Account Updater

Developer Guide

cybersource
A Visa Solution

Version: 24.01

Contents

Account Updater

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

Audience and Purpose

This guide is written for merchants and partners who want to keep stored card data updated for recurring payments and credentials-on-file payments. It describes tasks that a merchant or partner must complete to submit a batch of tokens using the REST API or to upload request files with new customer primary account numbers (PANs). Account Updater is intended to help you reduce the number of authorization declines to retain revenue and reduce the cost of manually updating payment data. You can store card data within your system or within the tokenization system, which includes Recurring Billing, the Token Management Service, and the legacy Payment Tokenization.

Conventions

The following special statements are used in this document:

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

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

Related Documentation

Refer to the Technical Documentation Hub in the Cybersource Developer Center for additional technical documentation:

https://developer.cybersource.com/docs.html

Customer Support

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

http://support.cybersource.com

Recent Revisions to This Document

24.01

Added a note about instrument identifier token length. See Tokenization (on page 8).

23.01

Updated the REST API authentication method for token updates. See Token Updates (on page 12).

Added information about the GET request header to Visa and Mastercard One-Time Updates (on page 18), American Express Daily and Token Harvest Update Reports (on page 20), and Retrieving a Batch with a Batch ID (on page 25).

Added information about using the date query to American Express Daily and Token Harvest Update Reports (on page 20).

Added new test cards to American Express Test Card Numbers (on page 45).

VISA Platform Connect: Specifications and Conditions for Resellers/Partners

The following are specifications and conditions that apply to a Reseller/Partner enabling its merchants through Cybersource for Visa Platform Connect ("VPC") processing. Failure to meet any of the specifications and conditions below is subject to the liability provisions and indemnification obligations under Reseller/Partner's contract with Visa/Cybersource.

1. Before boarding merchants for payment processing on a VPC acquirer's connection, Reseller/Partner and the VPC acquirer must have a contract or other legal agreement that permits Reseller/Partner to enable its merchants to process payments with the acquirer through the dedicated VPC connection and/or traditional connection with such VPC acquirer.

2. Reseller/Partner is responsible for boarding and enabling its merchants in accordance with the terms of the contract or other legal agreement with the relevant VPC acquirer.

3. Reseller/Partner acknowledges and agrees that all considerations and fees associated with chargebacks, interchange downgrades, settlement issues, funding delays, and other processing related activities are strictly between Reseller and the relevant VPC acquirer.

4. Reseller/Partner acknowledges and agrees that the relevant VPC acquirer is responsible for payment processing issues, including but not limited to, transaction declines by network/issuer, decline rates, and interchange qualification, as may be agreed to or outlined in the contract or other legal agreement between Reseller/Partner and such VPC acquirer.

DISCLAIMER: NEITHER VISA NOR CYBERSOURCE WILL BE RESPONSIBLE OR LIABLE FOR ANY ERRORS OR OMISSIONS BY THE VISA PLATFORM CONNECT ACQUIRER IN PROCESSING TRANSACTIONS. NEITHER VISA NOR CYBERSOURCE WILL BE RESPONSIBLE OR LIABLE FOR RESELLER/PARTNER BOARDING MERCHANTS OR ENABLING MERCHANT PROCESSING IN VIOLATION OF THE TERMS AND CONDITIONS IMPOSED BY THE RELEVANT VISA PLATFORM CONNECT ACQUIRER.

Introduction to Account Updater

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. Account Updater updates 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, card numbers, and brands.

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

Integration options are available depending on whether you are:

- Using TMS or Recurring Billing for tokenization. See Tokenization (on page 8).
- Storing PANs on your system. See Storing PANs (on page 9).

Tokenization

If you are already using TMS or Recurring Billing, Account Updater is simple to integrate. You will benefit from token 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.

Important: Account Updater is incompatible with 22-digit instrument identifier tokens. For more information about configuring the format of instrument identifier tokens in your token vault, see the Token Vault Management section of the *Token Management Service Developer Guide*.

To use this service you must generate the REST API keys 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.

Token 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 Token Harvest (on page 12).

Token Selective Batch Updates

Using the Account Updater REST API, you can add the specific tokens (also known as subscriptions) that you wish to batch update. For Visa and Mastercard, these batches produce one-time 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 Token Batch Updates (on page 13).

Storing PANs

When you directly manage customer card data, you create a file containing PANs that Account Updater 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 Updates (on page 31).

Enabling Account Updater

Contact your account representative who submits enrollment forms on your behalf to 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 use Account Updater for PAN upload updates. In the Account Updater request files, they must indicate 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 requests. For more information about PAN upload updates, see PAN Updates (on page 31).

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.
- Create REST API credentials for token batch updates.
- For PAN updates, add and activate a PGP Security Key.
- Access downloadable response files.

Terms of Use

By using the 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 notifications.

You may not:

- Request updates on accounts that have returned a response of Contact Card Holder (CCH). 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.

Token Updates

You can arrange for Account Updater to harvest and update all of your tokens on an agreed-upon date; this is called the *token harvest update* method. You must retrieve a monthly report for Visa and Mastercard updates and a daily update report if you are enrolled for American Express Cardrefresher.

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

The Account Updater REST API enables you to selectively send a POST request for a batch of tokens (subscription IDs) for American Express cards to be enrolled, processed, and updated. This is called the *token batch update* method.

Both options use the standard REST API HTTP signature authentication method.

Token Harvest

On an agreed-upon monthly date, Account Updater submits your tokenized cards to Visa and Mastercard for updates. American Express tokenized 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. For more information about reports, see Retrieving Update Reports (on page 17).

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.

Token Batch Updates

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

- Test: `https://apitest.cybersource.com/accountupdater/v1/batches`
- Production: `https://api.cybersource.com/accountupdater/v1/batches`

Submitting Visa and Mastercard One-Time Updates

You can submit a specific set of tokens or all tokens for a one-time 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.

To perform a one-time update, set the **type** field to `oneOff`.

Registering Tokens for American Express Daily Updates

You can register tokens for American Express cards for daily updates. Account Updater 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** field to `amexRegistration`.

Batch Creation Request Examples

TMS supports these 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.

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

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: 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: Creating a Batch of Two Instrument Identifier Tokens

```
{
  "type": "amexRegistration",
  "included": {
    "tokens": [
      {
        "id": "7B1F41664F08F6DD3BB1C63892907524",
          "expirationMonth": "12",
          "expirationYear": "2021"
      },
      {
        "id": "E8F44CFA7EBEADDB06A5A9625E7F8696",
          "expirationMonth": "12",
          "expirationYear": "2021"
      }
    ]
  },
  "merchantReference": "Merchant reference",
  "notificationEmail": "email@example.com"
}
```

Batch Creation Response Examples

Example: 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
85/status"
    }
  },
  batchId": "152699969452400021395943 85",
  batchItemCount": 2
}
```

Example: 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: HTTP 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 tokens in your batch, 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-Time Updates

To retrieve one-time updates, verify the batch status URL that was returned in the one-time batch creation. For more info about one-time updates, see Submitting Visa and Mastercard One-Time Updates (on page 13).

To get the status of the batch, send an authenticated GET request, including the header `ACCEPT=application/json`, to one of these resources:

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

One-Time Batch Update Response

Batch processing 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.

These batch statuses are possible:

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`	Account Updater received a response from the card association(s) and is updating the tokens.
`Completed`	Updates have been applied to the tokens. The batch report URL is now available.
`Failed`	Review specific error message.

Not all data is available immediately. As the batch status progresses from `Received` through `Processing` and `Updating` to `Completed`, 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 `Completed`.

Example: 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/status"
      }
    ]
    "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
    }
}
```

American Express Daily and Token Harvest Update Reports

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

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

For daily and token harvest update reports, the first step is to retrieve the batch ID itself by sending an authenticated GET request, including the header `ACCEPT=application/json`, to one of the following resources:

- Test: `https://apitest.cybersource.com/accountupdater/v1/batches`
- Production: `https://api.cybersource.com/accountupdater/v1/batches`

The response is 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, send `/v1/batches?offset=1&limit=50`.

To filter by date, add the **fromDate** and **toDate** fields as a query string using the UTC date format yyyymmddThhmmssZ. Example: `v1/batches?fromDate=20200315T000000Z&toDate=20200415T000000Z`.

Example: 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"
    },
    {
```

Example: HTTP 200: Successful Response (*continued*)


```
      "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/154160314794100020992121
2314/report"
          }
        ]
       },
       "batchId": "154160314794100020992123 14",
       "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
       }
     },
```

Example: HTTP 200: Successful Response (*continued*)


```
    {
      "_links": {
        "reports": [
          {
            "href":
 "https://apitest.cybersource.com/accountupdater/v1/batches/154160250107300016553438
3827/report"
              }
            ]
          },
          "batchId": "154160250107300016553438 27",
          "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/154160250107300016553438
3827/report"
            }
          ]
        },
        "batchId": "154022212730700016839845 45",
        "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 **batchCreatedDate** and the **batchSource** field values.

Batch Source Values

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-time 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 response. A successful response returns the status of the batch. For information about registering American Express tokens for daily updtes, see Registering Tokens for American Express Daily Updates (on page 13).

Example: American Express Registry Status Response


```
{
  "_links": {
    "self": {
      "href":
 "https://apitest.cybersource.com/accountupdater/v1/batches/158160235356200016468
4894/status"
      },
      "report": [
{
        "href":
 "https://apitest.cybersource.com/accountupdater/v1/batches/158160235356200016468
4894/report"
      }
    ]
  },
  "batchCaEndpoints": "AMEX",
  "batchCreatedDate": "2020-02-13T13.59.13Z",
  "batchId": "158160235356200016468548 94",
  "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

To access an individual batch report, send an authenticated GET request, including the header `ACCEPT=application/json`, using the URL returned in the batch status or batches resource described in American Express Daily Harvest Update Reports (on page 20).

Example: AMEX_REGISTRY_API Batch Method HTTP 200: Successful Response


```
{
  "version": "1.0",
  "reportCreatedDate": "2018-11-07T15:33:11Z",
  "batchId": "154160471643300015933142 31",
  "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": "2021"
  },
  "responseRecord": {
    "response": "DEC",
    "reason": "852"
  }
},
{
  "sourceRecord": {
    "token": "456",
    "cardExpiryMonth": "01",
    "cardExpiryYear": "2021"
  },
  "responseRecord": {
    "response": "DEC",
    "reason": "851"
  }
},
```

Example: AMEX_REGISTRY_API Batch Method HTTP 200: Successful Response (*continued*)

```
{
  "sourceRecord": {
     "token": "789",
     "cardExpiryMonth": "01",
     "cardExpiryYear": "2021"
  },
  "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 are set to a CLOSED status under these circumstances:

- 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 the TMS Developer Guide at https://developer.cybersource.com/docs/cybs/en-us/tms/developer/ctv/rest/tms/tms-overview.html.

Example: AMEX_MAINTENANCE Batch Method

```
{
  "version": "1.0",
  "reportCreatedDate": "2020-01-23T11:16:13Z",
  "batchId": "157977801370100005061820 90",
  "batchSource": "AMEX_MAINTENANCE",
  "batchCaEndpoints": "AMEX",
  "batchCreatedDate": "2020-01-23T11:13:33Z",
  "totals": {
    "updatedRecords": 3,
    "rejectedRecords": 0,
```

Example: AMEX_MAINTENANCE Batch Method (*continued*)


```
    "caResponses": 3,
    "caResponsesOmitted": 0
  },
  "billing": {
    "nan": 1,
    "ned": 1,
    "acl": 1,
    "cch": 0
  },
  "records": [
    {
      "id": "562239661",
      "sourceRecord": {
        "token": "9CCD3AE24DD9E254E0533F36CF0A356E",
        "cardNumber": "371449XXXXX2009",
        "cardExpiryMonth": "02",
        "cardExpiryYear": "2021",
        "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": "2021",
        "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."
          }
        ]
      }
    }
```

Example: AMEX_MAINTENANCE Batch Method (*continued*)


```
  {
    "id": "562239711",
    "sourceRecord": {
      "token": "9CCD3AE24DF7E254E0533F36CF0A356E",
      "cardNumber": "371449XXXXX1100",
      "cardExpiryMonth": "02",
      "cardExpiryYear": "2021",
      "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": "2021",
      "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: TOKEN_API Batch Method and Scheduler


```
{
  version": "1.0
  "reportCreatedDate": "2018-11-01T14:43:36Z",
  "batchId": "154108334734000000123332450",
  "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": "2021",
          "cardType": "002"
        },
        "responseRecord": {
          "response": "ACL",
          "reason": "800"
        }
      },
    {
      "id": "784311",
      "sourceRecord": {
      "token": "7020000000014008934",
      "cardNumber": "371000XXXXXX8115",
      "cardExpiryMonth": "01",
      "cardExpiryYear": "2021",
      "cardType": "003",
      "instrumentIdentifierId": "7020000000014008115"
      },
```

Example: TOKEN_API Batch Method and Scheduler (*continued*)

```
      "responseRecord": {
      "response": "NAN",
      "reason": "800",
      "token": "7020000000012513358",
      "cardNumber": "401000XXXXXX2753",
      "cardType": "001",
      "instrumentIdentifierId": "7020000000012512753",
      "instrumentIdentifierCreated": "true",
      "cardExpiryMonth": "08",
      "cardExpiryYear": "2021"
      }
    }
  ]
}
```

Example: Batch Retrieval Error

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

PAN Updates

Important: You must enroll in Account Updater and comply with the Terms of Use. See Terms of Use (on page 10).

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 (on page 41).

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* on the Developer Center.

PGP Public/Private Key

The 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 the "PGP Security Keys" in *Creating and Using Security Keys* on the Developer Center.

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 these components:

- 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.

Request Header Record

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

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 merchant ID. Format: sampleID2	Required	Alphanumeric (30)

Header Record Fields (continued)

Field Name	Description	Required or Optional	Data Type & Length
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)

Header Record Fields (continued)

Field Name	Description	Required or Optional	Data Type & Length
recordCount	The number of detail records in the file. Format: 12345	Required	Numeric
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	String (10)
Batch Info	Optional field that you can pass for reference. Format: sample12	Optional	Alphanumeric (50)

Request Detail Record

Each file must contain at least one detail record.

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)

Detail Record Fields (continued)

Field Name	Description	Required or Optional	Data Type & Length
Card Number	Card number to process.	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)

Request Footer Record

Each file should contain only one footer record.

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: Non-Billing Aggregator Merchants

```
H,cybs.au.request.pan,merchant1,001,2,notify@yourcompany.com,2019-03-23,My January
 Batch
D,1111222233334444,11,09,0001
D,2222333344445555,11,09,0002
      F
```

Example: Billing Aggregator Merchants

```
H,cybs.au.request.pan,merchant1,001,2,notify@yourcompany.com,2019-03-23,My January
 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, test your request files. Follow the instructions in Testing (on page 45).

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

- **Test:** `https://accountupdatertest.cybersource.com/upload/encrypted-pans`
- **Production:** `https://accountupdater.cybersource.com/upload/encrypted-pans`

The request is a POST form data request with the encrypted file keyed as a data file.

The request requires a basic authorization header containing your merchant ID and the merchant signature PGP public key fingerprint colon-delimited and Base64 encoded. For example:

```
Authorization: Basic
 base64Encode(<merchant-id>:<hex-formatted-merchant-public-key-fingerprint>)
```

A successful request results in an HTTP 200 response code. The following error codes are possible:

HTTP Error Codes

Code	Description	Failure Scenarios
400	Bad Request	Malformed request. Payload too large.
401	Unauthorized	Invalid request credentials.

HTTP Error Codes (continued)

Code	Description	Failure Scenarios
		Public key matching public key fingerprint not found.
403	Forbidden	Invalid merchant header in uploaded file.
500	Internal Server Error	Message fails to be put on queue for processing.
503	Server Unavailable	Internal database unavailable to retrieve public key.

For more information on creating a client certificate to upload request files, see Sample Java Code for Uploading PANs (on page 56).

Email Notification

After you upload the request file, Account Updater 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 will not receive an email confirmation, and you must go to the Business Center to view the status. Account Updater 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.

Email Notifications

Subject Line	Reason	Status Viewable in the Business Center
Received	The Account Updater request file was received. Account Updater processes the requests in the file. No action is required.	Yes
Rejected	The file was rejected. Read the contents of the email and follow the suggested remedy.	No
Validated	The file passed validation.	Yes
Declined	The file did not pass validation checks. All records are declined. Read the contents of the email and follow the suggested remedy.	Yes
Processing	The request file is being processed by Account Updater.	Yes
Completed	The response file has been generated and is ready for download.	Yes

Related information

Viewing the Batch File Status *(on page 40)*

Viewing the Status of a Batch File

1. Log in to the Business Center:
    - Live: https://businesscenter.cybersource.com
    - Test: https://businesscentertest.cybersource.com
2. On the left navigation pane, click the **Tools** icon.
3. Click **Account Updater**. The Account Updater page appears.
4. Click **Add filter**. The New Filter selection box appears.
5. Choose **Batch ID** or **Date Range Options**.
    - For Batch ID, enter the specific batch ID.
    - For Date Range Options, choose a time range. If you choose Custom, you can specify the start date, start time, end date, and end time.
6. The Search Results list displays matching results.
7. Click the batch ID to view the details of the file upload.
   If you need to create a new PGP key, click **API Credentials**. The Key Management page appears.

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 in the Developer Center.

1. Log in to the Business Center:
    - Live: https://businesscenter.cybersource.com
    - Test: https://businesscentertest.cybersource.com
2. On the left navigation pane, click the **Reports** icon.
3. Under Downloadable Reports, click **Available Reports**. The Available Reports page appears.
4. Click the tab containing the report you want to download.
5. In the Download column, click the file format link. Only reports that have successfully finished generating and that contain data include links.
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 Account Updater. 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 that you used to create the keys.

The format for a response file consists of these components:

- 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.

Response Header Record

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

Header Record Fields

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

Response Detail Record

Each file contains at least one detail record.

Response Detail Record Fields

Field Name	Description	Data Type & Length
Record Identifier	Constant value indicating the record type. Format: D	Alpha (1)
Request ID	Unique identifier for the record.	Numeric (30)
Old Card Number	Old card number.	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.	Numeric (19)
New Card Expiration Month	New expiration month. Format: MM	Numeric (2)
New Card Expiration Year	New expiration year. Format: YY	Numeric (2)

Response Detail Record Fields (continued)

Field Name	Description	Data Type & Length
Merchant Reference ID	This field is optional and is returned in the response if present in the request file.	Alphanumeric (50)
BA Sub Merchant ID	This field is returned in the response if sent in the request file.	Alphanumeric (10)
Response Code	Response code for the record.	Alpha (3)
Reason Code	Reason code for the record.	Numeric (3)

Related information

PAN Upload Response Codes and Reason Codes *(on page 51)*

Response Footer Record

Each file contains only one footer record.

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.	Numeric (10)
Response Code	Response code for the file.	Alpha (3)

Footer Record Fields (continued)

Field Name	Description	Data Type & Length
Reason Code	Reason code for the file.	Numeric (3)

Related information

PAN Upload Response Codes and Reason Codes *(on page 51)*

Response File Examples

Example: 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: 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,N
AN,800
F,2,COM,800
```

Testing

The Account Updater test environment provides a simulator in which the response from the card association can be triggered using test card numbers.

This simulator ensures that you can handle the possible response combinations when connecting to multiple card associations.

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.

American Express Test Card Numbers

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

For response code descriptions, see PAN Upload Response Codes and Reason Codes (on page 51).

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

American Express Test Numbers (continued)

Card Number	Response Code
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.)
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
BIN `1 4254 6639`	NAN (No New Account Number, No New Expiry Date)
BIN `6 7595 0950`	NAN (No New Account Number, New Expiry Date)

Related information

PAN Upload Response Codes and Reason Codes *(on page 51)*

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. Replace the BIN with `511111` and remove spaces when sending to Account Updater.

Mastercard Card Test Numbers

Card Number	Response Codes
BIN `10 4714 3086`	**Visa Response: NAN** | Mastercard Response: NAN
BIN `10 2999 7178`	Visa Response: ACL |**Mastercard Response: NAN**

Mastercard Card Test Numbers (continued)

Card Number	Response Codes
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

Mastercard Card Test Numbers (continued)

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

Related information

PAN Upload Response Codes and Reason Codes *(on page 51)*

Visa Test Card Numbers

The bold response codes represent the token updates for TMS, Recurring Billing, and Payment Tokenization merchants using the REST API batch update and harvest update. Replace the BIN with `400000` and remove spaces when sending to Account Updater.

Visa Card Test Numbers

Card Number	Response Codes
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**

Visa Card Test Numbers (continued)

Card Number	Response Codes
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**
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

Related information

PAN Upload Response Codes and Reason Codes *(on page 51)*

API Fields

The following fields can be used with Account Updater.

PAN Upload 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: Details Record

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

Record Response Codes and Reason Codes

Response Code	Response Code Description	Reason Code	Reason Code Description	Billable or Non-Billable Code
ACL	Match: account closed. 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.

Record Response Codes and Reason Codes (continued)

Response Code	Response Code Description	Reason Code	Reason Code Description	Billable or Non-Billable Code
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.
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.	Non-billable.

Record Response Codes and Reason Codes (continued)

Response Code	Response Code Description	Reason Code	Reason Code Description	Billable or Non-Billable Code
			This token has a status of CLOSED from a previous Account Updater batch.	
DEC	—	861	Cardholder is already enrolled or cannot cancel cardholder that is not enrolled.	Non-billable.
DEC	—	862	Rejected because cardholder 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.

Record Response Codes and Reason Codes (continued)

Response Code	Response Code Description	Reason Code	Reason Code Description	Billable or Non-Billable Code
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: Footer Record

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

Request File Response Codes and Reason Codes

Response Code	Response Code Description	Reason Code	Reason Code Description
COM	The merchant request file has been validated, 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.

Sample Java Code for Uploading PANs

This section explains how to use the sample Java code to upload your files to Account Updater.

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

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

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.
2. Copy the Bouncy Castle **.jar* files to the *$JAVA_HOME/jre/lib/ext* directory.
3. Edit the *$JAVA_HOME/jre/lib/security/java.security* file by inserting this security provider immediately after the Oracle provider:
   `Security.provider.2=org.bouncycastle.jce.provider.BouncyCastleProvider`

4. Be sure to increment the numbers of the other providers in the list.
   Your list of security providers should now look like this:

```
security.provider.1=sun.security.provider.Sun
security.provider.2=org.bouncycastle.jce.provider.BouncyCastleProvider
security.provider.3=com.sun.net.ssl.internal.ssl.Provider
security.provider.4=com.sun.rsajca.Provider
security.provider.5=com.sun.crypto.provider.SunJCE
security.provider.6=sun.security.jgss.SunProvider
```

5. Import your 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 merchant ID. For example, if your key is giraffe.p12, enter `giraffe` 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.

6. Create a key store file to contain your Simple Order API .p12 security key:

   a. Browse to one of these URLs:

      Test: `https://accountupdatertest.cybersource.com/upload/UploadAccountUpdaterFile`

      Production: `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 giraffe.p12, the pass phrase must be *giraffe*.

m. 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>**

Request Example: Creating the Keystore

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

The output looks like this example:

Response Example: 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
```

7. Modify the *SSLFileTransfer.props* file with your settings. The file is part of the download package and looks similar to this example:

Example: 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>
```

8. Set the JAVA_HOME environment variable to the location in which you installed J2SE.

   Example: Java Home Environment

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

9. Include *$JAVA_HOME/bin* in the PATH.
10. Compile and run the sample:

    a. Change to the directory containing the sample files.

    b. Enter this information:

    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: 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
    ```