Cashflows

Account Updater Integration Guide

Version 1.0 – May 2019

Table of contents

Introduction

CashFlows delivers a range of acquiring services designed to help businesses manage their payments. CashFlows enables businesses to offer their customers a full range of payment channels including, online, mobile and mail and telephone orders.

CashFlows Account Updater

Merchants offering recurring payments can use the Account Updater to check for changes in customers' card details and ensure they are up to date.

With recurring payments, a customer uses the same card to make the recurring payments over a period. But payments can stop when a customer changes their card details or cancels a payment card without contacting you.

The Account Updater gives you access to the VISA Account Updater (VAU) and Mastercard Automatic Billing Updater (ABU) service which allow you to ensure your customers payment details are up to date.

Requests can be supplied in XML or JSON format, and data received is dependent on the accept header which requests what response should be returned.

The mechanism used to sign Account Updater requests (SHA2-512), which is consistent with other CashFlows APIs, allowing re-use of integration code across CashFlows endpoints.

Benefits

When you use Account Updater to keep card details up to date, recurring payments can occur without interruption. By using Account Updater, you can:

- Reduce costs and need to contact customer to manually update payment data
- Minimise the rejected transaction threshold
- Update the cards on record with new expiry date if the card has expired
- Replace card information if the card has been reported lost or stolen

Payment service requirements

To use Account Updater, make sure you:

1. Process recurring or account on file payments
2. Use CashFlows Europe Limited as your acquirer
3. Register with your Implementation Manager to check if you qualify for the service tech tech-support@cashflows.com

Supported card schemes

The following card schemes are supported for the Account Updater service with CashFlows:

- Visa International
- Mastercard

Sensitive Authentication Data and PCI-DSS

Using the Account Updater to send payment data means that you will be capturing, transmitting, and possibly storing card data. The card schemes, Visa and Mastercard, do not permit the storage of Sensitive Authentication Data (track data and/or CVV2) post- authorisation, and it is prohibited under Requirement 3 of the Payment Card Industry Data Security Standard (PCI-DSS).

If you use the Account Updater you will need to demonstrate that your systems handle this data securely and that you are taking full responsibility for your PCI compliance. This includes, but is not limited to, providing your current Attestation of Compliance certificate and evidence of a recent clean vulnerability scan.

A list of approved Security Assessors can be found at:
https://www.pcisecuritystandards.org/assessors_and_solutions/qualified_security_assessors

For further information on PCI security standards, please visit the following web page:
https://www.pcisecuritystandards.org

Command Base URL

A single API endpoint is provided for all users for all gateway functions. Users must identify themselves in each request using a unique ApiKey and sign the message with a signature.

All calls use HTTPS on the standard port 443. TLS v1.2 must be used, in line with PCI requirements. Earlier versions of TLS and SSL of any version are not supported.

The base URL for the Account Updater is:

https://integration.cashflows.com/accountupdater/[commandname] for the Integration/Sandbox environment.

https://live.cashflows.com/accountupdater/[commandname] for the Production/live environment.

The [commandname] is the name of the service being requested, such as, RequestAccountUpdate or RetrieveAccountUpdate. The URLs are provided in each command detail section below.

Request Forms

Requests can be formatted in XML or JSON, and the Accept header should be set to “application/json” or “application/xml” to specify the format of the Response. If no Accept header is specified JSON is assumed.

All responses contain:

- the Version number of the Response being delivered (which will match that in the Request if that Version is valid)
- the current UTC DateTime field (in ISO-8601 format)
- readable Error node field (explaining the error if there is one) or
- a Response node with command-specific nodes within it

Online Documentation

Online documentation can be obtained on Swagger page:
https://integration.cashflows.com/accountupdater/documentation

RequestAccountUpdate Request

The RequestAccountUpdate request can be used to submit new account update requests to obtain updated card details from the schemes.

All new requests are made by POSTing to the command URI. Version, ApiKey, Request and Signature are supplied for Account Updater command:

JSON:

```
{
  "ApiKey": "12345678-ABCD-ABCD-ABCD-12345678ABCD",
  "Signature":
"35b4957df561b0bc0ad64582416e536f75a7f146d7c548e6df17496a351cecf3367c2d555e0d299657c09aea7f5ab311e3cf
c6a3b7d9ba35112a312efde585c4",
  "Request": {
    "MerchantId": "5912345",
    "Details": [
      {
        "CardNumber": "4444333322221111",
        "ExpiryDate": "0118"
      },
      {
        "CardNumber": "5555444433332222",
        "ExpiryDate": "0119"
      },
      {
        "CardNumber": "5555444433331111",
        "ExpiryDate": "0119"
      }
    ]
  },
  "Version": "1.0"
}
```

XML:

```
<SignedRequestAccountUpdateRequest>
<ApiKey>12345678-ABCD-ABCD-ABCD-12345678ABCD</ApiKey>
<Signature>3985e9dd101f89168fd53958d320505b76f021d719821c919773833952ef2c60b205c865f0c0f17d94e
4 2f3d8c bf6c7438f055d17bdea55a53025791b13c13b0</Signature>
<Version>1.0</Version>
<Request>
  <MerchantId>5912345</MerchantId>
  <Details>
    <RequestAccountUpdateRequestDetail>
      <CardNumber>4444333322221111</CardNumber>
      <ExpiryDate>0118</ExpiryDate>
    </RequestAccountUpdateRequestDetail>
    <RequestAccountUpdateRequestDetail>
      <CardNumber>5555444433332222</CardNumber>
      <ExpiryDate>0119</ExpiryDate>
    </RequestAccountUpdateRequestDetail>
    <RequestAccountUpdateRequestDetail>
      <CardNumber>5555444433331111</CardNumber>
      <ExpiryDate>0119</ExpiryDate>
    </RequestAccountUpdateRequestDetail>
  </Details>
</Request>
</SignedRequestAccountUpdateRequest>
```

If an invalid Version number is specified, the latest is assumed. The Signature is a hash of the user's security token and the contents of the Request node (see the section on calculating request signatures for examples). The contents of the request node are unique to each command and detailed later in the document.

The response for the RequestAccountUpdate looks like the example below:

```
{
      "Response" {
    "RequestId": "6fbc91f55d7e401ea89e91dd5fe52af7",
    "MerchantId": "5912345",
    "Results": null,
    "Status": 1
      },
      "Error": null,
      "Date": "2012-04-21T18:25:43Z",
      "Version": "1.0"
}
```

The status indicates the current state of the request. For RequestAccountUpdate requests, this is usually 1 (pending) as requests will take 24 hours to process.

RetrieveAccountUpdate Request

The RetrieveAccountUpdate request can be used to retrieve a request that was previously submitted to RequestAccountUpdate.

PLEASE NOTE: do not post a retrieve account update request until 24 hours have elapsed from the initial RequestAccountUpdate request for that card

All RetrieveAccountUpdate requests are made by POSTing to the command URI. Version, ApiKey, Request and Signature are supplied for the command:

JSON:


```
{
      "ApiKey": "12345678-ABCD-ABCD-ABCD-12345678ABCD",
  "Signature":
"2c6d6a6e134bebdfaf76034563e4c3975c5628cf715aa8eb509500976a82af0fc06c5072d1755dfee41ee81aa74c8cb34dac
d4b4ec8fa53a4ede209cad65cf56",
      "Request": {
    "RequestId": "6fbc91f55d7e401ea89e91dd5fe52af7"
      },
      "Version": "1.0"
}
```


The response for the RetrieveAccountUpdate looks like the example below:


```
{
  "Response" {
    "RequestId":
    "6fbc91f55d7e401ea89e91dd5fe52af7",
    "MerchantId": "5912345",
    "Results": [
      {
      "OldCardNumber": "4444333322221111",
      "OldExpiryDate": "0118",
      "NewCardNumber": "4444333332225555",
      "NewExpiryDate": "0221",
      "VauRawResponse": {
          "ResponseCode": "A"
      },
      "Status": 2
      },
      {
      "OldCardNumber": "5555444433332222",
      "OldExpiryDate": "0119",
      "NewCardNumber": "5555444433331000",
      "NewExpiryDate": "0122",
      "AbuRawResponse": {
          "ReasonIdentifier": "Update",
          "ResponseIndicator" : null
      },
      "Status": 2
      },
      {
      "OldCardNumber": "5555444433331111",
      "OldExpiryDate": "0119",
      "Status": 3
  }
],
    "Status": 3
  },
  "Date": "2012-04-21T18:25:43Z",
  "Version": "1.0"
}
```


The status in the response node indicates the current state of the request. The status in the result nodes indicates the state of each individual card number.

Calculating Request Signatures

To ensure that the server-server messages have been issued by valid users, request messages must be signed. The Signature field is calculated by concatenating the security token with the contents of the Request node (represented as a string) and calculating a SHA2-512 hash of that concatenated string.

The signature must be correct in order to receive a non-error response. Repeated use of an incorrect signature will lock out the user account, and an administrator will need to unlock it.

JSON Signatures

Using the JSON example for RequestAccountUpdate from the previous section, a signature is obtained by first isolating the “Request” node. For JSON requests the request string to be hashed begins after the { following the word “Request” and ends before the } at the end of the node:

```
"Request": {"MerchantId": "5912345","Details": [{"CardNumber": "4444333322221111","ExpiryDate":
"0120"},{"CardNumber": "5555444433332222","ExpiryDate": "0120"},{"CardNumber":
"5555444433331111","ExpiryDate": "0121"}]}
```

To this, the security token is pre-pended:

```
3031E5834AAD94B05C563292E6590ED13336501627EF1248036838C9BEBC08226A030134B3D791B488C086A97EA521FB192B
D
578CD41583DCB6DC21A896A497E"MerchantId": "5912345","Details": [{"CardNumber":
"4444333322221111","ExpiryDate": "0120"},{"CardNumber":
"5555444433332222","ExpiryDate": "0120"},{"CardNumber":
"5555444433331111","ExpiryDate": "0121"}]
```

The SHA-512 hash of the string is then generated:

```
35b4957df561b0bc0ad64582416e536f75a7f146d7c548e6df17496a351cecf3367c2d555e0d299657c09aea7f5ab311e3cfc
6a3b7d9ba35112a312efde585c4
```

This hash is used as the value for "Signature": The message is sent as an HTTP POST to:

```
https://live.cashflows.com/accountupdater/requestaccountupdate
```

With the body:

```
  {
    "ApiKey": "12345678-ABCD-ABCD-ABCD-12345678ABCD",
    "Signature":
  "35b4957df561b0bc0ad64582416e536f75a7f146d7c548e6df17496a351cecf3367c2d555e0d299657c09aea7f5ab311e3cf
  c6a3b7d9ba35112a312efde585c4",
    "Request": {
      "MerchantId":
      "5912345",
      "Details": [
        {
          "CardNumber": "4444333322221111",
          "ExpiryDate": "0118"
        },
        {
          "CardNumber": "5555444433332222",
          "ExpiryDate": "0119"
        },
        {
          "CardNumber": "5555444433331111",
          "ExpiryDate": "0119"
        }
 ]
}
,
    "Version": "1.0"
  }
```

XML Signatures

Using the XML example for RequestAccountUpdate from the previous section, a signature is obtained by first isolating the “Request” element. For XML requests, the request string starts after the > of <Request> and ends before the < of </Request>.

```
  <Request>
    <MerchantId>5912345</MerchantId>
    <Details>
      <RequestAccountUpdateRequestDetail>
        <CardNumber>4444333322221111</CardNumber>
        <ExpiryDate>0118</ExpiryDate>
      </RequestAccountUpdateRequestDetail>
      <RequestAccountUpdateRequestDetail>
        <CardNumber>5555444433332222</CardNumber>
        <ExpiryDate>0119</ExpiryDate>
      </RequestAccountUpdateRequestDetail>
      <RequestAccountUpdateRequestDetail>
        <CardNumber>5555444433331111</CardNumber>
        <ExpiryDate>0119</ExpiryDate>
      </RequestAccountUpdateRequestDetail>
    </Details>
  </Request>
```

To this, the security token is pre-pended (NOTE in this example there are whitespace linefeed characters, and these are included in the Hash calculation):

```
3031E5834AAD94B05C563292E6590ED13336501627EF1248036838C9BEBC08226A030134B3D791B488C086A97EA521FB
192BD
578CD41583DCB6DC21A896A497E
  <MerchantId>5912345</MerchantId>
  <Details>
    <RequestAccountUpdateRequestDetail>
      <CardNumber>4444333322221111</CardNumber>
      <ExpiryDate>0118</ExpiryDate>
    </RequestAccountUpdateRequestDetail>
    <RequestAccountUpdateRequestDetail>
      <CardNumber>5555444433332222</CardNumber>
      <ExpiryDate>0119</ExpiryDate>
    </RequestAccountUpdateRequestDetail>
    <RequestAccountUpdateRequestDetail>
      <CardNumber>5555444433331111</CardNumber>
      <ExpiryDate>0119</ExpiryDate>
    </RequestAccountUpdateRequestDetail>
  </Details>
```

The SHA-512 hash of the string is then generated:

```
D0A49D99BBC3835323FE2CA95DAE97710F40C19A42BC923ABCD970B218AFB64F56B71CD349937EEA11A198A9D9E06F01ADA2D7A
FF88A1 A811BA5FD 811104D7AD
```

This hash is used as the value for the <Signature> element. The message is sent as an HTTP POST to:

```
https://live.cashflows.com/accountupdater/requestaccountupdate
```

With the body:

```
      <SignedRequestAccountUpdateRequest>
      <ApiKey>12345678-ABCD-ABCD-ABCD-12345678ABCD</ApiKey>
      <Signature>D0A49D99BBC3835323FE2CA95DAE97710F40C19A42BC923ABCD970B218AFB64F56B71CD349937EEA11A198
      A9D9E0 6F01ADA2D7AFF88A1A811BA5FD811104D7AD</Signature>
      <Version>1.0</Version>
      <Request>
  <MerchantId>5912345</MerchantId>
  <Details>
    <RequestAccountUpdateRequestDetail>
      <CardNumber>4444333322221111</CardNumber>
      <ExpiryDate>0118</ExpiryDate>
    </RequestAccountUpdateRequestDetail>
    <RequestAccountUpdateRequestDetail>
      <CardNumber>5555444433332222</CardNumber>
      <ExpiryDate>0119</ExpiryDate>
    </RequestAccountUpdateRequestDetail>
    <RequestAccountUpdateRequestDetail>
      <CardNumber>5555444433331111</CardNumber>
      <ExpiryDate>0119</ExpiryDate>
    </RequestAccountUpdateRequestDetail>
  </Details>
      </Request>
      </SignedRequestAccountUpdateRequest>
```

IMPORTANT: All whitespace and new lines within the request nodes will be included in the calculation of the Hash Value. CashFlows uses CR-LF for line breaks; Unix systems often use just LF, and this can affect calculations. If you are unable to match signatures with the Account Updater and have the correct password hash, consider removing unnecessary whitespace from your Request nodes.

Example of POST Request Signature Calculation in C#

```
System.Security.Cryptography.SHA512 sha512 = System.Security.Cryptography.SHA512.Create();
System.Text.StringBuilder builder = new System.Text.StringBuilder();

byte[] bytes = System.Text.Encoding.UTF8.GetBytes(
      passwordHash + requestBody);

byte[] hashedBytes = sha512.ComputeHash( bytes );
foreach( byte b in hashedBytes )
{
      builder.Append( b.ToString( "x2" ) );
}
string signature = builder.ToString();
```

Testing

When integrating with the Account Updater API, or testing new code, the Integration environment should be used. The URL for the Integration end-point is:

https://integration.cashflows.com/accountupdater/[commandname]/

Some test data will need to be generated to fully build and test the Account Updater API. For example, a valid Merchant Id is required to attempt RequestAccountUpdate commands.

Please contact our Implementation team (contact details at the end of this document) for more information about test accounts and data.

Account Updater API Commands

This section lists all currently available Account Updater API commands with version-specific variants where they exist. Each command's subsection provides tabular Request and Response parameters, with a simple worked example of each. The Signatures on each request are calculated as outlined in the Calculating Request Signatures section.

List of Commands

- RequestAccountUpdate
- RetrieveAccountUpdate

Standard Request Parameters

Parameter	Requirement	Type & size	Details
ApiKey	Mandatory	String (36)	API Key responsible for calling the API
Version	Mandatory	String (10)	Version of the API being called
Signature	Mandatory	String (128)	SHA512 Hash of security token and request as described in Calculating Request Signatures
Request	Mandatory	object	A request object

Standard Response Parameters

Parameter	Type & size	Details
Version	String(10)	Confirmation of the version of the API used in the request
DateTime	Datetime	The date of the response
Error	ErrorResponse	Contains the details for all HTTP 4xx or 5xx responses. The structure is shown below
Response	Object	The response object for all HTTP 2xx responses

ErrorResponse

Parameter	Type & size	Details
Identifier	string	Returned with all HTTP 5xx responses, and can be used by customer services to help identify the exact nature of the error
Code	string	A code associated with the error type
Message	string	Summary of the error
Target	string	For validation errors, this states which property is in error
Details	ErrorResponse[]	Details about each individual error

RequestAccountUpdate

Used to submit new request to obtain updated card details.

Command URL

Live: https://live.cashflows.com/accountupdater/RequestAccountUpdate

Request Parameters

Parameter	Requirement	Type & size	Details
MerchantId	Mandatory	String(15)	Merchant Id (Acquirer MID) to send in requests
Details	Mandatory	Collection of Objects	See definition below **

Details object

Parameter	Requirement	Type & size	Details
CardNumber	Mandatory	String(20)	The full Primary Account Number (PAN) on the card
ExpiryDate	Optional	String(4)	The payment card's expiry date and month (example: 1225) (mmyy)

RetrieveAccountUpdate

Used to obtain update on a previous RetrieveAccountUpdate.

Command URL

Live: https://live.cashflows.com/accountupdater/RetrieveAccountUpdate

Request Parameters

Parameter	Requirement	Type & size	Details
RequestId	Mandatory	String(32	Unique ID returned in the response to the account update request (command RequestAccountUpdate)

Response Parameters

Response parameters for RequestAccountUpdate and RetreiveAccountUpdate are as listed below.

Parameter	Type & size	Details
MerchantId	String(15)	The merchant number from the original request
Status	String(20)	Unknown = 0 (please refer to Implementations team with date and time of Unknown status response) Pending = 1 (no response from the scheme as yet) Complete = 2 (request has been completed) Error = 3 (please refer to Implementations team with date and time of Error status response Expired = 4 (the result is no longer available. This is because either the result has been retrieved or have not been collected after 7 days since made available)
Results	Collection of Objects	See definition below**
RequestId	String(32)	Unique ID returned to be used to retrieve results of an account update request

Results Object

Parameter	Type & size	Details
OldCardNumber	String(20)	The original card number
OldExpiryDate	String(4)	The original expiry date
NewCardNumber	String(20)	The new card number
NewExpiryDate	String(4)	The new expiry date
VauRawResponse	Object	Visa only See object definition below
AbuRawResponse	Object	Mastercard only See object definition below
Status	String(20)	Unknown = 0, Complete = 2, Error = 3

The following http status codes may be returned in the header of the response, if there is an error:

Code	Description	Troubleshooting
400	Bad request	Check parameters for correct formation and that all mandatory items are present
403	Forbidden	Check the signature calculation
429	Too many requests	Either rate limits have been exceeded or replay protection has been triggered.
500	Internal server error	Please get in touch with your implementations contact if this issue arises on Integration, or your Relationship Manager or Support should it occur in Production

Account Updater Schemes Codes

Visa Service Identifier codes

The following codes will be returned in the standard response parameter when Visa account number has been queried VauRawResponse (Object)

Parameter	Type & size	Details
ResponseCode	String(1)	The code returned by Visa

Response Code	Description
A	Account number change (the account number or account number and expiration date are updated)
C	Closed account advice
E	Expiration date change
N	Non-participating BIN
P	Participating BIN, no match
Q	Contact cardholder advice (the merchant should contact the cardholder for additional information on the account)
V	Match made, account number and expiration date unchanged

Mastercard Response parameters

Automatic Billing Updater (ABU) for Mastercard will return reason identifier and response indicator AbuRawResponse (Object)

Reason Identifier	Description
UPDATE	Match made; update data provided (includes issuer Reason Codes R, B, and P) R – Replacement Card B – Brand Flip to Mastercard P - Mastercard to Mastercard Portfolio
CONTAC	Match made; account closed (includes Issuer Reason Code C) C - Closed Account
EXPIRY	Match made; expiration date changed (includes Issuer Reason Code E) E - Update Expiration Date (Update Only)
VALID	No updates were found but the account is valid
UNKNWN	The account number could not be found in the ABU database. (See the Response Indicator for further explanation.) If Mastercard detects an error in the input, this field will contain the error code value

Response Indicator

ABU Validation Response Indicator (displayed only when the Reason Identifier is VALID or UNKNWN)

Response Indicator	Description
V	Matches the account as reported by the issuer
P	No match, participating BIN/ issuer
N	No match, non-participating BIN/issuer
R	Replacement Card
B	Brand Flip to Mastercard match
C	Closed Account
E	Update Expiration Date

Cashflows

+44 (0)1223 550920

Cambridge
CPC1
Capital Park
Cambridge
CB21 5XE

London
2 Portman Street
London
W1H 6DU

The Netherlands
Noorderhof 24
5804 BV Venray