

# Account Updater

## User Guide

January 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](mailto:sales@cybersource.com) or call 650-432-7350 or 888-330-2300 (toll free in the United States).

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

## Copyright

© 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 Unpublished rights reserved under the copyright laws of the United States.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](#) 6

## [About This Guide](#) 7

[Audience](#) 7

[Purpose](#) 7

[Conventions](#) 7

[Text and Command Conventions](#) 8

[Related Documents](#) 8

[Customer Support](#) 8

---

## **Chapter 1** [Introduction](#) 9

[Options](#) 9

[REST API Batch Updates](#) 9

[Harvest Updates](#) 9

[PAN Upload Updates](#) 9

[Enrollment](#) 10

[Business Center Permissions](#) 10

[Terms of Use](#) 11

---

## **Chapter 2** [REST API Batch Updates](#) 12

[Retrieve OAuth Token](#) 12

[Endpoints](#) 13

[POST Authentication Payload](#) 13

[URL Encoded POST Example](#) 13

[Response Example](#) 14

[POST Batch Payload](#) 14

[Endpoints](#) 14

[Batch Payload](#) 15

[Headers](#) 15

[Batch Payload](#) 15

[Batch Payload Example](#) 15

|                             |    |
|-----------------------------|----|
| Successful Response Data    | 16 |
| Successful Response Example | 17 |
| Unsuccessful Response Codes | 17 |
| Error Response Example      | 18 |
| Error Message Codes         | 19 |
| Field Validation Messages   | 20 |
| GET Batch Status            | 20 |
| Endpoint                    | 20 |
| Status Responses            | 20 |
| Response Example            | 21 |
| GET Batch Report            | 22 |
| Endpoint                    | 22 |
| Report Fields               | 22 |
| Report Example              | 24 |
| Testing                     | 25 |
| Profile Update Report       | 25 |

---

|                  |                               |           |
|------------------|-------------------------------|-----------|
| <b>Chapter 3</b> | <b>Harvest Updates</b>        | <b>26</b> |
|                  | Profile Update Report         | 27        |
|                  | Profile Update Report Records | 28        |
|                  | Header Records                | 28        |
|                  | Details Records               | 29        |
|                  | Footer Records                | 30        |

---

|                  |                               |           |
|------------------|-------------------------------|-----------|
| <b>Chapter 4</b> | <b>PAN Upload Updates</b>     | <b>31</b> |
|                  | Creating Security Keys        | 31        |
|                  | Transaction Security Key      | 31        |
|                  | PGP Public/Private Key Pair   | 32        |
|                  | Formatting a Request File     | 32        |
|                  | Header Record                 | 32        |
|                  | Detail Record                 | 33        |
|                  | Footer Record                 | 34        |
|                  | Request File Examples         | 34        |
|                  | Uploading a Request File      | 35        |
|                  | Email Notification            | 35        |
|                  | Viewing the Batch File Status | 36        |
|                  | Downloading a Response File   | 37        |
|                  | Response File Records         | 39        |
|                  | Header Record                 | 40        |
|                  | Detail Record                 | 40        |
|                  | Footer Record                 | 41        |

[File Examples](#) 42

---

**Chapter 5    [Response Codes and Reason Codes](#)    43**

[Record Level](#) 43

[Request File Level](#) 44

---

**Chapter 6    [Testing](#)    46**

[Visa Card Numbers](#) 46

[MasterCard Card Numbers](#) 47

---

**Chapter 7    [Sample Java Code for Uploading Files](#)    49**

[Requirements](#) 49

[Using the Sample Code](#) 49

# Recent Revisions to This Document

| Release        | Changes   |
|----------------|---|
| January 2016   | Removed the note from the CCH entry in <a href="#">Table 22, page 43</a> .  |
| October 2016   | Added the 852 reason code. See <a href="#">page 43</a> .  |
| September 2016 | Updated the <b>card_expiry_year</b> field. See <a href="#">page 23</a> .  |
| July 2016      | This revision contains only editorial changes and no technical updates.   |
| May 2016       | <p>No content updates. Clarification on the initial release of the document:</p> <ul style="list-style-type: none"> <li>■ The “REST API Batch Updates” chapter was added (see <a href="#">page 12</a>).</li> <li>■ The “Testing” chapter was added (see <a href="#">page 46</a>).</li> <li>■ Harvest and PAN Upload information separated into specific chapters: <ul style="list-style-type: none"> <li>● “Harvest Updates” (see <a href="#">page 26</a>).</li> <li>● “PAN Upload Updates” (see <a href="#">page 31</a>).</li> </ul> </li> <li>■ Response codes and “Reason Codes” chapter was added (see <a href="#">page 43</a>).</li> </ul> |
| March 2016     | Initial release. This guide replaces the <i>CyberSource Account Updater User Guide</i> .  |

# About This Guide

## Audience

---

This guide is written for merchants who want to decrease the cost of manually updating customer account information. Account Updater enables merchants to manually or automatically incorporate changes made to a customer's payment card data: expiration date, new card number, account closures, and brand migrations between Visa and MasterCard.

## Purpose

---

This guide describes tasks that a merchant must complete in order to submit a batch of tokens using the REST API (see "[REST API Batch Updates](#)," page 12), automatically update customer subscriptions or profiles with the latest credit card data (see "[Harvest Updates](#)," page 26), or upload request files with new customer PANs (see "[PAN Upload Updates](#)," page 31). It is intended to help the merchant reduce the number of authorization declines to retain revenue and reduce the cost of manually updating payment data.

## Conventions

---



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

---



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

---

## Text and Command Conventions

| Convention             | Usage  |
|------------------------|--|
| <b>bold</b>            | Items that you are instructed to act upon; for example:<br>Click <b>Certificates</b> . |
| <code>monospace</code> | Code examples and samples.   |

## Related Documents

- *Getting Started with CyberSource Advanced for the SCMP API* ([PDF](#) | [HTML](#)) or *Getting Started with CyberSource Advanced for the Simple Order API* ([PDF](#) | [HTML](#)) describes how to get started using your CyberSource account.
- *Business Center Overview* ([PDF](#) | [HTML](#)) describes the features and options available with your CyberSource account using the Business Center.
- *Payment Tokenization Using the Business Center* ([PDF](#) | [HTML](#)) describes how to create and use on-demand customer profiles.
- *Recurring Billing Using the Business Center* ([PDF](#) | [HTML](#)) describes how to create and use customer recurring subscriptions.

Refer to the Support Center for complete CyberSource technical documentation:

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

## Customer Support

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

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



# Introduction

Account Updater notifies you of changes to stored customer credit card data. Such changes could affect expiration date, credit card number, or a brand. If you use the CyberSource Tokenization or Recurring Billing services, Account Updater automatically updates the stored card data on your behalf. Account Updater obtains credit card information from both the Visa INC's Account Updater Service and the MasterCard's Automatic Billing Updater Service.

## Options

---

### REST API Batch Updates

If you use CyberSource Recurring Billing or Payment Tokenization services, you can submit a batch of tokens using the REST API to be processed by the Account Updater service. See ["REST API Batch Updates," page 12](#).

### Harvest Updates

If you use CyberSource Recurring Billing or Payment Tokenization services, you can configure Account Updater to automatically update all your customer subscriptions or profiles with the latest credit card data. CyberSource creates a report detailing the updates made, including new expiry dates, masked primary account numbers (PANs), and card brands. See ["Harvest Updates," page 26](#).

### PAN Upload Updates

If you directly manage your customer's card data, you can create a file containing PANs, which CyberSource updates. Create a request file containing new PANs and POST it to the Account Updater URL. Download the response file using the Business Center or a client application. See ["PAN Upload Updates," page 31](#).

# Enrollment

---

Contact your account representative to enroll in Account Updater. CyberSource submits enrollment forms on your behalf to both MasterCard and Visa. The enrollment process can take up to 10 business days.

**Important**

If you are going to process Account Updater requests on behalf of merchants for whom you are not the merchant of record, you must enroll in Account Updater as a billing aggregator.

---

Billing aggregators can participate in Account Updater (see ["PAN Upload Updates," page 31](#)), but they must indicate in the Account Updater request files the merchant for whom the request is made. If you are a billing aggregator and fail to include the proper data in a record, CyberSource rejects the record and does not process your Account Updater requests.

## Business Center Permissions

As part of the enrollment process, an administrator must grant you permission in the Business Center to perform the actions below. If you are an administrator, you already have these permissions.

- View the status of a request file.
- Add and activate a PGP Security Key for PAN upload updates (see [page 31](#)).
- Create OAuth credentials for REST API batch updates (see [page 12](#)).
- Access downloadable response files.

# Terms of Use

---

By using the CyberSource Account Updater service, you agree to comply with the Visa U.S.A. Operating Regulations, Visa Account Updater Terms of Use, MasterCard rules and regulations, and all other applicable rules and regulations issued by any card association.

**Note**

If you are receiving Harvest updates (see [page 26](#)), your subscription is updated once every 30 days to ensure that you meet this requirement.

In addition, you must:

- Request an update for every participating Visa account in your customer database at least:
  - Once every 180 calendar days if you bill daily, weekly, monthly, quarterly, or bi-annually.
  - Once every 365 calendar days if you bill annually.
- Submit inquiries only for those accounts with which you have an ongoing customer relationship.
- Update your customer account database within 5 business days of receiving an update.
- Ensure that all update information you receive is properly, completely, and accurately incorporated into your data store for use in future transactions.
- Correct erroneous account information within 5 business days of receipt of error notification from CyberSource.

You may not:

- Request updates on accounts that have returned a response of *Contact Card Holder*. You must check your response file for CCH responses and take appropriate action such as removing the customer record from your billing cycle until you have contacted the cardholder.
- Submit update inquiries on behalf of any other entity unless you have enrolled in Account Updater as a billing aggregator.

# REST API Batch Updates

The Account Updater REST API enables you to POST a batch of tokens (subscription IDs) to the Account Updater service to be processed and updated. You must retrieve an OAuth token before creating the batch of tokens to be updated.

**Note**

The REST API supports [gzipped](#) requests for large payloads. To indicate to the server that you are sending a zipped request you must add the following to the request header (see [page 13](#)):

`Content-Encoding: gzip`

## Retrieve OAuth Token

The AU REST API uses OAuth 2.0 for authentication. The client ID and client secret credentials are required in order to create the access token that is used to authenticate each Account Updater REST API request.

### To create OAuth client credentials:

- Step 1** Log in to the Business Center:
- Live transactions: <https://ebc.cybersource.com>
  - Test transactions: <https://ebctest.cybersource.com>
- Step 2** In the left navigation panel, choose **Tools & Settings > Credit Card Account Updater > OAuth Credentials**.
- Step 3** Click **Create**.

**Important**

Make note of the generated values; the OAuth client credentials are displayed only once.

## Endpoints

To access endpoints, use an HTTP POST request with valid `x-www-form-urlencoded` authentication data:

- Test endpoint: `https://authtest.ic3.com/apiauth/v1/oauth/token`
- Live endpoint: `https://auth.ic3.com/apiauth/v1/oauth/token`

## POST Authentication Payload



### Note

Generate the OAuth client credentials to create the access token that is used to authenticate each Account Updater REST API request. See [page 12](#).

**Table 1 Required Authentication Payload**

| Variable      | Description   |
|---------------|---|
| platform      | Defines the Visa merchant platform.<br>Value: 1.  |
| client_id     | A merchant reference assigned to your account.<br>Created in the CyberSource Business Center. See <a href="#">page 12</a> . |
| grant_type    | Method by which the OAuth token is granted.<br>Value: <code>client_credentials</code>                                       |
| client_secret | Secret assigned to your account.<br>Created in the CyberSource Business Center. See <a href="#">page 12</a> .               |
| merchant_id   | Your CyberSource Merchant ID (MID)  |
| client_type   | Defines the client connection type.<br>Value: <code>confidential</code>   |

## URL Encoded POST Example

### Example 1 Header

```
POST /apiauth/v1/oauth/token HTTP/1.1
Host: authtest.ic3.com
Cache-Control: no-cache
Content-Type: application/x-www-form-urlencoded
```

**Example 2      Body**


---

```
platform=1&client_id=gfK2f0d552&grant_type=client_credentials&client_secret=d2dfdc225891fa719abe589b25afc2&merchant_id=myMID&client_type=confidential
```

---

## Response Example

Parse the `access_token` value from the response and use it within five minutes of being generated. CyberSource recommends generating a new OAuth token for each AU REST API request.

**Example 3      Response**


---

```
{
  "access_token": "00add7ee-9523-45b6-ac52-90f3e86962dd",
  "token_type": "bearer",
  "expires_in": 28799,
  "scope": "<comma_separated_list_of_scopes>",
  "client_status": "active"
}
```

---

## POST Batch Payload

Create a batch of tokens (subscription IDs) and POST it to the Account Updater service to be processed and updated.

## Endpoints

To access endpoints, use an HTTP POST request with a valid JSON payload:

- Test endpoint: <https://api.accountupdatertest.cybersource.com/v1/batch>
- Live endpoint: <https://api.accountupdater.cybersource.com/v1/batch>

## Batch Payload

### Headers

**Table 2 Request Headers**

| Variable      | Description  | Validation   |
|---------------|--|--|
| Authorization | Bearer \${ACCESS_TOKEN}                                  | The requests should be populated with a valid OAuth token. See <a href="#">"Retrieve OAuth Token," page 12</a> . |
| Content-Type  | Defines the content type. JSON is supported by this API. | Use application/json   |

## Batch Payload

**Table 3 Batch Payload**

| Variable           | Description   | Required | Validation  |
|--------------------|---|----------|---|
| notification_email | Email address to which batch status updates will be sent.   | Yes      | Valid email address   |
| merchant_reference | Your reference to identify the batch.   | No       | Min = 0<br>Max 255 characters   |
| included           | Elements to be included. Initially, the only element is <b>tokens</b> ; additional elements (such as credit card numbers) may be added in the future. | Yes      | —   |
| Tokens             | Comma-separated list of the tokens to be updated.   | Yes      | If the array is present, then it should not be empty (min length = 1) or contain null values. Maximum number of tokens is 10 million. |

## Batch Payload Example

**Example 4 Batch Payload**

```
{
  "notification_email": "test@cybersource.com",
  "merchant_reference": "Merchant Name",
  "included": {
    "tokens": ["321354654", "986546546"]
  }
}
```

## Successful Response Data

A successful response returns a 202 response code to the client along with the following payload.

**Table 4 Response Payload**

| Variable         | Description  | Always Returned? | Format            |
|------------------|--|------------------|-------------------|
| batch_id         | When the request is successful a batch ID is returned to the user.   | N                | Alphanumeric (26) |
| batch_item_count | When the request is successful this value is the number of items that were included in the request.<br><br>When the request is unsuccessful the value of this field is 0.  | Y                | Numeric (9)       |
| _links           | JSON object containing link elements relating to the request. Successful requests return the URI of the batch status.  | Y                |                   |
| self             | The resource address that was requested. Element within <b>_links</b> .  | Y                | URL               |
| status           | URI of the batch status resource. Do not hard-code the link to the batch status resource. Use the returned value to avoid errors if the URI structure changes.   |                  | URL               |
| correlation_id   | Returned when an error occurs. Provide this ID to Customer Support to help identify your transaction.  | N                | Alphanumeric      |
| code             | Returned when an error occurs. See <a href="#">"Unsuccessful Response Codes," page 17</a> .  | N                | Alphanumeric      |
| detail           | Returned when an error occurs. Detailed description of the error. See <a href="#">"Error Message Codes," page 19</a> .   | N                | Alphanumeric      |
| fields           | Returned when an error occurs. The array contains elements that describe the erroneous fields. See <a href="#">"Field Validation Messages," page 20</a> .  | N                | Alphanumeric      |
| path             | Returned when an error occurs. Element within the fields. Path of field name. See <a href="#">"Error Message Codes," page 19</a> .   | N                | Alphanumeric      |
| message          | Returned when an error occurs. This is a plain text error message and can be an element within the fields. This field can also appear with the fields JSON object.   | N                | Alphanumeric      |
| localizationKey  | Returned when an error occurs. An unique key which represents the error message and can be an element within fields. See <a href="#">"Error Message Codes," page 19</a> . This field can also appear with the JSON object. | N                | Alphanumeric      |



## Successful Response Example

### Example 5 Successful Response

---

```
{
  "_links": {
    "self": {
      "href": "https://api.accountupdatertest.cybersource.com/v1/batch"
    },
    "status": [
      {
        "href": "https://api.accountupdatertest.cybersource.com/v1/batch/14558965083280000539584782/status"
      }
    ]
  },
  "batch_id": "14558965083280000539584782",
  "batch_item_count": 2
}
```

---

## Unsuccessful Response Codes

If the response is unsuccessful, the client receives one of several different HTTP response codes to indicate an error. The body of the response contains an error message that specifically indicates what went wrong.

**Table 5 Response Codes**

| HTTP Response Code     | Description   | Error Message Code      |
|------------------------|---|-------------------------|
| 400 Bad Request        | A problem occurred during parsing of the request body.  | MALFORMED_PAYLOAD_ERROR |
| 401 Unauthorized       | The token passed in the header by calling client is not valid.  | INVALID_OAUTH_TOKEN     |
| 403 Forbidden          | The client does not have the required permissions to access the requested resource.<br><br>This error can occur when the role returned from the OAuth service is different from the role required to access the specified resource. | FORBIDDEN_RESPONSE      |
| 404 Not Found          | The requested URI is not valid.   | RESOURCE_NOT_FOUND      |
| 405 Method Not Allowed | The URI is valid, but the http operation is not permitted.<br><br>You might get this error when you try to perform a GET operation on an endpoint that only supports PUT, for example.  | UNSUPPORTED_HTTP_METHOD |

**Table 5 Response Codes (Continued)**

| HTTP Response Code         | Description  | Error Message Code  |
|----------------------------|--|---|
| 415 Unsupported Media Type | The Client has set a content type in the header which is not supported by the endpoint.<br><br>You may get this if, for example, you've set the content type to "application/xml" but the endpoint only supports "application/json". | UNSUPPORTED_MEDIA_TYPE  |
| 422 Unprocessable Entity   | Invalid data sent in payload   | VALIDATION_ERROR<br><br>See <a href="#">Table 7, "Field Validation Messages,"</a> on page 20. |
| 500 Internal Server Error  | Server error. This is normally thrown when something has happened on the server side which the calling client has no control over.<br><br>An example of this might be where a database connection is not working.                    | SERVER_ERROR  |
| 502 Bad Gateway            | There is an issue with a 3rd party system.   | REMOTE_SERVER_ERROR   |

## Error Response Example

### Example 6 Error Response

```
{
  "_links":{
    "self":{
      "href":"http://localhost:8080/"
    }
  },
  "code":"VALIDATION_ERROR",
  "correlationId":"98e30964397e49898a982a59b9338f5a",
  "detail":null,
  "fields":[
    {
      "path":"merchant_reference",
      "message":"Merchant reference cannot be null",
      "localizationKey":"cybsapi.ondemand.batch.merchantRef.notNull"
    }
  ],
  "localizationKey":"cybsapi.validation.errors",
  "message":"Field validation error"
}
```

## Error Message Codes

**Table 6 Response Level Error Messages**

| Code                    | Message                                      | Localization Key              | Detail  |
|-------------------------|--|-------------------------------|---|
| VALIDATION_ERROR        | Field validation error                       | cybsapi.validation.error      | One or more fields failed validation.   |
| MALFORMED_PAYLOAD_ERROR | Unable to parse payload                      | cybsapi.payload.error         | The payload could not be parsed.  |
| RESOURCE_NOT_FOUND      | Requested Resource Not Found                 | cybsapi.resource.notfound     | The requested resource is not found. Please try again later.  |
| SERVER_ERROR            | Error encountered while processing request   | cybsapi.server.error          | Internal Server Error. Contact the customer support.  |
| REMOTE_SERVER_ERROR     | Error encountered while processing request   | cybsapi.server.error          | Remote Server Error. Contact CyberSource customer support.  |
| FORBIDDEN_RESPONSE      | Unauthorized Access                          | cybsapi.forbidden.response    | You are not authorized to access this resource.   |
| INVALID_OAUTH_TOKEN     | Unauthorized Access. Invalid OAUTH token     | cybsapi.oauthtoken.invalid    | You are not authorized to access this resource. The request header should contain a valid OAuth token.  |
| OAUTH_TOKEN_NOT_PRESENT | Unauthorized Access. OAUTH token not present | cybsapi.oauthtoken.notpresent | The request header should contain a valid OAuth token.  |
| UNSUPPORTED_MEDIA_TYPE  | Unsupported media type                       | cybsapi.media.notsupported    | The header value for Content-type is not supported for the specified resource.                          |
| UNSUPPORTED_HTTP_METHOD | Method not allowed                           | cybsapi.method.notallowed     | The specified http method is not allowed. See the http response header for a list of permitted methods. |

## Field Validation Messages

If the error code is `VALIDATION_ERROR`, the field array is populated with one or more of the following validation errors.

**Table 7 Field Validation Messages**

| Path               | Message   | Localization Key                             |
|--------------------|---|--|
| notification_email | Email Address is invalid                                  | cybsapi.ondemand.batch.email.invalid         |
| merchant_reference | Merchant reference value should not exceed 255 characters | cybsapi.ondemand.batch.merchantRef.maxLength |
| included.tokens    | Token array should not be empty                           | cybsapi.ondemand.batch.tokens.notEmpty       |

## GET Batch Status

The Batch Status endpoint provides status information for the batch you created. The information is provided in JSON format.

## Endpoint

The URI is returned in the `_links` variable of the create batch response (see [page 16](#)), and CyberSource recommends using the dynamic link rather than hard-coding it. Access the status resource by sending a HTTP GET request to the endpoint. You need to authenticate the request with an OAuth `access_token` value in the Authorization header (see [page 12](#)).

## Status Responses

**Table 8 Status Responses**

| Status     | Description  |
|------------|--|
| Received   | The batch was received and is being checked for errors.                                      |
| Processing | The batch was sent to the card association(s) to be updated.                                 |
| Updating   | CyberSource has received a response from the card association(s) and is updating the tokens. |
| Complete   | Updates have been applied to the tokens.<br>A URI to view the batch report is now available. |
| Failed     | {specific error message}   |

## Response Example

### Example 7 Response

---

```
{
  "_links": {
    "self": {
      "href": "https://api.accountupdatertest.cybersource.com/v1/batch/{batchId}/status"
    },
    "report": [
      {
        "href": "https://api.accountupdatertest.cybersource.com/v1/batch/{batchId}/report"
      }
    ]
  },
  "version": "1.0",
  "status": "COMPLETE",
  "description": "Updates have been applied to your tokens. A batch report is available.",
  "batch_id": "14546840036390000725214624",
  "batch_source": "TOKEN_API",
  "batch_ca_endpoints": "VISA,MASTERCARD",
  "batch_created_date": "2016-04-23T18:25:43.511Z",
  "report_created_date": "2016-04-25T19:35:47.511Z",
  "merchant_reference": "acc7089",
  "totals": {
    "accepted_records": 10000,
    "rejected_records": 50,
    "updated_records": 487,
    "ca_responses": 500,

    "ca_responses_omitted": 13
  },
  "billing": {
    "nan": 175,
    "ned": 126,
    "acl": 124,
    "cch": 75
  }
}
```

---

# GET Batch Report

The Batch Report is in JSON format and contains details of the tokens that have been updated, such as new masked PANs and expiry details. If you are using format-preserving tokens (contain the last four digits of the PAN), the report also contains the new token (subscription ID) values.

To reduce the file size of the JSON report, CyberSource recommends that you specify Accept-Encoding: gzip in the header of the GET request.

## Endpoint

The URI is returned in the report link of the batch status response (see [page 20](#)) and CyberSource recommends using the dynamic link rather than hard-coding it. Access the endpoint by using an HTTP GET request with valid `x-www-form-urlencoded` authentication data to:

`https://api.accountupdater.cybersource.com/batch/<batch_id>/status`

## Report Fields

**Table 9 Batch Report Fields**

| Field Name          | Description  |
|---------------------|--|
| version             | <p>The version of the report.</p> <p>For example, v1.4-1 is the major version of the API used to create the batch and 4 is the minor version of the report.</p> <p><b>Note</b> You always receive the latest minor version of the report for the API you used to create the batch.</p> |
| batch_id            | The batch ID assigned by Account Updater.  |
| batch_source        | <p>The method used to create the batch.</p> <p>For example, token_api</p>  |
| batch_ca_endpoints  | <p>The endpoints to which the card numbers were sent.</p> <p>Possible values: VISA,MASTERCARD</p>  |
| batch_created_date  | <p>The date on which the batch was created.</p> <p>Format: ISO_8601 UTC date.</p>  |
| report_created_date | <p>The date on which the report was created.</p> <p>Format: ISO_8601 UTC date.</p>   |
| merchant_reference  | The merchant reference (if present in the request).  |

**Table 9 Batch Report Fields (Continued)**

| Field Name           | Description   |
|----------------------|---|
| totals               | JSON object containing the high-level summary of the batch:   |
| accepted_records     | The number of tokens that were identified and retrieved for the merchant ID.  |
| rejected_records     | The number of tokens that were not identified or retrieved.   |
| updated_records      | The number of updates that were applied to a token.   |
| ca_responses         | The number of updates that were received from the card associations. This value represents updates that may have or have not been applied to a token. |
| ca_responses_omitted | The number of updates that were not applied to a token.<br><br>For example, a response is returned by more than one card association.                 |
| billing              | JSON object containing the billing summary information:   |
| nan/ned/acl/cch      | The number of each billed response type.  |
| records              | JSON object containing additional objects that relate to the original tokens and the updates or errors that occurred:                                 |
| id                   | The CyberSource generated ID for the record.  |
| source_record        | JSON object containing details from the source token:   |
| token                | The subscription ID included in the request.  |
| card_number          | Masked card number before an update. First six digits and the last four digits are not masked.  |
| card_expiry_month    | Two-digit month in which the card expires.  |
| card_expiry_year     | Two-digit year in which the card expires.   |
| card_type            | Type of card.   |
| response_record      | JSON object containing the details that were made to the token:   |
| response             | The type of response. See <a href="#">Table 22, "Response Codes and Reason Codes,"</a> on page 43.  |
| reason               | Reason code for the response. See <a href="#">Table 22, "Response Codes and Reason Codes,"</a> on page 43.  |
| token                | If last-four-digit format-preserving tokens are used, a new token (subscription ID) can be returned that replaces the source record token.            |
| card_number          | Masked card number. First six and last four digits are not masked.  |
| card_expiry_month    | Two-digit month in which the card expires.  |
| card_expiry_year     | Four-digit year in which the card expires.  |
| card_type            | Type of card associated with the updated token.   |

## Report Example

### Example 8 Report

---

```
{
  "version": "1.0",
  "batch_id": "14546840036390000725214624",
  "batch_source": "TOKEN_API",
  "batch_ca_endpoints": "VISA,MASTERCARD",
  "batch_created_date": "2012-04-23T18:25:43.511Z",
  "report_created_date": "2012-04-25T19:35:43.511Z",
  "merchant_reference": "acc1234",
  "totals": {
    "accepted_records": 10000,
    "rejected_records": 50,
    "updated_records": 487,
    "ca_responses": 500,
    "ca_responses_omitted": 13
  },
  "billing": {
    "nan": 175,
    "ned": 126,
    "acl": 124,
    "cch": 75
  },
  "records": [
    {
      "id": "58996",
      "source_record": {
        "token": "423456732651111",
        "card_number": "411111xxxxxx1111",
        "card_expiry_month": "07",
        "card_expiry_year": "17",
        "card_type": "001"
      },
      "response_record": {
        "response": "NED",
        "reason": "800",
        "card_expiry_month": "07",
        "card_expiry_year": "19"
      }
    },
    {
      "id": "58997",
      "source_record": {
        "token": "423456732651111",
        "card_number": "411111xxxxxx1112",
        "card_expiry_month": "08",
        "card_expiry_year": "18",
        "card_type": "001"
      }
    }
  ]
}
```



```

    },
    "response_record": {
      "response": "ERR",
      "reason": "805"
    }
  },
  {
    "id": "58999",
    "source_record": {
      "token": "1234567890121111",
      "card_number": "411111xxxxx1111",
      "card_expiry_month": "07",
      "card_expiry_year": "17",
      "card_type": "001"
    },
    "response_record": {
      "response": "NAN",
      "reason": "800",
      "token": "5234567890124444",
      "card_number": "555555xxxxx4444",
      "card_expiry_month": "08",
      "card_expiry_year": "18",
      "card_type": "001"
    },
    "response_record": {
      "response": "DEC",
      "reason": "801"
    }
  }
]
}

```

---

## Testing

---

Before you submit files to the production server, CyberSource recommends that you first test your request files. See ["Testing," page 46](#).

## Profile Update Report

---

The Account Updater Profile Update report is available for download 24 to 48 hours after your scheduled updates. You can download the report from the Business Center or with a client application. See ["Profile Update Report," page 27](#).

# Harvest Updates

**Important**

You must enroll in Account Updater and comply with the Terms of Use (see [page 11](#)).

CyberSource Account Updater is integrated with the Recurring Billing and Payment Tokenization functionality so that your customer subscriptions or profiles are automatically kept up to date with the latest credit card data changes.

After your enrollment forms are processed, CyberSource:

- 1 Configures your account to automatically update your customer subscriptions or profiles once per month with updated credit card data. These updates can include a new expiration date, a new credit card number, or a brand change such as a change from Visa to MasterCard.
- 2 Requests updates only for customer subscriptions or profiles with a status of *active* or *on-hold*. Updates for customer subscriptions or profiles with a status of *completed* or *cancelled* are not requested.
- 3 Generates the Profile Update Report (see [page 27](#)), which can be downloaded by logging in to the Business Center or by using a client application.

To have your customer subscriptions or profiles updated on a particular day of the month to coincide with your billing cycle, contact Customer Support.

**Note**

It is best practice to request updates for your customer subscriptions or profiles 3 to 5 days before your billing cycle begins. You can choose any calendar day, 1 through 28.

# Profile Update Report

---

The Account Updater Profile Update report is available for download 24 to 48 hours after your scheduled updates. You can download the report from the Business Center or with a client application.

## To download the Profile Update report:

---

- Step 1** Log in to the Business Center:
    - Live transactions: <https://ebc.cybersource.com>
    - Test transactions: <https://ebctest.cybersource.com>
  - Step 2** In the left navigation panel, choose **Reports > Report Search**. The Report Search window appears.
  - Step 3** From the Report drop-down list, choose **All**.
  - Step 4** From the Frequency drop-down list, choose **Daily**.
  - Step 5** Choose the day that your reports were processed.
  - Step 6** Click **Submit**. The report is listed in the Downloadable Reports table.
  - Step 7** Click the **Download** link next to the report name.
- 

To connect to the report server, your client application must support HTTPS connections:

- HTTP/1.0 or HTTP/1.1
- TLS 1.2 or later

Your client application must use Basic Access Authentication to send the username and password. Many HTTPS client libraries implement this authentication method. For more information about Basic Access Authentication, see:

<http://www.ietf.org/rfc/rfc2617.txt>

To request a report, your client application must send an HTTP GET message to the report server. Use this URL format for the request:

**`https://<server_name>/DownloadReport/YYYY/MM/DD/<merchant_ID>/<report_name>.<report_format>`**

**Table 10 Report URL Values**

| Variable        | Description   |
|-----------------|---|
| <server_name>   | Name of the server from which to download the report. Use one of these values: <ul style="list-style-type: none"> <li>■ Test: ebctest.cybersource.com/ebctest</li> <li>■ Live: ebc.cybersource.com/ebc</li> </ul> |
| YYYY            | Four-digit year.  |
| MM              | Two-digit month.  |
| DD              | Two-digit day.  |
| <merchant_id>   | CyberSource merchant ID.  |
| <report_name>   | Name of the report to download:<br><merchant_id>au.response.ss  |
| <report_format> | Report format to download:<br>csv   |

## Profile Update Report Records

### Header Records

**Table 11 Header Records**

| Order | Field Name          | Description   | Max. No. of Characters |
|-------|---------------------|---|------------------------|
| 1     | Record Identifier   | Constant value indicating the record type.<br>Format: H                       | 1                      |
| 2     | File Classification | Indicates file type.<br>Format: cybs.au.response.ss                           | 30                     |
| 3     | Merchant ID         | CyberSource Merchant ID.<br>Format: Alphanumeric                              | 30                     |
| 4     | Batch ID            | Unique identifier for the batch, generated by CyberSource.<br>Format: Numeric | 30                     |

## Details Records

**Table 12 Details Records**

| Order | Field Name                 | Description   | Max. No. of Characters |
|-------|----------------------------|---|------------------------|
| 1     | Record Identifier          | Constant value indicating the record type.<br>Format: D   | 1                      |
| 2     | Account Updater Request ID | Unique CyberSource identifier for the record.<br>Format: Numeric  | 30                     |
| 3     | Profile ID                 | The value that identifies the customer subscription or profile. CyberSource returned this value when the subscription or profile was created.<br>Format: Numeric  | 16 or 22               |
| 4     | New Card Number            | New card number with eight digits masked.<br>Format: Alphanumeric<br><br>If no new card number is available, this field is populated with the current card number.  | 19                     |
| 5     | Response Code              | Return code for the record (see <a href="#">Table 22, "Response Codes and Reason Codes," on page 43</a> ).<br><br>Customer subscriptions or profiles with a response code of ACL are moved to a <i>cancelled</i> state.<br>Format: Alphabetic | 3                      |
| 6     | Reason Code                | Reason code for the record (see <a href="#">Table 22, "Response Codes and Reason Codes," on page 43</a> ).<br>Format: Numeric   | 3                      |
| 7     | Old Card Number            | Old card number with eight digits masked.<br>Format: Alphanumeric   | 19                     |
| 8     | Old Card Expiry Month      | Expiration month of the old card.<br>Format: MM   | 2                      |
| 9     | Old Card Expiry Year       | Expiration year of the old card.<br>Format: YY  | 2                      |
| 10    | New Card Expiry Month      | Expiration date of the new card.<br>Format: MM  | 2                      |
| 11    | New Card Expiry Year       | Expiration year of the new card.<br>Format: YY  | 2                      |
| 12    | New Profile ID             | The new value that identifies the customer subscription or profile. This value supersedes the previous subscription ID. CyberSource returns this value when the profile is updated.<br>Format: Numeric  | 16                     |

## Footer Records

**Table 13 Footer Records**

| Order | Field Name        | Description   | Max. No. of Characters |
|-------|-------------------|---|------------------------|
| 1     | Record Identifier | Constant value indicating the record type.<br>Format: F   | 1                      |
| 2     | Record Count      | The number of detail records in the file.<br>Format: Numeric  | —                      |
| 3     | Response Code     | Indicates the overall response (see <a href="#">Table 23, "Response Codes and Reason Codes," on page 44</a> ).<br>Format: Alphabetic<br>Possible value:<br>COM: Completed | 3                      |
| 4     | Reason Code       | Indicates the overall reason for the response code (see <a href="#">Table 23, "Response Codes and Reason Codes," on page 44</a> ).<br>Format: Numeric                     | 3                      |

# PAN Upload Updates

**Important**

You must enroll in Account Updater and comply with the Terms of Use. See ["Terms of Use," page 11.](#)

After the syntax of the request file is validated, CyberSource begins processing the file.

Account Updater files are processed once per day. You can expect your response file to be available 24-48 hours after you submit your request file. CyberSource recommends sending your Account Updater request file 3 to 5 days before your billing cycle starts to ensure that your file completes processing and that you have enough time to update your data store.

Responses from Visa and MasterCard are consolidated and returned in an encrypted response file. See ["Response File Records," page 39.](#)

## Creating Security Keys

To upload PAN updates you must create two types of security keys: a transaction security key and a PGP public/private key pair.

### Transaction Security Key

You must use the transaction security key to programmatically connect to CyberSource and upload request files.

If you use the Simple Order API to process transactions, you can use the same key for Account Updater.

If you have been using the SCMP API to process transactions, you must create a transaction security key that works with the Simple Order API. See "Simple Order API Security Keys" in [Creating and Using Security Keys](#).

## PGP Public/Private Key Pair

PGP public/private key pair is used to protect, by encryption, credit card data contained in the response files. The key pair contains both a public and a private key. You exchange the public part of this key pair with CyberSource, who uses it to encrypt the response files. You maintain the private part of the key pair to decrypt the response file.

To create a PGP key pair for encrypting and decrypting credit card data, see “PGP Security Keys” in [Creating and Using Security Keys](#).

## Formatting a Request File

Account Updater request files must be in CSV format with a maximum file size of 12 MB.

The format for a request file consists of:

- A header record.
- A detail record with one or more data records, each on a separate line.
- A footer record, which indicates the end of the file.

## Header Record

The header record consists of comma-separated values and uses the fields listed in the following table:

**Table 14 Header Record Fields**

| Field Name          | Description  | Required or Optional | Data Type (length) |
|---------------------|--|----------------------|--------------------|
| Record Identifier   | Constant value indicating the record type.<br>Format: H  | Required             | Alpha (1)          |
| File Classification | Indicates whether this is a request or response file.<br>Format: cybs.au.request.pan   | Required             | Alpha (30)         |
| merchantID          | Your CyberSource merchant ID.<br>Format: sampleID2   | Required             | Alphanumeric (30)  |
| batchID             | File (batch) identifier that you assign. The batch ID must be unique. If you send a file that contains a previously submitted batch ID, the file is rejected.<br>Format: 12345 | Required             | Numeric (30)       |



**Table 14 Header Record Fields (Continued)**

| Field Name   | Description  | Required or Optional | Data Type (length) |
|--------------|--|----------------------|--------------------|
| recordCount  | The number of detail records in the file.<br>Format: 12345   | Required             | Numeric            |
| statusEmail  | Email address to which status emails for the request are sent.<br>Format: aaa@aaa.aaa  | Required             | Alphanumeric (100) |
| creationDate | Optional field that can be passed by merchant for reference. If present, it appears in the Business Center Account Updater View Status window.<br>Format: YYYY-MM-DD | Optional             | (10)               |
| Batch Info   | Optional field that can be passed by the merchant for reference.<br>Format: sample12   | Optional             | Alphanumeric (50)  |

## Detail Record

Each file must contain at least one detail record.

**Table 15 Detail Record Fields**

| Field Name            | Description  | Required or Optional | Data Type (length) |
|-----------------------|--|----------------------|--------------------|
| Record Identifier     | Constant value indicating the record type.<br>Format: D  | Required             | Alpha (1)          |
| Card Number           | Card number to process.<br>Format: Numeric   | Required             | Numeric (19)       |
| Card Expiry Month     | Expiration month of the card.<br>Format: MM  | Required             | Alphanumeric (2)   |
| Card Expiry Year      | Expiration year of the card.<br>Format: YY   | Required             | Numeric (2)        |
| Merchant Reference ID | You can use this field to track your Account Updater request records. If this field is populated, the same value will be returned in the Account Updater response file.<br>Format: sampleID2 | Optional             | Alphanumeric (50)  |

**Table 15 Detail Record Fields (Continued)**

|                     |  |          |                   |
|---------------------|--|----------|-------------------|
| BA Sub Merchant ID* | This field is required for billing aggregator merchants only.<br><br>Format: sampleID2 | Optional | Alphanumeric (10) |
|---------------------|--|----------|-------------------|

## Footer Record

Each file should contain only one footer record.

**Table 16 Footer Record Field**

| Field Name        | Description   | Required or Optional | Data Type (length) |
|-------------------|---|----------------------|--------------------|
| Record Identifier | Constant value indicating the record type.<br><br>Format: F | Required             | Alpha (1)          |

## Request File Examples

### Example 9 Request File for Non-Billing Aggregator Merchants

```
H,cybs.au.request.pan,merchant1,001,2,notify@yourcompany.com,2009-03-23,My Jan Batch
D,111122233334444,11,09,0001
D,2222333344445555,11,09,0002
F
```

### Example 10 Request File for Billing Aggregator Merchants

```
H,cybs.au.request.pan,merchant1,001,2,notify@yourcompany.com,2009-03-23,My Jan Batch
D,111122233334444,11,09,0001,subId01
D,2222333344445555,11,09,0002,subId02
F
```

# Uploading a Request File

---

**Important**

For each PAN you upload, you can receive multiple responses. For example, if you upload one Visa card for an update, you can receive both a MasterCard and Visa response, or two Visa responses.

---

To upload the request file, use HTTPS. Your client application must support HTTP/1.0 or HTTP/1.1 and TLS 1.2 or later.

To access the Account Updater URL, you 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.

Before you submit files to the production server, CyberSource recommends that you first test your request files. Follow the instructions in ["Testing," page 46](#).

Use the following URLs for submitting test and live Account Updater request files:

- Testing:  
`https://accountupdatertest.cybersource.com/upload/UploadAccountUpdaterFile`
- Live:  
`https://accountupdater.cybersource.com/upload/UploadAccountUpdaterFile`

See ["Sample Java Code for Uploading Files," page 49](#), for more information on creating a client certificate to upload request files.

## Email Notification

---

After you upload the request file, CyberSource validates the syntax and sends you a confirmation email indicating whether the file passed this stage of validation. You must specify an email address in the **statusEmail** header field in order to receive this confirmation email. If this field is left blank, you do not receive an email confirmation and you must go to the Business Center to view the status (see ["Viewing the Batch File Status," page 36](#)). CyberSource sends the email notification within 30 minutes of receipt of the request file. However, actual timing depends on the system load at the time of file submission.

The table below lists possible subject lines of the email notifications.

**Table 17 Email Notifications**

| Subject Line | Reason  |
|--------------|---|
| Received     | <p>The Account Updater request file was received.</p> <p>CyberSource processes the requests in the file. No action is required.</p> <p>You can view the status of this request file in the Business Center. See <a href="#">page 36</a>.</p>              |
| Rejected     | <p>The file was rejected.</p> <p>Read the contents of the email and follow the suggested remedy.</p> <p>You cannot view the status of this request file in the Business Center.</p>   |
| Validated    | <p>The file passed validation.</p> <p>You can view the status of this request file in the Business Center. See <a href="#">page 36</a>.</p>   |
| Declined     | <p>The file did not pass validation checks. All records are declined.</p> <p>Read the contents of the email and follow the suggested remedy.</p> <p>You can view the status of this request file in the Business Center. See <a href="#">page 36</a>.</p> |
| Processing   | <p>The request file is being processed by Account Updater.</p> <p>You can view the status of this request file in the Business Center. See <a href="#">page 36</a>.</p>   |
| Completed    | <p>The response file has been generated and is ready for download.</p> <p>You can view the status of this request file in the Business Center. See <a href="#">page 36</a>.</p>   |

## Viewing the Batch File Status

### To view the status of a batch file in the Business Center:

- Step 1** Log in to the Business Center:
  - Live transactions: <https://ebc.cybersource.com>
  - Test transactions: <https://ebctest.cybersource.com>
- Step 2** In the navigation pane, choose **Tools & Settings > Credit Card Account Updater > Status**. The Account Updater Status Search window appears.
- Step 3** Choose the type of search: **batch ID** or **date**.

**Note**

You specified the batch ID search type in the request file.

**Step 4** Choose the number of batches that you want displayed on each page of results (50 to 100) and the sort order (newest results or oldest results first).

**Step 5** Click **Search**.

The Status Results window appears.

## Account Updater Status Results

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

| Batch ID      | Date of Request         | Status     | Origin      | Records In Request |
|---------------|-------------------------|------------|-------------|--------------------|
| 197713050006  | May 14 2009 06:49:59 AM | Declined   | File Upload | 2                  |
| 1976120509003 | May 12 2009 06:42:01 AM | Processing | File Upload | 3                  |
| 1976120509002 | May 12 2009 03:17:01 AM | Processing | File Upload | 3                  |
| 1976120509001 | May 12 2009 01:31:04 AM | Processing | File Upload | 3                  |
| 1976080509002 | May 08 2009 03:56:55 AM | Validated  | File Upload | 3                  |
| 1976080509001 | May 08 2009 01:23:04 AM | Received   | File Upload | 3                  |
| 2966002       | May 07 2009 07:12:27 AM | Validated  | File Upload | 7                  |
| 1976070509004 | May 07 2009 03:51:39 AM | Received   | File Upload | 3                  |
| 1976070509003 | May 07 2009 03:51:18 AM | Received   | File Upload | 3                  |
| 2966001       | May 07 2009 03:42:11 AM | Validated  | File Upload | 7                  |

**Step 6** To view the completed files, choose **Reports > Report Search** in the navigation pane.

## Downloading a Response File

You can download response files with a status of *Complete* from the Business Center or with a client application.

### To download a response file:

**Step 1** Log in to the Business Center.

**Step 2** In the left navigation pane, choose **Reports > Report Search**. The Report Search window appears.

**Step 3** Chose **All** from the Report menu and **Daily** from the Frequency menu.

**Step 4** Choose the day that your files were processed.

**Step 5** Click **Submit**.

The file appears in the **Downloadable Reports** table.

**Step 6** Click the **download** link next to your file.

Depending on your browser settings, the file either opens in your browser window, or you are prompted to save the file.

**Step 7** You must use the private PGP key to decrypt files. You can do it using the same third-party software you used to create the keys.



**Note**

If you do not activate the PGP public key in your CyberSource merchant account profile, the response file is not generated. See *Creating and Using Security Keys* ([PDF](#) | [HTML](#)).

To connect to the report server, your client application must support HTTPS connections:

- HTTP/1.0 or HTTP/1.1
- TLS 1.2 or later

Your client application must use Basic Access Authentication to send the username and password. Many HTTPS client libraries implement this authentication method. For more information about Basic Access Authentication, see:

<http://www.ietf.org/rfc/rfc2617.txt>

To send an API request with an HTML form or any script, include this endpoint:

`https://ebc.cybersource.com/ebc/DownloadReports`

To request a response file, your client application must send an HTTP GET message to the report server. The URL that you specify in your message indicates which report you want to download.

Use the following format for the URL:

`https://<server_name>/DownloadReport/YYYY/MM/DD/<merchant_ID>/<report_name>.<report_format>`

For example, if your merchant ID is *sample*, you would use the following URL to download the February 1, 2010 response file from the production system:

`https://ebc.cybersource.com/ebc/DownloadReport/2010/02/01/sample/sample.1234.au.response.pan.csv`

The following table describes each value in the URL.

**Table 18 Report URL Values**

| Value           | Description  |
|-----------------|--|
| <server_name>   | Name of the server from which to download the report.<br>Use one of these values: <ul style="list-style-type: none"> <li>■ Test server: ebctest.cybersource.com/ebctest</li> <li>■ Production server: ebc.cybersource.com/ebc</li> <li>■ Test reports server: downloadreportstest.cybersource.com</li> <li>■ Production reports server: downloadreports.cybersource.com</li> </ul> |
| YYYY            | Four-digit year  |
| MM              | Two-digit month  |
| DD              | Two-digit day  |
| <merchant_id>   | CyberSource merchant ID  |
| <report_name>   | Name of the report to download: merchantID.batchID.au.response.pan   |
| <report_format> | Report format: csv   |

## Response File Records

The response file is encrypted with the public part of the PGP Key that you generated and uploaded to CyberSource. To read a response file, you must decrypt it using the private part of the PGP key pair. You can do so with the same third-party software you used to create the keys.

The format for a request file consists of:

- A header record.
- A detail record with one or more data records, each on a separate line.
- A footer record, which indicates the end of the file.

## Header Record

The header record consists of comma-separated values and uses the fields listed in the following table:

**Table 19 Header Record Fields**

| Field Name          | Description   | Required or Optional | Data Type (length) |
|---------------------|---|----------------------|--------------------|
| Record Identifier   | Constant value indicating the record type.<br>Format: H   | Required             | Alpha (1)          |
| File Classification | Indicates whether this is a request or response file, and the type of service.<br>Formats: cybs.au.response.pan | Required             | Alphanumeric (30)  |
| MerchantID          | Your CyberSource merchant ID.<br>Format: Alphanumeric   | Required             | Alphanumeric (30)  |
| BatchID             | File (batch) identifier sent in the request file.<br>Format: Numeric  | Required             | Numeric (30)       |

## Detail Record

Each file contains at least one detail record.

**Table 20 Detail Record Fields**

| Field Name                | Description  | Data Type (length) |
|---------------------------|--|--------------------|
| Record Identifier         | Constant value indicating the record type.<br>Format: D          | Alpha (1)          |
| Request ID                | Unique CyberSource identifier for the record.<br>Format: Numeric | Numeric (30)       |
| Old Card Number           | Old card number with eight digits masked.<br>Format: Numeric     | Numeric (19)       |
| Old Card Expiration Month | Old expiration month.<br>Format: MM                              | Numeric (2)        |
| Old Card Expiration Year  | Old expiration year.<br>Format: YY                               | Numeric (2)        |
| New Card Number           | New card number with eight digits masked.<br>Format: Numeric     | Numeric (19)       |
| New Card Expiration Month | New expiration month.<br>Format: MM                              | Numeric (2)        |



**Table 20 Detail Record Fields (Continued)**

| Field Name               | Description   | Data Type (length) |
|--------------------------|---|--------------------|
| New Card Expiration Year | New expiration year.<br>Format: YY  | Numeric (2)        |
| Merchant Reference ID    | This field is optional and is returned in the response if present in the request file.<br>Format: Alphanumeric              | Alphanumeric (50)  |
| BA Sub Merchant ID       | This field is returned in the response if sent in the request file.<br>Format: Alphanumeric                                 | Alphanumeric (10)  |
| Response Code            | Response code for the record. See <a href="#">Table 22, "Response Codes and Reason Codes,"</a> on page 43.<br>Format: Alpha | Alpha (3)          |
| Reason Code              | Reason code for the record. See <a href="#">Table 22, "Response Codes and Reason Codes,"</a> on page 43.<br>Format: Numeric | Numeric (3)        |

## Footer Record

Each file contains only one footer record.

**Table 21 Footer Record Fields**

| Field Name        | Description   | Data Type (length) |
|-------------------|---|--------------------|
| Record Identifier | Constant value indicating the record type.<br>Format: F   | Alpha (1)          |
| Record Count      | The number of detail records in the file.<br>Format: Numeric  | Numeric (10)       |
| Response Code     | Response code for the file. See <a href="#">Table 23, "Response Codes and Reason Codes,"</a> on page 44.<br>Format: Alpha | Alpha (3)          |
| Reason Code       | Reason code for the file. See <a href="#">Table 23, "Response Codes and Reason Codes,"</a> on page 44.<br>Format: Numeric | Numeric (3)        |

## File Examples

### Example 11 Non-Billing Aggregator Response File

---

```
H,cybs.au.response.pan,merchant1,001
D,10000000000000000001,1111222233334444,11,09,,,0001,,NUP,800
D,10000000000000000002,2222333344445555,11,09,6666777788889999,11,11,0002,,NAN,800
F,2,COM,800
```

---

### Example 12 Billing Aggregator Response File

---

```
H,cybs.au.response.pan,merchant1,001
D,10000000000000000001,1111222233334444,11,09,,,0001,subId01,NUP,800
D,10000000000000000002,2222333344445555,11,09,6666777788889999,11,11,0002,subId02,NAN,800
F,2,COM,800
```

---

# Response Codes and Reason Codes

## Record Level

The response code and the reason code for the record appear in the details record of the request file.

### Example 13 Details Record

```
D,10000000000000000002,2222333344445555,11,09,6666777788889999,11,11,0002,,NAN,800
```

**Table 22 Response Codes and Reason Codes**

| Response Code | Response Code Description  | Reason Code | Reason Code Description   | Billable or Non-Billable Code |
|---------------|--|-------------|---|-------------------------------|
| ACL           | Match: account closed.<br><br><b>Note</b> The status of the customer subscription changes to <i>cancelled</i> and all recurring billing payments stop. | 800         | Success.  | Billable.                     |
| CCH           | Contact card holder.   | 800         | Success.  | Billable.                     |
| CUR           | Card data current.   | 800         | Success.  | Non-billable.                 |
| DEC           | —  | 801         | Invalid card number.  | Non-billable.                 |
| DEC           | —  | 802         | Invalid check digit.  | Non-billable.                 |
| DEC           | —  | 803         | Invalid expiration date.  | Non-billable.                 |
| DEC           | —  | 804         | Unsupported card type.  | Non-billable.                 |
| DEC           | —  | 805         | Invalid card type length.   | Non-billable.                 |
| DEC           | —  | 806         | Unknown card type.  | Non-billable.                 |
| DEC           | —  | 810         | Invalid BA sub merchant ID.   | Non-billable.                 |
| DEC           | —  | 852         | Unknown token.<br><br>This token does not exist or is not associated with your account. | Non-billable.                 |
| ERR           | —  | 801         | Invalid card number.  | Non-billable.                 |

**Table 22 Response Codes and Reason Codes (Continued)**

| Response Code | Response Code Description                                      | Reason Code | Reason Code Description                            | Billable or Non-Billable Code |
|---------------|--|-------------|--|-------------------------------|
| ERR           | —  | 802         | Invalid check digit.                               | Non-billable.                 |
| ERR           | —  | 803         | Invalid expiration date.                           | Non-billable.                 |
| ERR           | —  | 804         | Unsupported card type.                             | Non-billable.                 |
| ERR           | —  | 807         | Merchant not enrolled properly in Account Updater. | Non-billable.                 |
| ERR           | —  | 808         | Incorrect record indicator.                        | Non-billable.                 |
| ERR           | —  | 809         | Unknown error code received during processing.     | Non-billable.                 |
| ERR           | —  | 811         | New account number failed MOD-10 check.            | Non-billable.                 |
| NAN           | New account number. It may also include a new expiration date. | 800         | Success.   | Billable.                     |
| NED           | New expiration date.   | 800         | Success.   | Billable.                     |
| NUP           | No match, no update.   | 800         | Success.   | Non-billable.                 |
| UNA           | Inconsistent update received, not applicable.                  | 800         | Inconsistent update received, not applicable.      | Non-billable.                 |

## Request File Level

The response code and the reason code for the request file appear in the footer record of the request file.

### Example 14 Footer Record

F, 2, COM, 800

**Table 23 Response Codes and Reason Codes**

| Response Code | Response Code Description  | Reason Code | Reason Code Description |
|---------------|--|-------------|-------------------------|
| COM           | The merchant request file has been validated by CyberSource, processed, and the response received. | 800         | Success.                |

**Table 23    Response Codes and Reason Codes (Continued)**

| <b>Response Code</b> | <b>Response Code Description</b>  | <b>Reason Code</b> | <b>Reason Code Description</b>                                      |
|----------------------|---|--------------------|---|
| DEC                  | The merchant request file was not processed because each record failed record-level validation. | 801                | All records within the request file failed record-level validation. |

# Testing

The CyberSource CAS environment provides a simulator in which the response from the card association can be triggered using card numbers listed in [Table 24, "Visa Card Test Numbers," on page 46](#) and [Table 25, "MasterCard Card Test Numbers," on page 47](#). This simulator ensures you can handle the possible response combinations when connecting to multiple card associations.



**Note**

The CAS environment typically completes the process in a matter of minutes rather than the 24+ hour duration of the live environment when updates are sent to the actual card associations.

## Visa Card Numbers

The bold fields represent the token updates for Recurring Billing and Payment Tokenization merchants using the REST API batch update and harvest update. For a description of each response code, see [Table 22, "Response Codes and Reason Codes," on page 43](#).

**Table 24 Visa Card Test Numbers**

| Card Number<br>(Remove spaces when<br>sending to CyberSource.) | Response   |
|--|--|
| 4000 0071 0951 9220  | Visa Response: NAN   <b>MasterCard Response: NAN</b> |
| 4000 0015 3919 2096  | <b>Visa Response: NAN</b>   MasterCard Response: ACL |
| 4000 0018 6481 0239  | <b>Visa Response: NAN</b>   MasterCard Response: CUR |
| 4000 0091 9582 8465  | Visa Response: NED   <b>MasterCard Response: NAN</b> |
| 4000 0027 5765 7455  | <b>Visa Response: NED</b>   MasterCard Response: ACL |
| 4000 0071 1311 2087  | <b>Visa Response: NED</b>   MasterCard Response: CUR |
| 4000 0021 1752 4874  | Visa Response: ACL   <b>MasterCard Response: NAN</b> |
| 4000 0071 1629 4650  | Visa Response: ACL   <b>MasterCard Response: ACL</b> |
| 4000 0020 5548 7183  | <b>Visa Response: ACL</b>   MasterCard Response: CUR |

**Table 24 Visa Card Test Numbers (Continued)**

| <b>Card Number</b><br>(Remove spaces when<br>sending to CyberSource.) | <b>Response</b>                                      |
|---|--|
| 4000 0052 8063 4792   | Visa Response: CUR   <b>MasterCard Response: NAN</b> |
| 4000 0024 0631 2635   | Visa Response: CUR   <b>MasterCard Response: ACL</b> |
| 4000 0089 2339 9344   | Visa Response: CUR   <b>MasterCard Response: CUR</b> |
| 4000 0055 7908 8940   | Visa Response: NUP   <b>MasterCard Response: NAN</b> |
| 4000 0057 9875 5634   | Visa Response: NUP   <b>MasterCard Response: ACL</b> |
| 4000 0080 9110 0706   | Visa Response: CCH   <b>MasterCard Response: NAN</b> |
| 4000 0026 9567 5155   | Visa Response: CCH   <b>MasterCard Response: ACL</b> |
| 4000 0035 8627 6236   | Visa Response: CCH   <b>MasterCard Response: CUR</b> |

## MasterCard Card Numbers

The bold fields represent the token updates for Recurring Billing and Payment Tokenization merchants using the REST API batch update and harvest update. For a description of each response code, see [Table 22, "Response Codes and Reason Codes," on page 43](#).

**Table 25 MasterCard Card Test Numbers**

| <b>Card Number</b>  | <b>Response</b>                                      |
|---------------------|--|
| 5111 1110 4714 3086 | <b>Visa Response: NAN</b>   MasterCard Response: NAN |
| 5111 1110 2999 7178 | Visa Response: ACL   <b>MasterCard Response: NAN</b> |
| 5111 1110 1548 6814 | Visa Response: CUR   <b>MasterCard Response: NAN</b> |
| 5111 1110 5459 2548 | Visa Response: NUP   <b>MasterCard Response: NAN</b> |
| 5111 1110 4871 8571 | Visa Response: CCH   <b>MasterCard Response: NAN</b> |
| 5111 1110 5798 7356 | <b>Visa Response: NAN</b>   MasterCard Response: NED |
| 5111 1110 7450 2964 | Visa Response: ACL   <b>MasterCard Response: NED</b> |
| 5111 1110 6971 3154 | Visa Response: CUR   <b>MasterCard Response: NED</b> |
| 5111 1110 2030 4416 | Visa Response: NUP   <b>MasterCard Response: NED</b> |
| 5111 1110 4733 5823 | Visa Response: CCH   <b>MasterCard Response: NED</b> |
| 5111 1110 3135 3600 | <b>Visa Response: NAN</b>   MasterCard Response: ACL |
| 5111 1110 4816 3604 | <b>Visa Response: ACL</b>   MasterCard Response: ACL |
| 5111 1110 1867 3020 | Visa Response: CUR   <b>MasterCard Response: ACL</b> |
| 5111 1110 3056 0627 | Visa Response: NUP   <b>MasterCard Response: ACL</b> |

**Table 25    MasterCard Card Test Numbers (Continued)**

| <b>Card Number</b>  | <b>Response</b>                                      |
|---------------------|--|
| 5111 1110 0270 8865 | Visa Response: CCH   <b>MasterCard Response: ACL</b> |
| 5111 1110 6646 9396 | <b>Visa Response: NAN</b>   MasterCard Response: CUR |
| 5111 1110 5787 1816 | <b>Visa Response: ACL</b>   MasterCard Response: CUR |
| 5111 1110 7350 8855 | <b>Visa Response: CCH</b>   MasterCard Response: CUR |



# Sample Java Code for Uploading Files

## Requirements

---

- J2SE 1.5 or later.
- Unlimited Strength Jurisdiction Policy files from Oracle (*US\_export\_policy.jar* and *local\_policy.jar*):  
<http://www.oracle.com/technetwork/java/javase/documentation/index.html>
- Bouncy Castle, which includes *bcmail\*.jar*, *bcpkg\*.jar*, *bcprov\*.jar*, and *bctest\*.jar*.  
[www.bouncycastle.org](http://www.bouncycastle.org)
- Click [here](#) to view Cybersource's sample code package.

## Using the Sample Code

---

**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:
- a** Find your existing *US\_export\_policy.jar* and *local\_policy.jar* files in the `$JAVA_HOME/jre/lib/security` directory.
  - b** Rename or move your existing files to another directory.
  - c** Copy the new *US\_export\_policy.jar* and *local\_policy.jar* files that you downloaded from Oracle to the `$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 immediately after the Oracle provider. Be sure to increment the numbers of the other providers in the list.

Insert this line:

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

Your list of security providers will now look like this:

---

```
security.provider.1=sun.security.provider.Sun
security.provider.2=org.bouncycastle.jce.provider.BouncyCastlePr
ovider
security.provider.3=com.sun.net.ssl.internal.ssl.Provider
security.provider.4=com.sun.rsajca.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 Internet Explorer, and choose **Tools > Internet Options**.
- b** Click the **Content** tab.
- c** Click **Certificates**.
- d** Click **Import** to open the Certificate Import Wizard, and click **Next** to start the Wizard.
- e** Browse to the location of your .p12 security key, and click **Next**.
- f** 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.
- g** On this page, check the box for **Mark this key as exportable**, and click **Next**.
- h** Click **Next** on the Certificate Store page.
- i** Click **Finish**. A confirmation message appears indicating that the import was successful.

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

- a** Browse to one of the following URLs:
  - If the system is in test mode and is not live with CyberSource Account Updater:  
<https://accountupdater.test.cybersource.com/upload/UploadAccountUpdaterFile>
  - If the system is live with CyberSource Account Updater:  
<https://accountupdater.cybersource.com/upload/UploadAccountUpdaterFile>

- b** Choose **File > Properties**.
- c** Click **Certificates**.
- d** Click the **Certification Path** tab.
- e** Click **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 a location to save the file.
- j** Enter a name for the file, such as *MyCert*. Click **Save** and click **Next**.
- k** Click **Finish**.

Your file (*MyCert.cer*) has been created in the location you specified.

- l** Go to the `$JAVA_HOME/bin/keytool` file and use the J2SE keytool program to create a keystore file that contains this newly created certificate. You must provide a pass phrase for the keystore. You MUST use the same password that you used in [Step 5](#). For example, if your p12 key is *infodev.p12*, the pass phrase must be *infodev*.

To create the keystore, enter this 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>
```

#### **Example     Request: Creating the Keystore**

---

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

---

The output looks like this example:

#### Example 15 Response: Creating the Keystore

---

```

Owner: CN=accountupdatertest.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 example:

#### Example 16 Modifying the SSLFileTransfer.props File

---

```

# Upload host
host=accountupdatertest.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 the file (provided by CyberSource)
path=/upload/UploadAccountUpdaterFile
# 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 the location in which you installed J2SE.

#### Example 17 Java Home Environment

---

```
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 containing the CyberSource sample files.

- b** Enter the following:

```
javac SSLFileTransfer.java
```

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

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

**Example 18 Upload Response**

---

```
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
```

---