How to Accept c-gold on Your Website

18 June 2009

Overview

What you will need

Aside from the ability to edit your website, the minimum you need is:

- a c-gold account to accept payments, and
- an understanding of HTML forms to direct people to c-gold.com.

Optionally, for automatic payment verification, you need:

- the ability to process form data posted to your website.

The way it works is you put an HTML form on your website wherever you want people to be able to make c-gold payments to you. When they click the submit button on the form, they will be directed to c-gold.com, where they can make a payment from their c-gold account to yours. You can specify your c-gold holding account number (very important!), the payment amount, and other important things discussed later in this document. After they are finished at c-gold.com, they will be directed back to your website.

Caution

You should be aware that dishonest customers can make it appear that their web browser has been directed from c-gold.com back to your website. They could conceivably make it appear they have made a c-gold payment to you, ***even if they haven't done so***. For this reason, you should confirm payment directly with c-gold. For example, you could manually log into your c-gold account and verify the payment. Or you could use automatic payment validation, which is explained later in this document.

Not a shopping cart

This document explains how the interface between your website and c-gold.com works. Click To Pay is not a shopping cart system, only a mechanism for accepting payments. In other words, your website needs to handle any necessary collecting of order information and determining the total cost of the order. After your customer's order is complete and they are ready to pay for it, your website needs to submit information to c-gold.com using the form data described in this document.

The payment process

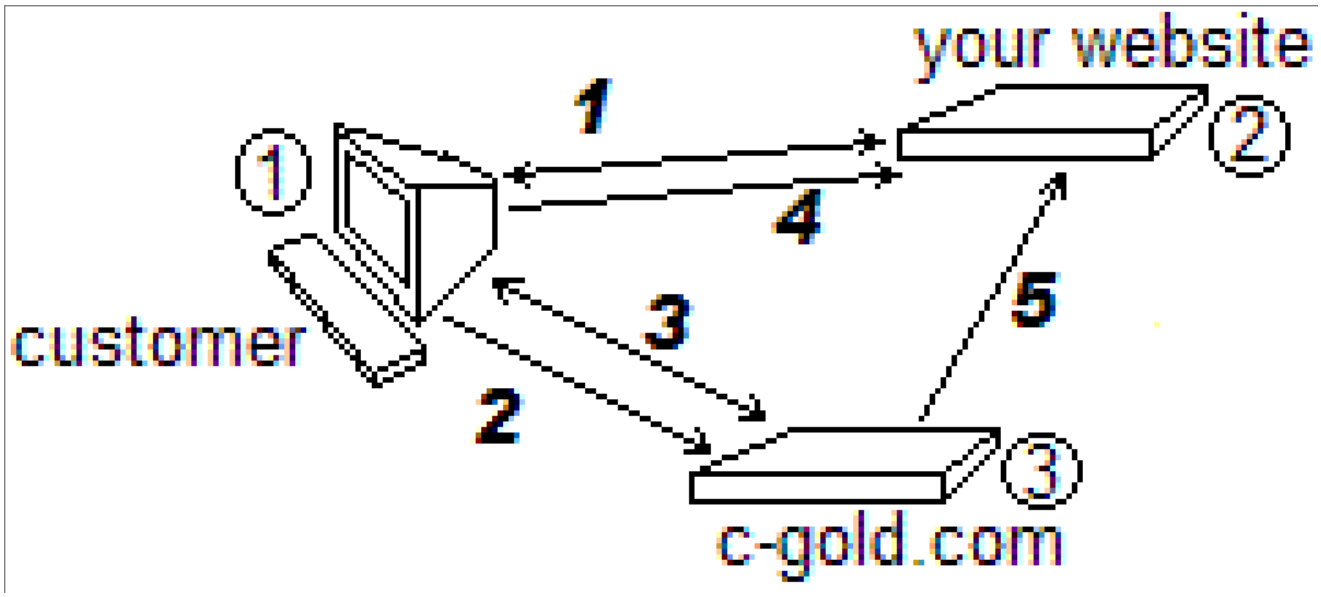
There are three computers involved in a Click To Pay transaction:

1) the customer's (which the customer uses to access your website)
2) your website host (where your website resides)
3) the c-gold.com server

There are five steps involved in the process (diagram below):

1. your customer visits your website
2. when ready to make payment, the customer is directed to c-gold.com
3. the customer makes payment at c-gold.com

4. the customer is directed back to your website
5. transaction details are transmitted directly from c-gold.com to your website or email



The HTML Form

The only step in this process that you need to implement is step 2. All other steps are parts of the process we have designed to allow you to accept c-gold payments. You need to create the form that will direct your customer from your website to the c-gold.com server. (Optionally – and recommended – you can automatically process, or validate, the transaction data transmitted in step 5. This is explained later in this document.)

Hidden Fields

Most of the fields in your HTML form should be “hidden” fields. In these form fields you will communicate with c-gold.com the information necessary for your customer to make the payment to you. For example, you must indicate your c-gold account number. All of the fields you need in your form are explained in the following table.

Field Name	Value	Notes
payee_account	numeric	your account number; will be displayed on preview/confirm pages
payee_name	text	this will be displayed on preview/confirm pages, regardless of the name listed in your account info
forced_payer_account	numeric (if present)	optional; will be displayed on preview page where customer would normally enter their account number, and on the confirm page
payment_amount	numeric, decimal allowed, but no other punctuation or units	will be displayed on preview/confirm pages; this might not be hidden if you want to accept donations or otherwise

		allow the payer to choose the amount

payment_units	must match one of the available payment units on the transaction order form in your c-gold account	will be displayed on preview/confirm pages; this might not be hidden if you want to accept donations or otherwise allow the payer to choose the amount
(any unrecognized field name)	any value	any unrecognized field name/value will be passed back to your website, but otherwise ignored
status_url	url, indicates where you want the data in step 5 sent	this value can start with "http://" or "https://", in which case the status data will be posted to the given url, it can also start with "mailto:" if you cannot process form data; this contact is attempted ONCE, and any response by your website is ignored; if your website is not available and able to process the data, you will have to log into c-gold.com to verify the payment manually
payment_url	url, indicates where you want your customer's browser directed in step 4	this is where your customer's browser should be directed after a successful payment
payment_url_method	must be "link", "get", or "post"	"link" indicates that no form data should be sent back, but only a link should be provided to direct the customer back to your website; for "get" and "post", Click To Pay will send fields back to your website using the indicated method
nopayment_url	url, alternate return path for step 4	this is where your customer's browser should be directed if for any reason the customer has not made the payment
nopayment_url_method	must be "link", "get", or "post"	"link" indicates that no form data should be sent back, but only a link should be provided to direct the customer back to your website; for "get" and "post", Click To Pay will send fields back to your website using the indicated method
suggested_memo	text	will be initially placed in the memo field of your customer's transaction order, but the customer will be able to change it

submit button	any name, any value	any name and/or value can be used to submit the form; as long as the form is posted, the value of the submit button will be ignored

Posting the data

These fields must be posted to `https://c-gold.com/clicktopay/`. So your form will (partially) look like this:

```
<form action='https://c-gold.com/clicktopay/' method='POST'>
...
</form>
```

That is all that is required in order for you to accept c-gold payments on your website.

Receiving Transaction Data Directly from c-gold.com

Step 5 of the Click To Pay transaction process is essentially the opposite of step 2. Instead of your website posting an HTML form to c-gold.com, c-gold.com will post an HTML form to your website (or c-gold.com will send an email, if you have so chosen).

Caution

In addition to making it appear as though their browser has been directed from c-gold.com back to your website (the previous caution), dishonest customers can also post a form to your server that looks like a c-gold payment confirmation, ***even if no such payment has never been made***. It also could be possible, depending on your security practices, for a dishonest person to ***duplicate a previous payment confirmation*** and make it appear that a new similar payment has come in.

It is easy to determine whether such a payment confirmation is valid. The simplest method is to manually log into your c-gold account and check if you really received the payment. (Make sure the transaction ID is not duplicated in your records - or you might end up providing goods or services twice when you only got paid once.)

You can also validate it automatically, using the verify_hash which c-gold.com provides you in step 5 of the Click To Pay transaction process. Note that you still need to make sure the transaction is not duplicated in your records.

Automatic Payment Validation

Optionally – and recommended – you can automatically validate transaction data transmitted to your website from the c-gold.com server.

First, you must make sure that you do not credit a customer more than once for the same transaction id.

Original verify_hash

To verify a payment, you must build a string as follows:
transaction_id:pay_from:payee_account:payment_amount:payment_units:merchant_passphrase

These are the specified fields posted to your status_url, plus your merchant_passphrase as set in your c-gold account, separated by colons. Any extra fields you send and receive back are NOT included.

You must then do an md5 hash of the string you have built.

Version 2 of the verify hash, v2_hash

Upon user request, we added a field to the postback data indicating the actual weight of gold transferred by the payer, ***but only if you requested a fiat denominanted payment.*** Otherwise, the field would be redundant. The field is called actual_weight. in order to allow the merchant to rely on this value, we also added a new hash to the postback data, called v2_hash.

The v2_hash field is calculated similarly to the original verify_hash, but you must add the actual_weight field to the colon separated list of field values, and then calculate the new hash in exactly the same way as the old hash.

Payment Validation Example

Suppose:
transaction_id = 136
pay_from = 1
payee_account = 2
payment_amount = 1
payment_units = "EUR worth"
actual_weight = 0.0455
merchant_passphrase = "test"

Original verify_hash

The string you must hash is "136:1:2:1:EUR worth:test". Important: there should not be any extra blank spaces in the string - there is one blank space in this string, because payment_units includes one between "EUR" and "worth".

When you do an md5 hash of this string, you should get:
verify_hash = "0ba2f6c828c37527351be2f07f3076d3"

Notice that the hex digits above 9 are lower case.

The verify_hash field must match the calculated md5 hash of the string you built. If it does not, then it is not a valid payment confirmation.

Version 2 of the verify hash, v2_hash

The v2 string you must hash is "136:1:2:1:EUR worth:test:0.0455".
The hash, v2_hash = "c7b55ec9539243ebb89ffeb1229b0940".

If your software depends on the actual_weight field to be valid (unmodified by a potentially malicious user), you should use the v2_hash instead of verify_hash. If you do not use the actual_weight field, or it's value is not critical, you can use either the original verify_hash or the new v2_hash.

Further Help

If you need more help, please use the c-gold.com contact form at
https://c-gold.com/contactus.php