



Peach Payments

API Specification

Technical document

Last updated: 30 May 2017

Contents

Introduction	4
Getting started	4
Response formats and structure	4
Successful request.....	4
Unsuccessful request	4
Duplicate checking	5
Using Tokens in place of account details.....	5
External links	5
Example	7
Payments.....	8
Overview.....	8
Request URL	8
Request structure	8
Request Example.....	10
Response structure.....	10
Response Example.....	11
Unpays.....	11
Unpays Example.....	12
Bank Account Verification (BANV)	13
Overview.....	13
Request URL	13
Request structure	13
Request Example.....	14
Response structure.....	15
Response Example.....	15
Validation Responses.....	15
Validations Response Example.....	16
Real-time Bank Account Verification (BANVR).....	17
Overview.....	17
Request URL	17
Request structure	17
Request Example.....	18

Response structure	19
Response Example.....	20
Check Digit Verification (CDV)	21
Overview.....	21
Request URL	21
Request structure	21
Request Example.....	22
Debit Orders	23
Overview.....	23
Request URL	23
Request structure	23
Request Example.....	24
Response structure	24
Response Example.....	25
Unpays.....	25
Unpays Example.....	26

Introduction

The Peach Payments API gives developers access to the processing platform used by Peach Payments. Developers can use the API to securely submit requests for services using their own software.

Getting started

Contact Peach Payments and request an API key. Your API key will be used with every request in order to identify your application. If no key is provided, then access to the data will not be granted. An incorrect key will result in the same error. Your API key will be appended to the request URL as a query string parameter with the key “key” followed by your unique key e.g.

<https://www.peachpay.co.za/API/Verification?key=6281ac86e87f485aa78bd4b90e14cf2c> where “6281ac86e87f485aa78bd4b90e14cf2c” is your unique API key.

All request and response data are submitted as a form data in a POST action (i.e. application/x-www-form-urlencoded) with the key “response”.

Response formats and structure

Successful request

The response is in a structured XML format and will always have the following basic structure:

```
<Response>
  <Result> OK </Result>
  <BatchCode> 30264 </BatchCode>
  <TotalFeeExcludingVAT> 4.50 </TotalFeeExcludingVAT>
</Response>
```

The *Result* element’s value is always “OK” if no errors were encountered. The *BatchCode* element’s value will display the batch code for that transaction. The batch code is a unique code that is returned with all responses so that you can determine which batch the response is for. The *TotalFeeExcludingVAT* element’s value will return the fees charged by Peach Payments for that batch. Other elements may be returned depending on the service.

Whenever a date is used it is always in the format yyyyMMdd e.g. 20120214 for 14 February 2012.

If a unique Id value has been included, it will be returned with the response as well.

Unsuccessful request

```
<Response>
  <Result> Error </Result>
  <ResultMessage> Your key is invalid </ResultMessage>
</Response>
```

If a request was an unsuccessful the *Result* element’s value will always be “Error”. The *ResultMessage* value will display the reason for the error.

Duplicate checking

All requests can include an optional *Uniqueld* field in the header of the request. When this field is included, it will be saved with the batch as a unique identifier. If another request arrives with the same unique Id value, it will be rejected.

```
<Response>
  <Result> Error </Result>
  <ResultMessage> This batch has the same unique reference as another batch and is rejected as a
  duplicate </ResultMessage>
</Response>
```

Using Tokens in place of account details

Each beneficiary record includes an API token that can be used as a shortcut for filling in the account details for that transaction. If the token field is present in the *FileContents* section, all other fields except for **FileAmount** and **AmountMultiplier** will be ignored and can be excluded.

The token will be matched to a beneficiary record for the client code specified in the request and the relevant information will be filled in from the beneficiary record.

For security reason, CDV results will not include the **account number** or **branch code** in the API response but will include the **customer code** and **token** so that you can match it the original transaction.

```
<Result>
  <Result> Valid </Result>
  <CustomerCode> Chri01 </CustomerCode>
  <Token> E939CF649768D5D9D478D2B0ABABE8 </Token>
</Result>
```

A CDV error will be returned with the **token** value if no matching beneficiary record can be found.

```
<Result>
  <Result> Invalid </Result>
  <Message> Token can not be matched to beneficiary. Ensure the beneficiary is active. </Message>
  <CustomerCode />
  <Token> 5252F632A616064D305A54F6FFE038 </Token>
</Result>
```

External links

External links are used to reference an object or page that is not part of the Peach Payments secure web interface. For example, a link to the original invoice for a payment can be included for display in reports. All requests can include an optional *ExternalLinks* section for each account (*FileContents*) field.

External links are displayed on the authorisation page of the secure web interface and is in the process of being rolled out to other reports.

Structure

An *ExternalLink* section consists of the following elements:

Element	Required	Description
Label	Yes	A description of the link. This will be displayed next to each link when reporting.
URL	Yes	The URL of the link

Multiple *ExternalLink* sections can be included under the *ExternalLinks* containing section.

Example

```

<FileContents>
  <Initials/>
  <IdNumber/>
  <AccountNumber>123456789</AccountNumber>
  <BranchCode>632005</BranchCode>
  <Reference>xxxxxxxx</Reference>
  <FirstNames>Test</FirstNames>
  <Surname>Example</Surname>
  <FileAmount>2000.0</FileAmount>
  <AmountMultiplier>1</AmountMultiplier>
  <ExternalLinks>
    <ExternalLink>
      <Label>Invoice</Label>
      <URL>http://www.example.com/Invoices?id=1</URL>
    </ExternalLink>
    <ExternalLink>
      <Label>Receipt</Label>
      <URL>http://www.example.com/Receipts</URL>
    </ExternalLink>
  </ExternalLinks>
</FileContents>

```

Payments

Overview

A payments request will submit data relating to a payments batch. A payments batch can be for Creditors, Salaries or Wages. Once a payment batch has been submitted, a CDV check will be performed on the account details provided and any accounts failing that check will be included in the response. Any accounts that passed the CDV check will continue to be processed.

If an account is rejected by a bank (usually because it has been closed) an unpaids response is created and POSTed back to the call-back URL provided. All response data are submitted as a form data in a POST action (i.e. application/x-www-form-urlencoded) with the key "response".

Request URL

<https://www.peachpay.co.za/API/Payments?key=yourkey>

Request structure

A payments request consists of 3 separate sections with a root element with the name *APIPaymentsRequest*. The *Header* section contains information relating the payments batch, the *Payments* sections is a collection of account details to be used as the destination for the payments and the *Totals* section is used to verify the information being sent is complete and correct.

Header

A payment request has a single *Header* section and it consists of the following elements:

Element	Required	Description
PsVer	Yes	The version of the Peach Payments file format. Currently 2.0.1.
Client	Yes	Your unique client code.
Service	Yes	The code for the service being used. See below for the service codes.
ServiceType	Yes	The service type of the payment. See below for the service types.
DueDate	Yes	The date the batch should be submitted to the bank for processing.
BankAccount	No	The bank account being used to deposit the funds. This is used to display the Peach Payments account details for your bank account.
CallBackUrl	Yes	The call-back URL to be used to send back unpaids.
Reference	Yes	The reference used to identify the batch for reporting
Uniqueld	No	Used to prevent duplicate requests

Payment service codes

The following service codes are available for payments.

Value	Description
Wages	For payments relating to wages.
Salaries	For payments relating to salaries.
Creditors	For payments relating to creditors.

Payment service types

The following service codes are available for payments.

Value	Description
-------	-------------

1Day	For 1 day payments.
SDV	For SDV payments.

Payments

A payment request has a single *Payments* section with multiple *FileContents* sections consisting of the following elements:

Element	Required	Description
Initials	No	The initials of the payment recipient.
FirstNames	No	The first name(s) of the payment recipient.
Surname	Yes	The surname or company name of the payment recipient.
BranchCode	Yes	The branch code of the account where the payment will be deposited.
AccountNumber	Yes	The account number of the account where the payment will be deposited.
FileAmount	Yes	The amount to be deposited.
AmountMultiplier	Yes	Usually 1 if the FileAmount is specified in Rands otherwise 100 if the FileAmount is specified in cents.
AccountType	No	The type of account where the funds will be deposited. This can be left blank or 0 can be used if you don't know. Only 0 – 6 can be used.
CustomerCode	No	The customer code used to identify the payment in your system.
Reference	Yes	The reference that will appear on the recipient's bank statement.

Totals

A payment request has a single *Totals* section and it consists of the following elements:

Element	Required	Description
Records	Yes	The total number of payment records being submitted.
Amount	Yes	The total value of the payment records being submitted.
BranchHash	Yes	The sum of all the branch codes for the payment records.
AccountHash	Yes	The sum of all the account numbers for the payment records.

Request Example

```

▼ <APIPaymentsRequest>
  <Header>
    <PsVer> 2.0.1 </PsVer>
    <Client> ZER001 </Client>
    <DueDate> 20120918 </DueDate>
    <Service> Wages </Service>
    <ServiceType> 1day </ServiceType>
    <Reference> Acme Wages w/e 11/19 </Reference>
    <CallbackUrl> http://example.com/API/CallBack </CallbackUrl>
    <BankAccount> Nedbank </BankAccount>
  </Header>
  ▼ <Payments>
    <FileContents>
      <Initials> AB </Initials>
      <FirstNames> Name 1 </FirstNames>
      <Surname> Surname 1 </Surname>
      <BranchCode> 632005 </BranchCode>
      <AccountNumber> 7912172416078 </AccountNumber>
      <FileAmount> 2409.00 </FileAmount>
      <AccountType> 0 </AccountType>
      <CustomerCode> MTO01 </CustomerCode>
      <AmountMultiplier> 1 </AmountMultiplier>
      <Reference> Acme Wages w/e 11/19 </Reference>
    </FileContents>
    ▼ <FileContents>
      <Initials> CD </Initials>
      <FirstNames> Name 2 </FirstNames>
      <Surname> Surname 2 </Surname>
      <BranchCode> 632006 </BranchCode>
      <AccountNumber> 12345689 </AccountNumber>
      <FileAmount> 12569 </FileAmount>
      <AccountType> 0 </AccountType>
      <CustomerCode> ZER001 </CustomerCode>
      <AmountMultiplier> 0.01 </AmountMultiplier>
      <Reference> Acme Wages w/e 11/19 </Reference>
    </FileContents>
  </Payments>
  <Totals>
    <Records> 2 </Records>
    <Amount> 2534.69 </Amount>
    <BranchHash> 1264011 </BranchHash>
    <AccountHash> 7912184761767 </AccountHash>
  </Totals>
</APIPaymentsRequest>

```

Response structure

The standard response structure will be returned with an extra element named `BatchValueSubmitted` which contains the total value of the batch submitted (less any accounts that failed the CDV check).

If any of the payment's *FileContents* records failed the CDV check they will be returned as a *CDVResults* element with a collection of *Result* elements outlining the reasons for the accounts rejection. The *Result* section consists of the following elements:

Filter	Description
Result	The result of the CDV check. Will always be Invalid for a failed account.
Message	A message giving the reason for the rejection.
AccountNumber	The account number that failed the CDV test.

Please note: If any of the accounts passed the CDV check the batch will be accepted and will proceed to be processed.

Response Example

```
<Response>
  <Result> OK </Result>
  <BatchCode> 30278 </BatchCode>
  <BatchValueSubmitted> 2409.00 </BatchValueSubmitted>
  <TotalFeeExcludingVAT> 2.54 </TotalFeeExcludingVAT>
  <CDVResults>
    <Result>
      <Result> Invalid </Result>
      <Message> Branch code not found. </Message>
      <AccountNumber> 12345689 </AccountNumber>
    </Result>
  </CDVResults>
</Response>
```

Unpays

An unpaid occurs when the bank rejects a payment. This can happen for many reasons like the recipients account being closed. When this happens, an unpays response is POSTed back to your server using the *CallBackUrl* provided in the initial request.

Note: The unpaid responses can continue for several days after the batch is submitted.

The *Response* root element will consist of the following elements:

Filter	Description
Result	Will always be OK.
BatchCode	The unique code for the batch the unpays belong to.
PaymentResults	The collection of <i>Results</i> elements with more information about each unpaid.

The *PaymentResult* section consists of the following elements:

Element	Description
FirstName	The first name(s) of the payment recipient.
Surname	The surname of the payment recipient.
BranchCode	The branch code of the account where the payment will be deposited.
AccountNumber	The account number of the account where the payment will be deposited.
CustomerCode	The customer code used to identify the payment in your system.
Reference	The reference that will appear on the recipient's bank statement.
Result	The result status which will always be Rejected for an unpaid.
ResultMessage	The reason for the unpaid.

Unpays Example

```
▼ <Response>
  <Result> OK </Result>
  <BatchCode> 30174 </BatchCode>
  ▼ <PaymentResults>
    <Result>
      <AccountNumber> 123456789 </AccountNumber>
      <BranchCode> 632005 </BranchCode>
      <FirstName> Name 2 </FirstName>
      <Surname> Surname 2 </Surname>
      <Reference> Acme Wages w/e 11/19 </Reference>
      <CustomerCode />
      <Result> Rejected </Result>
      <ResultMessage> ACCOUNT CLOSED </ResultMessage>
    </Result>
    ▼ <Result>
      <AccountNumber> 123456789 </AccountNumber>
      <BranchCode> 632005 </BranchCode>
      <FirstName> Name 1 </FirstName>
      <Surname> Surname 1 </Surname>
      <Reference> Acme Wages w/e 11/19 </Reference>
      <CustomerCode />
      <Result> Rejected </Result>
      <ResultMessage> INVALID ACCOUNT NO </ResultMessage>
    </Result>
  </PaymentResults>
</Response>
```

Bank Account Verification (BANV)

Overview

A BANV request will submit data relating to the verification of a batch of account numbers against the information provided. Once a batch has been submitted, a CDV check will be performed on the account details provided and any accounts failing that check will be included in the response. Any accounts that passed the CDV check will continue to be processed.

Request URL

<https://www.peachpay.co.za/API/Verification?key=yourkey>

Request structure

A BANV request consists of 3 separate sections with a root element with the name *APIVerificationRequest*. The *Header* section contains information relating to the BANV batch, the *Records* sections is a collection of account details to be used for verification and the *Totals* section is used to verify the information being sent is complete and correct.

Header

A BANV request has a single *Header* section and it consists of the following elements:

Element	Required	Description
PsVer	Yes	The version of the Peach Payments file format. Currently 2.0.1.
Client	Yes	Your unique client code.
Service	Yes	The code for the service being used. Must be BANV for bank account verification.
Reference	Yes	The batch reference. Allows you to identify the batch in the reports and the verification responses.
CallBackUrl	Yes	The call-back URL to be used to send back verification response.
Uniqueld	No	Used to prevent duplicate requests

Records

A BANV request has a single *Records* section with multiple *FileContents* sections consisting of the following elements:

Element	Required	Description
Initials	No	The initials of the account holder.
Name	Yes	The account holder name (could be a company name).
IdNumber	No	The ID/Passport or company registration number of the account holder.
AccountNumber	Yes	The account number of the account to be verified.
BranchCode	Yes	The branch code of the account number to be verified.
Reference	No	A transaction reference. Can be used to uniquely identify a verification record.

Totals

A BANV request has a single *Totals* section and it consists of the following elements:

Element	Required	Description
Records	Yes	The total number of BANV records being submitted.
BranchHash	Yes	The sum of all the branch codes for the BANV records.
AccountHash	Yes	The sum of all the account numbers for the BANV records.

Request Example

```

▼ <APIVerificationRequest xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Header>
    <PsVer> 2.0.1 </PsVer>
    <Client> CLI001 </Client>
    <Service> BANV </Service>
    <Reference> 5621-28/02/13 </Reference>
    <CallBackUrl> http://yourURL.com/Callback </CallBackUrl>
  </Header>
  ▼ <Records>
    ▼ <FileContents>
      <Initials> EX </Initials>
      <Name> Example Name </Name>
      <IdNumber> 5511255173085 </IdNumber>
      <AccountNumber> 123456789101 </AccountNumber>
      <BranchCode> 632005 </BranchCode>
      <Reference> 5621-123456789101 </Reference>
    </FileContents>
  </Records>
  <Totals>
    <Records> 1 </Records>
    <BranchHash> 632005 </BranchHash>
    <AccountHash> 123456789101 </AccountHash>
  </Totals>
</APIVerificationRequest>

```

Response structure

The standard response structure will be returned. All response data are submitted as a form data in a POST action (i.e. application/x-www-form-urlencoded) with the key "response".

If any of the BANV *FileContents* records failed the CDV check they will be returned as a *CDVResults* element with a collection of *Result* elements outlining the reasons for the account's rejection. The *Result* section consists of the following elements:

Filter	Description
Result	The result of the CDV check. Will always be Invalid for a failed account.
Message	A message giving the reason for the rejection.
AccountNumber	The account number that failed the CDV test.

Please note: If any of the accounts passed the CDV check the batch will be accepted and will proceed to be verified with the bank.

Response Example

```
<Response>
  <Result> OK </Result>
  <BatchCode> 31534 </BatchCode>
  <TotalFeeExcludingVAT> 0.00 </TotalFeeExcludingVAT>
  <CDVResults>
    <Result>
      <Result> Invalid </Result>
      <Message> Branch code not found. </Message>
      <AccountNumber> 009071574627 </AccountNumber>
    </Result>
  </CDVResults>
</Response>
```

Validation Responses

A validation response occurs when the bank returns the result of the bank account verification. When this happens, a validation response is POSTed back to your server using the *CallBackUrl* provided in the initial request.

Note: The validation responses are POSTed back to the *CallBackUrl* whenever a response is returned by the bank. This is likely to occur multiple times as each bank has different response times.

The *Response* root element will consist of the following elements:

Filter	Description
Result	Will always be OK.
BatchCode	The unique code for the batch the validations belong to.

The *BanvResults* section consists of the following elements:

Element	Description
AccountNumber	The account number of the account that was verified.
IDNumber	The ID/Passport or company registration number of the account holder.
Initials	The initials of the account holder.
Name	The account holder name (could be a company name).
Reference	The batch reference. Allows you to identify the batch in the reports.
Flags	Y/N/Blank characters indicating a corresponding result. See section below.
Result	The result status which is a text representation of the verification result.

BANV Flags

The flags of the verification result are a character indication of each of the 8 verification options. Not all banks are able to supply all the information in which case the character will be a blank space. An example Flags return would be: YYYYYYYY which is the equivalent to a result of "A/c exists, ID matches, Inits match, Name matches, A/c open, accepts Dr, accepts Cr & open more than 3 months".

If the first 5 flags are "Y" then then account can be considered valid.

Note: Names and Initials must be identical to the name or initial the bank has on record otherwise it will not match. If the Initials provided are "WJ" and the bank has just "W" on record then the initials flag will be displayed as an "N".

The 8 flag options are:

Position	Options
1	Account exists
2	ID number or Company registration match
3	Initials match (should be blank to match for company)
4	Surname or company name match
5	Account open
6	Account accepts debits
7	Account accepts credits
8	Accounts open for three or more months

Validations Response Example

```

<Response>
  <Result> OK </Result>
  <BatchCode> 34477 </BatchCode>
  <BanvResults>
    <Result>
      <AccountNumber> 123456789101 </AccountNumber>
      <IDNumber> 5511255173085 </IDNumber>
      <Initials> EX </Initials>
      <Name> Example Name </Name>
      <Reference> 5621-123456789101 </Reference>
      <Flags> YNNYYYYY </Flags>
      <Result>
        A/c exists, ID matches, Inits don't match, Name doesn't match, A/c open, accepts Dr, accepts Cr & open more than
        3 months
      </Result>
    </Result>
  </BanvResults>
</Response>

```

Real-time Bank Account Verification (BANVR)

Overview

A BANVR request will submit data relating to the verification of a single account number against the information provided. Real-time account verification has a guaranteed response time from the bank of 60 seconds. If no response has been received after 60 seconds, then no result is returned. Due to the real-time response, no callback URL is supplied as no callback will be sent.

Request URL

<https://www.peachpay.co.za/API/RealtimeVerification?key=yourkey>

Request structure

A BANVR request consists of 3 separate sections with a root element with the name *APIRealTimeVerificationRequest*. The *Header* section contains information relating to the BANVR batch, the *Records* section is a collection of account details to be used for verification (only a single account is valid for real-time verification) and the *Totals* section is used to verify the information being sent is complete and correct.

Header

A BANVR request has a single *Header* section and it consists of the following elements:

Element	Required	Description
PsVer	Yes	The version of the PaySoft file format. Currently 2.0.1.
Client	Yes	Your unique client code.
Reference	Yes	The batch reference. Allows you to identify the batch in the reports and the verification responses.
Uniqueld	No	Used to prevent duplicate requests
CallBackUrl	No	If a callback URL is supplied and the real-time verification fails, then a standard BANV service will be used and the results will be sent to the callback URL supplied

Records

A BANVR request has a single *Records* section with a single *FileContents* section consisting of the following elements:

Element	Required	Description
Initials	No	The initials of the account holder.
Name	Yes	The account holder name (could be a company name).
IdNumber	No	The ID/Passport or company registration number of the account holder.
AccountNumber	Yes	The account number of the account to be verified.
BranchCode	Yes	The branch code of the account number to be verified.
Reference	No	A transaction reference. Can be used to uniquely identify a verification record.

Totals

A BANVR request has a single *Totals* section and it consists of the following elements:

Element	Required	Description
Records	Yes	The total number of BANVR records being submitted.
BranchHash	Yes	The sum of all the branch codes for the BANVR record.
AccountHash	Yes	The sum of all the account numbers for the BANVR record.

Request Example

```
<?xml version="1.0" encoding="utf-8"?>
<APIRealTimeVerificationRequest xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                                xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Header>
    <PsVer>2.0.1</PsVer>
    <Client>ZER001</Client>
    <Reference>TEST BANVR</Reference>
  </Header>
  <Records>
    <FileContents>
      <Name>BARISHMAN</Name>
      <Initials>A</Initials>
      <IdNumber>4012155047088</IdNumber>
      <AccountNumber>1021278653</AccountNumber>
      <Reference>12345</Reference>
      <CustomerCode>C12345C</CustomerCode>
      <BranchCode>198765</BranchCode>
    </FileContents>
  </Records>
  <Totals>
    <Records>1</Records>
    <BranchHash>198765</BranchHash>
    <AccountHash>1021278653</AccountHash>
  </Totals>
</APIRealTimeVerificationRequest>
```

Response structure

The standard response structure will be returned with a *BanvRealtimeResult* element containing the validation results.

The *Response* root element will consist of the following elements:

Filter	Description
Result	Will always be OK.
BatchCode	The unique code for the batch the validation belong to.

The *BanvRealTimeResult* section consists of the following elements:

Element	Description
AccountNumber	The account number of the account that was verified.
IDNumber	The ID/Passport or company registration number of the account holder.
Initials	The initials of the account holder.
Name	The account holder name (could be a company name).
Reference	The batch reference. Allows you to identify the batch in the reports.
Flags	Y/N/Blank characters indicating a corresponding result. See section below.
Result	The result status which is a text representation of the verification result.

If the real-time was unable to be processed, the **Result** field will have the text “Unprocessed” and the **Flags** field will have the text “U”.

The *CDVResults* element will always be returned regardless of whether or not the account passed the CDV check.

The *Result* section consists of the following elements:

Filter	Description
Result	The result of the CDV check. Will always be Invalid for a failed account.
Message	A message giving the reason for the rejection.
AccountNumber	The account number that failed the CDV test.

BANV Flags

The flags of the verification result are a character indication of each of the 8 verification options. Not all banks are able to supply all the information in which case the character will be a blank space. An example Flags return would be: YYYYYYYY which is the equivalent to a result of “A/c exists, ID matches, Inits match, Name matches, A/c open, accepts Dr, accepts Cr & open more than 3 months”.

If the first 5 flags are “Y” then then account can be considered valid.

Note: Names and Initials must be identical to the name or initial the bank has on record otherwise it will not match. If the Initials provided are “WJ” and the bank has just “W” on record then the initials flag will be displayed as an “N”.

The 8 flag options are:

Position	Options
1	Account exists
2	ID number or Company registration match
3	Initials match (should be blank to match for company)
4	Surname or company name match
5	Account open

6	Account accepts debits
7	Account accepts credits
8	Accounts open for three or more months

Response Example

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<Response>
  <Result>OK</Result>
  <BatchCode>35308</BatchCode>
  <TotalFeeExcludingVAT>0.00</TotalFeeExcludingVAT>
  <BanvRealtimeResult>
    <AccountNumber>1021278653</AccountNumber>
    <IDNumber>4012155047088</IDNumber>
    <Initials>A</Initials>
    <Name>BARISHMAN</Name>
    <Reference>12345</Reference>
    <Flags>YYNYYYY</Flags>
    <Result>A/c exists, ID matches, Inits matches, Name doesn't match, A/c open,
    accepts Dr, accepts Cr & open more than 3 months</Result>
  </BanvRealtimeResult>
  <CDVResults>
    <Result>
      <Result>Valid</Result>
      <AccountNumber>1021278653</AccountNumber>
      <BranchCode>198765</BranchCode>
      <Reference>12345</Reference>
      <CustomerCode>C12345C</CustomerCode>
    </Result>
  </CDVResults>
</Response>
```

Check Digit Verification (CDV)

Overview

A CDV request will submit data relating to the verification of a batch of account numbers against a bank supplied algorithm to determine if the account numbers fall within a valid range for that bank.

Request URL

`https://www.peachpay.co.za/API/CDV?key=yourkey`

Request structure

A CDV request consists of 3 separate sections with a root element with the name *APICDVRequest*. The *Header* section contains information relating to the CDV batch, the *Records* sections is a collection of account details to be verified and the *Totals* section is used to verify the information being sent is complete and correct.

Header

A CDV request has a single *Header* section and it consists of the following elements:

Element	Required	Description
PsVer	Yes	The version of the Peach Payments file format. Currently 2.0.1.
Client	Yes	Your unique client code.
Service	Yes	The code for the service being used. Must be CDV for check digit verification.
Reference	Yes	The batch reference. Allows you to identify the batch on the history page on the secure website.
Uniqueld	No	Used to prevent duplicate requests

Records

A CDV request has a single *Records* section with multiple *FileContents* sections consisting of the following elements:

Element	Required	Description
AccountNumber	Yes	The account number of the account to be verified.
BranchCode	Yes	The branch code of the account number to be verified.
CustomerCode	No	The customer identifier. This will be returned with results, CDV results, etc.

Totals

A CDV request has a single *Totals* section and it consists of the following elements:

Element	Required	Description
Records	Yes	The total number of CDV records being submitted.
BranchHash	Yes	The sum of all the branch codes for the CDV records.
AccountHash	Yes	The sum of all the account numbers for the CDV records.

Request Example

```
<APICDVRequest xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Header>
    <PsVer> 2.0.1 </PsVer>
    <Client> EXA001 </Client>
    <Service> CDV </Service>
    <Reference> TEST CDV </Reference>
  </Header>
  <Records>
    <FileContents>
      <AccountNumber> 559071574627 </AccountNumber>
      <BranchCode> 632005 </BranchCode>
    </FileContents>
    <FileContents>
      <AccountNumber> 111111 </AccountNumber>
      <BranchCode> 632005 </BranchCode>
    </FileContents>
  </Records>
  <Totals>
    <Records> 2 </Records>
    <BranchHash> 1264010 </BranchHash>
    <AccountHash> 9071685738 </AccountHash>
  </Totals>
</APICDVRequest>
```

Debit Orders

Overview

A debit orders request will submit data relating to a debit order batch. Once a debit order batch has been submitted, a CDV check will be performed on the account details provided and any accounts failing that check will be included in the response. Any accounts that passed the CDV check will continue to be processed.

If an account is rejected by a bank (usually because it has been closed) an unpaids response is created and POSTed back to the call-back URL provided.

Request URL

<https://www.peachpay.co.za/API/DebitOrder?key=yourkey>

Request structure

A debit order request consists of 3 separate sections with a root element with the name *APIDebitOrdersRequest*. The *Header* section contains information relating to the debit order batch, the *DebitOrders* sections is a collection of account details to be used as the destination for the debit orders and the *Totals* section is used to verify the information being sent is complete and correct.

Header

A debit order request has a single *Header* section and it consists of the following elements:

Element	Required	Description
PsVer	Yes	The version of the Peach Payments file format. Currently 2.0.1.
Client	Yes	Your unique client code.
Service	Yes	Always DebitOrder.
ServiceType	Yes	Always 2Day.
DueDate	Yes	The date the batch should be submitted to the bank for processing.
CallBackUrl	Yes	The call-back URL to be used to send back unpaids.
Reference	Yes	The reference used to identify the batch for reporting
Uniqueld	No	Used to prevent duplicate requests

Debit Orders

A debit order request has a single *DebitOrders* section with multiple *FileContents* sections consisting of the following elements:

Element	Required	Description
Initials	No	The initials of the account holder.
FirstNames	Yes	The first name(s) of the account holder.
Surname	Yes	The surname of the account holder.
BranchCode	Yes	The branch code of the account to be debited.
AccountNumber	Yes	The account number of the account to be debited.
FileAmount	Yes	The amount to be debited
AmountMultiplier	Yes	Usually 1 if the FileAmount is specified in Rands otherwise 100 if the FileAmount is specified in cents.
CustomerCode	No	The customer code used to identify the debit order in your system.
Reference	Yes	The reference that will appear on the recipient's bank statement.

Totals

A debit order request has a single *Totals* section and it consists of the following elements:

Element	Required	Description
Records	Yes	The total number of debit order records being submitted.
Amount	Yes	The total value of the debit order records being submitted.
BranchHash	Yes	The sum of all the branch codes for the debit order records.
AccountHash	Yes	The sum of all the account numbers for the debit order records.

Request Example

```

▼<APIDebitOrdersRequest>
  ▼<Header>
    <PsVer>2.0.1</PsVer>
    <Client>ZER001</Client>
    <Duedate>20140918</Duedate>
    <Service>DebitOrder</Service>
    <ServiceType>2day</ServiceType>
    <Reference>Debit Order API Example</Reference>
    <CallBackUrl>http://example.com/APIResponse</CallBackUrl>
  </Header>
  ▼<DebitOrders>
    ▶<FileContents>...</FileContents>
    ▼<FileContents>
      <Initials>CD</Initials>
      <FirstNames>Name 2</FirstNames>
      <Surname>Surname 2</Surname>
      <BranchCode>632006</BranchCode>
      <AccountNumber>12345689</AccountNumber>
      <FileAmount>12569</FileAmount>
      <CustomerCode>ZER001</CustomerCode>
      <AmountMultiplier>0.01</AmountMultiplier>
      <Reference>Example D/O API 2</Reference>
    </FileContents>
    ▶<FileContents>...</FileContents>
  </DebitOrders>
  ▼<Totals>
    <Records>3</Records>
    <Amount>2660.38</Amount>
    <BranchHash>1896016</BranchHash>
    <AccountHash>7912308218556</AccountHash>
  </Totals>
</APIDebitOrdersRequest>

```

Response structure

The standard response structure will be returned with an extra element named *BatchValueSubmitted* which contains the total value of the batch submitted (less any accounts that failed the CDV check).

The results of the CDV check will be returned as a *CDVResults* element with a collection of *Result* elements outlining the reasons for the accounts rejection. The *Result* section consists of the following elements:

Filter	Description
Result	The result of the CDV check. Will always be Invalid for a failed account.
Message	A message giving the reason for the rejection.
AccountNumber	The account number that failed the CDV test.

Please note: If any of the accounts passed the CDV check the batch will be accepted and will proceed to be processed.

Response Example

```

▼<Response>
  <Result>OK</Result>
  <BatchCode>119146</BatchCode>
  <BatchValueSubmitted>125.69</BatchValueSubmitted>
  <TotalFeeExcludingVAT>0.00</TotalFeeExcludingVAT>
  ▼<CDVResults>
    ▼<Result>
      <Result>Invalid</Result>
      <Message>Branch code not found.</Message>
      <AccountNumber>12345689</AccountNumber>
      <BranchCode>632006</BranchCode>
    </Result>
    ▼<Result>
      <Result>Invalid</Result>
      ▼<Message>
        Account number is too long (appears to be a credit card).
      </Message>
      <AccountNumber>7912172416078</AccountNumber>
      <BranchCode>632005</BranchCode>
    </Result>
    ▼<Result>
      <Result>Valid</Result>
      <AccountNumber>123456789</AccountNumber>
      <BranchCode>632005</BranchCode>
    </Result>
  </CDVResults>
</Response>

```

Unpays

An unpaid occurs when the bank rejects a debit order. This can happen for many reasons like the account being closed. When this happens, an unpays response is POSTed back to your server using the *CallbackUrl* provided in the initial request. All response data are submitted as a form data in a POST action (i.e. application/x-www-form-urlencoded) with the key "response".

Note: The unpaid responses can continue for several days after the batch is submitted.

The *Response* root element will consist of the following elements:

Filter	Description
Result	Will always be OK.
BatchCode	The unique code for the batch the unpays belong to.
PaymentResults	The collection of <i>Results</i> elements with more information about each unpaid.

The *DebitOrderResults* section consists of the following elements:

Element	Description
FirstName	The first name(s) of the account holder.
Surname	The surname of the account holder.
BranchCode	The branch code of the account to be debited.
AccountNumber	The account number of the account to be debited.
CustomerCode	The customer code used to identify the debit order in your system.
Reference	The reference that would have appeared on the recipient's bank statement.
Result	The result status which will always be Rejected for an unpaid.
ResultMessage	The reason for the unpaid.

Unpays Example

```

▼<Response>
  <Result>OK</Result>
  <BatchCode>119138</BatchCode>
  ▼<DebitOrderResults>
    ▼<Result>
      <AccountNumber>12345689</AccountNumber>
      <BranchCode>632006</BranchCode>
      <FirstName>Name 2</FirstName>
      <Surname>Surname 2</Surname>
      <Reference>Test D/O API 2</Reference>
      <CustomerCode>ZER001</CustomerCode>
      <Result>Rejected</Result>
      <ResultMessage>AUTHORISATION CANCELLED</ResultMessage>
    </Result>
  </DebitOrderResults>
</Response>

```