Account Updater

User Guide

February 2020

CyberSource®

the power of payment

CyberSource Corporation HQ | P.O. Box 8999 | San Francisco, CA 94128-8999 | Phone: 800-530-9095

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

Copyright

© 2020 CyberSource Corporation. All rights reserved. CyberSource Corporation ("CyberSource") furnishes this document and the software described in this document under the applicable agreement between the reader of this document ("You") and CyberSource ("Agreement"). You may use this document and/or software only in accordance with the terms of the Agreement. Except as expressly set forth in the Agreement, the information contained in this document is subject to change without notice and therefore should not be interpreted in any way as a guarantee or warranty by CyberSource. CyberSource assumes no responsibility or liability for any errors that may appear in this document. The copyrighted software that accompanies this document is licensed to You for use only in strict accordance with the Agreement. You should read the Agreement carefully before using the software. Except as permitted by the Agreement, You may not reproduce any part of this document, store this document in a retrieval system, or transmit this document, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written consent of CyberSource.

Restricted Rights Legends

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

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 Unpublished rights reserved under the copyright laws of the United States.

Trademarks

Authorize.Net, eCheck.Net, and The Power of Payment are registered trademarks of CyberSource Corporation.

CyberSource, CyberSource Payment Manager, CyberSource Risk Manager, CyberSource Decision Manager, and CyberSource Connect are trademarks and/or service marks of CyberSource Corporation.

All other brands and product names are trademarks or registered trademarks of their respective owners.

Contents

Recent Revisions to This Document

Release	Changes
February 2020	Updated "Related Documents," on page 7. Added "American Express Registry Status Response," on page 21. Updated "American Express Daily Updates," on page 23. Updated "Viewing the Batch File Status," on page 32, and "Downloading a Response File," on page 32. Updated "American Express Test Card Numbers," on page 38. Added the **additionalUpdates** reply array. See additionalUpdates, page 46. Updated "Response Codes and Reason Codes," on page 47.
June 2019	Updated the **Old Card Number** and **New Card Number** fields. See "Detail Record," page 34. Updated the formatting of the BIN `35 8627 6236` test card number.
May 2019	Updated the example for creating a batch of tokens. See Example 1, "Creating a Batch of Two Customer or Payment Instrument Tokens," on page 14.
March 2019	Updated test and live endpoints. See "Batches Resource," page 13, "Visa and Mastercard One-Off Reports," page 16, and "Daily American Express and Harvest Update Reports," page 18.
January 2019	Updated the audience information. See "Audience," page 6. Updated the list of related documents. See "Related Documents," page 7. Added support for Token Management Service (TMS) throughout. Added support for American Express Cardrefresher updates throughout. Updated the integration options. See "Options," page 8. Updated the enrollment information. See "Enabling Account Updater," page 10. Replaced the REST API Batch Upload and Harvest Updates chapters with the Token Updates chapter. See Chapter 2, "Token Updates," on page 12. Added the REST API field names appendix. See Appendix A, "REST API Fields," on page 40.
March 2018	Updated maximum file size value for Account Updater request files. See "Formatting a Request File," page 28.

About This Guide

Audience

This guide is written for merchants and partners who want to keep stored card data updated for recurring payments and credentials-on-file payments. You can store card data within your system or within the CyberSource tokenization system, which includes Recurring Billing, Token Management Service, and the legacy Payment Tokenization.

Purpose

This guide describes tasks that a merchant or partner must complete in order to submit a batch of tokens using the REST API (see Chapter 2, "Token Updates," on page 12) or in order to upload request files with new customer primary account numbers (PANs) (see Chapter 3, "PAN Updates," on page 27). It is intended to help the merchant or partner reduce the number of authorization declines to retain revenue and reduce the cost of manually updating payment data.

Conventions

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.

Text and Command Conventions

Convention	Usage
bold	Field and service names in text; for example: Include the **merchantReference** field.
`screen text`	■ XML elements ■ Code examples ■ Values for API fields; for example: Set the **type** to `amexRegistration`.

Related Documents

- *Getting Started with CyberSource Advanced for the Simple Order API* (PDF | HTML) or *Getting Started with CyberSource Advanced for the SCMP API* (PDF | HTML) describes how to get started using your CyberSource account.
- Business Center Online Help describes the features and options available with your CyberSource account in the Business Center.
- *Business Center Reporting User Guide* (PDF | HTML)
- *Recurring Billing Using the Business Center* (PDF | HTML) describes how to create and use customer recurring subscriptions.
- *Creating and Using Security Keys* (PDF | HTML) describes how to create and update security keys.
- Secure File Share API
- *Token Management Service Using the Simple Order API* (PDF | HTML)
- *Token Management Service Using the SCMP API* (PDF | HTML) describes how to use the Token Management Service.
- Token Management Service using REST APIs

Refer to the Support Center for more CyberSource technical documentation:

https://www.cybersource.com/en-us/support/technical-documentation.html

Customer Support

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

http://www.cybersource.com/support/s/

CHAPTER 1

Introduction

Account Updater helps you to keep stored card data up to date so that you can improve authorization success rates by cutting down on declined payments related to lost, stolen, or expired cards. CyberSource supports updates for card data stored on your servers or in the CyberSource tokenization system, which includes Recurring Billing, Token Management Service (TMS), and the legacy Payment Tokenization. The updates include expiration dates, credit card numbers, and brands.

CyberSource provides a single interface to access updates from the Visa Account Updater and Mastercard Automatic Billing Updater services. If you are using the Token Management Service, CyberSource also provides updates to you from the American Express Cardrefresher service.

Options

Integration options are available depending on whether you are:

- Using TMS or Recurring Billing for tokenization. See "Tokenization Merchants," page 8.
- Storing PANs on your system. See "Merchants Storing PANs," page 9.

Tokenization Merchants

If you are already using TMS or Recurring Billing, Account Updater is simple to integrate. You will benefit from updates from Visa, Mastercard, and American Express.

Important If you are using tokens that preserve the last four digits, the new PAN updates that you receive from card networks result in the token (subscription ID) being changed. When you receive the new tokens in the update report, you should update them in your system immediately to avoid authorization failures.

To use this service you need REST API keys generated in the key management section of the Business Center. Contact your CyberSource representative if you do not have this option enabled for your account. Find more details on authenticating API requests on the Developer Center.

Integration Options for Tokenization Merchants

Selective Updates

Using the Account Updater REST API, add the specific tokens (also known as subscriptions) that you wish to batch update. For Visa and Mastercard, these batches produce one-off update reports.

For tokens containing American Express cards, card numbers are enrolled for automatic updates for which reports are generated daily. Tokens are removed automatically when deleted or updated to a different card type. See "Batch Update," page 13.

Harvest Updates

You can configure Account Updater to automatically update all of your tokens with the latest credit card data. For Visa and Mastercard, update reports are generated monthly. For American Express, update reports are generated daily. See "Options," page 8.

Merchants Storing PANs

If you directly manage customer card data, you create a file containing PANs that 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 Chapter 3, "PAN Updates," on page 27.

Enabling Account Updater

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

For American Express Cardrefresher, contact your American Express representative to ensure that your organization is enabled for Cardrefresher using your existing American Express credentials.

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 Chapter 3, "PAN Updates," on page 27), 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 following actions. If you are an administrator, you already have these permissions.

- View the status of a request file.
- Tokenization merchants: create CyberSource REST API credentials. See "Batch Update," page 13.
- PAN upload merchants: add and activate a PGP Security Key for PAN upload updates. See Chapter 3, "PAN Updates," on page 27.
- 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, American Express rules and regulations, and all other applicable rules and regulations issued by any card association.

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 biannually.
  - 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 review 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.

CHAPTER 2

Token Updates

You can arrange for CyberSource to harvest and update all of your tokens on an agreed-upon date. You must retrieve a monthly report for Visa and Mastercard updates and a daily update report if you are enrolled for American Express Cardrefresher.

Note Account Updater requires card number and expiration dates, so the harvest option is available only when you use the *customer* or *payment instrument* tokens. *Instrument identifier* tokens that do not belong to a *customer* or *payment instrument* token are not updated.

The Account Updater REST API enables you to selectively POST a batch of tokens (subscription IDs) to the Account Updater service to be enrolled (American Express only), processed, and updated.

Both options use the standard CyberSource REST API authentication methods:

- JSON web token
- HTTP signature

For information about REST API authentication methods, see the *Developer Center*.

Token Harvest Option

On an agreed-upon monthly date, your tokenized cards are submitted to Visa and Mastercard for updates. Any tokens containing American Express cards are automatically enabled for Cardrefresher daily, and deleted or updated tokens are de-enrolled automatically.

You must retrieve American Express reports daily and/or Visa and Mastercard reports monthly. See the "Retrieving Update Reports," page 16, for more details.

Note It is best practice to request updates for your tokens 3 to 5 days before your billing cycle begins. You can choose any calendar day, from the 1st through the 28th.

Batch Update

Batches Resource

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

- Test endpoint: https://apitest.cybersource.com/accountupdater/v1/batches
- Live endpoint: https://api.cybersource.com/accountupdater/v1/batches

Submitting Visa and Mastercard One-Off Updates

Tokens can be submitted for a one-off update to Visa and Mastercard. Your update report is generated in 24 to 48 hours. A successful response to the batch creation returns a batch ID. You can check the status of the batch, which returns the URI of the batch update report when available.

Set the **type** field to `oneOff` in order to perform this type of update.

Registering Tokens for American Express Daily Updates

You register tokens containing American Express cards for daily updates. CyberSource receives updates from American Express daily, applies them to your tokens, and produces a daily report that is available to you through the REST API.

To indicate that the batch contains tokens to be enrolled with American Express Cardrefresher, set the **type** to `amexRegistration`.

Batch Creation Request Examples

TMS supports different types of tokens:

- Customer
- Payment instrument
- Instrument identifier

Customer tokens and *payment instrument* tokens store the expiration date in addition to the PAN. *Instrument identifier* tokens store only the PAN.

Note

Each batch request should contain only one token type: *customer*, *payment instrument*, or *instrument identifier*.

For more information about TMS tokens, see *Token Management Service Using the Simple Order API* (PDF | HTML), *Token Management Service Using the SCMP API* (PDF | HTML), or Token Management Service on the Developer Center.

Account Updater requires the existing PAN and expiration date. If you are using *instrument identifier* tokens, you must also to specify the expiration date.

Example 1 Creating a Batch of Two Customer or Payment Instrument Tokens

```
{
  "type": "oneOff",
  "included": {
    "tokens": [
      {
        "id": "3FA02EB4E49B65FDA194B38994B1F3F3"
      },
      {
        "id": "D1944BD9A7F9052BE431A276EB492C39"
      }
    ]
  },
  "merchantReference": "Merchant reference",
  "notificationEmail": "email@example.com"
}
```

Example 2 Creating a Batch of Two Instrument Identifier Tokens

```
{
  "type": "amexRegistration",
  "included": {
    "tokens": [
      {
        "id": "7B1F41664F08F6DD3BB1C63892907524",
        "expirationMonth": "12",
        "expirationYear": "2018"
      },
      {
        "id": "E8F44CFA7EBEADDB06A5A9625E7F8696",
        "expirationMonth": "12",
        "expirationYear": "2018"
      }
    ]
  },
  "merchantReference": "Merchant reference",
  "notificationEmail": "email@example.com"
}
```

Batch Creation Response Examples

Example 3 HTTP 202 – Successful Batch Creation

```
{
   "_links": {
      "self": {
      "href": "https://api.cybersource.com/accountupdater/v1/batches"
   },
   "status": {
      "href": "https://api.cybersource.com/accountupdater/v1/batches/
15269996945240002139594385/status"
   }
   },
   "batchId": "15269996945240002139594385",
   "batchItemCount": 2
}
```

Example 4 HTTP 401 – Not authorized to access resource.

```
{
   "_links": {
      "self": {
         "href": "https://api.cybersource.com/accountupdater/v1/batches"
      }
   },
   "code": "FORBIDDEN_RESPONSE",
   "correlationId": "c7b74452a7314f9ca28197d1084447a5",
   "detail": "You are not authorized to access this resource",
   "fields": null,
   "localizationKey": "cybsapi.forbidden.response",
   "message": "Unauthorized Access"
}
```

Action: Verify that the credentials that you are using are correct for the environment you are accessing. Ensure that your credentials have not expired and that your authentication process is correct.

Example 5 422 – Failure to process request

```
{
   "_links": {
      "self": {
         "href": "https://api.cybersource.com/accountupdater/v1/batches"
      }
   },
   "code": "VALIDATION_ERROR",
   "correlationId": "c7b74452a7314f9ca28197d1084447a5",
   "detail": "One or more fields failed validation",
   "fields": [
      {
         "path": "notificationEmail",
         "message": "Email address provided should not be 'null'",
         "localizationKey": "cybsapi.ondemand.batch.email.null"
      }
   ],
   "localizationKey": "cybsapi.validation.error",
   "message": "Field validation error"
}
```

Action: Examine the message to learn what failed validation. Verify that the structure of your JSON format is correct.

Retrieving Update Reports

The update reports contain details of updates that have been applied to the token and include a masked version of new card numbers and/or expiration dates.

To retrieve the batch, obtain the batch ID. The process for retrieving the batch depends on how the batch was created.

Visa and Mastercard One-Off Reports

One-off updates are retrieved by checking the batch status URL that was returned in the one-off batch creation process. See "Submitting Visa and Mastercard One-Off Updates," page 13. An authenticated GET on the following resources returns the status of the batch:

- Test endpoint:
  https://apitest.cybersource.com/accountupdater/v1/batches/{batchId}/status

- Live endpoint:
  https://api.cybersource.com/accountupdater/v1/batches/{batchId}/status

Response Example

The processing of a batch by Visa and Mastercard can take up to 48 hours; therefore, reports are not available immediately. A successful response returns the status of the batch and additional information relating to the batch as it becomes available.

The following batch statuses are possible:

Table 1 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 received a response from the card association(s) and is updating the tokens.
Complete	Updates have been applied to the tokens. The batch report URL is now available.
Failed	*Review specific error message.*

Note

Not all data is available immediately. As the batch status progresses from *Received* through *Processing* and *Updating* to *Complete*, additional data becomes available in the batch status. Check the status after submitting the batch to catch early errors that might result in a *Failed* status or incorrect **acceptedRecords** or **rejectedRecords** counts. The URL of the batch report appears when the status is *Complete*.

Example 6 HTTP 200 – Successful Response

```
{
   "_links": {
      "self": {
         "href": "https://api.cybersource.com/accountupdater/v1/batches/
15269996945240002139594385/status"
      },
      "report": [
         {
         "href": "https://api.cybersource.com/accountupdater/v1/batches/
15269996945240002139594385/report"
         }
      ]
   },
   "batchCaEndpoints": "VISA,MASTERCARD",
   "batchCreatedDate": "2018-05-22T14.38.57Z",
   "batchId": "15269996945240002139594385",
   "batchSource": "TOKEN_API",
   "billing": {
      "nan": "0,",
      "ned": "9,",
      "acl": "5,",
      "cch": 0
   },
   "description": "Batch processing complete. Report URL now available.",
   "merchantReference": "Merchant reference",
   "status": "COMPLETED",
   "totals": {
      "acceptedRecords": "8,",
      "rejectedRecords": "7,",
      "updatedRecords": "8,",
      "caResponses": "14,",
      "caResponsesOmitted": 6
   }
}
```

Daily American Express and Harvest Update Reports

American Express update reports are generated daily, so the batch ID is not known in advance.

Similarly, harvest updates are scheduled by the Account Updater service on a date you agree upon with your CyberSource account representative.

For both integration options, the first step is to retrieve the batch ID itself by sending an authenticated GET request on the following resources:

- Test endpoint: https://apitest.cybersource.com/accountupdater/v1/batches
- Live endpoint: https://api.cybersource.com/accountupdater/v1/batches

This step returns an array of batches. Paging is supported with offset and limit query parameters. For example, to return the second page of results with 50 per page you need to send GET /v1/batches?offset=1&limit=50.

Example 7 HTTP 200 – Successful Response

```
{
   "_links": [
      {
         "rel": "self",
         "href": "https://apitest.cybersource.com/accountupdater/v1/
batches?offset=0&limit=1"
      },
      {
         "rel": "first",
         "href": "https://apitest.cybersource.com/accountupdater/v1/
batches?offset=0&limit=1"
      },
      {
         "rel": "next",
          "href": "https://apitest.cybersource.com/accountupdater/v1/
batches?offset=1&limit=1"
       },
       {
          "rel": "last",
          "href": "https://apitest.cybersource.com/accountupdater/v1/
batches?offset=114&limit=1"
       }
   ],
   "object": "collection",
   "offset": 0,
   "limit": 3,
   "count": 1,
   "total": 3,
   "_embedded": {
       "batches": [
          {
              "_links": {
               "reports": [
                   {
                       "href": "https://apitest.cybersource.com/
accountupdater/v1/batches/15416031479410002099212314/report"
                     }
              ]
          },
          "batchId": "15416031479410002099212314",
          "batchCreatedDate": "2018-11-07T07:05:48Z",
          "batchModifiedDate": "2018-11-07T07:05:50Z",
          "batchSource": "SCHEDULER",
          "tokenSource": "TMS",
          "merchantReference": "Merchant Name",
          "batchCaEndpoints": [
              "VISA",
              "MASTERCARD"
          ],
          "status": "COMPLETE",
          "totals": {
              "acceptedRecords": 1,
              "rejectedRecords": 0,
              "updatedRecords": 1,
              "caResponses": 1,
              "caResponsesOmitted": 0
              }
          },
       {
          "_links": {
              "reports": [
                  {
                      "href": "https://apitest.cybersource.com/
accountupdater/v1/batches/15416025010730001655343827/report"
                      }
                  ]
              },
              "batchId": "15416025010730001655343827",
              "batchCreatedDate": "2018-11-07T06:55:01Z",
              "batchModifiedDate": "2018-11-07T06:56:52Z",
              "batchSource": "AMEX_REGISTRY_API",
              "tokenSource": "TMS",
              "batchCaEndpoints": [
                  "AMEX"
              ],
              "status": "COMPLETE"
          },
       {
          "_links": {
              "reports": [
                  {
                      "href": "https://apitest.cybersource.com/
accountupdater/v1/batches/15416025010730001655343827/report"
                  }
              ]
             },
       "batchId": "15402221273070001683984545",
       "batchCreatedDate": "2018-10-22T08:28:47Z",
       "batchModifiedDate": "2018-10-22T08:29:19Z",
       "batchSource": "AMEX_MAINTENANCE",
       "tokenSource": "TMS",
       "batchCaEndpoints": [
       "AMEX"
       ],
       "status": "COMPLETE",
       "totals": {
       "acceptedRecords": 0
       }
       }
       ]
   }
}
```

Batches are identified by the batch creation date (**batchCreatedDate**) and the batch method (**batchSource**) field values. The following table provides the **batchSource** possible values:

Table 2 Batch Methods

batchSource Value	Description
AMEX_REGISTRY_API	Batch for American Express token registration. American Express generates a report only when the registration batch contains errors.
AMEX_MAINTENANCE	Daily updates for tokens enrolled in the American Express Cardrefresher service.
TOKEN_API	Updates relating to a one-off request to Visa or Mastercard.
SCHEDULER	Updates relating to a monthly harvest of all tokens.

After you submit a batch for American Express token registration, you can access the batch status through the authenticated GET request using the URL returned in the submission. A successful response of the authenticated GET returns the status of the batch. See "Registering Tokens for American Express Daily Updates," page 13.

Example 8 American Express Registry Status Response

```
{
  "_links": {
    "self": {
        "href": "https://apitest.cybersource.com/accountupdater/v1/
batches/15816023535620001646854894/status"
      },
    "report": [
{
        "href": "https://apitest.cybersource.com/accountupdater/v1/
batches/15816023535620001646854894/report"
      }
    ]
  },
  "batchCaEndpoints": "AMEX",
  "batchCreatedDate": "2020-02-13T13.59.13Z",
  "batchId": "15816023535620001646854894",
  "batchSource": "AMEX_REGISTRY_API",
  "description": "Updates have been applied to your tokens. A batch report
is available.",
  "merchantReference": "Merchant Name",
  "status": "COMPLETE",
  "totals":
  {
    "acceptedRecords": 999,
    "rejectedRecords": 123,
    "updatedRecords": 0,
    "caResponses": 0,
    "caResponsesOmitted": 0
  }
}
```

Retrieving a Batch with a Batch ID

You can access an individual batch report through an authenticated GET request using the URL returned in the batch status or batches resource described in the previous sections.

HTTP 200 – Successful Response

Example 9 AMEX_REGISTRY_API Batch Method

```
{
    "version": "1.0",
    "reportCreatedDate": "2018-11-07T15:33:11Z",
    "batchId": "15416047164330001593314231",
    "batchSource": "AMEX_REGISTRY_API",
    "batchCaEndpoints": "AMEX",
    "batchCreatedDate": "2018-11-07T15:31:56Z",
    "merchantReference": "Merchant Name",
    "totals": {
        "acceptedRecords": 0,
        "rejectedRecords": 3
    },
    "records": [
        {
            "sourceRecord": {
                "token": "12345678901234567890",
                "cardExpiryMonth": "01",
                "cardExpiryYear": "2001"
            },
            "responseRecord": {
                "response": "DEC",
                "reason": "852"
            }
        },
        {
            "sourceRecord": {
                "token": "456",
                "cardExpiryMonth": "01",
                "cardExpiryYear": "2001"
            },
            "responseRecord": {
                "response": "DEC",
                "reason": "851"
            }
        },
        {
            "sourceRecord": {
                "token": "789",
                "cardExpiryMonth": "01",
                "cardExpiryYear": "2001"
            },
            "responseRecord": {
                "response": "DEC",
                "reason": "851"
            }
        }
    ]
}
```

American Express Daily Updates

Card numbers in TMS are represented by *instrument identifier* tokens. A card number and its associated *instrument identifier* token will be set to a CLOSED status when:

- The card network sends a direct account closed notification (response code ACL).
- A new card number is issued to replace a cancelled card (response code NAN).

Account Updater updates *customer* and *payment instrument* tokens only when you specify them in the request. When you specify a *customer* token for update or harvest, only the customer's default *payment instrument* token is updated. When you do not specify the *customer* and *payment instrument* tokens, they can become associated with a closed *instrument identifier* token in the update batch or harvest. These results are detailed in the additionalUpdates section of the update report. To update *customer* tokens and *payment instrument* tokens, include them in a subsequent Account Updater batch API request, or send a direct call to the TMS REST API. See Token Management Service on the Developer Center.

Example 10 AMEX_MAINTENANCE Batch Method

```
{
  "version": "1.0",
  "reportCreatedDate": "2020-01-23T11:16:13Z",
  "batchId": "15797780137010000506182090",
  "batchSource": "AMEX_MAINTENANCE",
  "batchCaEndpoints": "AMEX",
  "batchCreatedDate": "2020-01-23T11:13:33Z",
  "totals": {
    "updatedRecords": 3,
    "rejectedRecords": 0,
    "caResponses": 3,
    "caResponsesOmitted": 0
  },
  "billing": {
    "nan": 1,
    "ned": 1,
    "acl": 1,
    "cch": 0
  },
  "records": [
    {
      "id": "562239661",
      "sourceRecord": {
        "token": "9CCD3AE24DD9E254E0533F36CF0A356E",
        "cardNumber": "371449XXXXX2009",
        "cardExpiryMonth": "02",
        "cardExpiryYear": "2020",
        "cardType": "003",
        "customerId": "9CCD3AE24DD9E254E0533F36CF0A356E",
        "paymentInstrumentId": "9CCD3AE24DD8E254E0533F36CF0A356E",
        "instrumentIdentifierId": "9CCD3AE24DD7E254E0533F36CF0A356E"
      },
      "responseRecord": {
        "response": "NAN",
        "reason": "800",
        "token": "9CCD3AE24DD9E254E0533F36CF0A356E",
        "cardNumber": "371449XXXXX0102",
        "cardType": "003",
        "instrumentIdentifierId": "9CCDC4D08BE0C16BE0533F36CF0A9916",
        "instrumentIdentifierCreated": "true",
        "cardExpiryMonth": "07",
        "cardExpiryYear": "2020",
        "additionalUpdates": [
          {
            "customerId": "8CCD3AE24DD8E254E0533F36CF0A355E",
            "paymentInstrumentId": "9CCD3AE24DD8E254E0533F36CF0A356D",
            "creator": "aura_regress_tms_report",
            "state": "CLOSED",
            "message": "This Payment Instrument contains the source card
number, which is now closed. If required, you can update manually or
through the AU REST API."
          }
        ]
      }
    },
  {
    "id": "562239711",
    "sourceRecord": {
      "token": "9CCD3AE24DF7E254E0533F36CF0A356E",
      "cardNumber": "371449XXXXX1100",
      "cardExpiryMonth": "02",
      "cardExpiryYear": "2020",
      "cardType": "003",
      "customerId": "9CCD3AE24DF7E254E0533F36CF0A356E",
      "paymentInstrumentId": "9CCD3AE24DF6E254E0533F36CF0A356E",
      "instrumentIdentifierId": "9CCD3AE24DF5E254E0533F36CF0A356E"
    },
    "responseRecord": {
      "response": "NED",
      "reason": "800",
      "cardExpiryMonth": "12",
      "cardExpiryYear": "2021"
      }
    },
  {
    "id": "562239751",
    "sourceRecord": {
      "token": "9CCD3AE24E0FE254E0533F36CF0A356E",
      "cardNumber": "371449XXXXX1226",
      "cardExpiryMonth": "02",
      "cardExpiryYear": "2020",
      "cardType": "003",
      "customerId": "9CCD3AE24E0FE254E0533F36CF0A356E",
      "paymentInstrumentId": "9CCD3AE24E0EE254E0533F36CF0A356E",
      "instrumentIdentifierId": "9CCD3AE24E0DE254E0533F36CF0A356E"
    },
    "responseRecord": {
      "response": "ACL",
      "reason": "800",
      "additionalUpdates": [
        {
          "customerId": "7CCD3AE24DD8E254E0533F36CF0A356A",
          "paymentInstrumentId": "9CCD3AE24E0EE254E0533F36CF0A356D",
          "creator": "aura_regress_tms_report",
          "state": "CLOSED",
          "message": "This Payment Instrument contains the source card
number, which is now closed. If required, you can update manually or
through the AU REST API."
          }
        ]
      }
    }
  ]
}
```

Example 11 TOKEN_API Batch Method & Scheduler

```
{
    "version": "1.0",
    "reportCreatedDate": "2018-11-01T14:43:36Z",
    "batchId": "15410833473400000123332450",
    "batchSource": "SCHEDULER",
    "batchCaEndpoints": "VISA,MASTERCARD",
    "batchCreatedDate": "2018-11-01T14:42:27Z",
    "merchantReference": "Merchant Name",
    "totals": {
        "acceptedRecords": 2,
        "caResponses": 3,
        "rejectedRecords": 0,
        "updatedRecords": 2,
        "caResponsesOmitted": 1
    },
    "billing": {
        "nan": 1,
        "ned": 0,
        "acl": 1,
        "cch": 0
    },
    "records": [
        {
            "id": "4451434614",
            "sourceRecord": {
                "token": "4682345889876532701018",
                "cardNumber": "511111XXXXXX3604",
                "cardExpiryMonth": "09",
                "cardExpiryYear": "21",
                "cardType": "002"
            },
            "responseRecord": {
                "response": "ACL",
                "reason": "800"
            }
       },
       {
          "id": "784311",
          "sourceRecord": {
          "token": "7020000000014008934",
          "cardNumber": "371000XXXXXX8115",
          "cardExpiryMonth": "01",
          "cardExpiryYear": "2016",
          "cardType": "003",
          "instrumentIdentifierId": "7020000000014008115"
          },
          "responseRecord": {
          "response": "NAN",
          "reason": "800",
          "token": "7020000000012513358",
          "cardNumber": "401000XXXXXX2753",
          "cardType": "001",
          "instrumentIdentifierId": "7020000000012512753",
          "instrumentIdentifierCreated": "true",
          "cardExpiryMonth": "08",
          "cardExpiryYear": "2021"
          }
       }
    ]
}
```

Example 12 Batch Retrieval Error

```
{
   "_links": {
      "self": {
         "href": "https://api.cybersource.com/accountupdater/v1/batches/
15410833473400333245 0/report"
      }
   },
   "code": "FORBIDDEN_RESPONSE",
   "correlationId": "0386623ab0eb47dfae61d273032f8202",
   "detail": "You are not authorized to access this resource",
   "localizationKey": "cybsapi.forbidden.response",
   "message": "Unauthorized Access"
```

CHAPTER 3

PAN 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 to 48 hours after you submit your request file. CyberSource recommends that you send 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 33.

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* (PDF | HTML).

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* (PDF | HTML).

Formatting a Request File

Account Updater request files must be in CSV format with a maximum file size of 10 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 1 Header Record Fields

Field Name	Description	Required or Optional	Data Type (length|)
Record Identifier	Constant value indicating the record type. Format: H	Required	Alpha (1)
File Classification	Indicates whether this is a request or response file. Format: cybs.au.request.pan	Required	Alpha (30)
merchantID	Your CyberSource merchant ID. 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. Format: 12345	Required	Numeric (30)
recordCount	The number of detail records in the file. Format: 12345	Required	Numeric

Table 1 Header Record Fields (Continued)

Field Name	Description	Required or Optional	Data Type (length|)
statusEmail	Email address to which status emails for the request are sent. Format: aaa@aaa.aaa	Required	Alphanumeric (100)
creationDate	Optional field that you can pass for reference. If present, it appears in the Business Center Account Updater View Status window. Format: YYYY-MM-DD	Optional	(10)
Batch Info	Optional field that you can pass for reference. Format: sample12	Optional	Alphanumeric (50)

Detail Record

Each file must contain at least one detail record.

Table 2 Detail Record Fields

Field Name	Description	Required or Optional	Data Type (length)
Record Identifier	Constant value indicating the record type. Format: D	Required	Alpha (1)
Card Number	Card number to process. Format: Numeric	Required	Numeric (19)
Card Expiration Month	Expiration month of the card. Format: MM	Required	Alphanumeric (2)
Card Expiration Year	Expiration year of the card. 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 is returned in the Account Updater response file. Format: sampleID2	Optional	Alphanumeric (50)
BA Sub Merchant ID	This field is required for billing aggregator merchants only. Format: sampleID2	Optional	Alphanumeric (10)

Footer Record

Each file should contain only one footer record.

Table 3 Footer Record Field

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

Request File Examples

Example 1 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,1111222233334444,11,09,0001
D,2222333344445555,11,09,0002
F
```

Example 2 Request File for Billing Aggregator Merchants

```
H,cybs.au.request.pan,merchant1,001,2,notify@yourcompany.com,2009-03-23,My Jan Batch
D,1111222233334444,11,09,0001,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 Chapter 4, "Testing," on page 36.

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 Appendix C, "Sample Java Code for Uploading PANs," on page 50 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 32). CyberSource sends the email notification within 30 minutes of receiving the request file. However, actual timing depends on the system load when the file is submitted.

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

Table 4 Email Notifications

Subject Line	Reason
Received	The Account Updater request file was received. CyberSource processes the requests in the file. No action is required. You can view the status of this request file in the Business Center. See "Viewing the Batch File Status," page 32.
Rejected	The file was rejected. Read the contents of the email and follow the suggested remedy. You cannot view the status of this request file in the Business Center.
Validated	The file passed validation. You can view the status of this request file in the Business Center. See "Viewing the Batch File Status," page 32.
Declined	The file did not pass validation checks. All records are declined. Read the contents of the email and follow the suggested remedy. You can view the status of this request file in the Business Center. See "Viewing the Batch File Status," page 32.

Table 4 Email Notifications (Continued)

Subject Line	Reason
Processing	The request file is being processed by Account Updater. You can view the status of this request file in the Business Center. See "Viewing the Batch File Status," page 32.
Completed	The response file has been generated and is ready for download. You can view the status of this request file in the Business Center. See "Viewing the Batch File Status," page 32.

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://ebc2.cybersource.com/ebc2/
- Test transactions: https://ebctest.cybersource.com/ebc2/

Step 2 On the left navigation pane, click the **Tools** icon.

Step 3 Click **Account Updater**. The Account Updater page appears.

Step 4 Use the filters on the Search toolbar to locate the batch you want to view. The Search Results list shows matching results.

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 it programmatically, see Secure File Share API at the Developer Center.

To download a response file:

Step 1 Log in to the Business Center:

- Live transactions: https://ebc2.cybersource.com/ebc2/
- Test transactions: https://ebctest.cybersource.com/ebc2/

Step 2 On the left navigation pane, click the **Reports** icon.

Step 3 Under Downloadable Reports, click **Available Reports**. The Available Reports page appears.

Step 4 Click the **Third-Party Reports** tab. The Third-Party Reports page appears.

Step 5 In the **Download** column, click the file format link.

Only reports that have successfully completed generating and that contain data include links.

Step 6 Follow your browser's instructions to open and save the file.

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 5 Header Record Fields

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

Detail Record

Each file contains at least one detail record.

Table 6 Detail Record Fields

Field Name	Description	Data Type (length)
Record Identifier	Constant value indicating the record type. Format: D	Alpha (1)
Request ID	Unique CyberSource identifier for the record. Format: Numeric	Numeric (30)
Old Card Number	Old card number. Format: Numeric	Numeric (19)
Old Card Expiration Month	Old expiration month. Format: MM	Numeric (2)
Old Card Expiration Year	Old expiration year. Format: YY	Numeric (2)
New Card Number	New card number. Format: Numeric	Numeric (19)
New Card Expiration Month	New expiration month. Format: MM	Numeric (2)
New Card Expiration Year	New expiration year. Format: YY	Numeric (2)
Merchant Reference ID	This field is optional and is returned in the response if present in the request file. Format: Alphanumeric	Alphanumeric (50)
BA Sub Merchant ID	This field is returned in the response if sent in the request file. Format: Alphanumeric	Alphanumeric (10)
Response Code	Response code for the record. See Table 1, "Response Codes and Reason Codes," on page 47. Format: Alpha	Alpha (3)
Reason Code	Reason code for the record. See Table 1, "Response Codes and Reason Codes," on page 47. Format: Numeric	Numeric (3)

Footer Record

Each file contains only one footer record.

Table 7 Footer Record Fields

Field Name	Description	Data Type (length)
Record Identifier	Constant value indicating the record type. Format: F	Alpha (1)
Record Count	The number of detail records in the file. Format: Numeric	Numeric (10)
Response Code	Response code for the file. See Table 2, "Response Codes and Reason Codes," on page 49. Format: Alpha	Alpha (3)
Reason Code	Reason code for the file. See Table 2, "Response Codes and Reason Codes," on page 49. Format: Numeric	Numeric (3)

File Examples

Example 3 Non-Billing Aggregator Response File

```
H,cybs.au.response.pan,merchant1,001
D,1000000000000000001,1111222233334444,11,09,,,,0001,,NUP,800
D,1000000000000000002,2222333344445555,11,09,6666777788889999,11,11,0002,,NAN,800
F,2,COM,800
```

Example 4 Billing Aggregator Response File

```
H,cybs.au.response.pan,merchant1,001
D,1000000000000000001,1111222233334444,11,09,,,,0001,subId01,NUP,800
D,1000000000000000002,2222333344445555,11,09,6666777788889999,11,11,0002,subId02,NAN,
800
F,2,COM,800
```

CHAPTER 4

Testing

The CyberSource test environment provides a simulator in which the response from the card association can be triggered using card numbers listed in "Visa Test Card Numbers," page 36, "Mastercard Test Card Numbers," page 37, and "American Express Test Card Numbers," page 38. This simulator ensures that you can handle the possible response combinations when connecting to multiple card associations.

Note The test environment typically completes the process in a matter of minutes rather than the 24-hour (or longer) duration of the live environment when updates are sent to the actual card associations.

Visa Test Card Numbers

The bold fields represent the token updates for TMS, Recurring Billing, and Payment Tokenization merchants using the REST API batch update and harvest update. For a description of each response code, see Table 2, "Response Codes and Reason Codes," on page 49. Replace BIN with `400000` and remove spaces when sending to CyberSource.

Table 1 Visa Card Test Numbers

Card Number	Response Code
BIN `71 0951 9220`	Visa Response: NAN | **Mastercard Response: NAN**
BIN `15 3919 2096`	**Visa Response: NAN** | Mastercard Response: ACL
BIN `18 6481 0239`	**Visa Response: NAN** | Mastercard Response: CUR
BIN `91 9582 8465`	Visa Response: NED | **Mastercard Response: NAN**
BIN `27 5765 7455`	**Visa Response: NED** | Mastercard Response: ACL
BIN `71 1311 2087`	**Visa Response: NED** | Mastercard Response: CUR
BIN `21 1752 4874`	Visa Response: ACL | **Mastercard Response: NAN**
BIN `71 1629 4650`	Visa Response: ACL | **Mastercard Response: ACL**
BIN `20 5548 7183`	**Visa Response: ACL** | Mastercard Response: CUR
BIN `52 8063 4792`	Visa Response: CUR | **Mastercard Response: NAN**

Table 1 Visa Card Test Numbers (Continued)

Card Number	Response Code
BIN 24 0631 2635	Visa Response: CUR | **Mastercard Response: ACL**
BIN 89 2339 9344	Visa Response: CUR | **Mastercard Response: CUR**
BIN 55 7908 8940	Visa Response: NUP | **Mastercard Response: NAN**
BIN 57 9875 5634	Visa Response: NUP | **Mastercard Response: ACL**
BIN 80 9110 0706	Visa Response: CCH | **Mastercard Response: NAN**
BIN 26 9567 5155	Visa Response: CCH | **Mastercard Response: ACL**
BIN 35 8627 6236	**Visa Response: CCH** | Mastercard Response: CUR

Mastercard Test Card Numbers

The bold fields represent the token updates for TMS, Recurring Billing, and Payment Tokenization merchants using the REST API batch update and harvest update. For a description of each response code, see Table 2, "Response Codes and Reason Codes," on page 49. Replace BIN with 511111 and remove spaces when sending to CyberSource.

Table 2 Mastercard Card Test Numbers

Card Number	Response Code
BIN 10 4714 3086	**Visa Response: NAN** | Mastercard Response: NAN
BIN 10 2999 7178	Visa Response: ACL | **Mastercard Response: NAN**
BIN 10 1548 6814	Visa Response: CUR | **Mastercard Response: NAN**
BIN 10 5459 2548	Visa Response: NUP | **Mastercard Response: NAN**
BIN 10 4871 8571	Visa Response: CCH | **Mastercard Response: NAN**
BIN 10 5798 7356	**Visa Response: NAN** | Mastercard Response: NED
BIN 10 7450 2964	Visa Response: ACL | **Mastercard Response: NED**
BIN 10 6971 3154	Visa Response: CUR | **Mastercard Response: NED**
BIN 10 2030 4416	Visa Response: NUP | **Mastercard Response: NED**
BIN 10 4733 5823	Visa Response: CCH | **Mastercard Response: NED**
BIN 10 3135 3600	**Visa Response: NAN** | Mastercard Response: ACL
BIN 10 4816 3604	**Visa Response: ACL** | Mastercard Response: ACL
BIN 10 1867 3020	Visa Response: CUR | **Mastercard Response: ACL**
BIN 10 3056 0627	Visa Response: NUP | **Mastercard Response: ACL**
BIN 10 0270 8865	Visa Response: CCH | **Mastercard Response: ACL**
BIN 10 6646 9396	**Visa Response: NAN** | Mastercard Response: CUR

Table 2 Mastercard Card Test Numbers (Continued)

Card Number	Response Code
BIN 10 5787 1816	**Visa Response: ACL** | Mastercard Response: CUR
BIN 10 7350 8855	**Visa Response: CCH** | Mastercard Response: CUR

American Express Test Card Numbers

American Express card updates through the Cardrefresher are available only if you are using TMS. Use the numbers listed in the following tables to simulate various scenarios. Replace BIN with `371449` and remove spaces when sending to CyberSource.

Table 3 American Express Test Numbers

Card Number	Response Code
BIN 0 0002 0115	NAN (No New Expiry Date)
BIN 1 0211 2216	NAN (No New Expiry Date)
BIN 2 0121 2206	NAN (No New Expiry Date)
BIN 1 0021 1119	NAN (No New Expiry Date)
BIN 1 0101 0023	NAN (No New Expiry Date)
BIN 0 0100 2112	NAN (New Expiry Date)
BIN 1 2101 2009	NAN (New Expiry Date)
BIN 0 2201 2009	NAN (New Expiry Date)
BIN 2 1000 0113	NAN (New Expiry Date)
BIN 0 2100 0229	NAN (New Expiry Date)
BIN 2 2210 0224	NED
BIN 0 0112 0203	NED
BIN 0 2102 1100	NED
BIN 2 0121 2107	NED
BIN 0 1121 0119	NED
BIN 0 1022 1109	ACL
BIN 1 0112 1226	ACL
BIN 2 0201 0005	ACL
BIN 1 2121 0207	ACL
BIN 0 1012 2109	ACL
BIN 1 2120 0224	DEC
BIN 2 1010 0020	861 attempt to enroll customer already enrolled.

Table 3 American Express Test Numbers (Continued)

Card Number	Response Code
BIN `1 1202 1118`	862 registry rejected due to card member opt out
BIN `1 0122 0218`	ERR
BIN `0 0010 2004`	ERR
BIN `1 1111 2108`	861

APPENDIX

A

REST API Fields

Request Fields

Table 1 Request Fields

Field Name	Description	Used By & Required (R)/ Optional (O)	Validation
notificationEmail	Email address to which batch status updates are sent.	(R) POSTs to /batches	Valid email address
merchantReference	Your reference to identify the batch.	(O) POSTs to /batches	0 to 255 characters
type	Indicates whether batch is a one-off update for Visa and/or Mastercard, or an enrollment in American Express Cardrefresher. Possible values: ▪ `oneOff` (default): Visa or Mastercard ▪ `amexRegistration`: American Express	(O) POSTs to /batches	
included	Elements to be included. Must include one of the following: ▪ `tokens` ▪ `instrument_identifier`	(R) POSTs to /batches	
tokens	Comma-separated list of subscription IDs, Token Management Service (TMS) *customer* or *payment instrument* tokens.	(O) POSTs to /batches	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.
instrumentIdentifiers	Token Management Service (TMS) *instrument identifier* token assigned to the tokenized PAN and its associated expiration dates.	(O) POSTs to /batches	
id	ID for the *instrument identifier* token.	(R) POSTs to /batches	String (32)

Table 1 Request Fields (Continued)

Field Name	Description	Used By & Required (R)/ Optional (O)	Validation
expirationMonth	Two-digit month in which the card expires.	(R) POSTs to /batches	String (2)
expirationYear	Four-digit year in which the card expires.	(R) POSTs to /batches	String (4)

Reply Fields

Table 2 Reply Fields

Field Name	Description	Returned By	Data Type & Length
batchId	When the request is successful, a batch ID is returned to the user.	/batches /…/status /…/report	Alphanumeric (26)
batchItemCount	When the request is successful, this value is the number of items that were included in the request. When the request is unsuccessful, the value of this field is `0`.	/batches /…/status /…/report	Numeric (9)
_links	JSON object containing link elements relating to the request. Successful requests return the URI of the batch status.	/batches /…/status /…/report	
self	The resource address that was requested. Element within **_links**.	/batches /…/status /…/report	URL
first	First page in the result set.	/batches	URL
next	Next page in the result set.	/batches	URL
last	Last page in the result set.	/batches	URL
status	URI of the batch status resource. **Note** Do not hard-code the link to the batch status resource. Use the returned value to avoid errors if the URI structure changes.	/batches	URL
reports	URI of the batch associated with the **batchId**.	/batches /status	URL

Table 2 Reply Fields (Continued)

Field Name	Description	Returned By	Data Type & Length
correlationId	Returned when an error occurs. Provide this ID to Customer Support to help identify your transaction.	/batches /…/status /…/report	String (36)
code	HTTP Response code. Returned when an error occurs. See Appendix B, "Response Codes and Reason Codes," on page 47.	/batches /…/status /…/report	String (3)
detail	Returned when an error occurs. Detailed description of the error. See Appendix B, "Response Codes and Reason Codes," on page 47.	/batches /…/status /…/report	String (1024)
fields	Returned when an error occurs. The array contains elements that describe the erroneous fields. See Appendix B, "Response Codes and Reason Codes," on page 47.	/batches	
path	Returned when an error occurs. Element within the fields. Path of field name. See Appendix B, "Response Codes and Reason Codes," on page 47.	/batches	String (36)
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.	/batches	String (256)
localizationKey	Returned when an error occurs. A unique key that represents the error message and can be an element within fields. See Appendix B, "Response Codes and Reason Codes," on page 47. This field can also appear with the JSON object.	/batches	String (128)
version	Version of the report. 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. **Note** You always receive the latest minor version of the report for the API you used to create the batch.	/report	

Table 2 Reply Fields (Continued)

Field Name	Description	Returned By	Data Type & Length

batchSource	Method used to create the batch. For example, `TOKEN_API`.	/batches
		/status
		/report
			`TOKEN_API`
			`SCHEDULER`
			`AMEX_REGISTRY`
			`AMEX_MAINTENANCE`

batchCaEndpoints	Card associations to which the card numbers were sent.	/batches /status /report	Array containing one or more of the following: ■ `VISA` ■ `MASTERCARD` ■ `AMEX`
batchCreatedDate	Date on which the batch was created.	/batches /status /report	ISO_8601 UTC date
reportCreatedDate	Date on which the report was created.	/batches /status /report	ISO_8601 UTC date
merchantReference	Your reference, if present in the request.	/batches /status /report	0 to 255 characters
totals	JSON object containing the high-level summary of the batch.	/batches /status /report	
acceptedRecords	Number of tokens that were identified and retrieved for the merchant ID.	/batches /status /report	String (9)
rejectedRecords	Number of tokens that were not identified and retrieved.	/batches /status /report	String (9)
updatedRecords	Number of updates that were applied to a token.	/batches /status /report	String (9)

Table 2 Reply Fields (Continued)

Field Name	Description	Returned By	Data Type & Length
caResponses	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.	/batches /status /report	String (9)
caResponsesOmitted	Number of updates that were not applied to a token. For example, a response is returned by more than one card association.	/batches /status /report	String (9)
billing	JSON object containing the billing summary information.	/status /report	
nan/ned/acl/cch	Number of each billed response type.	/status /report	String (3)
records	JSON object containing additional objects that relate to the original tokens and the updates or errors that occurred.	/report	
id	CyberSource generated ID for the record.	/report	
sourceRecord	JSON object containing details from the source token.	/report	
token	Subscription ID included in the request.	/report	
cardNumber	Masked card number before an update. First six digits and the last four digits are not masked.	/report	
cardExpiryMonth	Two-digit month in which the card expires.	/report	
cardExpiryYear	Four-digit month in which the card expires.	/report	
cardType	Type of card. Possible values: `001`: Visa `002`: Mastercard `003`: American Express	/report	String (3)
customerId	Value of the *customer* token assigned to the tokenized shipping information and merchant defined data. **Note** This field is for TMS merchants only.	/report	

Table 2 Reply Fields (Continued)

Field Name	Description	Returned By	Data Type & Length
paymentInstrumentId	The value of the *payment instrument* token assigned to the tokenized billing information and card expiration dates. **Note** This field is for TMS merchants only.	/report	
instrumentIdentifierId	Value of the *instrument identifier* token assigned to the tokenized PAN. **Note** This field is for TMS merchants only.	/report	
responseRecord	JSON object containing the details that were made to the token.	/report	
response	Type of response. See Appendix B, "Response Codes and Reason Codes," on page 47.	/report	
reason	Reason code for the response. See Appendix B, "Response Codes and Reason Codes," on page 47.	/report	
token	If last-four-digit format-preserving tokens are used, a new token (subscription ID) can be returned that replaces the source record token.	/report	
cardNumber	Masked card number. First six and last four digits are not masked.	/report	
cardExpiryMonth	Two-digit month in which the card expires.	/report	
cardExpiryYear	Four-digit month in which the card expires.	/report	
cardType	Type of card. Possible values: `001`: Visa `002`: Mastercard `003`: American Express	/report	String (3)
instrumentIdentifierId	Value of the *instrument identifier* token assigned to the updated tokenized PAN. **Note** This field is for TMS merchants only.	/report	
instrumentIdentifierIdCreated	Indicates whether this is the first time the PAN has been tokenized for you. Possible values: ▪ `true` ▪ `false`	/report	String (5)

Table 2 Reply Fields (Continued)

Field Name	Description	Returned By	Data Type & Length
additionalUpdates	Details associated with a closed *instrument identifier* token.	/report	
customerId	Value of the *customer* token not present in the batch and associated with a closed *instrument identifier* token.	/report	String
paymentInstrumentId	Value of the *payment instrument* token not present in the batch and associated with a closed *instrument identifier* token.	/report	String
creator	Merchant that created the *payment instrument* token	/report	String
state	State of the token.	/report	“CLOSED”
message	Information about the tokens.	/report	String

APPENDIX

B

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 1 Details Record

```
D,1000000000000000002,2222333344445555,11,09,6666777788889999,11,11,0002,,NAN,800
```

Table 1 Response Codes and Reason Codes

Response Code	Response Code Description	Reason Code	Reason Code Description	Billable or Non-Billable Code
ACL	Match: account closed. **Note** The status of the customer subscription changes to *cancelled* 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	—	850	Invalid token format.	Non-billable.
DEC	—	851	Invalid token length.	Non-billable.

Table 1 Response Codes and Reason Codes (Continued)

Response Code	Response Code Description	Reason Code	Reason Code Description	Billable or Non-Billable Code
DEC	—	852	Unknown token. This token does not exist, is not associated with your account, or might be superseded.	Non-billable.
DEC	—	853	Invalid token status. This token has a status of CLOSED from a previous Account Updater batch.	Non-billable.
DEC	—	861	Cardmember is already enrolled or cannot cancel cardmember that is not enrolled.	Non-billable.
DEC	—	862	Rejected because cardmember opted out.	Non-billable.
ERR	—	801	Invalid card number.	Non-billable.
ERR	—	802	Invalid check digit.	Non-billable.
ERR	—	803	Invalid expiration date.	Non-billable.
ERR	—	804	Unsupported card type or cancelled card.	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 might 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 2 Footer Record

```
F,2,COM,800
```

Table 2 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.
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.

APPENDIX C

Sample Java Code for Uploading PANs

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*, *bcpg*.jar*, *bcprov*.jar*, and *bctest*.jar*:

  www.bouncycastle.org

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.BouncyCastleProv
ider
```

Your list of security providers should 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://accountupdatertest.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 **Detail**s 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 1 Response: Creating the Keystore

```
Owner: CN=accountupdatertest.cybersource.com, OU=Operations,
O=Cybersource Corporation, L=Foster City, 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 2018 until: Tue Jan 31 17:51:24
PST 2020
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 2 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 3 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 4 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
```