NETBANX Product Management

UPP
Functional specification

Version **1.6**

Date: 18/01/10

3rd Floor, Mount Pleasant House, Mount Pleasant, Cambridge CB3 0RN
Telephone +44 (0)1223 446030
Facsimile +44 (0)1223 446021
http://www.netbanx.com

Copyright Notice

NETBANX UPP Integration Guide
Version - 1.6
Publication Date – January 2010

Table of contents

1 Purpose

This document will constitute a generic requirements specification document for the merchant's system on NETBANX.

2 Version changes

Version no	Change	By whom	Date
1	Initial version	NT	12/09/08
1.1	Re-formatted throughout whilst keeping existing functionality: Added definitions for CNP and MOTO (section 3) Updated Summary to explain workflows (section 4) Changed NETBANX target URLs (sections 5.1, 5.2, 10) Enhanced required fields section to separate into mandatory & optional with descriptions/usage (section 5.1.3) Fuller explanations for passed/failed/error scenarios (section 5.2) Updated process flow diagram (section 10) New sections for: • Redirections (new section 7) • Use of 3rd party tracking tools (new section 8) • HTML Template Usage (new section 11)	NT	30/12/08
1.2	Changed merchant reference name to **nbx_merchant_ref**	NT	31/12/08
1.3	Renamed **nbx_** fields to allow greater flexibility (Section 5.1.3) Added restrictions on CGI call return data for added security (Section 6) Added feature to retry failed CGI calls (Section 6) Changed workflow so that failed transactions have their own failure page (Sections 4, 5.2, 10) 3rd party tracking tools under review (section 8)	NT	26/01/09
1.4	Corrected workflow examples (Section 4) Made nbx_currency_code mandatory (Section 5.1.3) Made ; reserved character (Section 5.1.3) Added optional fields nbx_success_show_content and nbx_failure_show_content (Sections 5.1.3, 6) Changed values for payment types (Section 5.1.3) Clarified position on 3rd party tracking tools (Section 8) Added detail on CGI queuing (Section 6)	NT	06/03/09
1.5	Added copyright; amended header and footer throughout (all pages). Amended typo in Section 4 Clarified note on restricted fields (Section 5.1) Added option for Major currency units (Section 5.1.3) Removed Poli as a payment option (Section 5.1.3) Clarified note on Added facility to exclude certain fields from CGI call and Redirection (Sections 5.1.3, 6.1, 6.2) Added distinction between nbx_payment_types and nbx_payment_type (Section 5.1.3, 6.1, 7)	NT	01/12/09

	Removed semicolon restriction (Section 5.1.3) Added note about cross site scripting (Section 5.1.4) Added facility to control which card types are accepted on a per-transaction basis (Section 5.1.3, 5.1.8) Added facility to manage multiple merchant numbers within the same integration (Section 5.1.3) Added extra security options to block certain transactions based on IP/country (5.1.2) Added restriction to nbx_success_url and nbx_failure_url (Section 5.1.3) Added section on Cross Site Scripting (New section 5.1.4) Amended NETBANX link description (Section 5.2.1) Removed nbx_payment_amount from parameter list (since this is included in "all parameters) (Sections 6.1, 6.2) Amended value of nbx_status for successful transactions (Section 6.1) Added option to pass back masked PAN (card number) (Section 6.1, 7) Added workflow option to limit the number of transaction attempts (Sections, 5.1.4, 10) Added facility to monitor failed CGI calls (New Section 6.3. Moved some elements of 6.1 and 6.2 to 6.3) Added extra field passed back in background call nbx_eci (Sections 6.1, 6.2) Added more checksum encryption options (section 9) Added facility to control which fields are included in the checksum (Section 9) Amended "Payment Page Structure" (Section 11)		
1.6	Added Store card checkbox on payment page Added Pay by last card Option Added "local" option for nbx_auth_type	TC	18/01/10

3 Definitions

Customer The end user of the system, who is buying services or products from the client

Client The merchant

Products The services on offer from the client

Browser The piece of software used to navigate the web sites involved

NETBANX domain netbanx.com unless advised to the contrary by NETBANX

Product selection The process of the customer following a specified path within client Website, so as to arrive at the NETBANX having made a choice to buy a specified service from the client.

Payment process The entering of payment details on the NETBANX secure web server, the subsequent charging of the payment type (if authorised), the displaying of confirmation messages on the NETBANX server, and the sending of emails from the NETBANX server

Post authorisation Any process which occurs outside the NETBANX domain following payment authorisation.

Payment page/form The payment details capture page on the NETBANX server which contains a form prompting for details specific to the payment type.

NETBANX reference The string of 18 alphanumeric characters generated by NETBANX on authorisation. This reference is given to the customer and used by NETBANX customer services to locate transaction details.

Transaction ID/Trans ID The unique, numerical ID for each transaction. This ID is passed to the acquiring bank in the authorisation and settlement request.

CGI Call A request made over HTTP or HTTPS, made by the NETBANX server to simulate a browser GET or POST request to a web page or service.

We Unless otherwise specified, NETBANX

Us Unless otherwise specified, NETBANX

UPP Unified Payment Page. The latest NETBANX system which handles multiple payment types within a single payment page.

CNP Cardholder Not Present. Refers to all transactions where the *card* is not physically present at the terminal device which is taking the payment.

MOTO A CNP merchant number provided by the bank for Mail Order / Telephone Order use, where the merchant is initiating the transaction on behalf of the customer.

4 Summary of process control flow

The NETBANX UPP system is based on workflows. A workflow is a sequence of events in a particular order. This specification describes the workflow that comes with the standard system. Functionality can be added or moved around on request, by the addition of a module, or rearrangement of the workflow. If a different workflow is requested, a different target URL will be provided to you (see section 5.1)

For example, the standard system consists of the following steps (and is visually depicted in the diagram in section 10)

1. Product selection page
2. NETBANX payment page
3. NETBANX payment authorisation
4. Optional CGI Call (dependent on field passed to NETBANX)
5. Optional wallet creation (dependent on payment types requested)
6. NETBANX Error/Receipt page OR Redirection

A revised workflow may redisplay the payment page instead of a failure page or redirecting to the merchants website, for step 6.

Another addition to the workflow, could be to keep track of the number of transaction attempts made, and display/hide specific methods after each attempt.

Workflows may also need to be changed to accommodate different shopping carts.

Please email `integrations@netbanx.com` if you want to make any changes to the standard workflow. A charge may be involved depending on the nature of the change.

5 Details of the process

5.1 PRODUCT SELECTION

The customer will navigate the client website by means of a browser to view the client services.

The activities of a customer inside the client website are of no importance to NETBANX *from a technical point of view*. However, when a customer wishes to purchase products via the NETBANX payment gateway, it is important that certain information is passed to the system.

Therefore, at the end of the product selection process, links to the NETBANX domain are detailed as follows:

There shall be an HTML form, the action of which is to call the following URL. Either **POST** or **GET** methods may be used, and data can also be appended in the query string, although **POST** is recommended: **`https://pay.netbanx.com/MERCHANT`**

Where **`MERCHANT`** is a value we provide to you once you are integrated.

If a different workflow has been requested (e.g. you require some functionality that falls outside of this specification) then a different URL and supplementary specification will be given to you.

Each instance of the system, where the target URL is different, is known as an integration. The integration will be fixed according to this specification. Where functionality is required such that the integration is either not flexible enough to accommodate multiple scenarios, or that the requirements do not logically belong to the same group of merchant numbers, then a new integration will be required for each group of requirements.

5.1.1 Referring page

For bureau merchants, there is a bank requirement to perform a check on the referring URL, by verifying the contents of the environment variable HTTP_REFERER against a list of authorised websites. Direct merchants can have this check disabled. To verify that the customer has not tampered with the input fields (for example the payment amount), both bureau and direct merchants may use a checksum to "sign" the fields. This feature is described in section 9.

5.1.2 IP Checking

Virtual Terminal integrations will have a check for the REMOTE_ADDR environment field against a known set of IPs. This is to ensure that payments are not attempted from outside of the merchant's controlled environment where the customer's identity can be correctly identified. An error sequence will be invoked as per section 5.2.4 if the IP does not match any of the IPs in the list pre-agreed with NETBANX.

Additionally, some merchants may wish to insist on the geographical location of the customer (based on the IP address) matching the card issuing country. An error sequence will be invoked as per section 5.2.5 if this check fails. Please indicate in the Integration Questionnaire if you want this facility. You should be aware that proxy servers may mask the actual location of the customer; we can also block transactions from known anonymous proxy servers.

5.1.3 Passing of parameters

Please note that variable names are case sensitive and should be passed as specified below.

The client should pass any variables that are required for them to check the payment and order since they will be returned following the transaction (See sections 6.1 & 6.2)

The names of these variables must **not** be any of the following as they can potentially interfere with NETBANX systems.

Field name	Reason
`nbx_<anything>`	Fields beginning with `nbx_` are reserved so must not be passed, unless they have been explicitly listed as required/optional further n the section below.
`submit`	Passing this field will interfere with the automatic form submission process used during 3D Secure. When declaring a submit button on your site, you should give it a name other than submit. E.g. `<input type="submit" name="submitform" value="Pay now">`

For the system to operate NETBANX will be expecting the following variables to be passed: Please note that variable names are case sensitive and should be passed as specified below. Any other variables may be passed for tracking purposes, as they will appear in the merchant email and CGI call automatically. NETBANX need not be notified in the future to accommodate such extra variables unless some specific processing is required on them.

Mandatory fields

Field name	Usage	Format
nbx_payment_amount	The amount to authorise against the customer's account	Minor units by default. I.e. Digits only, so 999 will authorise £9.99 (or other currency as appropriate). Please indicate on the questionnaire if major currency units are to be used instead.
nbx_currency_code	Used to determine the currency against which to authorise the card. If not passed, then GBP will be used. If an unrecognised[‡] currency is passed, then an error page will be displayed and the payment will not be processed. As it is used in deriving the value for nbx_checksum it is mandatory if the checksum method is required.	ISO4217 e.g. GBP or USD
nbx_merchant_reference	Recorded in the reporting tool within search and download facilities. For example the merchant may wish to record the customer's order number. Also used in deriving the value for nbx_checksum hence if the checksum feature is required then it is mandatory.	Any character may be used. The length is limited to 50 characters. A delimiter can be used if the field is used to record multiple pieces of information. The delimiter character is configured in the reporting tool.
nbx_checksum	To verify the request has come from a legitimate system. If the merchant wishes to use this feature then it will be a mandatory field.	20 hex pairs using lower case letters. See section 9.2

Optional fields

Field name	Usage	Format
nbx_language	Used to determine which language in which to display the payment page. If no value is passed, then English will be used.	ISO 639-1 E.g. en or de This field may instead be passed as part of the URL e.g. https://pay.netbanx.com/MERCHANT/de or https://pay.netbanx.com/de/MERCHANT
nbx_payment_types	Used to determine which payment types to be made available to the customer. Where no value is passed, all available types are displayed.	See table below. The order in which the fields are listed determines the order they are displayed on the screen.
nbx_success_url	The URL to call back in the event of a successful authorisation (see section 6.1)	Internet-visible fully qualified URL. Standard ports must be used (80 for http/443 for https)

[‡] Note that NETBANX will only be able to send authorisation to the bank in the requested currency if the appropriate merchant account has been set up and approved for that particular currency.

nbx_failure_url	The URL to call back in the event of a failed authorisation (see section 6.2)	Internet-visible fully qualified URL Standard ports must be used (80 for http/443 for https)
nbx_return_url	The URL for the customer to link back to in the case of an aborted payment (see sections 5 and 6	Fully qualified URL (may not necessarily be internet visible for internal testing - e.g. http://localhost)
nbx_success_redirect_url	The URL to redirect to in the event of a successful authorisation (see section 7)	Internet-visible fully qualified URL
nbx_failure_redirect_url	The URL to redirect to in the event of a failed authorisation (see section 7)	Internet-visible fully qualified URL
nbx_email	If passed, the value is used to pre-fill the payment form with the primary account holders email address.	As per RFC5322 http://www.rfc-archive.org/getrfc.php?rfc=5322&tag=Internet-Message-Format
nbx_cardholder_name	If passed, the value is used to pre-fill the payment form with the primary account holder's name.	Any character. No limit.
nbx_houseno	If passed, the value is used to pre-fill the payment form with the first line of the address of the primary account holder. This is used for AVS checking with the issuing bank where available. It is the bank's decision to authorise or decline the transaction based on this information.	Any character. No limit. Only the numeric values are sent to the bank.
nbx_postcode	If passed, the value is used to pre-fill the payment form with the postcode of the primary account holder. This is used for AVS checking with the issuing bank where available. It is the bank's decision to authorise or decline the transaction based on this information.	Any character. No limit. Only the numeric values are sent to the bank. If the customer lives in an area where there is no postcode, then the value "NONE" may be passed.
nbx_success_show_content	If passed with a value of 1, the background call as defined by nbx_success_url is triggered immediately. The receipt page is not displayed (or redirection is not invoked) until the remote server has responded. If the remote server cannot be reached or an error is returned, then the call is queued.	Boolean - 1 or 0
nbx_failure_show_content	As above, except for failed transactions and nbx_failure_url	Boolean - 1 or 0
nbx_redirect_exclude	Used to determine which fields should be excluded from the redirection.	comma-separated list of fields to be excluded from the redirection (Refer to section7) e.g. "field1,field2" If required, these values may also be hardcoded within the merchant's configuration file on the NEBANX server instead of being passed for each transaction.
nbx_cgi_exclude	Used to determine which fields should be excluded from the CGI Call.	comma-separated list of fields to be excluded from the CGI call (Refer to section 6) e.g. "field1,field2" If required, these values may also be hardcoded within the merchant's configuration file on the NEBANX server instead of being passed for each transaction.
nbx_card_types	Used to determine which card types should be displayed and allowed to the customer This facility cannot be used to allow cards which have been excluded at NETBANX's end. Instead it is used to choose cards out of the merchant's available list. Any types that are passed which are not available to the merchant will be ignored.	comma-separated list of card types If no value is passed, then all allowable card types for the merchant will be allowed. Possible values: 21 => "UK Maestro", 22 => "Int. Maestro", 23 => "Solo", 24 => "Visa Debit", 25 => "Visa Electron", 26 => "Visa Credit", 27 => "Mastercard", 28 => "Amex", 29 => "Diners", 30 = "JCB"
nbx_interface	Will determine which merchant ID to use. Instead of passing the actual ID, a lookup table will be used.	Alphanumeric characters e.g. SITE1 The merchant number will then be determined using a combination of nbx_interface, nbx_currency_code and the card type used by customer.

`nbx_auth_type`	Will determine whether the transaction is settled immediately. Auth-only transactions will need to be settled either via Netcentre or the API. Valid for credit/debit card payments only.	Possible values: "auth" (perform authorisation only) "bill" (perform authorisation & immediate settlement) “local” (perform a local authorisation to verify card format details, no request is sent to the acquirer)
`store_card_prompt`	If passed with a value of 1 a checkbox will be displayed asking the customer “Store card details for future use?” if passed with 0 or not sent then no checkbox will be displayed The result of the checkbox will be returned within “store_card_indicator”	“store_card_indicator” Boolean - 1 or 0 “store_card_prompt” Boolean - 1 or 0
`nbx_previous_reference`	If passed with a valid “nbx_netbanx_reference” then the payment page will be displayed with a masked card number and the expiry date of the previous transaction. The Credit Card box will be locked not allowing the customer to make any changes. If an invalid “nbx_netbanx_reference” is passed then an error page will be displayed to the customer	The unique NETBANX reference for a previous transaction (“nbx_netbanx_reference)

The **`nbx_payment_types`** field will determine which payment type is displayed to the end customer and may contain one of the following values:

VALUE	**Payment type**
`card`	Credit/Debit Card
`directpay24`	DirectPay24/Direct e-banking
`neteller`	NETELLER wallet
`ukash`	Ukash
`paypal`	PayPal
`ideal`	iDeal

Please contact our sales team for confirmation on which payment types are currently available.

If multiple options are to be displayed, the values must be separated by commas. e.g. **`nbx_payment_types="ukash,card".`** The ordering of these values determines the order in which they are displayed, the first being on the top.

If **`nbx_payment_types`** is not passed, or is passed with no value, then all available types for the merchant will be displayed.

The actual method chosen by the customer will be passed back using the field **`nbx_payment_type`** (See Sections 6.1, 7)

For cards, all potentially available cards (the full set available to the merchant) will be displayed. To restrict the customer to using a particular card or set of cards, **`nbx_card_types`** must be passed with a comma-separated list of acceptable card types. If **`nbx_card_types`** is passed with no value, no cards will be accepted; to accept all cards the field must either contain all possible values, or the field must not be passed at all.

If the merchant passes a card type that hasn't been configured by NETBANX, the card type will still be displayed on the payment page, but will be rejected once submitted by the customer, so this feature cannot be used to start accepting new types of card.

5.1.4 Cross-site scripting

To prevent the risk of malicious code being introduced into the system which could interfere with cardholder data, certain characters will be stripped from field values. At the time of writing these characters are as follows (more may be added as and when further exploits are discovered)

" (double-quotation mark)

```
< (less than)
> (more than)
```

Prior arrangement needs to be made with the integrations team to exclude any fields from this check. Such fields are advised to be included in the checksum to ensure data integrity.

5.2 NETBANX AUTHORISATION

5.2.1 Display of payment page and entering of payment details

When the form POST or GET request to **`https://pay.netbanx.com/MERCHANT`** is made with the mandatory fields, a payment form will be displayed, allowing the customer to enter their payment details. This form may be designed by the client, based on the template file supplied by NETBANX, or a default page may be used.

If any of the following checks fail, then the payment page will not be displayed and instead we will proceed directly to the error sequence in section 5.2.4

- For MOTO transactions, IP check failure (refer to section 5.1.2)
- Mandatory fields not passed or incorrectly formatted (refer to section 5.1.3).
- Checksum failure (refer to section 9)
- For any transaction type, if the IP has been blacklisted from previous prohibited activity with any NETBANX merchant (e.g. persistent fraud/DDOS attempts).
- If the number of payment attempts allowed by the merchant has been reached (this may be specified in the Integration Questionnaire)

The client may elect to have certain form fields mandatory and/or non-editable on the payment form. If items are to be marked as non-editable, then they need to have been passed to the payment form if they are also mandatory. The field names that can be pre-filled, such as ***`nbx_postcode`*** are indicated in italics in section 5.1.3.

If the fields are editable on the payment page, when they are reported back to the merchant, the values carried will be those as entered by the customer. Therefore if the merchant wishes to get back 2 values, the value as passed from the site, and the value entered by the customer, then they need to choose a different name to carry the cardholders details and we will use that field to pre-fill the payment form instead.

Once the customer has entered their payment details and submitted them to NETBANX, we will attempt carry out preliminary checks prior to account authorisation with the relevant payments provider or third party system.

5.2.2 In the event of a successful transaction

- An email may be sent to **THE MERCHANT**, containing all the information that was originally passed to the payment form, with the exception of sensitive payment details, and additionally the NETBANX Reference for the transaction.

- An email from NETBANX will be sent to the customer, detailing the amount of the transaction, and the corresponding NETBANX reference.

- A CGI Call to a script on your server may be performed, as detailed in section 6.1.

- The receipt page is displayed to the customer on the **NETBANX** server. Alternatively a redirection may be performed as detailed in section 7.

- Where applicable to the merchant, payment method, and where the customer has requested it, a NETELLER wallet is created.

5.2.3 In the event of a failed transaction

- An email may be sent to **THE MERCHANT**, containing all the information that was originally passed to the payment form, with the exception of sensitive payment details. *There will be no NETBANX Reference number for failed transaction emails*.

- No email will be sent to the customer

- A CGI Call to a script on your server may be performed, as detailed in section 6.2.

- EITHER The customer is redirected back to the payment page on the **NETBANX** server. The customer has the option to go back to the merchant's website using the **`nbx_return_url`** link.

  OR The customer is shown an error/decline page. The customer has the option to go back to the merchant's website using the **`nbx_return_url`** link.

  The default workflow will display the error/decline page. Redirecting back to the payment page following a failure is not appropriate for all payment types.

5.2.4 In the event of an pre-payment-page error

- No email will be sent to the merchant

- No email will be sent to the customer.

- No CGI call will take place.

- An Error page is displayed to the customer by the **NETBANX** server. The customer has the option to go back to the merchant's website using the **`nbx_return_url`** link.

A pre-payment page error occurs when the input or security checks are rejected by NETBANX.

5.2.5 In the event of an pre-transaction error

This sequence of events will be followed in the event that the customer attempts to pay using a card type that the merchant has excluded for that transaction, or that NETBANX has generally excluded for that merchant (e.g. BIN ranges from a certain country). In the case where a card type is theoretically allowed for the merchant but excluded for that transaction, appropriate wording will explain that the card is potentially allowable by returning to the merchant site.

- No email will be sent to the merchant.

- No email will be sent to the customer.

- No CGI call will take place.

- EITHER The customer is redirected back to the payment page on the **NETBANX** server. The customer has the option to go back to the merchant's website using the **`nbx_return_url`** link.

  OR The customer is shown an error/decline page. The customer has the option to go back to the merchant's website using the **`nbx_return_url`** link.

  The default workflow will display the error/decline page. Redirecting back to the payment page following a failure is not appropriate for all payment types.

A pre-transaction error is defined as a transaction attempt being rejected by NETBANX rather than by the payment network, after the user has attempted a payment, e.g. missing mandatory field, or a security check.

6 CGI Callback

6.1 Successful Authorisation

After a successful transaction authorisation has been attempted the NETBANX script will perform a CGI Call to URL as defined in the value of the field **nbx_success_url**

NETBANX is able to hard-code a value for this field in a configuration file on the NETBANX server. If the field **nbx_success_url** isn't passed to NETBANX, this hard-coded value will be used instead. If no value has been pre-arranged, then no CGI call will take place.

The following parameters are to be passed to the script using the POST method:

If different fields are required to be sent, then the field **nbx_cgi_exclude** needs to be used to prevent all fields mentioned in Section 5.1.3 from being sent. The fields to be excluded may also be hardcoded in a configuration file on the NETBANX server.

Please note that variable names are case sensitive, so will be passed exactly as specified below.

• ***All mandatory and optional parameters if passed by the merchant as described in section 5.1.3***	
• **nbx_status**	**- the transaction status**
• **nbx_email**	- as entered by the customer on the payment page unless the fields are non-editable in which case they are passed back to the merchant unaltered.
• **nbx_cardholder_name**	
• **nbx_houseno**	
• **nbx_postcode**	
• **nbx_netbanx_reference**	**- the unique[1] NETBANX reference for the transaction**
• **nbx_masked_pan**	**- The first 4 and the last 4 digits of the card number[2]**
• **nbx_checksum**	**- A new checksum as described in section 9**
• **nbx_CVV_auth**	**- The result of the CVV check as received from the bank**
• **nbx_houseno_auth**	**- The result of the AVS check as received from the bank**
• **nbx_postcode_auth**	**- The result of the AVS check as received from the bank**
• **nbx_eci**	**- The ECI status as received from the bank**
• **nbx_retries_remaining**	**- The number of retry attempts remaining on this CGI call**

The **nbx_status** parameter will contain the value “passed”.

Possible values for **nbx_eci**. Please consult with your acquirer concerning chargeback liability shift.

- ECI absent or empty - Failed Authentication[3], or not a 3D secure transaction, no liability shift.
- *01* - Incomplete Authentication (attempted cardholder not enrolled) (MasterCard)
- *02* - Successful Authentication (MasterCard)
- *05* - Successful Authentication (Visa)
- *06* - Incomplete Authentication (attempted cardholder not enrolled) (Visa)
- *07* - Authentication Couldn't Be Performed (Visa)

The script may return some text to be displayed within the receipt page. At the time of writing, full HTML may not be used. Please refer to the HTML section of the online procedure guide for details on the positioning of this text.

The script may return some output as acknowledge that it has completed correctly, but the output is not displayed to the customer. In order for the output to be parsed and recognised by NETBANX, a pre-agreed format must be specified. Unless stated otherwise in the questionnaire, the format is as follows.

```
NETBANXOK:1
```

This text must be on a line on its own. Any other output is ignored.

The output will be used to determine whether to queue the call, to be retried (see section 6.3)

[1] During testing, the NETBANX reference will not be unique; a constant value of 1234567890ABCDEFGH is used.

[2] Example format: "4921 xxxx xxxx 1230". The number of x's between the digits are not necessarily representative of the actual card length; there will always be 2 sets of 4 x's. This feature is not provided as standard but available on request.

[3] Failed authentication will result in a Failed transaction. For successful payments, empty will indicate not 3D Secure.

6.2 FAILED AUTHORISATION

After a failed transaction authorisation has been attempted the NETBANX script will perform a CGI Call to URL as defined in the value of the field **`nbx_failure_url`**

NETBANX is able to hard-code a value for this field in a configuration file on the NETBANX server. If the field **`nbx_failure_url`** isn't passed to NETBANX, this hard-coded value will be used instead. If no value has been pre-arranged, then no CGI call will take place.

The following parameters are to be passed to the script using the POST method:

If different fields are required to be sent, then the field **`nbx_cgi_exclude`** needs to be used to prevent all fields mentioned in Section 5.1.3 from being sent. The fields to be excluded may also be hardcoded in a configuration file on the NETBANX server.

Please note that variable names are case sensitive, so will be passed exactly as specified below.

• ***All mandatory and optional parameters if passed by the merchant as described in section 5.1.3***	
• **`nbx_status`**	**- the transaction status**
• **`nbx_email`**	- as entered by the customer on the
• **`nbx_cardholder_name`**	payment page unless the fields are
• **`nbx_houseno`**	non-editable in which case they are passed
• **`nbx_postcode`**	back to the merchant unaltered.
• **`nbx_checksum`**	**- A new checksum as described in section 9**
• **`nbx_CVV_auth`**	**The result of the CVV check as received from the bank**
• **`nbx_houseno_auth`**	**- The result of the AVS check as received from the bank**
• **`nbx_postcode_auth`**	**- The result of the AVS check as received from the bank**
• **`nbx_eci`**	**- The ECI status as received from the bank.**
• **`nbx_retries_remaining`**	**- The number of retry attempts remaining on this CGI call**

The **`nbx_status`** parameter will contain the value “declined", "error”, or “pending”. If the result is “pending”, the get_status API called can be used to check the status at a later time.

The script may return some text to be displayed within the receipt page. At the time of writing, full HTML may not be used. Please refer to the HTML section of the online procedure guide for details on the positioning of this text.

The script may return some output as acknowledge that it has completed correctly, but the output is not displayed to the customer. In order for the output to be parsed and recognised by NETBANX, a pre-agreed format must be specified. Unless stated otherwise in the questionnaire, the format is as follows.

```
NETBANXOK:1
```

This text must be on a line on it's own. Any other output is ignored.

The output will be used to determine whether to queue the call, to be retried (see section
If the call to your server fails, the call will queued and re-tried every 5 minutes. Once the call has failed for a certain period of time, the call will no longer be attempted. These settings can be altered on a per-integration basis.

A hyperlink will be present on the failure page to allow the customer to navigate back to the merchant's website, so that they do not get stuck on the NETBANX website. The target of the hyperlink is defined in the field name **`nbx_return_url`**

The value of **`nbx_return_url`** must be the full URL, including the http:// or https:// path and any relevant fields required to be passed back. E.g.

`https://www.merchant.com/paynow?customer_id=12345`

6.3 CGI CALL QUEUEING

The merchant may return some specific data (defined in the sections above) in the body of the response. The purpose of is to assist in both NETBANX and merchant support staff in ensuring that the script executed successfully. It also allows some extra complexity in the logic to be applied, so that even if the webserver returns a positive response, the application still needs to have completed its internal tasks to satisfy a positive outcome.

By default, the background call is not made immediately. Instead it is put in a queue, which is run each 5 minutes (this frequency value is a system-wide setting). Items in the queue can consist of new calls and calls to be retried. The queue manager will keep track of how many times the call has been attempted. Each integration can have a limit, after which point no further attempts will be made and emails are sent out to alert the relevant parties. By default this limit is set to 3 for each transaction. The email recipients may include relevant merchant and/or 3rd party staff who need to be alerted to potential connectivity problems or application failure

To execute the call immediately, the field **`nbx_success_show_content`** must be passed (or **`nbx_failure_show_content`** for failed transactions). If this field is passed with any value other than "0", the system will wait until the call has completed (either successfully or timed out) before going on to the next step (display of receipt or failure page). If this field is used and the call is unsuccessful, then the call will be queued for retrying later.

A failed call is defined as an HTTP return code of anything other than 200 (OK), or if a timeout value has been exceeded.

There are likely to be 4 different reasons for a CGI call failing.
1) Transient exception due to input particular to that call (e.g. unexpected character)
2) Permanent exception due to code failure
3) Transient connectivity issue
4) Permanent connectivity issue

In the case of 2) and 4), notifications for each individual event is likely to swamp the recipient therefore the following policy is applied:

Each merchant integration will have a threshold which will define the maximum number of failures permitted within a 1 hour period. Each hour, a process will run and look at the status of terminally failed calls (where the number of attempts has been reached). If the number of terminally failed calls has not exceeded the threshold, the email will go to a particular recipient list.

If the number of terminally failed calls exceeds the threshold, the Subject line will change and a different recipient list may be used. The body of the email will be the same.

So for example: number of failures does not exceed threshold *(the fields listed below are examples and not representative of previously agreed names/values/formats)*

```
To: helpdesk@merchant.com
From: support@netbanx.com
Subject: CGI Failure report for MERCHANT

There were 2 failures in the last hour:

POST to https://www.merchant.com/netbanx-
callback/response.asp?nbx_netbanx_reference=ABCDEFGH1234567890&nbx_merchant_ref=ABC

POST to https://www.merchant.com/netbanx-
callback/response.asp?nbx_netbanx_reference=ACEGIKMO1234567890&nbx_merchant_ref=DEF
```

And if the threshold is exceeded

```
To: helpdesk@merchant.com, helpdesk@technical3rdparty.com, operations@netbanx.com
From: support@netbanx.com
Subject: CRITICAL CGI Failure report for MERCHANT. Threshold of 100 failures exceeded.

There were 101 failures in the last hour:

POST to https://www.merchant.com/netbanx-
callback/response.asp?nbx_netbanx_reference=ABCDEFGH1234567890&nbx_merchant_ref=ABC
```

```
POST to https://www.merchant.com/netbanx-
callback/response.asp?nbx_netbanx_reference=ACEGIKMO1234567890&nbx_merchant_ref=DEF

... repeated for each call ...
```

The following elements may be provided to NETBANX (via the Integration Questionnaire):

- Number of attempts for each individual call before it is considered to have terminally failed (max_retries)
- Number of terminally failed calls per hour, per integration before the integration is considered in a critical state (threshold)
- Email recipient list to be notified of terminally failed calls (email_list)
- Email recipient list to be notified when number of terminally failed calls reaches a critical state (critical_email_list)

The following elements are global and cannot be altered

- The frequency between attempts may not be changed from 5 minutes.
- Notification cannot be sent out for each individual failed attempt
- Notification cannot be sent out for each individual terminally failed call (they will be sent as a group per hour)

7 Redirection

NETBANX does not recommend this method as a substitute for automatic background call, however for navigational purposes, merchants may opt to have a redirection to their server instead of the NETBANX hosted receipt page. If the field `<nbx_success_redirect_url>` or `<nbx_failure_redirect_url>` are not passed to NETBANX, or are passed with a NULL value, then no redirection will occur.

The redirection to your server is achieved by the NETBANX script sending the following response to the browser:

```
Status: 302 Moved\n
Location: <nbx_success_redirect_url>?<fields>\n
\n
```

Where `<nbx_success_redirect_url>` is the value of this field as passed to NETBANX. `<nbx_failure_redirect_url>` is used for failed transactions.

`<fields>` represent name=value pairs as described in section 6.1, except that they are sent within the query string rather than within the POST body.

If different fields are required to be sent, then the field **nbx_redirect_exclude** needs to be used to prevent all fields mentioned in Section 5.1.3 from being sent. The fields to be excluded may also be hardcoded in a configuration file on the NETBANX server.

`\n` represents a new line character.

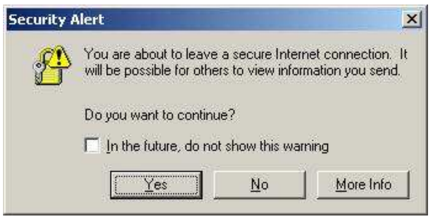
For example (the entire list of fields and correct values are not shown in this example)

```
Status: 302 Moved\n
Location: https://www.merchant.com/from_netbanx.aspx?ref=123&nbx_netbanx_ref=123456789\n
\n
```

There are 3 issues that you should be aware of as follows:

1) Fields can only be appended to the query string. The POST method cannot be used. Therefore relevant encoding and length restrictions as per the appropriate RFC guidelines need to be considered

2) If the URL we are redirecting to is an http address instead of https, then most browsers, if the user has not disabled this option, will display a warning that they are being redirected to a non-secure area, as shown in the image on the right. Some users may find this warning disconcerting and hence elect not to continue.



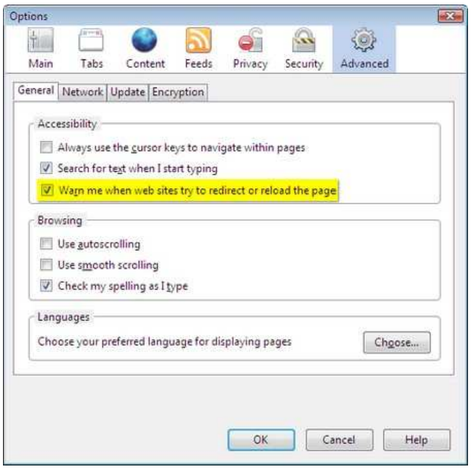
Security Alert

You are about to leave a secure Internet connection. It will be possible for others to view information you send.

Do you want to continue?

□ In the future, do not show this warning

3) Merchant should be aware of browser compatibility issues and perform regular testing within all possible environments to minimize potential customer problems, for example should certain browser settings may prevent the redirection from occurring (see image on the right)



Options

Accessibility

□ Always use the cursor keys to navigate within pages
☑ Search for text when I start typing
☑ Warn me when web sites try to redirect or reload the page

Browsing

□ Use autoscrolling
□ Use smooth scrolling
☑ Check my spelling as I type

Languages

Choose your preferred language for displaying pages

8 Use of 3rd party tracking tools

In general terms, any third party tracking tool can be used, however you should be aware of the following:

- The code may only be placed on pages where card data is not collected (i.e. receipt or decline/error pages)
- The code must be hosted on a secure domain
- Some affiliate packages require sub-totals, being the grand total minus tax and shipping. This will need to be passed as a separate field to the `nbx_payment_amount`

8.1 GOOGLE ANALYTICS

Google analytics has been successfully integrated before into NETBANX. This is achieved using a piece of JavaScript code hosted on the secure Google server (`XXXXXXX` below is replaced with the merchant's ID supplied by Google).

```
<!-- Begin Google Analytics code -->

<script src="https://ssl.google-analytics.com/urchin.js" type="text/javascript">
</script>
<script type="text/javascript">
  _uacct = "UA-XXXXXXX";
  _udn="none";
  _ulink=1;
  urchinTracker();
</script>

<!-- End Google Analytics code -->
```

8.2 WEB TRENDS

There are many different versions of this package, so more information would be required. Provided the code is similar to the above, i.e. hosted on a secure site, there is no problem.

Any custom fields, e.g. an order ID passed by the merchant, can be inserted using the convention below

```
[% order_id %]
```

The system will render the code above to display the value of the field "`order_id`".

9 Checksum requirements

9.1 INTRODUCTION AND CONCEPTS

Checksum technology adds a signature of authentication to data that is passed between two systems. As a result, the receiving system has more assurance that the data received from the sender is authentic.

This complements SSL technology since SSL technology ensures privacy of data in transit and the integrity of data in transit. By employing both technologies therefore, it is possible to ensure that nothing has occurred to the data in transit, what is received is what was sent by the originating system and that the data was sent by the originating system and not another system pretending to be the original.

Without this extra checksum it would be possible for a malicious user to spoof the IP address of The Merchant and send valid SSL packets that would be legitimate but containing malicious data. A malicious user would need to know the secret (checksum) key which, along with the source data forms the checksum, in order to circumvent this additional measure of security.

The standard method used is SHA1. Merchants may also elect to use MD5, HMAC MD5 or HMAC SHA1.

9.2 OVERVIEW OF PROCESS AND EXAMPLE

An encryption module is required on both source and destination servers in order to perform the necessary functions described below. Such modules are available for most platforms free of charge.

The checksum is created by signing a piece of text with the secret key.

When **passing** information to NETBANX, the piece of text to be signed is the values of the following fields:
nbx_payment_amount nbx_currency_code nbx_merchant_reference

If other sensitive information is being passed which should not be tampered with (e.g. acceptable card types) then these can also be included to make up the checksum. Please indicate this on the Integration questionnaire.

E.g. if you have these fields

```
<input type="hidden" name="nbx_payment_amount" value="100">
<input type="hidden" name="nbx_currency_code" value="GBP">
<input type="hidden" name="nbx_merchant_reference" value="12345">
```

And the secret key is "SECRETKEY" (without the quotes)

The string to be signed will be	**100GBP12345SECRETKEY**
giving a result of	**f284b69678d2659de47b4951f7f8a55d53de246f**

When **receiving** information from NETBANX (from the CGI callback) the same method is used except the value of netbanx_reference will be added after the merchant_ref for passed transactions. Additionally, other fields may be included if they are deemed to be sensitive to the merchant. Again, please specify this in the Integraiton Questionnaire.

e.g.	**100GBP12345ABCDEFGH1234567890SECRETKEY**
giving a result of	**12f4e9d17ed53e7367a7216488711f8257a7cba5**

Perl example

```
use Digest::SHA qw(sha1_hex);
 my $signature = sha1_hex($amount.$currency.$ref.$secret_key); # incoming
 my $signature = sha1_hex($amount.$currency.$ref.$nbx_reference.$secret_key); # outgoing
```

PHP example

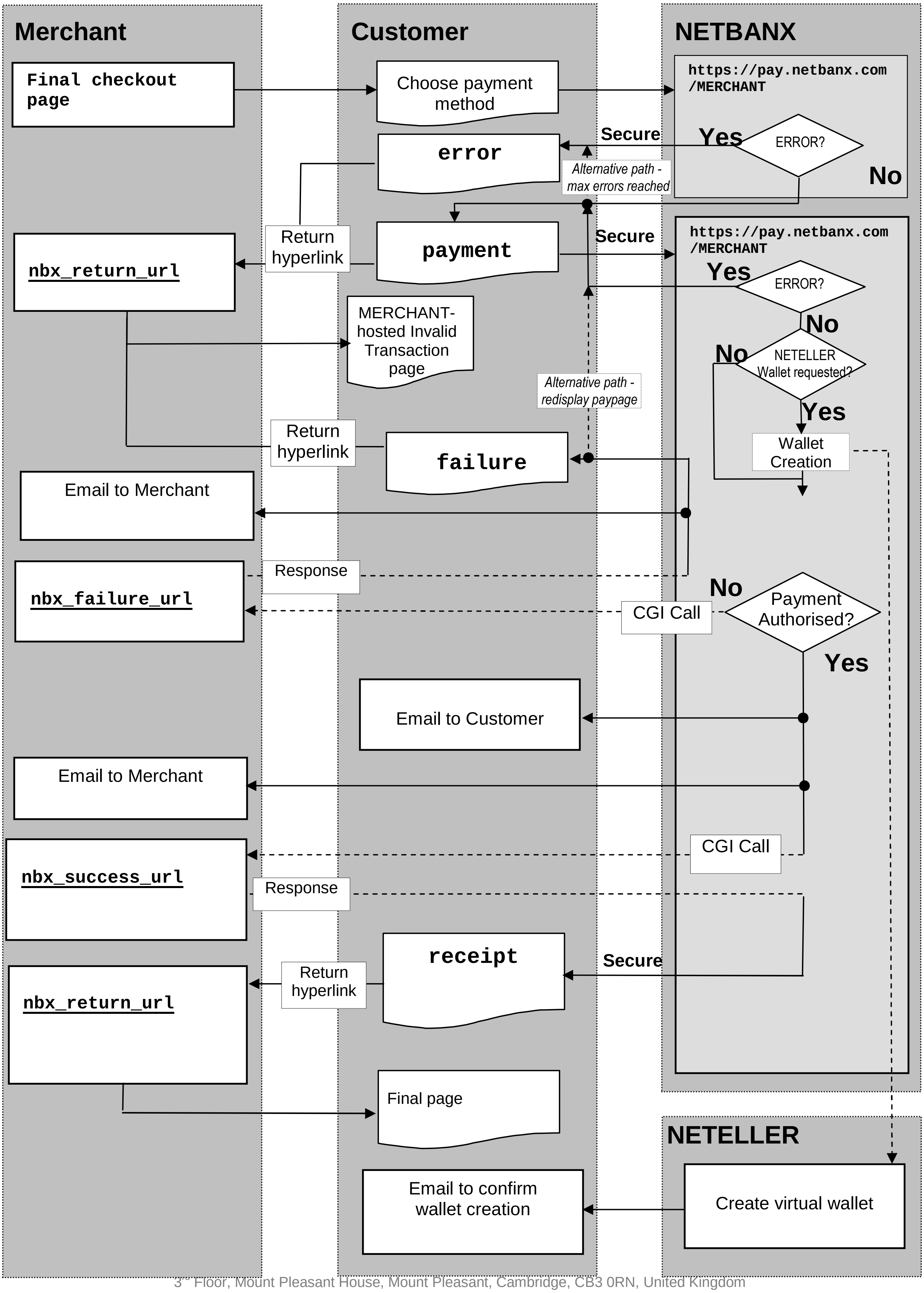
```
 $signature = sha1($amount.$currency.$ref.$secret_key); # incoming
 $signature = sha1($amount.$currency.$ref.$nbx_reference.$secret_key); # outgoing
```

When the destination server receives the checksum, it will produce a checksum in the same way, i.e. using the same secret key. The resulting checksum should match the checksum sent by the source server. If not, then the input should be considered as malicious

Further examples can be checked online at the following URLs: Please note that the contents of these 3rd party sites are out of our control and may be subject to change at any time.

`http://pajhome.org.uk/crypt/md5/`
`http://www.movable-type.co.uk/scripts/sha1.html`

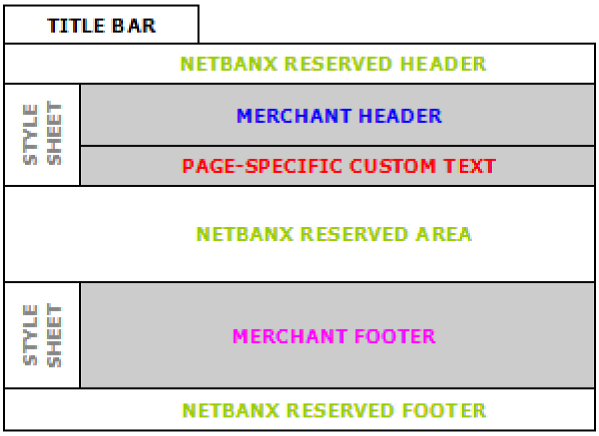
10 Diagram of Process Control Flow (workflow)



11 Payment page structure

The NETBANX payment pages are very flexible. You have the power to add you own HTML code above, below and to either side of the pay form itself. There are certain areas of the page that are controlled by NETBANX and therefore cannot be edited.

This is a "map" of the page:

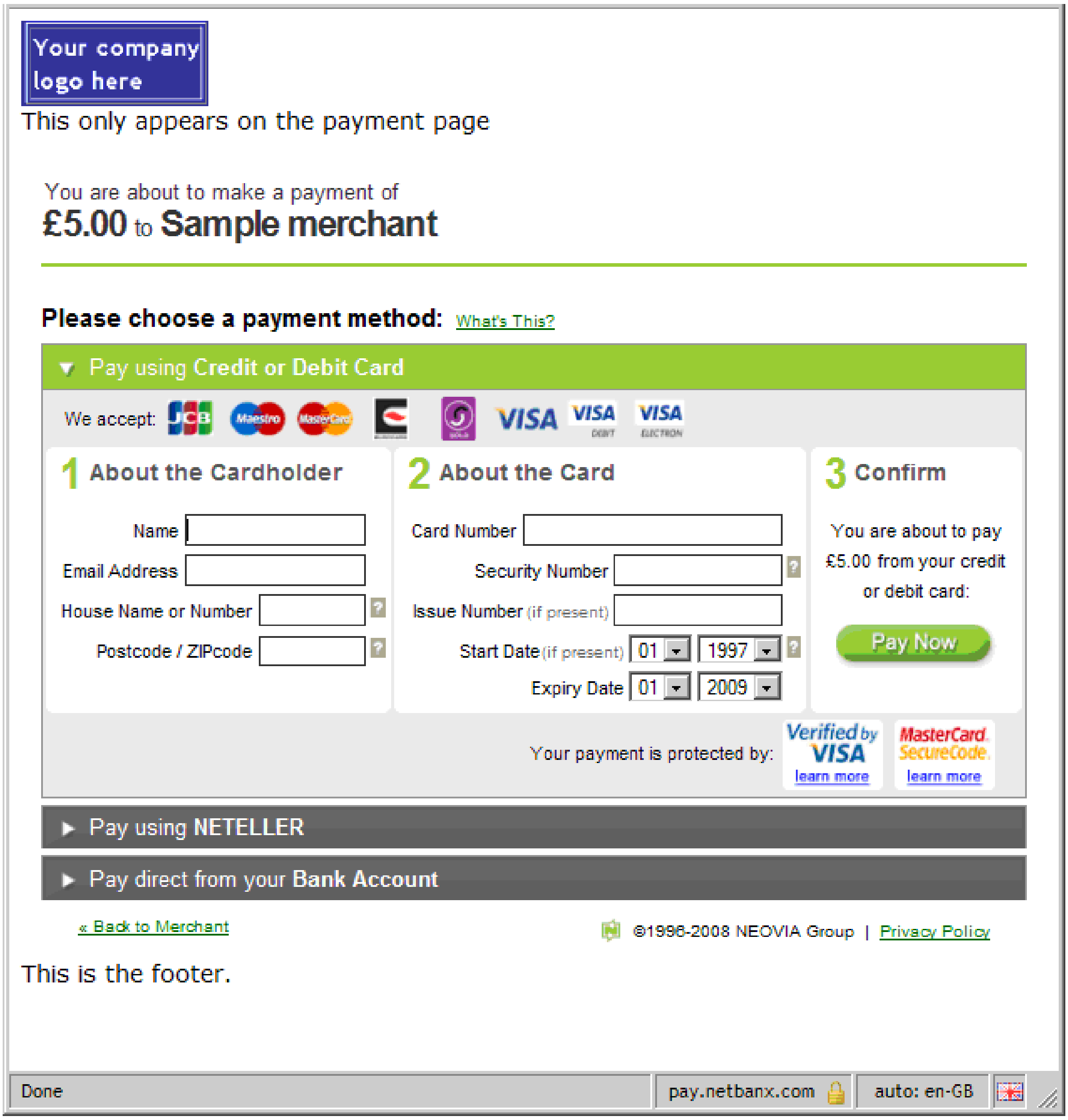


Below is an example of the simplest possible template

You will see that each element has been colour-coded so that you can see how each section of the page is used within the template. For example the first item is in black and this controls the merchant_page_title that appears in the browser window.

```
[%- merchant_page_title = BLOCK -%]
        <!-- This block defines the title bar for the browser window -->
        NETBANX Payment
[%- END -%]

[%- merchant_css = BLOCK -%]
        <!-- Merchant style sheets, applied to all pages -->
        <link rel="stylesheet" type="text/css" href="/MERCHANT/style.css">
[%- END -%]

[%- merchant_header = BLOCK -%]
        <!-- Merchant specific header, displayed on all pages -->
        <img src="/MERCHANT/logo.gif">
[%- END -%]

[%- merchant_payment_html = BLOCK -%]
        <!-- Merchant specific payment page text -->
        <p>This only appears on the payment page</p>
[%- END -%]

[%- merchant_receipt_html = BLOCK -%]
        <!-- Merchant specific receipt page text -->
        <p>This only appears on the receipt page</p>
[%- END -%]

[%- merchant_error_html = BLOCK -%]
        <!-- Merchant specific error page text -->
        <p>This only appears on the error page</p>
[%- END -%]

[%- merchant_footer = BLOCK -%]
        <!-- Merchant specific footer, displayed on all pages -->
        This is the footer.
[%- END -%]
```

So using the file above, each element is inserted into our page as shown in the rendered page below:



Your company logo here

This only appears on the payment page

You are about to make a payment of

£5.00 to Sample merchant

Please choose a payment method: What's This?

Pay using Credit or Debit Card

We accept:

1 About the Cardholder

Name:

Email Address:

House Name or Number:

Postcode / ZIPcode:

2 About the Card

Card Number:

Security Number:

Issue Number (if present):

Start Date (if present): 01 1997

Expiry Date: 01 2009

3 Confirm

You are about to pay £5.00 from your credit or debit card:

Your payment is protected by:

Pay using NETELLER

Pay direct from your Bank Account

« Back to Merchant

©1996-2008 NEOVIA Group | Privacy Policy

This is the footer.

Done | pay.netbanx.com | auto: en-GB

- The NETBANX header has been loaded (but is not visible - this contains our reserved JavaScript and style sheets)
- The css file has been applied to the merchant areas
- The merchant logo has been added to the head of the page
- The payment page text has been added below the header (with the merchant's style applied)
- The NETBANX payment form has been inserted
- The merchant footer has been added below the payment area (with the merchant's style applied)
- The NETBANX footer has been loaded (but is not visible)

Further examples may be found in the setup guide (https://www.netbanx.com/setup/guide) under the sections "HTML Templates & Setup" and "Template Customisation Reference".