



Seller # 35128 [Admin HomePage](#) [Support Forums](#)

Account Details	Products	Shopping Cart	Sales Activity	Other Resources	Logout
		Design Resources	Search Engine Overview	Search Engine List	

Database Payment Specifications

This page covers the specifications for the database payment routine on the 2Checkout.com system. It is provided for clients that want to use their own specialized program and have sufficient knowledge to code the procedures.

Payment Routines:

<https://www.2checkout.com/cgi-bin/sbuyers/purchase.2c> ([Example](#))

<https://www.2checkout.com/cgi-bin/ccbuyers/purchase.2c> ([Example](#))

The following parameters must be passed from your routine to purchase.2c.

sid - your 2Checkout.com account number

product_id - Product ID of the product/service from the 2Checkout system to be billed. If sending a second product/service to be billed product_id1, third product_id2, etc..

quantity - Quantity of the product/service from the 2Checkout system to be billed. If sending a second product/service to be billed quantity1, third quantity2, etc..

Optional Parameters

merchant_order_id - order id from your system that you would like recorded and passed back.

demo - set to **Y** for testing or set the purchase button/link routine to demo mode [here](#).

DO NOT PASS DEMO IN OR LEAVE IT BLANK FOR LIVE PROCESSING.

Example of total string:

https://www.2checkout.com/cgi-bin/ccbuyers/purchase.2c?sid=124&product_id=1&quantity=1&merchant_order_id=test1&demo=Y

Hint: You can build the above string in a routine if you wanted and then use the following directive (written in perl) to pass control to 2Checkout for payment.

```
print "Location: https://www.2checkout.com/cgi-bin/ccbuyers/purchase.2c..... ";
```

The following parameters will be passed back to your routine. Optionally a confirmation email with your merchant order number and ours will be sent to you and the buyer when the credit card transaction is approved.

order_number - 2Checkout.com order number

card_holder_name - Card holders name

street_address - Card holders address

city - Card holders address

state - Card holders address

zip - Card holders address

country - Card holders address

email - Card holders email

phone - Card holders phone number

merchant_order_id - Your order number passed back

credit_card_processed - "Y" or "K" : "Y" if successful, "K" indicates a check, which will take 2-3 days to verify

total - Total billed to credit card.

ship_name - Shipping Info

ship_street_address - Shipping Info

ship_city - Shipping Info

ship_state - Shipping Info

ship_zip - Shipping Info

ship_country - Shipping Info

product_id - 2Checkout Product ID for Item Purchased

quantity - Quantity Product Purchase

merchant_product_id - Merchant Product ID for Item Purchased

product_description - Merchant Product Description for Item Purchased

If two or more times are purchased the fields product_id, quantity, merchant_product_id, product_description will be followed by a number 1 for second item 2 for third

i.e. product_id1,quantity1,merchant_product_id1,product_description1 for purchase item 2.

If you send your customer a sales confirmation email. The confirmation email should contain the follow text:

Charges for the transaction will appear on your credit card statement under the name **2Checkout.com**.

Questions regarding your order should be directed to:

Your Company
www.yourcompany.com
youremail@youremailaddress.com

If you have problems with the order and "Your Company" cannot help you, you may contact custservice@2checkout.com.

Recommendation: The following perl code can be used to verify that the passback to your site was initiated by 2Checkout.

#####Begin Code Sample#####

```
@referers=("www.2checkout.com","2checkout.com");
&check_url;
sub check_url {
# Localize the check_referer flag which determines if user is valid.
local($check_referer) = 0;

# If a referring URL was specified, for each valid referer, make sure
# that a valid referring URL was passed to .

if ($ENV{'HTTP_REFERER'}) {
    foreach $referer (@referers) {
        if ($ENV{'HTTP_REFERER'} =~ m|https?:/[^\]*$referer|i) {
            $check_referer = 1;
            last;
        }
    }
}
else {
    $check_referer = 1;
}
# If the HTTP_REFERER was invalid, send back an error.
# It is possible that the referrer variable was simply not available for a variety of reasons.
if ($check_referer != 1) {
    print "Bad Referrer - Merchant should put in code to write to a log or mail themselves a
    note to verify order was billed in 2Checkout.com system"; }
}
```

#####End Code Sample#####