical Account Manager.

API Fields

The Offline Transaction File Submission system uses API fields from the CyberSource Simple Order API; it does not use fields from the SCMP API. For the mapping between the SCMP API fields and the Simple Order API fields, see [Appendix A, "API Field Mapping," on page 62](#).

Downloading a Template

There is a template for each type of transaction that you can submit in a batch file. The following table lists these templates.

Table 1 Batch File Templates

Card Transactions	Check Transactions	Subscriptions
<ul style="list-style-type: none"> ■ Authorizations ■ Sales ■ Captures ■ Credits ■ Sales with Level III ■ Captures with Level III ■ Credits with Level III 	<ul style="list-style-type: none"> ■ Electronic Check Debits ■ Electronic Check Credits 	<ul style="list-style-type: none"> ■ Create Subscriptions ■ Update Subscriptions ■ Cancel Subscriptions

Step 1 To download a template, go to **Tools & Settings > Batch Transactions > Templates**.

Transaction Batch Templates

[I need help with this page.](#)

Use this page to download the templates for the types of transaction batches that you want to send with CyberSource. Each template is a spreadsheet that lists the fields that you can include in your batch files.

Transaction Batch Templates

Template

Step 2 Select a template from the list of templates.

Step 3 Click **Download**.

Step 4 Enter the destination for the template.

Creating a Custom Template

Custom templates enable you to send batch files that contain information for other transaction services not included with the standard template, such as direct debit and direct debit refunds. When you create a custom template from a standard template, request fields do not have to be in a specific order.

Step 1 Follow the file formatting rules in "[Creating and Uploading Batch Files Manually](#)," page 13 to generate a batch file for upload.

- Step 2** At the end of the file header, add another comma and type "Template=custom".
- Step 3** Save the file.
-

Example File Header

```
merchantID=infodev,batchID=12345,creationDate=2007-06-12,recordCount=3,purchaseTotals_
currency=EUR,statusEmail=nobody@example.com,targetAPIVersion=1.90,
Template=custom
```

Creating a Batch File

- Step 1** Make a copy of the template you downloaded.
- Step 2** Open the copy, which is in CSV format. CyberSource recommends that you use Excel or a similar program. If you use Excel, you must *import* the file. If you *open* the file, Excel will not process the CSV formatting correctly.
- Step 3** Enter a batch ID in the file header. This value is a file (batch) identifier that you assign. The batch ID must be unique. Format: alphanumeric with a maximum of 8 characters.
-



Important

You can enter up to 10 elements in the header of the batch file.

- Step 4** Load your transaction data into the file starting in row 4.
- The information that you provide for each request in the batch file is the same information that you would provide for an individual service request. The template specifies the required and optional fields for the batch file's transaction type. For a description of each field, see the template descriptions:

http://apps.cybersource.com/library/documentation/sbc/SB_Batch_Submission_UG/Batched_Template_Guides.pdf

Page through the template descriptions to find the pertinent information for your transaction type.

- Step 5** You can enter the data in one of three ways:
- Enter it manually.
 - Copy and paste it.

- Write a program to load it into the template.

**Note**

Files that include non-ISO-8859-1 characters (letters with diacritical markings) will be processed, but the characters will appear as question marks when the transaction details are viewed in the Business Center and in the reports.

- Step 6** Count the number of records in the file and enter this value for the record count in the file header.

**Note**

The system will set the trailer record as well as additional fields in the file header.

- Step 7** Save the file.

Uploading a Batch File

- Step 1** Choose **Tools & Settings > Batch Transactions > Upload**.

Transaction Batch Upload

[Page help](#) [Page feedback](#)

Use this page to send to CyberSource transaction batch files instead of individual transactions. Before uploading your files, make sure that they follow the format described in the transaction batch templates.

Transaction Batch Upload

Reference Notes for Batch

Email Address for Status

File Name No file chosen

- Step 2** Optional: Enter reference notes for the batch and an email address for receiving status messages.
- Step 3** Click **Browse** and navigate to the batch file.
- Step 4** Click **Submit**.

**Note**

You can submit multiple batch files by repeating Steps **b** through **d** for each batch file.

CyberSource reads the file, verifies that it conforms to the template, and sends you an email indicating whether the file passed the verification test.

- Step 5** Optional: View the batch file's status on the **Batch Upload Search** page. Specify a search date range and click **Search**. Your results appear in the status grid.



Note

Batch status is available on batches uploaded within 31 calendar days. After processing is completed, the information will be available in the Batch Files Detail Report and Batch Files Daily Summary Report as described in [Chapter 2, "Batch File Reports," on page 31](#).

Table 2 Status Values

Status	Description
Validating	<p>After the file is uploaded, the system displays Validating in the status grid. CyberSource will usually update the status in the grid and send a batch status alert within 30 minutes of receiving the file. However, actual timing depends on the system load and the number of files ahead of yours.</p> <p>CyberSource will not process any of the requests in the file if there is any type of syntax error. If multiple records in the file have errors, CyberSource will send only one email with the line number of the first failed record. Typical errors are:</p> <ul style="list-style-type: none"> ■ The recordCount you specified in the file header does not match the number of data records in the file. ■ A data record in the file does not have the correct number of fields as specified in the data header. The batch status alert will indicate the line number of the problem data record.
Rejected	<p>If the validation fails, the system displays Rejected in the status grid. To handle a failed validation, follow the suggested remedy in the batch status alert. If you need to resend the file, use the same batchID that you used for the original file unless otherwise instructed in the batch status alert.</p>
Processing	<p>If the validation succeeds, the system displays Processing in the status grid and CyberSource begins processing the transactions. The transaction processing time depends on the time of day and the size of your file. You need to submit the batch file early enough in the day to allow plenty of time for validation and processing before your batch cutoff time.</p>
Completed	<p>When the processing has been completed, the system displays Completed in the status grid and the date in the Uploaded column becomes a link to a Batch Files Daily Summary report, which is described in Chapter 2, "Batch File Reports," on page 31.</p>

- Step 6** Receive the batch status alerts.

The system sends you an email when batch file validation succeeds or fails and when batch file processing is complete.

**Note**

You can also view validation files as described in "[Viewing the Status of Your Batch File Submissions](#)," page 20.

Step 7 View the reports and response files for your transaction requests:

- After all the requests in a batch file are processed, CyberSource creates three batch reports:

Batch Files Daily Summary Report—Shows a summary of the batched transactions.

Batch Files Detail Report—Shows details of the batched transactions.

Batch Submission Detail Report—Shows real-time detailed status information of the batched transactions.

See [Chapter 2, "Batch File Reports,"](#) on page 31.

- **Payment Submission Detail Report**—Shows all the transactions that were submitted to your processor for settlement. You can download the report daily. If the batch file has an error, the file is not processed and so the file's requests are not included in the report. See the [Reporting Developer's Guide](#).
- **Response files**—After CyberSource processes all of the requests in the batch file, CyberSource creates two types of CSV response files that you can use to determine the results of the requests. See "[Response Files](#)," page 29.

Creating and Uploading Batch Files Manually

When creating and uploading a batch file manually, you can include requests for different CyberSource services in the same file. For example, you can include authorization, capture, and credit transactions in one file. You can include a capture and credit for the same order in a single file, but if you do, the credit must be a stand-alone credit. This means you must provide the customer's name, billing address, and credit card information in the credit request record.

Security Keys

When creating and uploading a batch file manually, you must use a transaction security key:

- If you are using the SCMP API to process transactions, you cannot use the key that you already have. You need to create a separate transaction security key that works with the Simple Order API.
- If you are already using the Simple Order API to process transactions, you do not need to create another key.

**Important**

You must generate two transaction security keys—one for the CyberSource production environment and one for the test environment. For information about generating and using security keys, see *Creating and Using Security Keys* ([PDF](#) | [HTML](#)).

File Format

Batch files must be in CSV format.

**Note**

Files that include non-ISO-8859-1 characters are processed, but the characters appear as question marks when you view the transaction details in the Business Center and in the reports.

The format for a batch file is:

- A file header followed by a new line
- A data header followed by a new line
- One or more data records, each on a separate line
- A trailer record that indicates the end of the file

The file follows CSV field formatting rules. Use quotation marks (") to enclose any field that contains a comma. For example:

```
"reference=UK office, John Smith"
```

File Header

The file header consists of comma-separated name-value pairs using the fields listed in the following table.

Table 3 File Header Fields

Field Name	Description	Required or Optional
merchantID	Your CyberSource merchant ID.	Required
batchID	File (batch) identifier that you assign. The batchID must be unique. Format: alphanumeric with a maximum of 8 characters.	Required
creationDate	Creation date of the file. Format: YYYY-MM-DD.	Optional
recordCount	Total number of data records in the file.	Required
reference	Your reference notes for the file.	Optional
statusEmail	Email address or alias where CyberSource sends the file receipt notification. See " File Receipt Response ," page 19.	Required
targetAPIVersion	<p>CyberSource updates the ICS transaction API periodically. When you request an ICS service through the Simple Order API, you must specify which version of the API you want to use. CyberSource recommends that you set this field to the same value that you configure your CyberSource Simple Order API client to use. This is the client you use to send individual ICS requests with the Simple Order API. To determine the latest API version available, go to https://ics2ws.ic3.com/commerce/1.x/transactionProcessor/</p> <p>If you are enabled for partial authorization, the value for targetAPIVersion, set in each OLP input file, must be set to 1.52 or higher to ensure the OLP disables Partial Authorization for all Authorization transactions sent.</p> <p>Important Setting this field to a value that does not exist will prevent your file from being processed.</p>	Required

In the file header, you can also include any CyberSource service API fields that will have a constant value throughout the file. For example, if you plan to use only euros as the currency for all of the requests, instead of declaring a column in the data header for purchaseTotals_currency, you can include purchaseTotals_currency=EUR in the file header. It does not matter where you place it in the header. Any field that you include in the header that is not in the preceding table will be treated as an API field.

Example File Header

```
merchantID=infodev,batchID=12345,creationDate=2007-06-12,recordCount=3,reference=Additional information,purchaseTotals_currency=EUR,statusEmail=nobody@example.com,targetAPIVersion=1.29
```

Data Header

Use the data header to name the Simple Order API fields that are in the requests in the file. The data header needs to consist of a comma-separated list of the fields. The order in which you list the fields in the data header does not matter. You can include any API field that the ICS service uses.



Important

Include all of the fields that the CyberSource service requires. If any required fields are missing from the data header or file header, then all of the requests in the batch file will be rejected.

CyberSource recommends that you use the **purchaseTotals_grandTotalAmount** API field even though it is optional for your requests. CyberSource uses this field when calculating a checksum to validate that the entire file was received. See "[Trailer Record](#)," [page 16](#). You are not required to include a value for **purchaseTotals_grandTotalAmount** for all of the records. To omit the value for a particular record, leave it blank; do not set it to 0.

A request for the capture service might include these API fields:

- **ccCaptureService_run**
- **ccCaptureService_authRequestID**
- **purchaseTotals_currency**
- **merchantReferenceCode**
- **purchaseTotals_grandTotalAmount**

Example Data Header

```
ccCaptureService_run,ccCaptureService_authRequestID,purchaseTotals_
currency,merchantReferenceCode,purchaseTotals_grandTotalAmount
```

Data Records

Each data record consists of a comma-separated list of the values that corresponds to the API fields in the data header. The maximum allowable file size is 60 MB. The order of your requests in the file does not matter. Requests for different ICS services can be mixed together.

Make sure that the first three letters of every data record are not `END` because `END` indicates the trailer record.

Trailer Record

The last item in the file is the trailer record, which consists of two items:

- The word `END` followed by a comma
- The word `SUM` followed by an equal sign and the sum of all of the **purchaseTotals_grandTotalAmount** values in all the records in the file. When you calculate the sum,

treat credits as positive values and ignore the fact that there might be different currencies. If you leave all of the **purchaseTotals_grandTotalAmount** values blank in the file, then set SUM to 0. CyberSource uses the sum that you provide to validate that the entire file was received.

Example Trailer Record When Using purchaseTotals_grandTotalAmount Values

```
END,SUM=1014.37
```

Example Trailer Record When Leaving purchaseTotals_grandTotalAmount Values Blank

```
END,SUM=0
```

File Examples

Example Captures

```
merchantID=infodev,batchID=12345,creationDate=2007-06-12,recordCount=3,"reference=UK office, John Smith",statusEmail=jsmith@example.com,targetAPIVersion=1.29
ccCaptureService_run,ccCaptureService_authRequestID,purchaseTotals_currency,merchantReferenceCode,purchaseTotals_grandTotalAmount
true,1234567891234567,EUR,ABC12320398,327.49
true,1234567891234568,GBP,ABC97611927,187.65
true,1234567891234569,EUR,ABC09177294,499.23
END,SUM=1014.37
```

Example Captures and Credits

```
merchantID=infodev,batchID=12345,creationDate=2007-06-12,recordCount=4,"reference=UK office, John Smith",statusEmail=jsmith@example.com,targetAPIVersion=1.29
ccCaptureService_run,ccCaptureService_authRequestID,ccCreditService_run,ccCreditService_captureRequestID,purchaseTotals_currency,merchantReferenceCode,purchaseTotals_grandTotalAmount
true,1234567891234567,false,,EUR,ABC12320398,327.49
true,1234567891234568,false,,GBP,ABC97611927,187.65
true,1234567891234569,false,,EUR,ABC09177294,499.23
false,,true,1234567891999994,CAD,ABC39882097,14.99
END,SUM=1029.36
```

This example has the same capture requests as the "Captures" example and it also includes a credit request for \$14.99. The data header includes additional fields used for credits. For the credit data record, `ccCaptureService_run=false` and `ccCreditService=true`.

Example Credit Card and Electronic Check Transactions

```

merchantID=infodev,batchID=12345,creationDate=2007-06-
12,recordCount=6,"reference=UK office, John Smith",statusEmail=j
smith@example.com,targetAPIVersion=1.29

ccAuthService_run,ccCaptureService_run,ccCaptureService_
authRequestID,ccCreditService_run,ccCreditService_
captureRequestID,ecDebitService_run,purchaseTotals_
currency,merchantReferenceCode,purchaseTotals_
grandTotalAmount,billTo_firstName,billTo_lastName,billTo_
street1,billTo_city,billTo_state,billTo_postalCode,billTo_
country,billTo_email,card_accountNumber,card_
expirationMonth,card_expirationYear,check_accountNumber,check_
accountType,check_bankTransitNumber
false,true,1234567891234567,false,,false,EUR,ABC12320398,327.49,
.....
false,true,1234567891234568,false,,false,GBP,ABC97611927,187.65,
.....
false,true,1234567891234569,false,,false,EUR,ABC09177294,499.23,
.....
false,false,,true,1234567891999994,false,CAD,ABC39882097,14.99,,,,,,
.....
true,true,,false,,false,USD,ABC47283891,299.12,John,Doe,123 Elm
St.,Richmond,CA,95102,US,doe@example.com,4111111111111111,05,2009,,,
false,false,,false,,true,USD,ABC26341197,39.75,Jane,Smith,988 Grape
Way,Napa,CA,96293,US,jane@example.com,,,4100,c,7293920

END,SUM=1368.23

```

This example includes the same transactions as the "Captures and Credits" example and it also includes a credit card sale, which is an authorization and capture together, for \$299.12, and an electronic check debit request for \$39.75.

File Submission Protocol

Submit the batch file to CyberSource using HTTPS. The security for the file submission is based on HTTPS and SSL and uses client certificates as part of the SSL handshake. To access the URLs listed below, your client must provide the same Simple Order API client certificate that you use to request regular individual ICS Simple Order API transactions. The client certificate is stored in a PKCS12 file named `<merchantID>.p12` and is protected by a single password.

When testing, your client should submit the file to the test URL at `https://batchtest.cybersource.com/upload/UploadBatchFile`. See "Testing," page 20.

When in production, your client should submit the file to the production URL at `https://batch.cybersource.com/upload/UploadBatchFile`.

See [Chapter 3, "Using Java Sample Code to Upload Files," on page 53](#) for more information about how to upload files to CyberSource.

File Receipt Response

After the batch file has been received, CyberSource validates the syntax and sends you a confirmation email indicating whether the file passed the validation. There is a field in the batch file's file header that specifies the email address or alias where you want the email notifications to be sent. CyberSource usually sends the email notification within 30 minutes of receiving the file. However, actual timing depends on the system load.

The following table lists the possible subject lines of the email notification.

Table 4 Possible Email Notification Subject Lines

Subject Line	Reason and Remedy
SUCCESS: Batch ID <Batch ID #> - Validation	The file was received and passed the validation checks. CyberSource will process the requests in the file. No action is required on your part.
FAILED: Batch ID <Batch ID #> - Validation	<p>Reason: The file did not pass the validation checks.</p> <p>Remedy: Look at the contents of the email and follow the suggested remedy. If you need to resend the file, use the same batch ID that you used for the original file unless otherwise instructed in the email.</p>

CyberSource will not process any of the requests in the file if there is any type of syntax error. If multiple records in the file have errors, CyberSource will send only one email with the line number of the first failed record. Typical errors that might occur are:

- CyberSource does not recognize the merchant ID that you specified in the file.
- The recordCount that you specified in the file header does not match the actual number of data records in the file.
- The file is missing the trailer record.
- A specific data record in the file does not have the correct number of fields as specified in the data header. The email will indicate the line number of the problem data record.

Transaction Processing

After the batch file is validated, CyberSource begins processing the transactions. Transaction processing time depends on the time of day and the size of your file. CyberSource begins submitting the day's settlement information to the payment processors starting at midnight Pacific time. Submit your file early enough to allow CyberSource enough time to process your file before the midnight submission deadline.

To view the results of the requested transaction processing, see "[Viewing the Results of Your Batch File Requests](#)," page 28.

Testing

When testing your system, upload your batch file to the test URL and not the production URL. The test URL is: <https://batchtest.cybersource.com/upload/UploadBatchFile>.

You will receive file receipt emails for your test files. You can also download the corresponding response files from the **Reports** area of the test version of the Business Center at <https://ebctest.cybersource.com>.

Viewing the Status of Your Batch File Submissions

Before processing the requests in your batch file, Offline Transaction File Submission validates the batch file. After attempting to validate your batch file, CyberSource creates the following types of XML-formatted response files that indicate whether the validation succeeded or failed:

- **Successful validation file**—This file indicates that your batch file is valid.
Filename format:
`<merchantID>.<batchID>.validate.Success.xml`
Example filename:
`CyberVacations.39762.validate.Success.xml`
- **Failed validation file**—This file indicates that your batch file is invalid and provides information about why the file failed validation.
Filename format:
`<merchantID>.<batchID>.validate.xml`
Example filename:
`CyberVacations.39768.validate.xml`

The response files are available in the **Reports** area on the [Business Center](#). You can download them the same way you download CyberSource reports, which is explained in the [Reporting Developer's Guide](#).

**Note**

These validation files are an optional feature. Contact Customer Support to have your account configured for these files.

Elements

<Batch>

The <Batch> element is the root of the response file.

```
<Batch BatchID=CDATA
    MerchantID=CDATA
    Name=CDATA
    Version=NMTOKEN>
    (ValidationStatus)
    (ValidatedRecords)
    (ValidationTime)
    (ValidationErrors)
</Batch>
```

Table 5 Attributes of <Batch>

Attribute Name	Description	Data Type & Length
BatchID	Batch file identifier that you assigned.	Alphanumeric (8)
MerchantID	Your CyberSource merchant ID.	Alphanumeric (30)
Name	Name of the report. This value will always be OLP Validation Report.	Alphanumeric (25)
Version	Version number of the report. The current version number is 1.0.	Numeric (10)

Table 6 Child Elements of <Batch>

Element Name	Description	Data Type & Length
ValidationStatus	Status of the attempted validation. Possible values: <ul style="list-style-type: none"> ■ Failed ■ Success ■ OnHold ■ ToBeResolved 	Alphanumeric (10)
ValidatedRecords	Number of records validated. This value is present only if the validation was successful.	Numeric (10)
ValidationTime	Timestamp for the validation process. Format: yyyy-MM-ddTHH:mm:ss	DateTime (25)
ValidationErrors	List of the errors in the batch file. This element is present only if the validation failed. See "<ValidationErrors>," page 22.	Element

<ValidationErrors>

The <ValidationErrors> element contains the errors associated with the batch file.

```
<ValidationErrors>
  (Error)*
</ValidationErrors>
```

Table 7 Child Element of <ValidationErrors>

Element Name	Description	Data Type & Length
Error	Information about an error. See "<Error>," page 23.	Element

<Error>

The <Error> element contains information about an error.

```
<Error>

    (RecordNumber)

    (ErrorCode)

    (ErrorMessage)

    (ErrorRemedy)

</Error>
```

Table 8 Child Elements of <Error>

Element Name	Description	Data Type & Length
RecordNumber	Number of the record that has an error.	Numeric (10)
ErrorCode	Code that identifies the error. See " Error Codes ," page 23.	Numeric (10)
ErrorMessage	Message that describes the error.	Alphanumeric (250)
ErrorRemedy	Message that describes the remedy for the error.	Alphanumeric (250)

Error Codes

These error codes, messages, and remedies are included in the "<Error>" element. In the following table, the %s in the error messages will be replaced with dynamic values, such as batch IDs and record numbers, in the actual error messages.

Table 9 Error Codes, Messages, and Remedies

Error Code	Error Message	Remedy
101	Missing required field in File Header record: %s.	Include the required field in the file and try again.
102	MerchantID (%s) is not in our database.	Check to make sure merchantID is valid.
103	MerchantID (%s) does not match with the ID used to log in for uploading the file.	Check merchantID and try again.
104	batchID exceeds max length of %s.	Generate a shorter batchID and try again.

Table 9 Error Codes, Messages, and Remedies (Continued)

Error Code	Error Message	Remedy
105	%s had already passed validation stage.	batchID has previously been submitted. If this is a new batch, generate a new batchID and send the file again.
106	Invalid recordCount value: %s.	Check recordCount to make sure it is numeric and try again.
107	The file does not contain any data records.	The file must contain at least one data record. Correct and try again.
108	The recordCount exceeds the maximum number of records allowed per batch (%s).	Generate a smaller batch file and try again.
109	Invalid creationDate format: %s.	Correct the date format and try again (use yyyy-mm-dd).
110	Unsupported service: %s.	The service is not supported through batching. You may only request that service directly online.
111	Missing required field in Data Header record: %s.	Include the missing required field in the file and try again.
112	Missing required field in Data Record: %s.	Include the required field in the file and try again.
113	merchantID '%s' is not valid or has not been configured correctly.	Contact CyberSource to confirm that the MerchantID you are using is valid or is configured to be submitted by another merchantID.
114	SUM in Trailer record (%s) does not match with actual total (%s).	Make sure the SUM in the Trailer record matches with the actual sum of all data records.
115	SUM in Trailer record is invalid (%s).	Check to make sure the SUM value is numeric.
116	Missing required field in File Trailer record: %s.	Pass in required field and try again.
117	Duplicate file (batchID: %s).	Make sure that the batchID is unique.
118	Duplicate file. The data content in this file seems to have been submitted in batchID=%s.	Make sure the content of the file has not been previously submitted.
119	File Trailer record is missing.	Correct file format and try again.
120	There was a problem reading the input file.	Contact CyberSource to verify that file %s was received and is readable.
121	Found Data Header record but missing File Header record.	Correct file format and try again.
122	Found Data record but missing Data Header record.	Correct file format and try again.

Table 9 Error Codes, Messages, and Remedies (Continued)

Error Code	Error Message	Remedy
123	Number of fields (%s) does not match with number of field names (%s).	Correct file format and try again.
124	Found trailer record but missing File/Data Header record.	Correct file format and try again.
201	There was a problem inserting the batch status in the database.	Contact CyberSource.
202	Problem updating: batchID: %s, status: %s"	Contact CyberSource.
203	There was a problem retrieving the merchant configuration from the database.	Contact CyberSource.
204	There was a problem validating the batchID against the database.	Contact CyberSource.
205	There was a database problem while doing the checksum.	Contact CyberSource.
301	There was an internal error while validating the file.	Contact CyberSource.

DTD

```

<!ELEMENT Batch (ValidationStatus, ValidatedRecords, ValidationTime,
                ValidationErrors)>

<!ATTLIST Batch BatchID CDATA #REQUIRED
                MerchantID CDATA #REQUIRED
                Name CDATA #REQUIRED
                Version NMTOKEN #REQUIRED>

<!ELEMENT ValidationStatus (#PCDATA)>
<!ELEMENT ValidatedRecords (#PCDATA)>
<!ELEMENT ValidationTime (#PCDATA)>
<!ELEMENT ValidationErrors (Error)*>
<!ELEMENT Error (RecordNumber, ErrorCode, ErrorMessage, ErrorRemedy)>
<!ELEMENT RecordNumber (#PCDATA)>
<!ELEMENT ErrorCode (#PCDATA)>
<!ELEMENT ErrorMessage (#PCDATA)>
<!ELEMENT ErrorRemedy (#PCDATA)>

```

Examples

Example Success

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Batch SYSTEM "olp_validation_report_1_0.dtd">
<Batch BatchID="11111" MerchantID="gpntest" Name="OLP Validation
Report" Version="1.0">
    <ValidationStatus>Success</ValidationStatus>
    <ValidatedRecords>15</ValidatedRecords>
    <ValidationTime>2009-07-31T14:41:12</ValidationTime>
</Batch>

```

Example Failure Scenario 1

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE Batch SYSTEM "olp_validation_report_1_0.dtd">

<Batch BatchID="22222" MerchantID="gpntest" Name="OLP Validation
Report" Version="1.0">

    <ValidationStatus>Failed</ValidationStatus>

    <ValidationTime>2009-07-31T14:40:14</ValidationTime>

    <ValidationErrors>

        <Error>

            <RecordNumber>3</RecordNumber>

            <ErrorCode>123</ErrorCode>

            <ErrorMessage>Number of fields (17) does not match with
                number of field names (18).</ErrorMessage>

            <ErrorRemedy>Correct file format and try again.
                </ErrorRemedy>

        </Error>

    </ValidationErrors>

</Batch>
```

Example Failure Scenario 2

```

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE Batch SYSTEM "olp_validation_report_1_0.dtd">

<Batch BatchID="33333" MerchantID="gpntest" Name="OLP Validation Report"
Version="1.0">

    <ValidationStatus>Failed</ValidationStatus>

    <ValidationTime>2009-07-31T14:40:36</ValidationTime>

    <ValidationErrors>

        <Error>

            <RecordNumber>0</RecordNumber>

            <ErrorCode>114</ErrorCode>

            <ErrorMessage>SUM in Trailer record (14.00) does not match
                with actual total (15.00).</ErrorMessage>

            <ErrorRemedy>Make sure the SUM in the Trailer record matches
                with the actual sum of all data records.</ErrorRemedy>

        </Error>

    </ValidationErrors>

</Batch>

```

Viewing the Results of Your Batch File Requests

Reports and response files provide information about the requests in your batch files.

Reports

The following reports provide information about your batch file requests:

- **Batch Files Detail Report**—Provides details about the transactions that were batched using the upload process. See ["Batch Files Detail Report," page 31](#).
- **Batch Files Daily Summary Report**—Provides a summary of the batch transactions. See ["Batch Files Daily Summary Report," page 45](#).
- **Payment Submission Detail Report**—Shows all the transactions that were submitted to your processor for settlement. You can download the report daily. If the

batch file has an error, the file is not processed and so the file's requests are not included in the report. See the [Reporting Developer's Guide](#).

- **Batch Submission Detail Report**—Provides similar information as the Batch Files Detail Report, but returns a real-time response with details about the transactions that were batched using the upload process. See "[Batch Submission Detail Report: Overview](#)," page 47.

Response Files

After CyberSource processes all the requests in the batch file, CyberSource creates the following types of CSV-formatted response files that you can use to determine the results of the requests in the file:

- **Full file**—This file includes the results for all the requests in the batch file.
Filename format:
`<merchantID>.<batchID>.reply.all`
Filename example: `CyberVacations.12345.reply.all`
- **Exception file**—This file includes the results for the failed requests.
Filename format:
`<merchantID>.<batchID>.reply.rejected`
Filename example: `CyberVacations.12345.reply.rejected`

The response files are available in the **Reports** area on the [Business Center](#). You can download them the same way you download CyberSource reports, which is explained in the [Reporting Developer's Guide](#).

The format for these files is:

- A "File Header" followed by a blank line
- One or more "Data Records", each on a separate line

File Header

The file header consists of a list of comma-separated name-value pairs, including:

- merchantID
- batchID

Example Response File Header

```
merchantID=infodev,batchID=12345
```

Data Records

The data records provide the API reply information for the requests in the batch file. Each data record consists of a comma-separated list of name-value pairs containing the API reply information for a single request. The name-value pairs can be in any order.



Note

The order of the data records in the response file might not correspond to the order of the requests in your file. Use the value of the **merchantReferenceCode** field to link the result in the response file to the corresponding request from the batch file.

Example Response File

This example shows a full file, including two successful requests and one failed request. The failed request is the second data record in the example.

```
merchantID=infodev,batchID=12345

merchantReferenceCode=ABC12320398,ccCaptureReply_reasonCode=100,
reasonCode=100,decision=ACCEPT,ccCaptureReply_reconciliationID=
1018546244150167904178,requestID=1018546244150167904178,ccCaptureReply_
amount=327.49,ccCaptureReply_requestDateTime=2007-06-13T22:43:53Z,
purchaseTotals_currency=EUR

merchantReferenceCode=ABC141854,ccCaptureReply_reasonCode=241,
reasonCode=241,decision=REJECT,requestID=1018546227570167904150

merchantReferenceCode=ABC39882097,ccCreditReply_reasonCode=100,
reasonCode=100,decision=ACCEPT,ccCreditReply_reconciliationID=
1018546230720167904150,requestID=1018546230720167904150,ccCreditReply_
amount=14.99,ccCreditReply_requestDateTime=2005-09-23T22:44:33Z,
purchaseTotals_currency=CAD
```

Batch File Reports

There are three reports for batch files: the Batch Files Detail Report shows details of the transactions that were batched using the upload process, Batch Files Daily Summary Report displays a summary of the batch transactions, and the Batch Submission Detail Report returns a real-time response with details about the transactions that were batched using the upload process.

**Note**

If you request a report too soon after submitting a batch, you may receive an error because the batch has not finished processing. Try the report request again later.

Batch Files Detail Report

The Batch Files Detail Report provides detailed information about the transactions that were batched. See the [Reporting Developer's Guide](#) for information about:

- Downloading CyberSource reports
- XML conventions
- CSV conventions

XML Format

Elements

<Report>

The <Report> element is the root element of the report.

```

<Report Name=CDATA
    Version=NMTOKEN
    xmlns=CDATA
    MerchantID=CDATA
    ReportStartDate=CDATA
    ReportEndDate=CDATA>
    (BatchFiles)
</Report>

```

Table 10 Attributes of <Report>

Attribute Name	Description	Data Type & Length
Name	Name of the report. This element always contains the text <code>Batch Files Detail Report</code> .	Alphanumeric (100)
Version	Version number of the report. The current version number is 1.0.	Numeric (10)
xmlns	XML namespace for the report. The namespace for the current version is <code>http://reports.cybersource.com/reports/bfdr/1.0</code> .	Alphanumeric (100)
MerchantID	CyberSource merchant ID used for the transactions in the report.	Alphanumeric (30)
ReportStartDate	First date included in the report.	DateTime (25)
ReportEndDate	Last date included in the report.	DateTime (25)

Table 11 Child Elements of <Report>

Element Name	Description
<BatchFiles>	Batch files that are included in the report. See " <BatchFiles> ," page 33 for a list of child elements.

Example <Report> Element

```
<Report Name="Batch Files Detail Report"
  Version="1.0"
  xmlns="http://reports.cybersource.com/reports/bfdr/1.0"
  MerchantID="pcpawncshop"
  ReportStartDate="2006-09-29T05:00:00-05:00"
  ReportEndDate="2006-09-30T05:00:00-05:00">
  <BatchFiles>
    ...
  </BatchFiles>
</Report>
```

<BatchFiles>

The <BatchFiles> element contains all of the batch files that are included in the report.

```
<BatchFiles>
  (BatchFile)*
</BatchFiles>
```

Table 12 Child Elements of <BatchFiles>

Element Name	Description
<BatchFile>	Payment processors for the transactions in the batch file. See " <BatchFile> ," page 34 for a list of attributes and child elements.

Example <BatchFiles> Element

```

<BatchFiles>

    <BatchFile BatchFileID="123">

        ...

    </BatchFile>

</BatchFiles>

```

<BatchFile>

The <BatchFile> element contains the payment processors for the transactions in the batch file.

```

<BatchFile BatchFileID=CDATA>

    (PaymentProcessor)*

</BatchFile>

```

Table 13 Attributes of <BatchFile>

Attribute Name	Description	Data Type & Length
BatchFileID	CyberSource batch file in which the transactions were sent.	Numeric (39)

Table 14 Child Elements of <BatchFile>

Element Name	Description
<PaymentProcessor>	Requests associated with the payment processor. See " <PaymentProcessor> ," page 35 for a list of attributes and child elements.

Example <BatchFile> Element

```

<BatchFile BatchFileID="10101">

    <PaymentProcessor PaymentProcessorName="vital">

        ...

    </PaymentProcessor>

</BatchFile>

```

<PaymentProcessor>

The <PaymentProcessor> element contains the requests associated with a payment processor.

```

<PaymentProcessor PaymentProcessorName=CDATA>

    (Request)*

</PaymentProcessor>

```

Table 15 Attributes of <PaymentProcessor>

Attribute Name	Description	Data Type & Length
PaymentProcessorName	Name of a payment processor.	Alphanumeric (30)

Table 16 Child Elements of <PaymentProcessor>

Element Name	Description
<Request>	Information about a payment transaction. See " <Request> ," page 36 for a list of attributes.

Example <PaymentProcessor> Element

```

<PaymentProcessor PaymentProcessorName="vital">

    <Request>

        ...

    </Request>

</PaymentProcessor>

```

<Request>

The <Request> element contains information about a payment transaction.

```

<Request RequestID=CDATA>

    (TransactionReferenceNumber)

    (MerchantReferenceNumber)

    (TransactionStatus)

    (Amount)

    (CurrencyCode)

    (PaymentStatus)

</Request>

```

Table 17 Attributes of <Request>

Attribute Name	Description	Data Type & Length
RequestID	Unique identifier generated by CyberSource for the transaction.	Numeric (26)

Table 18 Child Elements of <Request>

Element Name	Description	Data Type & Length
Transaction Reference Number	Reference number that you use to reconcile your CyberSource reports with your processor reports. This field corresponds to the <service>_reconciliationID (Simple Order API) and to the <service>_trans_ref_no (SCMP API) reply fields.	Alphanumeric (60)

Table 18 Child Elements of <Request> (Continued)

Element Name	Description	Data Type & Length
Merchant Reference Number	Merchant-generated order reference or tracking number.	Alphanumeric (50)
Transaction Status	One-word description of the result of the transaction request.	Alphanumeric (50)
Amount	Amount of the transaction.	Amount (19)
CurrencyCode	ISO currency code used for the transaction.	Alphanumeric (5)
PaymentStatus	One-word description of the current status of the transaction. Possible values: <ul style="list-style-type: none"> ■ BATCH_ERROR ■ BATCH_RESET ■ BATCHED ■ CANCELED_REVERS ■ CANCELLED ■ DENIED ■ FAILED ■ PENDING ■ REFUNDED ■ REVERSED ■ TRXN_ERROR ■ VOIDED 	Alphanumeric (50)

Example <Request> Element

```
<Request RequestID="1580782287420174065733">  
  
    <TransactionReferenceNumber>5533830406</  
TransactionReferenceNumber>  
  
    <MerchantReferenceNumber>1158078228539</MerchantReferenceNumber>  
  
    <TransactionStatus>SOK</TransactionStatus>  
  
    <Amount>25.00</Amount>  
  
    <CurrencyCode>USD</CurrencyCode>  
  
    <PaymentStatus>PENDING</PaymentStatus>  
  
</Request>
```

DTD

```

<!ELEMENT Report (BatchFiles)>

<!ATTLIST Report Name CDATA #REQUIRED

                Version NMTOKEN #REQUIRED

                xmlns CDATA #REQUIRED

                MerchantID CDATA #REQUIRED

                ReportStartDate CDATA #REQUIRED

                ReportEndDate CDATA #REQUIRED>

<!ELEMENT BatchFiles (BatchFile)*>

<!ELEMENT BatchFile (PaymentProcessor)*>

<!ATTLIST BatchFile BatchFileID CDATA #REQUIRED>

<!ELEMENT PaymentProcessor (Request)*>

<!ATTLIST PaymentProcessor PaymentProcessorName CDATA #REQUIRED>

<!ELEMENT Request (TransactionReferenceNumber, MerchantReferenceNumber,
                TransactionStatus, Amount, CurrencyCode, PaymentStatus)>

<!ATTLIST Request RequestID CDATA #REQUIRED>

<!ELEMENT TransactionReferenceNumber (#PCDATA)>

<!ELEMENT MerchantReferenceNumber (#PCDATA)>

<!ELEMENT TransactionStatus (#PCDATA)>

<!ELEMENT Amount (#PCDATA)>

<!ELEMENT CurrencyCode (#PCDATA)>

<!ELEMENT PaymentStatus (#PCDATA)>

```

Example

The following example shows a report that contains two batch files. The first batch file contains three requests and the second batch file contains one request.

```

<?xml version="1.0" encoding="utf-8"?>

<!DOCTYPE Report SYSTEM "https://ebctest.cybersource.com/ebctest/
reports/dtd/bfdr.dtd">

<Report Name="Batch Files Detail Report"

    Version="1.0"

    xmlns="https://ebctest.cybersource.com/ebctest/reports/dtd/
bfdr.dtd"

    MerchantID="pcpawshop"

    ReportStartDate="2006-09-29T05:00:00-05:00"

    ReportEndDate="2006-09-30T05:00:00-05:00">

  <BatchFiles>

    <BatchFile BatchFileID="127788">

      <PaymentProcessor PaymentProcessorName="vital">

        <Request RequestID="1595558344253232243215">

          <TransactionReferenceNumber>7242635150</
TransactionReferenceNumber>

          <MerchantReferenceNumber>1158078228539</
MerchantReferenceNumber>

          <TransactionStatus>SOK</TransactionStatus>

          <Amount>25.00</Amount>

          <CurrencyCode>USD</CurrencyCode>

          <PaymentStatus>PENDING</PaymentStatus>

        </Request>

        <Request RequestID="1595558354743232243215">

          <TransactionReferenceNumber>7242636613</
TransactionReferenceNumber>

          <MerchantReferenceNumber>1158078892610</
MerchantReferenceNumber>

          <TransactionStatus>SOK</TransactionStatus>

          <Amount>100.00</Amount>

```

```
<CurrencyCode>USD</CurrencyCode>

    <PaymentStatus>PENDING</PaymentStatus>

</Request>

<Request RequestID="1595558364563232243215">

    <TransactionReferenceNumber>7242637653</
TransactionReferenceNumber>

    <MerchantReferenceNumber>1158079157035</
MerchantReferenceNumber>

    <TransactionStatus>SOK</TransactionStatus>

    <Amount>99.00</Amount>

    <CurrencyCode>USD</CurrencyCode>

    <PaymentStatus>VOIDED</PaymentStatus>

</Request>

</PaymentProcessor>

</BatchFile>

<BatchFile BatchFileID="123987">

    <PaymentProcessor PaymentProcessorName="smartfdc">

        <Request RequestID="1595564779663232243215">

            <TransactionReferenceNumber>7243278653
            </TransactionReferenceNumber>

            <MerchantReferenceNumber>1159429157035
            </MerchantReferenceNumber>

            <TransactionStatus>SOK</TransactionStatus>

            <Amount>4.00</Amount>

            <CurrencyCode>USD</CurrencyCode>

            <PaymentStatus>PENDING</PaymentStatus>

        </Request>

    </PaymentProcessor>

</BatchFile>

</BatchFiles>

</Report>
```

CSV Format

Records

First Header Record

The first header record describes the name and version of the report and indicates which dates are included in the report.

Example First Header Record

```
Batch Files Detail Report,1,2006-09-29 to 2006-09-30,,,,,,,,
```

Table 19 Fields in the First Header Record

Position	Field Name	Description	Data Type & Length
1 (A)	report_name	Name of the report. This field always contains the text <i>Batch Files Detail Report</i> .	Alphanumeric (100)
2 (B)	version_number	Version number of the report. The current version number is 1.	Numeric (10)
3 (C)	date_range	Dates included in the report in the format <i>YYYY-MM-DD to YYYY-MM-DD</i> . The first date is the start date; the second date is the end date.	Alphanumeric (100)

Second Header Record

The second header record indicates the name of each field in the report. The fields in the second header record follow these rules:

- The content of each field is the same as the field name.
- The data type and length of each field is alphanumeric (100).

Example Second Header Record

```
merchant_id,txn_batch_id,payment_processor,request_id,trans_ref_  
no,merchant_ref_number,ics_rflag,amount,currency,action
```

Transaction Record

Each transaction record contains information about a CyberSource payment transaction.

Example Transaction Record

```
pcpawshop,127788,vital,9979040000003515181891,7242635150,1158078228539
,SOK,25.00,USD,PENDING
```

Table 20 Fields in the Transaction Record

Position	Field Name	Description	Data Type & Length
1	merchant_id	CyberSource merchant ID used for the transaction.	Alphanumeric (30)
2	txn_batch_id	CyberSource batch file in which the transactions were sent.	Numeric (39)
3	payment_processor	Name of a payment processor.	Alphanumeric (30)
4	request_id	Identifier for the transaction.	Numeric (26)
5	trans_ref_no	Reference number that you use to reconcile your CyberSource reports with your processor reports. This field corresponds to the <service>_reconciliationID (Simple Order API) and to the <service>_trans_ref_no (SCMP API) reply fields.	Alphanumeric (60)
6	merchant_ref_number	Merchant-generated order reference or tracking number.	Alphanumeric (50)
7	ics_rflag	One-word description of the result of the transaction request.	Alphanumeric (50)
8	amount	Amount of the transaction.	Amount (19)
9	currency	ISO currency code used for the transaction.	Alphanumeric (5)

Table 20 Fields in the Transaction Record (Continued)

Position	Field Name	Description	Data Type & Length
10	action	One-word description of the current status of the transaction. Possible values: <ul style="list-style-type: none"> ■ BATCH_ERROR ■ BATCH_RESET ■ BATCHED ■ CANCELED_REVERS ■ CANCELLED ■ DENIED ■ FAILED ■ PENDING ■ REFUNDED ■ REVERSED ■ TRXN_ERROR ■ VOIDED 	Alphanumeric (50)

Example

The following example shows a report that contains two batch files. The first batch file contains three requests and the second batch file contains one request.

```

Batch Files Detail Report,1,2006-09-29 to 2006-09-30,,,,,,
merchant_id,txn_batch_id,payment_processor,request_id,trans_ref_no,
merchant_ref_number,ics_rflag,amount,currency,action
pcpawncshop,127788,vital,1598344253232243215,7242635150,1158078228539,SO
K,25.00,USD,PENDING
pcpawncshop,127788,vital,1598354743232243226,7242636613,1158078892610,SO
K,100.00,USD,PENDING
pcpawncshop,127788,vital,1598364563232243237,7242637653,1158079157035,SO
K,99.00,USD,VOIDED
pcpawncshop,123987,smartfdc,1594779663232243248,7243278653,1159429157035
,SOK,4.00,USD,PENDING

```

Batch Files Daily Summary Report

The Batch Files Daily Summary Report is a daily report that summarizes batch transactions as shown in the following figure:

Figure 1 Batch Files Daily Summary Report

Batch Files Daily Summary Report				
struong_acct September 21, 2006				
Batch ID	123123			104
Processor	paymenttech	Total Count		70
Debits				Total Count 70
Transaction Count	Payment Status	Amount	Currency	
1	ERROR	50.00	USD	
67	PENDING	8,976.37	USD	
2	TRANSMITTED	100.00	USD	
Processor	vital	Total Count		34
Debits				Total Count 31
Transaction Count	Payment Status	Amount	Currency	
4	ERROR	250.00	USD	
18	PENDING	1,500.00	USD	
9	TRANSMITTED	1,874.95	USD	
Credits				Total Count 3
Transaction Count	Payment Status	Amount	Currency	
3	TRANSMITTED	100.00	USD	
Batch ID	123678			13
Processor	paymenttech	Total Count		13
Debits				Total Count 13
Transaction Count	Payment Status	Amount	Currency	
2	ERROR	50.00	USD	
9	PENDING	1,432.93	USD	
2	TRANSMITTED	100.00	USD	

- The report consists of a main section (1) for each batch file. The heading for each batch file displays the batch file ID and the total number of transactions in the batch file.
- For each batch file, the report is divided into sections for each processor (2) included in the batch file. The heading for each processor displays the name of the processor and the total number of transactions that were performed for the processor.
- For each processor, the report shows the types of transactions (3) that were performed, such as credits and debits. The heading for each type of transaction displays the total number of transactions for that type of transaction.
- For each type of transaction, the report shows the totals for each different payment status (4), such as ERROR, PENDING, or TRANSMITTED. The line for each payment status displays the total number of transactions with that payment status, the total

amount for all the transactions with that payment status, and the currency used for the transactions with that payment status.

To obtain the Batch Files Daily Summary Report:

- Step 1** Log in to the Business Center.
- Step 2** In the navigation pane, choose **Reports > Report Search**.
- Step 3** In the Report pull-down menu, choose **Batch Files Daily Summary Report**.
- Step 4** Choose a start date and click **Submit**.
- Step 5** On the Report Search Results page, click the link to view the report.
You can download the report in PDF and CSV formats. Links for these downloads are in the upper right corner of the Report View area.

Batch Submission Detail Report: Overview

The Batch Submission Detail Report provides real-time detailed status information about the transactions that you previously uploaded in the Business Center or processed with the Offline Transaction File Submission service.

This report can be downloaded from the Business Center, or programmatically. See the [Reporting Developer's Guide](#) for information about:

- Downloading CyberSource reports
- XML conventions
- CSV conventions

Requesting Report with Query API

The query uses a POST method with search parameters to obtain the details of a batch file. The content of the report is described on [page 30](#), the DTDs on [page 49](#), and the samples on [page 51](#).

Formulating the Query

You can request this report as many times as you wish each day. [Table 21](#) describes the request information.

To request the report, send the required data described in [Table 21](#) to one of these URLs:

Production <https://ebc.cybersource.com/ebc/batchupload/ResponseFileWrite.do>
 Test <https://ebctest.cybersource.com/ebctest/batchupload/ResponseFileWrite.do>

Table 21 Required Data for Report Request

Parameter	Value	Required/Optional
merchantID	Your CyberSource merchant ID.	Required
username	This optional field can be different from the merchant ID. If this field is empty, the merchant ID will be used to generate the report. Important If you use this field, make sure that the user name has the permission to download reports. Otherwise, you will receive an error message and will not be able to download the report.	Optional
password		Required
batchID	CyberSource batch file in which the transactions were sent	Required

Table 21 Required Data for Report Request (Continued)

Parameter	Value	Required/ Optional
Format	xml csv	Required

XML format

To use the XML format, write a program that can send the required fields in a POST request.

HTML format

To use the HTML format, write an HTML form with one of the URLs listed above. The following sample shows the `form` tag with the action to send a report request to the production URL. The form includes the required fields and a Submit button.

```
<html>

<body>

<form action="https://ebc.cybersource.com/ebc/batchupload/
ResponseFileWrite.do" method="post">

<input type="hidden" name="merchantID" value="examplemerchantid"/>

<input type="hidden" name="username" value="exampleusername"/>

<input type="password" name="password" value="password"/>

<input type="hidden" name="batchID" value="12345678"/>

<input type="hidden" name="format" value="csv"/>

<button type="submit">

</form>

</body>

</html>
```

Viewing and Saving the Report

After sending a request, you receive a response immediately. The report contains real-time information on batched transactions.

XML format

If the query is successful, the results appear as a document of mime type `application/xml`. To use this report, write a program to save or process the XML data in the report.

HTML format

If you use a browser that supports XSLT, such as Microsoft Internet Explorer 6.0+ or Firefox 2.0, the XML file that you receive can be converted to HTML and formatted into a table that you can see in your browser. Save the report as follows:

Step 1 Right-click the report.

Step 2 Click **View Source**.

If you do not save the source, only the URL for the request is saved. The report appears in your default text editor.

Step 3 Save the text file as an XML file.

The table representation is restored when you view the report again in the browser.

For a sample report, see [page 51](#).

DTD

```
<!ELEMENT Report (Transaction)*>

<!ATTLIST Report MerchantID CDATA #REQUIRED
                Name CDATA #REQUIRED
                SubmissionFileID CDATA #REQUIRED
                SubmissionDateTime CDATA #REQUIRED
                Version NMTOKEN #REQUIRED>

<!ELEMENT Transaction (LinkToRequest?, RequestID, TransactionDate,
CybsMID, ProcessorMID?, HierarchyID?, TransRefNumber?, MerchantRefNumber?,
TransactionType?, Amount?, TransactionAmountCurrency?, PaymentMethod?,
PaymentType?, AccountSuffix?, Decision?, ReasonCode?, Auth?,
MerchantDefinedData1?, MerchantDefinedData2?, MerchantDefinedData3?,
MerchantDefinedData4?)>

<!ELEMENT LinkToRequest (#PCDATA)>

<!ELEMENT RequestID (#PCDATA)>

<!ELEMENT TransactionDate (#PCDATA)>

<!ELEMENT CybsMID (#PCDATA)>

<!ELEMENT ProcessorMID (#PCDATA)>

<!ELEMENT HierarchyID (#PCDATA)>

<!ELEMENT TransRefNumber (#PCDATA)>
```

```
<!ELEMENT MerchantRefNumber (#PCDATA)>
<!ELEMENT TransactionType (#PCDATA)>
<!ELEMENT Amount (#PCDATA)>
<!ELEMENT TransactionAmountCurrency (#PCDATA)>
<!ELEMENT PaymentMethod (#PCDATA)>
<!ELEMENT PaymentType (#PCDATA)>
<!ELEMENT AccountSuffix (#PCDATA)>
<!ELEMENT Decision (#PCDATA)>
<!ELEMENT ReasonCode (#PCDATA)>
<!ELEMENT Auth (TransRefNumber?, TransactionDate?, RequestID?, Amount?,
Currency?, AuthCode?, ReasonCode?, RCode?)>
<!ELEMENT TransRefNumber (#PCDATA)>
<!ELEMENT TransactionDate (#PCDATA)>
<!ELEMENT RequestID (#PCDATA)>
<!ELEMENT Amount (#PCDATA)>
<!ELEMENT Currency (#PCDATA)>
<!ELEMENT AuthCode (#PCDATA)>
<!ELEMENT ReasonCode (#PCDATA)>
<!ELEMENT RCode (#PCDATA)>
<!ELEMENT MerchantDefinedData1 (#PCDATA)>
<!ELEMENT MerchantDefinedData2 (#PCDATA)>
<!ELEMENT MerchantDefinedData3 (#PCDATA)>
<!ELEMENT MerchantDefinedData4 (#PCDATA)>
```

Example

The following example shows a report that contains one batch file.

```

<?xml version="1.0" encoding="utf-8"?>

<!DOCTYPE Report SYSTEM "https://ebctest.cybersource.com/ebctest/
reports/dtd/bsd.dtd">

<Report Name="Batch Submission Detail Report"

    Version="1.0"

    xmlns="https://ebctest.cybersource.com/ebctest/reports/dtd/
bsd.dtd"

    Name="Batch Submission Detail Report"

    SubmissionFileID="12345678"

    SubmissionDateTime="2010-03-18 15:15:40 GMT"

    Version="1.0">

<Transaction>

    <LinkToRequest>2689254011060008415089</LinkToRequest>

    <RequestID>2689254011060008415089</RequestID>

    <TransactionDate>2010-03-18 15:16:41 GMT</TransactionDate>

    <CybsMID>examplemerchantid</CybsMID>

    <TransRefNumber>81389795F6RU7QH0</TransRefNumber>

    <MerchantRefNumber>12345-8569-8794654</MerchantRefNumber>

    <TransactionType>ics_bill,ics_auth</TransactionType>

    <Amount>1.00</Amount>

    <TransactionAmountCurrency>USD</TransactionAmountCurrency>

    <PaymentMethod>credit card</PaymentMethod>

    <PaymentType>MasterCard</PaymentType>

    <AccountSuffix>1234</AccountSuffix>

    <ReasonCode>231</ReasonCode>

    <Auth>

        <RequestID>2689254011060008415089</RequestID>

    </Auth>

</Transaction>

</Report>

```

Downloading a Report from the Business Center

You can access the Batch Submission Detail Report from the Tools & Settings section of the Business Center. Follow the steps below to download the report.

- Step 1** Open a Web browser:
- Microsoft Internet Explorer 4.0 or greater
 - Netscape Navigator 6.2 or greater
- Step 2** Enter the production URL for the Business Center: <https://ebc.cybersource.com>.
- Step 3** Choose **Tools & Settings > Batch Transactions > Detail Report**.
- Step 4** Enter the batch ID and choose format: XML or CSV.
- Step 5** Click **Get Report**.
You receive a response immediately.

Using Java Sample Code to Upload Files

Requirements

- J2SE 1.4 or higher
- Unlimited Strength Jurisdiction Policy files from Oracle (`US_export_policy.jar` and `local_policy.jar`), available at <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- Bouncy Castle, which includes `bcmail*.jar`, `bcpg*.jar`, `bcprov*.jar` and `bctest*.jar`, available at www.bouncycastle.org
- View CyberSource's sample code package at http://apps.cybersource.com/cgi-bin/pages/additional.cgi?kit=Offline_Transaction_File_Submission_Sample_Code

Using Sample Code for Basic Authentication

**Note**

The sample code was developed and tested on a Solaris platform.

- Step 1** Replace your Java installation's existing security policy files with the new ones you downloaded from Oracle's site:
- Find your existing `US_export_policy.jar` and `local_policy.jar` files in the `$JAVA_HOME/jre/lib/security` directory.
 - Rename or move your existing files to another directory.
 - Copy the new `US_export_policy.jar` and `local_policy.jar` files that you downloaded from Oracle to the `$JAVA_HOME/jre/lib/security` directory.

Step 2 Copy the Bouncy Castle *.jar files to the \$JAVA_HOME/jre/lib/ext directory.

Step 3 Edit the \$JAVA_HOME/jre/lib/security/java.security file and insert the security provider right after the Oracle provider. Make sure to increment the numbers of the other providers in the list. The line to insert is this:

```
security.provider.2=org.bouncycastle.jce.provider.BouncyCastleProvider
```

Your list of security providers will now look similar to this:

```
security.provider.1=sun.security.provider.Sun
```

```
security.provider.2=org.bouncycastle.jce.provider.BouncyCastleProvider
```

```
security.provider.3=com.sun.net.ssl.internal.ssl.Provider
```

```
security.provider.4=com.sun.rsa.jca.Provider
```

```
security.provider.5=com.sun.crypto.provider.SunJCE
```

```
security.provider.6=sun.security.jgss.SunProvider
```

Step 4 Import your CyberSource Simple Order API .p12 security key into Internet Explorer.

- a** Open an Internet Explorer Web browser, go to **Tools > Internet Options...**, and click the **Content** tab.
- b** Click **Certificates...**
- c** Click **Import...** to open the Certificate Import Wizard and click **Next** to start the Wizard.
- d** Browse to the location of your .p12 security key and click **Next**.
For the password for the private key, enter your CyberSource merchant ID. For example, if your key is infodev.p12, enter **infodev** as the password.
- e** On this page, click the check box for **Mark this key as exportable** and then click **Next**.
- f** Click **Next** on the Certificate Store page.
- g** Click **Finish** and you will see a confirmation that the import was successful.

Step 5 Create a key store file to contain your CyberSource Simple Order API .p12 security key:

- a** In a Web browser, go to one of the following URLs:

If you are in test mode and have not gone live with CyberSource:

<https://batchtest.cybersource.com/upload/UploadBatchFile>

If you have gone live with CyberSource:

<https://batch.cybersource.com/upload/UploadBatchFile>

- b** Go to File > Properties.
- c** Click **Certificates**.
- d** Click the **Certification Path** tab.
- e** Select **Entrust.net Secure Server Certification Authority**.
- f** Click **View Certificate**.

- g** Click the **Details** tab.
- h** Click **Copy to File** and then **Next**.
- i** Click Browse and navigate to where you want to save the file.
- j** Enter the name you want to use for the file, such as `MyCert`. Click **Save** and click **Next**.
- k** Click Finish.

Your file, for example `MyCert.cer`, has been created in the location you specified.

- l** Go to `$JAVA_HOME/bin/keytool` and use the J2SE `keytool` program to create a keystore file that contains this newly created certificate. You will need to provide a pass phrase for the keystore. You **MUST** use the same password that you used in Step d on [page 54](#), which is part of [Step 4](#) above. For example, if your p12 key is `infodev.p12`, the pass phrase must be `infodev`.

To create the keystore, type the following command:

```
$JAVA_HOME/bin/keytool -import -file <path to certificate>/
<name of certificate file> -keystore <name of keystore
file>.jks -storepass <pass phrase of keystore>
```

For example:

```
$JAVA_HOME/bin/keytool -import -file /home/bluu/MyCert.cer -
keystore MyKeystore.jks -storepass myMerchantID
```

If successful, the output will be similar to this:

```
Owner: CN=batchtest.cybersource.com, OU=Operations,
O=Cybersource Corporation, L=Mountain View, ST=California, C=US
Issuer: CN=Entrust.net Secure Server Certification Authority,
OU=(c) 1999 Entrust.net Limited, OU=www.entrust.net/CPS incorp.
by ref. (limits liab.), O=Entrust.net, C=US
Serial number: 374e1b7b
Valid from: Thu Nov 18 17:15:34 PST 2004 until: Tue Jan 31
17:51:24 PST 2006
Certificate fingerprints:
    MD5:  BE:BF:B0:91:69:C4:7B:10:45:EC:D6:0F:16:AA:3D:77
    SHA1: 07:F8:41:DC:B2:FC:F5:DA:FC:EE:09:7A:33:B8:29:15:31:18
Trust this certificate? [no]: yes
Certificate was added to keystore
```

- Step 6** Modify the `SSLFileTransfer.props` file with your settings. The file is part of the CyberSource download package and looks similar to this:

```
# Upload host
host=batchtest.cybersource.com

# Upload port
port=<Upload port>

# Username to log into the Business Center
```

```

bcUserName=<Business Center login name>

# Password to log into the Business Center
bcPassword=<Business Center login password>

# File to upload
uploadFile=<path to your file>/<file name>

# Path where to upload it to (provided by CyberSource)
path=/upload/UploadBatchFile

# Your CyberSource security key
key=<key location path>/<key file name>

# New key store you just created that contains the certificate
keyStore=<key store location>/<new key store name>

# Pass phrase is the string you used in -storepass option when you
# created the key store file earlier
passPhrase=<pass phrase>

```

Step 7 Set the `JAVA_HOME` environment variable to where you installed J2SE. For example:

```
JAVA_HOME=/home/j2se
```

Step 8 Include `$JAVA_HOME/bin` in the `PATH`.

Step 9 Compile and run the sample:

a Change to the directory where you stored the CyberSource sample files.

b Type the following:

```
javac SSLFileTransfer.java
```

```
java SSLFileTransfer <path to props file>/SSLFileTransfer.props
```

If the upload is successful, the output should look similar to this:

```

HTTP/1.1 200 OK
Date: Wed, 26 Jan 2005 17:26:31 GMT
Server: Apache Coyote/1.0
Content-Type: text/plain
Content-Length: 0
X-Cache: MISS from <your host>
Connection: close
UPLOAD FILE SUCCESSFUL

```


Sample Code for Basic Authentication

The sample below shows how you can upload a batch file to a batch server using Basic Authentication.



Note

You need to install Bouncy Castle to run this sample code.

```
import java.util.*;
import java.net.*;
import java.io.*;
import javax.net.ssl.*;
import javax.security.cert.X509Certificate;
import java.security.*;

/*
 * This class is to upload files but can be expanded to download
 * files also.
 */
public class SSLFileTransfer {

    Properties props = new Properties(); // stores properties from
    property file

    /*
     * SSLFileTransfer(): constructor
     */
    public SSLFileTransfer() {
    }

    /*
     * init(): initialization (load property file)
     *
     * @param propsFileproperties needed for file transfer
     */
    public void init(String propsFile) {
        try {
            props.load(new BufferedInputStream(new FileInputStream(new
            File(propsFile))));
        } catch (Exception e) {
            e.printStackTrace();
            System.exit(-1);
        }
    }
}
```

```

/*
 * usage()
 */
public static void usage() {
    System.out.println("USAGE: java SSLFileTransfer <full path
property file name>");
    System.exit(-1);
}

/*
 * getFactory(): get factory for authentication
 *
 * @throws IOException if exception occurs
 */
private SSLSocketFactory getFactory() throws IOException {
    try {
        SSLContext ctx;
        KeyManagerFactory kmf;
        KeyStore ks, ks1;
        char[] passphrase =
props.getProperty("passPhrase").toCharArray();

        ctx = SSLContext.getInstance("TLS");
        kmf = KeyManagerFactory.getInstance("SunX509");
        ks = KeyStore.getInstance("PKCS12", "BC");
        ks1 = KeyStore.getInstance("JKS");

        ks.load(new FileInputStream(props.getProperty("key")),
passphrase);
        ks1.load(new FileInputStream(props.getProperty("keyStore")),
passphrase);

        kmf.init(ks, passphrase);
        TrustManagerFactory tmf =
TrustManagerFactory.getInstance("SunX509");
        tmf.init(ks1);
        ctx.init(kmf.getKeyManagers(), tmf.getTrustManagers(), null);

        return ctx.getSocketFactory();
    } catch (Exception e) {
        e.printStackTrace();
        throw new IOException(e.getMessage());
    }
}

/*
 * getHost(): Get host from property file
 */

```

```

private String getHost() {
    return props.getProperty("host", "localhost");
}

/*
 * getPort(): Get port from property file
 */
private int getPort() {
    return Integer.parseInt(props.getProperty("port"));
}

/*
 * sendRequest(): Send request (file) to the server
 *
 * @param outstream to send the data to the server
 *
 * @throws Exception if an error occurs.
 */
private void sendRequest(PrintWriter out)
    throws Exception
{
    String path = props.getProperty("path");

    out.println("POST " + path + " HTTP/1.0");

    final String BOUNDARY = "7d03135102b8";
    out.println("Host: " + props.getProperty("host"));
    out.println("Content-Type: multipart/form-data;
boundary="+BOUNDARY);

    String uploadFile = props.getProperty("uploadFile");
    String authString = props.getProperty("bcUserName") + ":" +
props.getProperty("bcPassword");
    String encodedAuthString = "Basic " + new
sun.misc.BASE64Encoder().encode(authString.getBytes ());
    out.println("Authorization: " + encodedAuthString);

    final String CRLF = "\r\n";
    StringBuffer sbuf = new StringBuffer();
    sbuf.append("--"+BOUNDARY+CRLF);
    sbuf.append("Content-Disposition: form-data; name=\"upfile\";
filename=\"\" + uploadFile + "\""+CRLF);
    sbuf.append("Content-Type: text/plain"+CRLF+CRLF);

    FileReader fi = new FileReader(uploadFile);
    char[] buf = new char[1024000];
    int cnt = fi.read(buf);
    sbuf.append(buf, 0, cnt);
}

```

```

sbuf.append(CRLF);
sbuf.append("--"+BOUNDARY+"--"+CRLF);

int sz = sbuf.length();
out.println("Content-Length: "+ sz);

out.println();
out.println(sbuf);
out.flush();

// Make sure there were no surprises
if (out.checkError())
    System.out.println("SSLFileTransfer: java.io.PrintWriter
error");
}

/*
 * readResponse(): reads response from the server
 *
 * @param instream to get the data from the server
 *
 * @throws Exception if an error occurs.
 */
private void readResponse(BufferedReader in)
    throws Exception
{
    boolean successful = false;
    String inputLine;
    while ((inputLine = in.readLine()) != null) {
        if (inputLine.startsWith("HTTP") && inputLine.indexOf("200")
>= 0)
            successful = true;
        System.out.println(inputLine);
    }

    System.out.println("UPLOAD FILE " + (successful? "SUCCESSFUL" :
"FAILED") + "!!!\n");
}

/*
 * upload(): upload file to server
 *
 * @throws Exception if an error occurs.
 */
public void upload()
    throws Exception
{
    try {

```

```

        SSLSocketFactory factory = getFactory();
        SSLSocket socket = (SSLSocket)factory.createSocket(getHost(),
getPort());

        PrintWriter out = new PrintWriter(new BufferedWriter(new
OutputStreamWriter(socket.getOutputStream())));
        BufferedReader in = new BufferedReader(new
InputStreamReader(socket.getInputStream()));

        socket.startHandshake();
        sendRequest(out);
        readResponse(in);

        out.close();
        in.close();
        socket.close();
    } catch (Exception e) {
        e.printStackTrace();
        throw e;
    }
}

/*
 * main(): main method to start file transfer
 *
 * @param argscommand line arguments (property file, see
usage())
 *
 * @throws Exceptionif an error occurs.
 */
public static void main(String[] args) throws Exception {
    if (args == null || args.length != 1)
        usage();

    SSLFileTransfer fileXfer = new SSLFileTransfer();
    fileXfer.init(args[0]);
    fileXfer.upload();
}
}

```

API Field Mapping

The Offline Transaction File Submission system uses fields from the CyberSource Simple Order API, not from the SCMP API. If you use the SCMP API to process individual transactions, you will need to know which Simple Order API fields correspond to the SCMP API fields with which you are familiar. This appendix provides an alphabetical list of the SCMP API field names and the corresponding Simple Order API field names.

The tables include a comprehensive list of the SCMP API fields that have corresponding Simple Order API fields. Some of the fields might be for CyberSource services that you are not allowed to request in a batch file. See ["Offline Transaction File Submission," page 8](#) for a list of those services.

To request CyberSource services in the SCMP API, set the **ics_applications** field to a comma-separated list of the services you want to run. For example, if you want to request a credit card authorization and capture, you set the requests for the services that you want as follows:

- SCMP API: **ics_applications=ics_auth,ics_bill**
- Simple Order API: set **ccAuthService_run=true** and **ccCaptureService_run=true**

Note also that the core reply fields for the SCMP API and the Simple Order API are different. The following table describes the fields. For more information about interpreting each API's reply, see [Credit Card Services with the Simple Order API](#) and [Credit Card Services with the SCMP API](#).

See the examples in ["File Examples," page 17](#) to help you understand how to use the Simple Order API fields in the requests in a batch file.

Reply-Level Fields

Table 1 Core Reply Fields for the SCMP API and the Simple Order API

SCMP API Reply Field and Possible Values	SO API Reply Field and Possible Values
ics_rcode and <service>_rcode <ul style="list-style-type: none"> ▪ -1 (error) ▪ 0 (declined) ▪ 1 (accepted) 	decision <ul style="list-style-type: none"> ▪ ERROR ▪ REJECT ▪ ACCEPT
ics_rflag and <service>_rflag <ul style="list-style-type: none"> ▪ SOK ▪ DINVALIDDATA ▪ ESYSTEM ▪ and so on... <p>This field returns a one-word description that gives a general reason for a decline or error. The implementation guide for the ICS service that you are implementing contains the list and description of the flags that can be returned for that service.</p>	reasonCode and <service>_reasonCode <ul style="list-style-type: none"> ▪ 100 ▪ 101 ▪ and so on... <p>This field returns a 3-digit number that gives the reason for a REJECT or ERROR decision. The implementation guide for the ICS service that you are implementing contains the list and description of the reason codes that can be returned for that service.</p>
ics_rmsg and <service>_rmsg <p>This field gives more information about why you received the particular rflag.</p>	<p>No corresponding field returned in the Simple Order API, although you can see the information in the transaction details screen on the Business Center.</p>

Request-Level Fields

Table 2 Request-Level Fields

SCMP API	SO API
account_encoder_id	check_accountEncoderID (for electronic checks) card_accountEncoderID (for credit cards)
airline_agent_code	airlineData_agentCode
airline_agent_name	airlineData_agentName
airline_booking_reference	airlineData_bookingReference
airline_carrier_name	airlineData_carrierName

Table 2 Request-Level Fields (Continued)

SCMP API	SO API
airline_charge_details	airlineData_chargeDetails
airline_check_digit	airlineData_checkDigit
airline_customer_code	airlineData_customerCode
airline_document_number	airlineData_documentNumber
airline_document_number_of_parts	airlineData_documentNumberOfParts
airline_document_type	airlineData_documentType
airline_extended_payment_code	airlineData_extendedPaymentCode
airline_invoice_number	airlineData_invoiceNumber
airline_leg#_carrier_code	airlineData_leg_#_carrierCode
airline_leg#_class	airlineData_leg_#_class
airline_leg#_depart_tax	airlineData_leg_#_departTax
airline_leg#_destination	airlineData_leg_#_destination
airline_leg#_fare_basis	airlineData_leg_#_fareBasis
airline_leg#_flight_number	airlineData_leg_#_flightNumber
airline_leg#_leg_departure_date	airlineData_leg_#_departureDate
airline_leg#_originating_airport_code	airlineData_leg_#_originatingAirportCode
airline_leg#_stopover_code	airlineData_leg_#_stopoverCode
airline_passenger_name	airlineData_passengerName
airline_restricted_ticket_indicator	airlineData_restrictedTicketIndicator
airline_ticket_issuer_city	airlineData_ticketIssuerCity
airline_ticket_issuer_code	airlineData_ticketIssuerCode
airline_ticket_issuer_country	airlineData_ticketIssuerCountry
airline_ticket_issuer_name	airlineData_ticketIssuerName
airline_ticket_issuer_postal_code	airlineData_ticketIssuerPostalCode
airline_ticket_issuer_state	airlineData_ticketIssuerState
airline_ticket_number	airlineData_ticketNumber
airline_transaction_type	airlineData_transactionType
alternate_tax_amount	otherTax_alternateTaxAmount
alternate_tax_amount_indicator	otherTax_alternateTaxIndicator
alternate_tax_id	otherTax_alternateTaxID
amexdata_taa1	invoiceHeader_amexDataTAA1
amexdata_taa2	invoiceHeader_amexDataTAA2
amexdata_taa3	invoiceHeader_amexDataTAA3
amexdata_taa4	invoiceHeader_amexDataTAA4

Table 2 Request-Level Fields (Continued)

SCMP API	SO API
auth_code	ccAuthService_verbalAuthCode (for credit card authorization) ccCaptureService_verbalAuthCode (for credit card capture)
auth_request_id	ccCaptureService_authRequestID (for credit card capture) ccAuthReversalService_authRequestID (for credit card full authorization reversal)
auth_type	ccAuthService_authType (for credit card authorization) ccCaptureService_authType (for credit card capture)
authorization_id	directDebitService_authorizationID
avs	afsService_avsCode
avs_level	ccAuthService_avsLevel
bank_account_name	fundTransfer_accountName
bank_account_number	fundTransfer_accountNumber
bank_address	bankInfo_address
bank_city	bankInfo_city
bank_code	bankInfo_bankCode
bank_country	bankInfo_country
bank_name	bankInfo_name
bank_sortcode	bankInfo_sortCode
bank_swiftcode	bankInfo_swiftCode
bank_transfer_request_id	bankTransferRefundService_bankTransferRequestID
bank_transfer_trans_ref_no	bankTransferRefundService_reconciliationID
batch_id	batch_batchID
batch_record_id	batch_recordID
bill_address1	billTo_street1
bill_address2	billTo_street2
bill_address3	billTo_street3
bill_address4	billTo_street4
bill_city	billTo_city
bill_company_tax_id	billTo_companyTaxID
bill_country	billTo_country
bill_county	billTo_county

Table 2 Request-Level Fields (Continued)

SCMP API	SO API
bill_payment	ccAuthService_billPayment (for credit card authorization) ccCreditService_billPayment (for credit card credit)
bill_request_id	ccCreditService_captureRequestID
bill_state	billTo_state
bill_zip	billTo_postalCode
bml_customer_billing_address_change	bml_customerBillingAddressChange
bml_customer_email_change	bml_customerEmailChange
bml_customer_has_checking_account	bml_customerHasCheckingAccount
bml_customer_has_savings_account	bml_customerHasSavingsAccount
bml_customer_password_change	bml_customerPasswordChange
bml_customer_phone_change	bml_customerPhoneChange
bml_customer_registration_date	bml_customerRegistrationDate
bml_customer_type_flag	bml_customerTypeFlag
bml_gross_household_income	bml_grossHouseholdIncome
bml_household_income_currency	bml_householdIncomeCurrency
bml_item_category	bml_itemCategory
bml_merchant_promotion_code	bml_merchantPromotionCode
bml_preapprovalNumber	bml_preapprovalNumber
bml_product_delivery_type_indicator	bml_productDeliveryTypeIndicator
bml_residence_status	bml_residenceStatus
bml_tc_version	bml_tcVersion
bml_years_at_current_residence	bml_yearsAtCurrentResidence
bml_years_with_current_employer	bml_yearsWithCurrentEmployer
branch_check_digit	fundTransfer_bankCheckDigit
branch_code	bankInfo_branchCode
button_type	paypalButtonCreateService_buttonType
buyer_registration	taxService_buyerRegistration
card_present	pos_cardPresent
card_type	card_cardType
cat_level	pos_catLevel
cavv	ccAuthService_cavv
cc_bin	card_bin

Table 2 Request-Level Fields (Continued)

SCMP API	SO API
comments	comments
company_name	billTo_company
cost_center	invoiceHeader_costCenter
currency	purchaseTotals_currency
customer_account_id	billTo_customerID
customer_cc_cv_indicator	card_cvIndicator
customer_cc_cv_number	card_cvNumber
customer_cc_expmo	card_expirationMonth
customer_cc_expyr	card_expirationYear
customer_cc_issue_number	card_issueNumber
customer_cc_number	card_accountNumber
customer_cc_startmo	card_startMonth
customer_cc_startyr	card_startYear
customer_cookies_accepted	billTo_httpBrowserCookiesAccepted
customer_email	billTo_email
customer_firstname	billTo_firstName
customer_gift_wrap	invoiceHeader_isGift
customer_hostname	billTo_hostname
customer_ipaddress	billTo_ipAddress
customer_lastname	billTo_lastName
customer_phone	billTo_phoneNumber
customer_phone_type	bml_billToPhoneType
customer_pin	card_pin
customer_ssn	billTo_ssn
cv_result	afsService_cvCode
date_collect	directDebitService_dateCollect
date_of_birth	billTo_dateOfBirth
decision_manager_enabled	decisionManager_enabled
decision_manager_profile	decisionManager_profile
decline_avs_flags	businessRules_declineAVSFlags
direct_debit_request_id	directDebitRefundService_directDebitRequestID
direct_debit_text	directDebitService_directDebitText
direct_debit_trans_ref_no	directDebitRefundService_reconciliationID
direct_debit_type	directDebitService_directDebitType
disable_avs	afsService_disableAVSScoring

Table 2 Request-Level Fields (Continued)

SCMP API	SO API
domain	billTo_domainName
driver_license_no	billTo_driversLicenseNumber
driver_license_state	billTo_driversLicenseState
duty_amount	purchaseTotals_dutyAmount
e_commerce_indicator	ccAuthService_commerceIndicator (for credit card authorization) ccCreditService_commerceIndicator (for credit card credit) pinlessDebitService_commerceIndicator (for PIN-less debit cards)
eci_raw	ccAuthService_eciRaw
eCP_account_no	check_accountNumber
eCP_account_type	check_accountType
eCP_check_no	check_checkNumber
eCP_debit_request_id	ecCreditService_debitRequestID
eCP_payment_key	ecDebitService_transactionToken (for electronic check debit) ecCreditService_transactionToken (for electronic check credit)
eCP_payment_mode	ecDebitService_paymentMode
eCP_rdfi	check_bankTransitNumber
eCP_ref_no	ecDebitService_referenceNumber (for electronic check debit) ecCreditService_referenceNumber (for electronic check credit)
eCP_settlement_method	ecDebitService_settlementMethod (for electronic check debit) ecCreditService_settlementMethod (for electronic check credit)
eCP_verification_level	ecDebitService_verificationLevel
employer_address1	bml_employerStreet1
employer_address2	bml_employerStreet2
employer_city	bml_employerCity
employer_company_name	bml_employerCompanyName
employer_country	bml_employerCountry
employer_phone	bml_employerPhoneNumber
employer_phone_type	bml_employerPhoneType
employer_state	bml_employerState

Table 2 Request-Level Fields (Continued)

SCMP API	SO API
employer_zip	bml_employerPostalCode
export_address_operator	exportService_addressOperator
export_address_weight	exportService_addressWeight
export_company_weight	exportService_companyWeight
export_name_weight	exportService_nameWeight
freight_amount	purchaseTotals_freightAmount
fxrates_funding_currency	fundingTotals_currency
fxrates_quote_id	ccAuthService_fxQuoteID
gecc_line1...7	gecc_line_0...6
gecc_plan_number	gecc_planNumber
gecc_promotion_end_date	gecc_promotionEndDate
gecc_promotion_plan	gecc_promotionPlan
gecc_sale_type	gecc_saleType
gecc_sequence_number	gecc_sequenceNumber
grand_total_amount	purchaseTotals_grandTotalAmount
http_browser_email	billTo_httpBrowserEmail
http_browser_type	billTo_httpBrowserType
ics_applications	<serviceName>_run
ignore_avs	businessRules_ignoreAVSResult
ignore_bad_cv	businessRules_ignoreCVResult
ignore_dav_result	businessRules_ignoreDAVResult
ignore_export_result	businessRules_ignoreExportResult
ignore_validate_result	businessRules_ignoreValidateResult
industry_datatype	ccCaptureService_industryDatatype (for credit card capture) ccCreditService_industryDatatype (for credit card credit)
invoice_date	invoiceHeader_invoiceDate
local_tax	otherTax_localTaxAmount
local_tax_indicator	otherTax_localTaxIndicator
merchant_defined_data1	merchantDefinedData_field1
merchant_defined_data2	merchantDefinedData_field2
merchant_defined_data3	merchantDefinedData_field3
merchant_defined_data4	merchantDefinedData_field4
merchant_descriptor	invoiceHeader_merchantDescriptor
merchant_descriptor_contact	invoiceHeader_merchantDescriptorContact

Table 2 Request-Level Fields (Continued)

SCMP API	SO API
merchant_id	merchantID
merchant_ref_number	merchantReferenceCode
merchant_vat_registration_number	invoiceHeader_ merchantVATRegistrationNumber
middleman_registration	taxService_middlemanRegistration
national_tax	otherTax_nationalTaxAmount
national_tax_indicator	otherTax_nationalTaxIndicator
nexus	taxService_nexus
no_nexus	taxService_noNexus
order_acceptance_city	taxService_orderAcceptanceCity
order_acceptance_country	taxService_orderAcceptanceCountry
order_acceptance_county	taxService_orderAcceptanceCounty
order_acceptance_state	taxService_orderAcceptanceState
order_acceptance_zip	taxService_orderAcceptancePostalCode
order_discount_amount	purchaseTotals_discountAmount
order_origin_city	taxService_orderOriginCity
order_origin_country	taxService_orderOriginCountry
order_origin_county	taxService_orderOriginCounty
order_origin_state	taxService_orderOriginState
order_origin_zip	taxService_orderOriginPostalCode
pa_http_accept	payerAuthEnrollService_httpAccept
pa_http_user_agent	payerAuthEnrollService_httpUserAgent
pa_merchant_name	payerAuthEnrollService_merchantName
pa_merchant_url	payerAuthEnrollService_merchantURL
pa_purchase_description	payerAuthEnrollService_purchaseDescription
pa_purchase_time	payerAuthEnrollService_purchaseTime
pa_signedpares	payerAuthValidateService_signedPAREs
partial_payment_id	ccCaptureService_partialPaymentID (for credit card capture) ccCreditService_partialPaymentID (for credit card credit) ecDebitService_partialPaymentID (for electronic check debit) ecCreditService_partialPaymentID (for electronic check credit)
payment_method	subscription_paymentMethod

Table 2 Request-Level Fields (Continued)

SCMP API	SO API
payment_request_id	paySubscriptionCreateService_paymentRequestID
payment_type	invoiceHeader_tenderType
paypal_mp_id	paypalPreapprovedPaymentService_mpID (for PayPal preapproved payment) paypalPreapprovedUpdateService_mpID (for PayPal preapproved payment update)
paypal_payment_request_id	paypalCreditService_payPalPaymentRequestID
pos_condition_code	pos_conditionCode
pos_entry_mode	pos_entryMode
pos_transaction_security	pos_transactionSecurity
purchaser_code	invoiceHeader_purchaserCode
purchaser_order_date	invoiceHeader_purchaserOrderDate
purchaser_vat_registration_number	invoiceHeader_purchaserVATRegistrationNumber
purchasing_level	ccCaptureService_purchasingLevel (for credit card capture) ccCreditService_purchasingLevel (for credit card credit)
record_id	riskUpdateService_recordID
record_name	riskUpdateService_recordName
recurring_approval_required	recurringSubscriptionInfo_approvalRequired
recurring_automatic_renew	recurringSubscriptionInfo_automaticRenew
recurring_frequency	recurringSubscriptionInfo_frequency
recurring_number_of_payments	recurringSubscriptionInfo_numberOfPayments
recurring_number_of_payments_to_add	recurringSubscriptionInfo_numberOfPaymentsToAdd
recurring_payment_amount	recurringSubscriptionInfo_amount
recurring_payment_event_action	paySubscriptionEventUpdateService_action
recurring_payment_event_amount	recurringSubscriptionInfo_event_amount
recurring_payment_event_approved_by	recurringSubscriptionInfo_event_approvedBy
recurring_payment_event_number	recurringSubscriptionInfo_event_number
recurring_start_date	recurringSubscriptionInfo_startDate
returns_accepted	invoiceHeader_returnsAccepted
seller_registration	taxService_sellerRegistration
ship_from_city	shipFrom_city

Table 2 Request-Level Fields (Continued)

SCMP API	SO API
ship_from_country	shipFrom_country
ship_from_county	shipFrom_county
ship_from_state	shipFrom_state
ship_from_zip	shipFrom_postalCode
ship_to_address1	shipTo_street1
ship_to_address2	shipTo_street2
ship_to_address3	shipTo_street3
ship_to_address4	shipTo_street4
ship_to_city	shipTo_city
ship_to_country	shipTo_country
ship_to_county	shipTo_county
ship_to_email	shipTo_email
ship_to_firstname	shipTo_firstName
ship_to_lastname	shipTo_lastName
ship_to_phone	shipTo_phoneNumber
ship_to_phone_type	bml_shipToPhoneType
ship_to_state	shipTo_state
ship_to_zip	shipTo_postalCode
shipping_method	shipTo_shippingMethod
subscription_end_date	recurringSubscriptionInfo_endDate
subscription_id	recurringSubscriptionInfo_subscriptionID
subscription_status	recurringSubscriptionInfo_status
subscription_title	subscription_title
summary_commodity_code	invoiceHeader_summaryCommodityCode
supplier_order_reference	invoiceHeader_supplierOrderReference
tax_indicator	invoiceHeader_taxable
terminal_capability	pos_terminalCapability
terminal_id	pos_terminalID
terminal_location	pos_terminalLocation
terminal_type	pos_terminalType
total_funding_amount	fundingTotals_grandTotalAmount
total_tax_amount	purchaseTotals_taxAmount
track_data	pos_trackData
transaction_type	directDebitService_transactionType
ucaf_authentication_data	ucaf_authenticationData

Table 2 Request-Level Fields (Continued)

SCMP API	SO API
ucaf_collection_indicator	ucaf_collectionIndicator
user_po	invoiceHeader_userPO
vat_invoice_ref_number	invoiceHeader_vatInvoiceReferenceNumber
vat_tax_amount	otherTax_vatTaxAmount
vat_tax_rate	otherTax_vatTaxRate
void_request_id	voidService_voidRequestID
xid	ccAuthService_xid

Offer-Level Fields

Table 3 Offer-Level Fields

SCMP API	SO API
alternate_tax_id	item_#_alternateTaxID
amount	item_#_unitPrice
buyer_registration	item_#_buyerRegistration
city_override_amount	item_#_cityOverrideAmount
city_override_rate	item_#_cityOverrideRate
commodity_code	item_#_commodityCode
country_override_amount	item_#_countryOverrideAmount
country_override_rate	item_#_countryOverrideRate
county_override_amount	item_#_countyOverrideAmount
county_override_rate	item_#_countyOverrideRate
discount_amount	item_#_discountAmount
discount_indicator	item_#_discountIndicator
discount_rate	item_#_discountRate
district_override_amount	item_#_districtOverrideAmount
district_override_rate	item_#_districtOverrideRate
export	item_#_export
gross_net_indicator	item_#_grossNetIndicator
merchant_product_sku	item_#_productSKU
middleman_registration	item_#_middlemanRegistration
national_tax	item_#_nationalTax

Table 3 Offer-Level Fields (Continued)

SCMP API	SO API
no_export	item_#_noExport
order_acceptance_city	item_#_orderAcceptanceCity
order_acceptance_country	item_#_orderAcceptanceCountry
order_acceptance_county	item_#_orderAcceptanceCounty
order_acceptance_state	item_#_orderAcceptanceState
order_acceptance_zip	item_#_orderAcceptancePostalCode
order_origin_city	item_#_orderOriginCity
order_origin_country	item_#_orderOriginCountry
order_origin_county	item_#_orderOriginCounty
order_origin_state	item_#_orderOriginState
order_origin_zip	item_#_orderOriginPostalCode
point_of_title_transfer	item_#_pointOfTitleTransfer
product_code	item_#_productCode
product_name	item_#_productName
product_risk	item_#_productRisk
quantity	item_#_quantity
score_category_time	item_#_timeCategory
score_category_gift	item_#_giftCategory
score_host_hedge	item_#_hostHedge
score_threshold	businessRules_scoreThreshold (score_threshold is an offer-level field in the SCMP API; businessRules_scoreThreshold is a request field in the Simple Order API)
score_time_hedge	item_#_timeHedge
score_velocity_hedge	item_#_velocityHedge
seller_registration	item_#_sellerRegistration
ship_from_city	item_#_shipFromCity
ship_from_country	item_#_shipFromCountry
ship_from_county	item_#_shipFromCounty
ship_from_state	item_#_shipFromState
ship_from_zip	item_#_shipFromPostalCode
state_override_amount	item_#_stateOverrideAmount
state_override_rate	item_#_stateOverrideRate
tax_amount	item_#_taxAmount
tax_rate	item_#_taxRate
tax_type_applied	item_#_taxTypeApplied

Table 3 Offer-Level Fields (Continued)

SCMP API	SO API
total_amount	item_#_totalAmount
unit_of_measure	item_#_unitOfMeasure
vat_rate	item_#_vatRate

Reply Fields

Table 4 Reply Fields

SCMP API	SO API
auth_auth_amount	ccAuthReply_amount
auth_auth_avs	ccAuthReply_avsCode
auth_auth_code	ccAuthReply_authorizationCode
auth_auth_record	ccAuthReply_authRecord
auth_auth_response	ccAuthReply_processorResponse
auth_auth_time	ccAuthReply_authorizedDateTime
auth_avs_raw	ccAuthReply_avsCodeRaw
auth_customer_cc_number	ccAuthReply_bmlAccountNumber
auth_cv_result	ccAuthReply_cvCode
auth_cv_result_raw	ccAuthReply_cvCodeRaw
auth_factor_code	ccAuthReply_authFactorCode
auth_fxrates_funding_currency	ccAuthReply_fundingTotals_currency
auth_fxrates_quote_exp	ccAuthReply_fxQuoteExpirationDateTime
auth_fxrates_quote_id	ccAuthReply_fxQuoteID
auth_fxrates_quote_rate	ccAuthReply_fxQuoteRate
auth_fxrates_quote_type	ccAuthReply_fxQuoteType
auth_rcode	N/A
auth_reversal_amount	ccAuthReversalReply_amount
auth_reversal_auth_code	ccAuthReversalReply_authorizationCode
auth_reversal_auth_response	ccAuthReversalReply_processorResponse
auth_reversal_rcode	N/A
auth_reversal_request_time	ccAuthReversalReply_requestDateTime
auth_reversal_rflag	N/A
auth_reversal_rmsg	N/A

Table 4 Reply Fields (Continued)

SCMP API	SO API
auth_rflag	N/A
auth_rmsg	N/A
auth_total_funding_amount	ccAuthReply_fundingTotals_grandTotalAmount
auth_trans_ref_no	ccAuthReply_reconciliationID
bank_transfer_account_holder	bankTransferReply_accountHolder
bank_transfer_account_number	bankTransferReply_accountNumber
bank_transfer_amount	bankTransferReply_amount
bank_transfer_bank_city	bankTransferReply_bankCity
bank_transfer_bank_country	bankTransferReply_bankCountry
bank_transfer_bank_name	bankTransferReply_bankName
bank_transfer_payment_reference	bankTransferReply_paymentReference
bank_transfer_rcode	N/A
bank_transfer_refund_amount	bankTransferRefundReply_amount
bank_transfer_refund_rcode	N/A
bank_transfer_refund_response_code	bankTransferRefundReply_processorResponse
bank_transfer_refund_rflag	N/A
bank_transfer_refund_rmsg	N/A
bank_transfer_refund_time	bankTransferRefundReply_requestDateTime
bank_transfer_refund_trans_ref_no	bankTransferRefundReply_reconciliationID
bank_transfer_response_code	bankTransferReply_processorResponse
bank_transfer_rflag	N/A
bank_transfer_rmsg	N/A
bank_transfer_special_id	bankTransferReply_bankSpecialID
bank_transfer_swiftcode	bankTransferReply_bankSwiftCode
bank_transfer_time	bankTransferReply_requestDateTime
bank_transfer_trans_ref_no	bankTransferReply_reconciliationID
bill_bill_amount	ccCaptureReply_amount
bill_bill_request_time	ccCaptureReply_requestDateTime
bill_enhanced_data_enabled	ccCaptureReply_enhancedDataEnabled
bill_fxrates_funding_currency	ccCaptureReply_fundingTotals_currency
bill_fxrates_quote_exp	ccCaptureReply_fxQuoteExpirationDateTime
bill_fxrates_quote_id	ccCaptureReply_fxQuoteID
bill_fxrates_quote_rate	ccCaptureReply_fxQuoteRate
bill_fxrates_quote_type	ccCaptureReply_fxQuoteType

Table 4 Reply Fields (Continued)

SCMP API	SO API
bill_purchasing_level3_enabled	ccCaptureReply_purchasingLevel3Enabled
bill_rcode	N/A
bill_rflag	N/A
bill_rmsg	N/A
bill_total_funding_amount	ccCaptureReply_fundingTotals_grandTotalAmount
bill_trans_ref_no	ccCaptureReply_reconciliationID
credit_credit_amount	ccCreditReply_amount
credit_credit_request_time	ccCreditReply_requestDateTime
credit_enhanced_data_enabled	ccCreditReply_enhancedDataEnabled
credit_purchasing_level3_enabled	ccCreditReply_purchasingLevel3Enabled
credit_rcode	N/A
credit_rflag	N/A
credit_rmsg	N/A
credit_trans_ref_no	ccCreditReply_reconciliationID
currency	purchaseTotals_currency
dav_address_type	davReply_addressType
dav_apt_info	davReply_apartmentInfo
dav_bar_code	davReply_barCode
dav_bar_code_chkdigit	davReply_barCodeCheckDigit
dav_ca_error_info	davReply_caErrorInfo
dav_ca_info	davReply_caInfo
dav_careof	davReply_careOf
dav_city_info	davReply_cityInfo
dav_country_info	davReply_countyInfo
dav_directional_info	davReply_directionalInfo
dav_intl_error_info	davReply_intlErrorInfo
dav_intl_info	davReply_intlInfo
dav_lvr_info	davReply_lvrInfo
dav_match_score	davReply_matchScore
dav_overall_info	davReply_overallInfo
dav_rcode	N/A
dav_rflag	N/A
dav_rmsg	N/A
dav_standard_address	davReply_standardizedAddress1

Table 4 Reply Fields (Continued)

SCMP API	SO API
dav_standard_address_noapt	davReply_standardizedAddressNoApt
dav_standard_address2	davReply_standardizedAddress2
dav_standard_address3	davReply_standardizedAddress3
dav_standard_address4	davReply_standardizedAddress4
dav_standard_city	davReply_standardizedCity
dav_standard_country	davReply_standardizedCountry
dav_standard_county	davReply_standardizedCounty
dav_standard_csz	davReply_standardizedCSP
dav_standard_iso_country	davReply_standardizedISOCountry
dav_standard_state	davReply_standardizedState
dav_standard_zip	davReply_standardizedPostalCode
dav_state_info	davReply_stateInfo
dav_street_info	davReply_streetInfo
dav_suffix_info	davReply_suffixInfo
dav_us_error_info	davReply_usErrorInfo
dav_us_info	davReply_usInfo
dav_zip_info	davReply_postalCodeInfo
decision_active_profile	decisionReply_activeProfileReply_name
decision_active_profile_destination_queue	decisionReply_activeProfileReply_destinationQueue
decision_active_profile_rule_#_decision	decisionReply_activeProfileReply_rulesTriggered_ruleResultItem_#_decision
decision_active_profile_rule_#_evaluation	decisionReply_activeProfileReply_rulesTriggered_ruleResultItem_#_evaluation
decision_active_profile_rule_#_id	decisionReply_activeProfileReply_rulesTriggered_ruleResultItem_#_ruleID
decision_active_profile_rule_#_name	decisionReply_activeProfileReply_rulesTriggered_ruleResultItem_#_name
decision_active_profile_selector_rule	decisionReply_activeProfileReply_selectedBy
direct_debit_refund_response_code	directDebitRefundReply_processorResponse
direct_debit_amount	directDebitReply_amount
direct_debit_rcode	N/A
direct_debit_rcode	directDebitReply_reasonCode
direct_debit_refund_amount	directDebitRefundReply_amount

Table 4 Reply Fields (Continued)

SCMP API	SO API
direct_debit_refund_rcode	N/A
direct_debit_refund_rflag	N/A
direct_debit_refund_rmsg	N/A
direct_debit_refund_time	directDebitRefundReply_requestDateTime
direct_debit_refund_trans_ref_no	directDebitRefundReply_reconciliationID
direct_debit_response_code	directDebitReply_processorResponse
direct_debit_rflag	N/A
direct_debit_rmsg	N/A
direct_debit_time	directDebitReply_requestDateTime
direct_debit_trans_ref_no	directDebitReply_reconciliationID
direct_debit_validate_rcode	N/A
direct_debit_validate_rflag	N/A
direct_debit_validate_rmsg	N/A
ecp_credit_processor_trans_id	ecCreditReply_processorTransactionID
ecp_credit_rcode	N/A
ecp_credit_ref_no	ecCreditReply_reconciliationID
ecp_credit_result_code	ecCreditReply_processorResponse
ecp_credit_rflag	N/A
ecp_credit_rmsg	N/A
ecp_credit_settlement_method	ecCreditReply_settlementMethod
ecp_credit_submit_time	ecCreditReply_requestDateTime
ecp_credit_total_amount	ecCreditReply_amount
ecp_debit_avs	ecDebitReply_avsCode
ecp_debit_avs_raw	ecDebitReply_avsCodeRaw
ecp_debit_processor_trans_id	ecDebitReply_processorTransactionID
ecp_debit_rcode	N/A
ecp_debit_ref_no	ecDebitReply_reconciliationID
ecp_debit_result_code	ecDebitReply_processorResponse
ecp_debit_rflag	N/A
ecp_debit_rmsg	N/A
ecp_debit_settlement_method	ecDebitReply_settlementMethod
ecp_debit_submit_time	ecDebitReply_requestDateTime
ecp_debit_total_amount	ecDebitReply_amount
ecp_debit_verification_level	ecDebitReply_verificationLevel
export_ip_country_confidence	exportReply_ipCountryConfidence

Table 4 Reply Fields (Continued)

SCMP API	SO API
export_match1_address1	deniedPartiesMatch_0_address_0
export_match1_address2	deniedPartiesMatch_0_address_1
export_match1_address3	deniedPartiesMatch_0_address_2
export_match1_list	deniedPartiesMatch_0_list
export_match1_name1	deniedPartiesMatch_0_name_0
export_match1_name2	deniedPartiesMatch_0_name_1
export_match2_address1	deniedPartiesMatch_1_address_0
export_match2_address2	deniedPartiesMatch_1_address_1
export_match2_address3	deniedPartiesMatch_1_address_2
export_match2_list	deniedPartiesMatch_1_list
export_match2_name1	deniedPartiesMatch_1_name_0
export_match2_name2	deniedPartiesMatch_1_name_1
export_match3_address1	deniedPartiesMatch_2_address_0
export_match3_address2	deniedPartiesMatch_2_address_1
export_match3_address3	deniedPartiesMatch_2_address_2
export_match3_list	deniedPartiesMatch_2_list
export_match3_name1	deniedPartiesMatch_2_name_0
export_match3_name2	deniedPartiesMatch_2_name_1
export_rcode	N/A
export_rflag	N/A
export_rmsg	N/A
fxrates_currency#	fxRatesReply_quote_#_currency
fxrates_funding_currency#	fxRatesReply_quote_#_fundingCurrency
fxrates_quote_date#	fxRatesReply_quote_#_receivedDateTime
fxrates_quote_exp#	fxRatesReply_quote_#_expirationDateTime
fxrates_quote_id#	fxRatesReply_quote_#_id
fxrates_quote_rate#	fxRatesReply_quote_#_rate
fxrates_quote_type#	fxRatesReply_quote_#_type
fxrates_rcode	N/A
fxrates_rflag	N/A
fxrates_rmsg	N/A
ics_rcode	N/A
ics_rflag	N/A
ics_rmsg	N/A
merchant_ref_number	merchantReferenceCode

Table 4 Reply Fields (Continued)

SCMP API	SO API
pa_enroll_acs_url	payerAuthEnrollReply_acsURL
pa_enroll_e_commerce_indicator	payerAuthEnrollReply_commerceIndicator
pa_enroll_pareq	payerAuthEnrollReply_paReq
pa_enroll_proofxml	payerAuthEnrollReply_proofXML
pa_enroll_proxypan	payerAuthEnrollReply_proxyPAN
pa_enroll_rcode	N/A
pa_enroll_rflag	N/A
pa_enroll_rmsg	N/A
pa_enroll_ucaf_collection_indicator	payerAuthEnrollReply_ucafCollectionIndicator
pa_enroll_xid	payerAuthEnrollReply_xid
pa_validate_authentication_result	payerAuthValidateReply_authenticationResult
pa_validate_authentication_status_msg	payerAuthValidateReply_authenticationStatusMessage
pa_validate_cavv	payerAuthValidateReply_cavv
pa_validate_e_commerce_indicator	payerAuthValidateReply_commerceIndicator
pa_validate_eci	payerAuthValidateReply_eci
pa_validate_eci_raw	payerAuthValidateReply_eciRaw
pa_validate_rcode	N/A
pa_validate_rflag	N/A
pa_validate_rmsg	N/A
pa_validate_xid	payerAuthValidateReply_xid
pay_subscription_create_rcode	N/A
pay_subscription_create_rflag	N/A
pay_subscription_create_rmsg	N/A
pay_subscription_create_subscription_id	paySubscriptionCreateReply_subscriptionID
pay_subscription_retrieve_approval_required	paySubscriptionRetrieveReply_approvalRequired
pay_subscription_retrieve_automatic_renew	paySubscriptionRetrieveReply_automaticRenew
pay_subscription_retrieve_bill_address1	paySubscriptionRetrieveReply_street1
pay_subscription_retrieve_bill_address2	paySubscriptionRetrieveReply_street2
pay_subscription_retrieve_bill_city	paySubscriptionRetrieveReply_city
pay_subscription_retrieve_bill_country	paySubscriptionRetrieveReply_country

Table 4 Reply Fields (Continued)

SCMP API	SO API
pay_subscription_retrieve_bill_state	paySubscriptionRetrieveReply_state
pay_subscription_retrieve_bill_zip	paySubscriptionRetrieveReply_postalCode
pay_subscription_retrieve_card_type	paySubscriptionRetrieveReply_cardType
pay_subscription_retrieve_comments	paySubscriptionRetrieveReply_comments
pay_subscription_retrieve_company_name	paySubscriptionRetrieveReply_companyName
pay_subscription_retrieve_currency	paySubscriptionRetrieveReply_currency
pay_subscription_retrieve_customer_account_id	paySubscriptionRetrieveReply_customerAccountID
pay_subscription_retrieve_customer_cc_expmo	paySubscriptionRetrieveReply_cardExpirationMonth
pay_subscription_retrieve_customer_cc_expyr	paySubscriptionRetrieveReply_cardExpirationYear
pay_subscription_retrieve_customer_cc_issue_number	paySubscriptionRetrieveReply_cardIssueNumber
pay_subscription_retrieve_customer_cc_number	paySubscriptionRetrieveReply_cardAccountnumber
pay_subscription_retrieve_customer_cc_startmo	paySubscriptionRetrieveReply_cardStartMonth
pay_subscription_retrieve_customer_cc_startyr	paySubscriptionRetrieveReply_cardStartYear
pay_subscription_retrieve_customer_email	paySubscriptionRetrieveReply_email
pay_subscription_retrieve_customer_firstname	paySubscriptionRetrieveReply_firstname
pay_subscription_retrieve_customer_lastname	paySubscriptionRetrieveReply_lastname
pay_subscription_retrieve_customer_phone	paySubscriptionRetrieveReply_phoneNumber
pay_subscription_retrieve_ecp_account_no	paySubscriptionRetrieveReply_checkAccountNumber
pay_subscription_retrieve_ecp_account_no	paySubscriptionRetrieveReply_checkAccountType
pay_subscription_retrieve_ecp_rdfi	paySubscriptionRetrieveReply_checkBankTransitNumber
pay_subscription_retrieve_end_date	paySubscriptionRetrieveReply_endDate
pay_subscription_retrieve_frequency	paySubscriptionRetrieveReply_frequency
pay_subscription_retrieve_merchant_ref_number	paySubscriptionRetrieveReply_merchantReferenceCode

Table 4 Reply Fields (Continued)

SCMP API	SO API
pay_subscription_retrieve_payment_method	paySubscriptionRetrieveReply_paymentMethod
pay_subscription_retrieve_payments_left	paySubscriptionRetrieveReply_paymentsRemaining
pay_subscription_retrieve_rcode	N/A
pay_subscription_retrieve_recurring_amount	paySubscriptionRetrieveReply_recurringAmount
pay_subscription_retrieve_rflag	N/A
pay_subscription_retrieve_rmsg	N/A
pay_subscription_retrieve_setup_amount	paySubscriptionRetrieveReply_setupAmount
pay_subscription_retrieve_start_date	paySubscriptionRetrieveReply_startDate
pay_subscription_retrieve_status	paySubscriptionRetrieveReply_status
pay_subscription_retrieve_subscription_id	paySubscriptionRetrieveReply_subscriptionID
pay_subscription_retrieve_title	paySubscriptionRetrieveReply_title
pay_subscription_retrieve_total_payments	paySubscriptionRetrieveReply_totalPayments
pay_subscription_update_rcode	paySubscriptionUpdateReply_reasonCode
pay_subscription_update_subscription_id	paySubscriptionUpdateReply_subscriptionID
paypal_button_create_button_type	paypalButtonCreateReply_buttonType
paypal_button_create_encrypted_form_data	paypalButtonCreateReply_encryptedFormData
paypal_button_create_rcode	N/A
paypal_button_create_rflag	N/A
paypal_button_create_rmsg	N/A
paypal_button_create_time	paypalButtonCreateReply_requestDateTime
paypal_button_create_trans_ref_no	paypalButtonCreateReply_reconciliationID
paypal_button_create_unencrypted_form_data	paypalButtonCreateReply_unencryptedFormData
paypal_credit_amount	paypalCreditReply_amount
paypal_credit_rcode	N/A
paypal_credit_response_code	paypalCreditReply_processorResponse
paypal_credit_rflag	N/A
paypal_credit_rmsg	N/A
paypal_credit_time	paypalCreditReply_requestDateTime
paypal_credit_trans_ref_no	paypalCreditReply_reconciliationID

Table 4 Reply Fields (Continued)

SCMP API	SO API
paypal_preapproved_payment_desc	payPalPreapprovedPaymentReply_desc
paypal_preapproved_payment_exchange_rate	payPalPreapprovedPaymentReply_exchangeRate
paypal_preapproved_payment_fee_amount	payPalPreapprovedPaymentReply_feeAmount
paypal_preapproved_payment_mp_max	payPalPreapprovedPaymentReply_mpMax
paypal_preapproved_payment_mp_status	payPalPreapprovedPaymentReply_mpStatus
paypal_preapproved_payment_payer	payPalPreapprovedPaymentReply_payer
paypal_preapproved_payment_payer_business	payPalPreapprovedPaymentReply_payerBusiness
paypal_preapproved_payment_payer_country	payPalPreapprovedPaymentReply_payerCountry
paypal_preapproved_payment_payer_id	payPalPreapprovedPaymentReply_payerID
paypal_preapproved_payment_payer_name	payPalPreapprovedPaymentReply_payerName
paypal_preapproved_payment_payer_status	payPalPreapprovedPaymentReply_payerStatus
paypal_preapproved_payment_payment_date	payPalPreapprovedPaymentReply_paymentDate
paypal_preapproved_payment_payment_gross_amount	payPalPreapprovedPaymentReply_paymentGrossAmount
paypal_preapproved_payment_payment_source_id	payPalPreapprovedPaymentReply_paymentSourceID
paypal_preapproved_payment_payment_status	payPalPreapprovedPaymentReply_paymentStatus
paypal_preapproved_payment_payment_type	payPalPreapprovedPaymentReply_paymentType
paypal_preapproved_payment_pending_reason	payPalPreapprovedPaymentReply_pendingReason
paypal_preapproved_payment_rcode	N/A
paypal_preapproved_payment_rflag	N/A
paypal_preapproved_payment_rmsg	N/A
paypal_preapproved_payment_settle_amount	payPalPreapprovedPaymentReply_settleAmount
paypal_preapproved_payment_tax_amount	payPalPreapprovedPaymentReply_taxAmount

Table 4 Reply Fields (Continued)

SCMP API	SO API
paypal_preapproved_payment_time	payPalPreapprovedPaymentReply_requestDateTime
paypal_preapproved_payment_trans_ref_no	payPalPreapprovedPaymentReply_reconciliationID
paypal_preapproved_payment_transaction_id	payPalPreapprovedPaymentReply_transactionID
paypal_preapproved_payment_transaction_type	payPalPreapprovedPaymentReply_transactionType
paypal_preapproved_update_desc	payPalPreapprovedUpdateReply_desc
paypal_preapproved_update_mp_max	payPalPreapprovedUpdateReply_mpMax
paypal_preapproved_update_mp_status	payPalPreapprovedUpdateReply_mpStatus
paypal_preapproved_update_payer	payPalPreapprovedUpdateReply_payer
paypal_preapproved_update_payer_business	payPalPreapprovedUpdateReply_payerBusiness
paypal_preapproved_update_payer_country	payPalPreapprovedUpdateReply_payerCountry
paypal_preapproved_update_payer_id	payPalPreapprovedUpdateReply_payerID
paypal_preapproved_update_payer_name	payPalPreapprovedUpdateReply_payerName
paypal_preapproved_update_payer_status	payPalPreapprovedUpdateReply_payerStatus
paypal_preapproved_update_payment_source_id	payPalPreapprovedUpdateReply_paymentSourceID
paypal_preapproved_update_rcode	N/A
paypal_preapproved_update_rflag	N/A
paypal_preapproved_update_rmsg	N/A
paypal_preapproved_update_time	payPalPreapprovedUpdateReply_requestDateTime
paypal_preapproved_update_trans_ref_no	payPalPreapprovedUpdateReply_reconciliationID
pinless_debit_amount	pinlessDebitReply_amount
pinless_debit_auth_code	pinlessDebitReply_authorizationCode
pinless_debit_processor_response	pinlessDebitReply_processorResponse
pinless_debit_rcode	N/A
pinless_debit_receipt_number	pinlessDebitReply_receiptNumber
pinless_debit_rflag	N/A

Table 4 Reply Fields (Continued)

SCMP API	SO API
pinless_debit_rmsg	N/A
pinless_debit_time	pinlessDebitReply_requestDateTime
pinless_debit_trans_ref_no	pinlessDebitReply_reconciliationID
pinless_debit_validate_rcode	N/A
pinless_debit_validate_rflag	N/A
pinless_debit_validate_rmsg	N/A
pinless_debit_validate_status	pinlessDebitValidateReply_status
pinless_debit_validate_time	pinlessDebitValidateReply_requestDateTime
request_id	requestID
score_address_info	afsReply_addressInfoCode
score_factors	afsReply_afsFactorCode
score_host_severity	afsReply_hostSeverity
score_hotlist_info	afsReply_hotlistInfoCode
score_internet_info	afsReply_internetInfoCode
score_phone_info	afsReply_phoneInfoCode
score_rcode	N/A
score_rflag	N/A
score_rmsg	N/A
score_score_result	afsReply_afsResult
score_suspicious_info	afsReply_suspiciousInfoCode
score_time_local	afsReply_consumerLocalTime
score_velocity_info	afsReply_velocityInfoCode
tax_city_name	taxReply_city
tax_city_tax#	taxReply_item_#_cityTaxAmount
tax_county_name	taxReply_county
tax_county_tax#	taxReply_item_#_countyTaxAmount
tax_district_tax#	taxReply_item_#_districtTaxAmount
tax_rcode	N/A
tax_rflag	N/A
tax_rmsg	N/A
tax_state_name	taxReply_state
tax_state_tax#	taxReply_item_#_stateTaxAmount
tax_tax_amount#	taxReply_item_#_totalTaxAmount
tax_total_city_tax	taxReply_totalCityTaxAmount
tax_total_county_tax	taxReply_totalCountyTaxAmount

Table 4 Reply Fields (Continued)

SCMP API	SO API
tax_total_district_tax	taxReply_totalDistrictTaxAmount
tax_total_grand	taxReply_grandTotalTaxAmount
tax_total_state_tax	taxReply_totalStateTaxAmount
tax_total_tax	taxReply_totalTaxAmount
tax_zip	taxReply_postalCode
ucaf_authentication_data	payerAuthValidateReply_ucafAuthenticationData
ucaf_collection_indicator	payerAuthValidateReply_ucafCollectionIndicator
void_rcode	N/A
void_rflag	N/A
void_rmsg	N/A
void_void_amount	voidReply_amount
void_void_currency	voidReply_currency
void_void_request_time	voidReply_requestDateTime

Index

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

A

API field mapping [62](#)
API fields used [8](#)

B

Batch Files Daily Summary Report [45](#)

C

creating batch files
in the Business Center [9](#)
manually [13](#)

P

Payment Submission Detail Report [28](#)

R

reports
Batch Files Daily Summary Report [45](#)
Payment Submission Detail Report [28](#)
request results [29](#)
response files
for requests in batch files [29](#)
for status of batch files [20](#)

S

sample code [53](#)

T

testing [20](#)

U

uploading batch files
in the Business Center [9](#)
manually [13](#)
sample code for [53](#)

V

validation results [20](#)