Virtual Terminal / Pay Button

User Guide – V1.04

Table of Contents

Version Control

Version	Date	Update information
1.01	13/11/2020	Updated formatting.
1.02	25/08/2023	Updated example code to use ‘examplemid’ and MID to use for testing changed from ‘TEST’ to using Merchant’s test MID.
1.03	03/10/2023	Added Expiry Date field to Pay by Link section and updated Advanced Link section to include a disclaimer. Removed Amount Mode and added Custom Amount.
1.04	16/04/2024	Added Expiry Date field to the Advanced Pay Button section.

1 About Pay Button

The Pay Button allows a Merchant to create a simple HTML link that can be integrated into a webpage or e-mail. When clicked, the user is taken to the hosted payment form with pre-populated product information such as the amount, product description and order reference.

We offer the Pay Button as a QR Code, as a styled and un-styled button and a simple link.

The advantage of using the Pay Button compared to our hosted integration is its simplicity. It has the features of the hosted form without needing server-side scripting knowledge. Just place the link under your products and you're done. All the card holder must do is click a button, enter their payment details and they have paid.

2 Creating a Pay Button

The easiest way to generate a Pay Button is to allow our MMS to do it for you. We've integrated the Pay Button into the virtual terminal so it's all in one place.

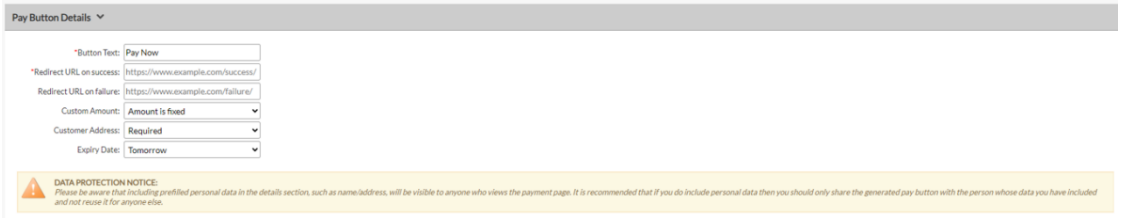
If you want more out of your Pay Button, we offer that too with our advanced integration. This requires more server-side scripting knowledge, but allows more control over what information your Pay Button contains.

Virtual Terminal

Under '**Transaction Details**', change the Payment Mode to '**Generate a Pay Button**'.

Pay Button Options

With the Pay Button mode selected, a new section called 'Pay Button Details' appears in the Virtual Terminal.



2.1.1 Button Text

This will determine what text appears on the Pay Button itself. This field accepts alphanumeric characters, e.g. Pay Now, Pay For 1, etc. and is mandatory.

2.1.2 Redirect URL on success

The redirect URL is the place the cardholder's browser will be returned to after they have made a payment (this must be an https URL, configured to receive post data). This is typically the website where the Merchant sells their products. This field is mandatory.

2.1.3 Redirect URL on failure

If you'd like the cardholder to be redirected to a different endpoint if their payment fails, specify a URL in this box (this must be an https URL, configured to receive post data). If no value is entered in this box and a payment fails, then the cardholder is taken to the value specified in the 'Redirect URL on success' box.

2.1.4 Custom Amount

This confirms if the payment value is fixed or if the customer can alter the payment

amount.

2.1.5 Customer Address

Set if the customer is required to provide their address. If set to 'Required', the 'Address' and 'Postcode' sections on the hosted form are marked as required and cannot be omitted. If set to 'Not Required', the 'Address' and 'Postcode' fields are optional.

2.1.6 Expiry Date

Set an expiry date for the link. This defaults to ‘tomorrow’. The other options are ‘one week’, ‘four weeks’, ‘twelve weeks’ and the option to set a ‘custom expiry’ date. There is also the option for the link to ‘never expire’.

Please be aware that data retention is the responsibility of the Merchant. This includes prefilled personal data in the details section, such as name/address, as this will be visible to anyone who views the payment page. It is recommended that if you do include personal data, then you should only share the generated Pay Button with the person whose data you have included and not reuse it for anyone else. Adding an expiry date helps to ensure data is not stored for longer than necessary.

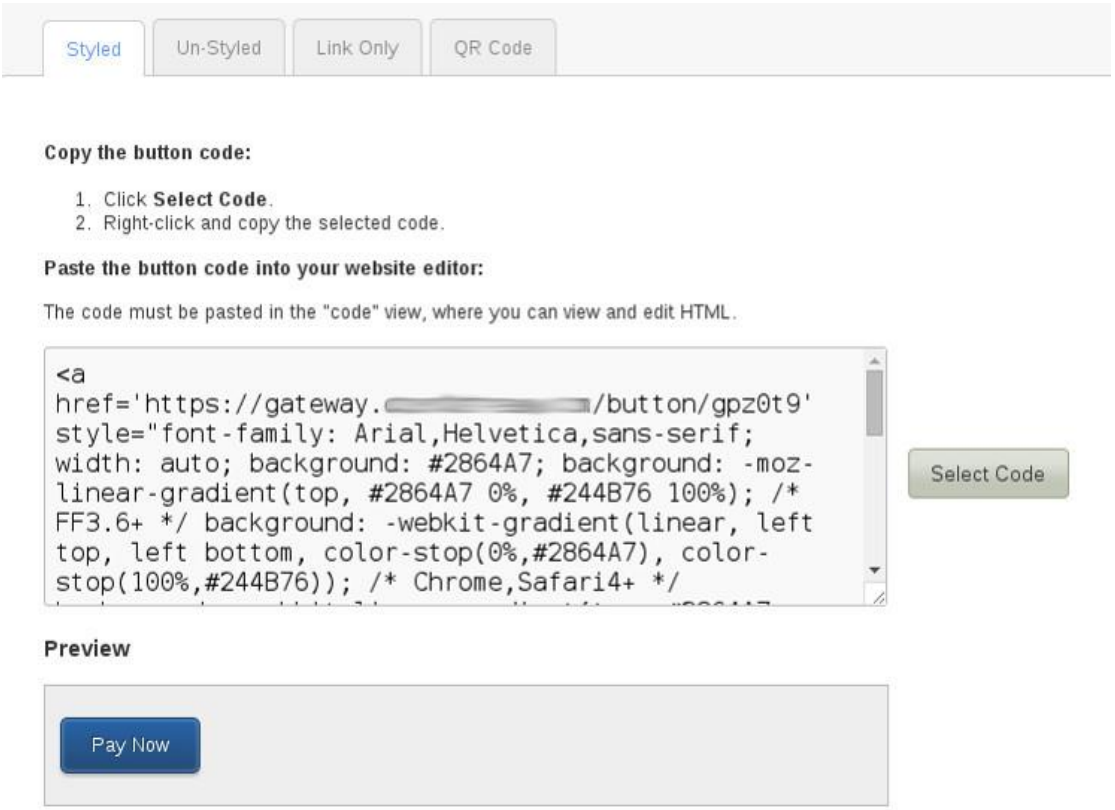
3 Inserting the Link/QR Code

Once 'Create Button' has been clicked, the link will be formatted and displayed on the next page.

To insert the code into HTML content, click the Select Code button and then right click and copy the selected code. The code must be pasted into the code view of your HTML content, where you can view and edit the raw HTML and edit any inline CSS. The link will then be ready for use.

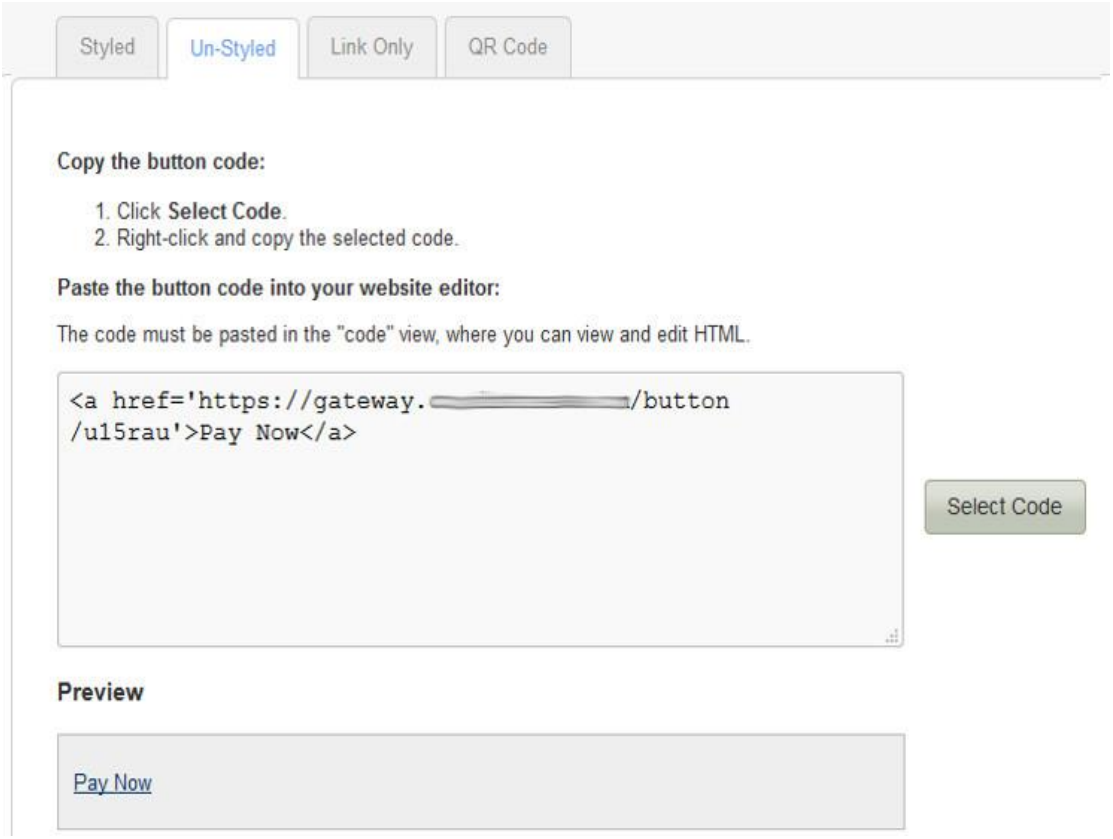
There are 4 options presented: Styled, Un-Styled, Link only and QR Code which can be selected using the tabs on the page. Previews of all the styling options can be seen in the Preview window.

Styled



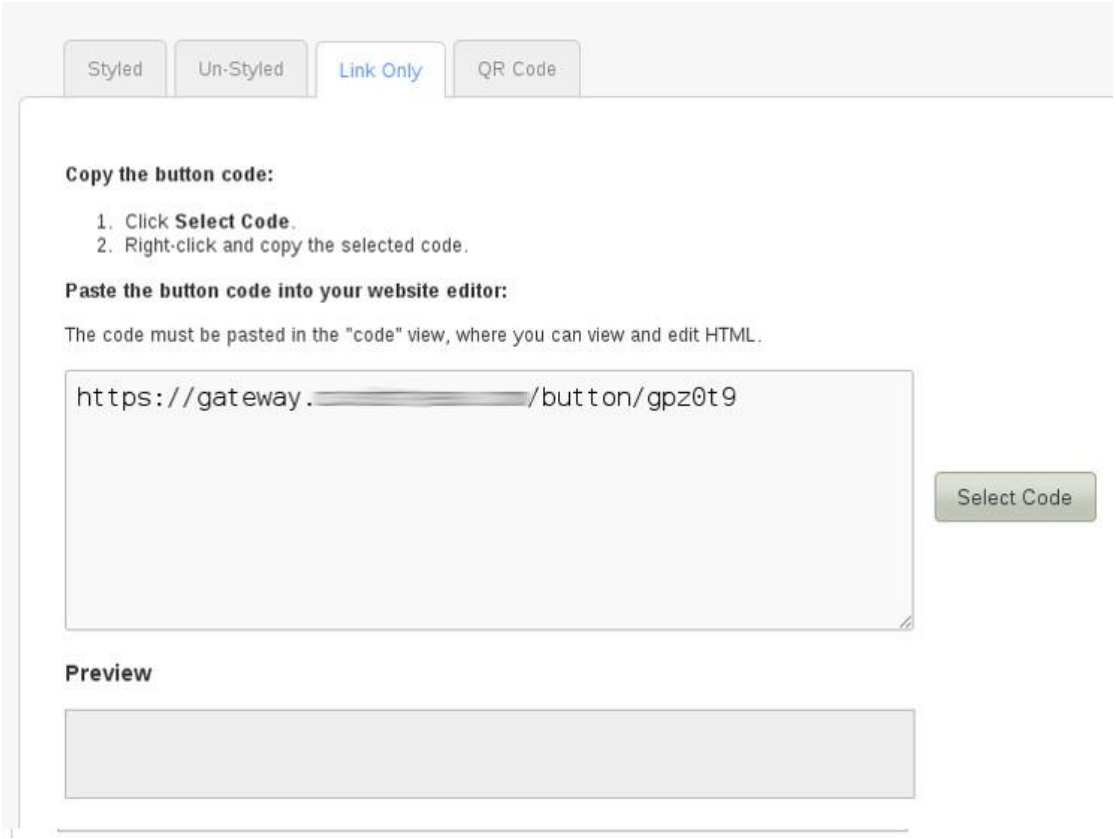
The styled option makes the link look like a pre-formatted button. This is useful for displaying on websites to fit of the style/content of a web-page. It isn't always suitable to send across e-mail clients as the client may not support formatted CSS.

Un-Styled



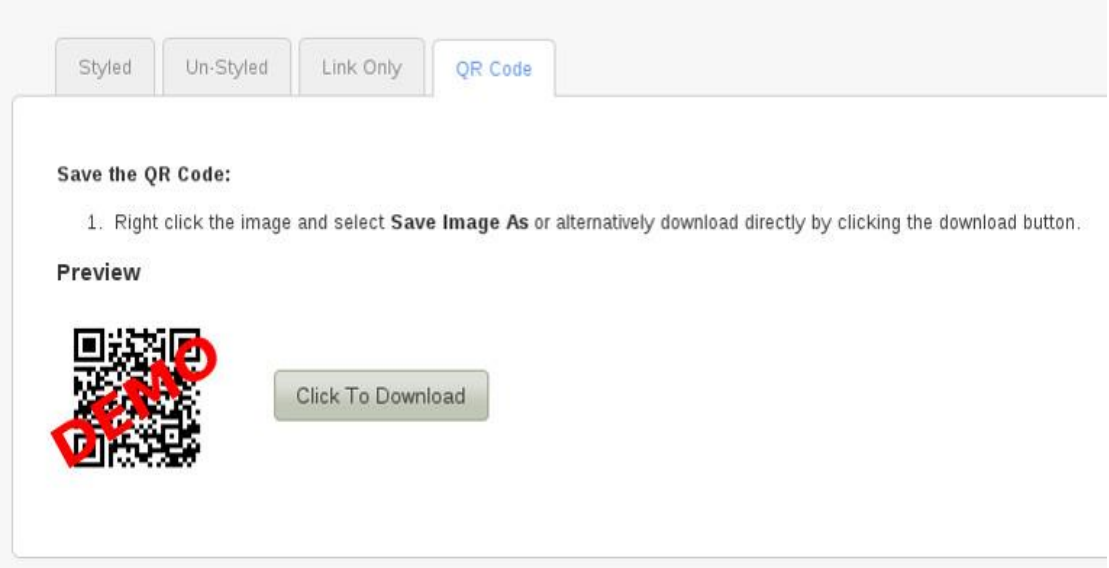
The Un-Styled option creates a standard link that can be styled later using CSS or used as is. This is useful if you want a base to work from to create a button of your choice using CSS.

Link Only



The link only option provides the raw URL without embedding it in an HTML link. If you want a simple link to send to customers, this is the best option.

QR Code



The QR Code option adds the ability for a QR Code to be downloaded. It is automatically generated and is displayed on the page for testing/preview purposes. Click the download button to save it in a file which can then be uploaded or placed wherever you wish. Alternatively, you can right click the preview image and save it from your browser.

A QR Code scanner must be used to 'read' the code, which will then send the link of the payment form to the device.

Send the Pay Button in an E-mail

In addition to the above, you can also send the Pay Button to a customer's e-mail address here.

If successful, you will be presented with an 'Email sent' message. You can enter an additional e-mail if you wish after this has been sent.

4 Advanced Pay Button Integration

The advanced integration method requires knowledge of server-side scripting languages, such as PHP, ASP, Java, etc.

It will allow you to have full control over what information is passed to our gateway via the Pay Button and use the full list of hosted integration fields outlined in our Hosted Guide.

Please be aware that data retention is the responsibility of the Merchant. This includes prefilled personal data in the details section, such as name/address, as this will be visible to anyone who views the payment page. It is recommended that if you do include personal data, then you should only share the generated Pay Button with the person whose data you have included and not reuse it for anyone else. Adding an expiry date helps to ensure data is not stored for longer than necessary.

Pre-Requisites

Merchant ID	Your Merchant ID enables you to access and communicate with the payment gateway. Please note that these details will differ to the login supplied to access the administration panel. You should have received these details when your account was set up. **NB:** You can also enter your test MID as the Merchant ID to test the system. This allows prospective customers to test the integration prior to signing up for an account.
Integration URL	«Gateway_URL»button/?fields={BASE 64 ENCODED STRING}
Signature	Provided with account setup details. **NB:** This is not necessary if you're using your Merchant test ID.

Gateway Request

To create the button the details should be URL encoded Name=Value fields separated by '&' characters (refer to RFC 1738 and the application/x-wwwform-urlencoded media type).

This is then base64 encoded with all padding characters (=) stripped and the following characters +, / replaced with – and _ respectively.

This string is then appended to the gateway URL via a GET parameter called **fields** to give the final link replacing the {base 64 encoded string} above.

Mandatory Request Fields

The below fields MUST be sent to the gateway via the advanced method.

Field Name	Mandatory	Description
merchantID	Yes	The six-digit id provided to you during set-up, e.g. 123456. For testing, please use your test MID..
amount	Yes	The amount of the transaction in minor currency. For the UK this is in pence, e.g. £10.99 is sent as 1099. **Numeric values only – no decimal points or currency symbols.**
action	Yes	The action of the transaction. Values are: **SALE** This will charge a customer's card with the value provided in the **amount** field. **VERIFY** This can be used to validate the card used exists. An **amount** of **0 must** be passed to the gateway. **PREAUTH** Reserves funds from the cardholder's card but doesn't collect them. For a period of up to 30 days depending on the card issuing bank, you can use the cross reference of a transaction to collect those funds using **SALE**. **Note: It is recommended VERIFY is used instead of PREAUTH. Details of why are listed in the HOSTED guide.**
type	Yes	The type of transaction. Passed as a single digit. Possible values are: **1** - Cardholder Not Present: Ecommerce. **2** - Cardholder Not Present: Mail Order.

		3 - Point of Sale: Card Keyed. **4** - Point of Sale: Card Swiped. **5** - Point of Sale: Card Chip & Pin.
redirectURL	Yes	The URL to which the customer will be redirected after the transaction with the transaction result sent via POST. We recommend the integration also contain a **callback<u>URL</u>** to ensure the transaction details are sent to the website in case the cardholder's browser fails to redirect them.
countryCode	Yes	Merchant's Location. **Valid ISO-3166 alpha or numeric code, e.g. 826 for U.K.**
currencyCode	Yes	Transaction Currency. **Valid ISO-3166 alpha or numeric code, e.g. 826 for U.K.**
signature	Yes	A hashed string of the request containing the signature key unique to the merchant ID. More details on message signing can be found in our core API guide.

Optional Request Fields

The below Pay Button fields are optional and do NOT have to be sent in the request.

Field Name	Mandatory	Description
redirectURLFail	No	The URL to which the customer will be redirected and the transaction result will be POSTed if the transaction fails. If left blank, the **redirectURL** will be used.
formAmountEditable	No	Accepts the following value: '**Y**' – Allows the cardholder to enter the amount to pay in the hosted form and is useful for donations. **Note:** Passing 'N', or null to this field causes the amount in the hosted form to act as default, i.e. A static value the user can't change.
buttonExpiry	No	The date the Pay Button link will expire. If the Pay Button request includes this field and is submitted after the date specified, the transaction will be aborted. Accepts dates in the 'YYYY-MM-DD[HH:MM:SS]' format. The time is optional and, if omitted, will use '00:00:00'.

A-1 Example Request

The following is an example test request built in PHP using the TEST Merchant. (Use your Merchant test MID in place of the ‘examplemid’ in the code below):

```php
<?php

function createSignature(array $data, $key, $algo = null) {
    if ($algo === null) {
        $algo = 'SHA512';
    }
    ksort($data);
    // Create the URL encoded signature string
    $ret = http_build_query($data, '', '&');
    // Normalise all line endings (CRNL|NLCR|NL|CR) to just NL (%0A)
    $ret = preg_replace('/%0D%0A|%0A%0D|%0A|%0D/i','%0A', $ret);
    // Hash the signature string and the key together
    $ret = hash($algo, $ret . $key);
    // Prefix the algorithm if not the default
    if ($algo !== 'SHA512') {
        $ret = '{' . $algo . '}' . $ret;
    }
    return $ret;
}

If (!isset($_GET['payment_response'])) {
    $gateway_url =
    'https://gateway.example.com/button/';
    $signature_key = 'examplesignature;
    $fields = array(
        'merchantID' => examplemid,
        'amount' => 1100,
        'action' => 'SALE',
        'type' => 1,
        'countryCode' => 'GB',
        'currencyCode' => 'GBP',
        'transactionUnique' => uniqid(),
        'orderRef' => 'XXXXXX Payments',
        'redirectURL' => ($_SERVER['HTTPS'] == 'on' ? 'https' : 'http') .
'://' . $_SERVER['HTTP_HOST'] . $_SERVER['REQUEST_URI'] . '?payment_response',
    );
    $fields['signature'] = createSignature($fields, $signature_key);
    //Convert array to Query String
    $link = http_build_query($fields);
    $link = gzdeflate($link, 9);
    //Base64 encode the string and remove any padding or invalid
    //URL characters including =,+,/
    $link = strtr(trim(base64_encode($link), '='), '+/', '-_');
```

```
    echo $link = $gateway_url . $link;

    echo "<p><a href='{$link}'>Pay Now</a></p>";

} else if (isset($_GET['payment_response'])){

    //Handle the response

    echo '<h1>Response from hosted form</h1>';
    print_r($_POST);
}
?>
```