Advanced e-Commerce

Technical Integration Guide for e-Commerce – Version 4.4

www.ogone.com

Contents

1 Introduction

Advanced e-Commerce explains the advanced integration of Ogone e-Commerce into your website. This document complements the **Basic e-Commerce** document.

For the configuration and functionality of the administration site, please refer to the **Back-Office User Guide**.

2 Test Environment

We recommend you to develop your integration in our test environment before going live in the production environment. Our test environment works almost identically to our production environment, except we do not send the transactions to the card acquirer and the usage is free of charge.

Our test environment allows you to simulate payments, change your account configuration and fine-tune the integration of our payment system into your website.

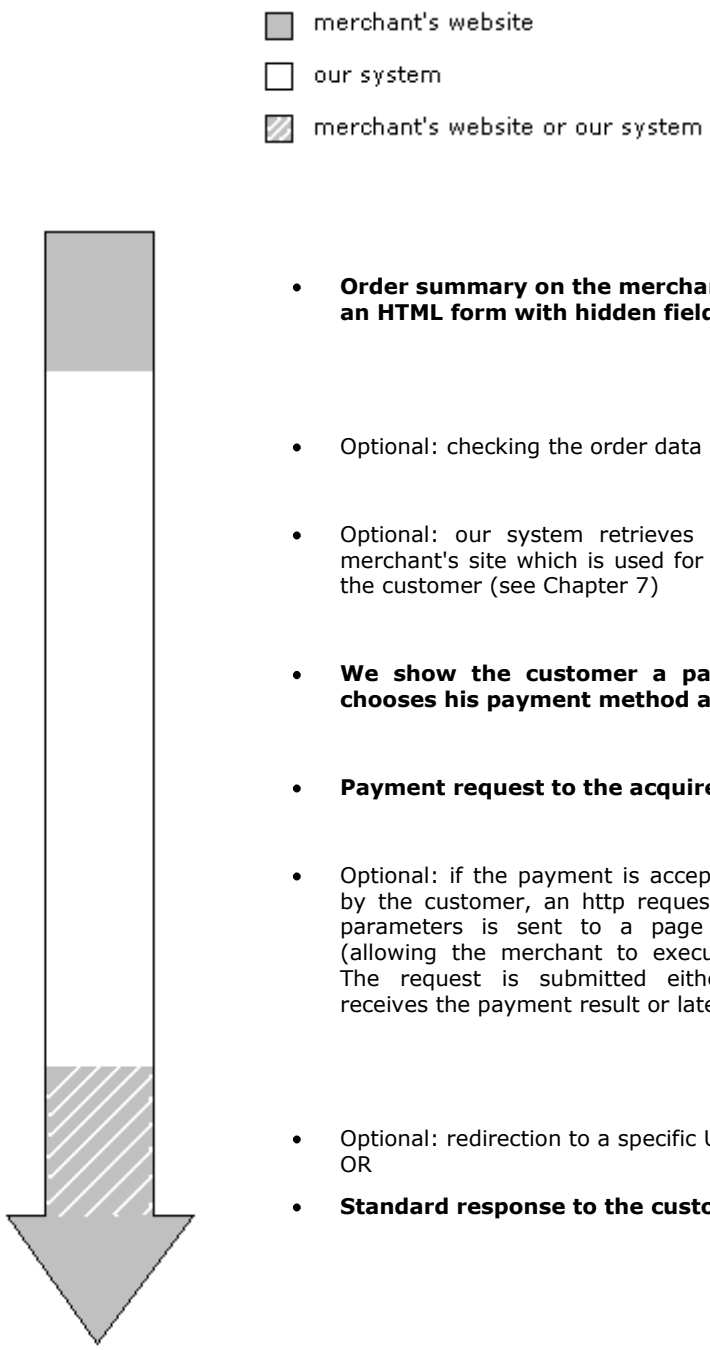
2.1 Configuring your test account

When you first log into your account, you will see a list of steps to complete on the homepage. These steps concern the administrative, payment method and technical details of your test account. For details on how to create, access and configure your test account, please refer to the **Basic e-Commerce** documentation. The configuration of the technical details will be explained in the following chapters.

The technical details are to be configured in the Technical Information page in your account. You can access the technical parameters via the link "Technical Information" in your account menu.

3 Sale Process

The following workflow represents a transaction with basic steps (in bold) and optional steps:



- **Order summary on the merchant's website containing an HTML form with hidden fields**
- Optional: checking the order data (see Chapter 6)
- Optional: our system retrieves a template page on the merchant's site which is used for the page layout shown to the customer (see Chapter 7)
- **We show the customer a payment page where he chooses his payment method and enters his details**
- **Payment request to the acquirer**
- Optional: if the payment is accepted, declined or cancelled by the customer, an http request containing the payment parameters is sent to a page on the merchant's site (allowing the merchant to execute automatic processes). The request is submitted either before the customer receives the payment result or later (see Chapter 8).
- Optional: redirection to a specific URL at the merchant's side OR
- **Standard response to the customer**

The merchant has the possibility to extend his integration, securing the order data, personalizing the payment pages, picking up feedback after a transaction and personalizing the response to his customer.

This manual explains the advanced e-commerce integration with the optional steps to personalize the transaction flow and fine-tune the integration.

For a screenshot representation of a sale process following a basic e-commerce integration please refer to the **Basic e-Commerce** documentation.

4 General Payment Parameters

For some payment methods (mainly credit cards), transactions are performed in two steps: the authorization and the data capture (payment request). (See Chapter 4.1 and 4.2)

During the authorization step, the transaction amount is either reserved on the customer's card or the account, or the request is matched against a blacklist (AUT operation).

In the data capture (payment request) step, the merchant's acquirer is requested to take the reserved or blacklist matched amount on the customer's card or account and transfer it to the merchant's bank account (DCP operation).

Additional payment methods (mainly credit cards) allow either online or offline transaction processing. (See Chapter 4.3)

The merchant can instruct our system to request the payment or authorization immediately from the acquirer (online processing), or simply confirm the receipt of the transaction and save it for capture by the acquirer at a later time (offline processing).

The payment behavior depends on general parameters the merchant defines in the "Global transaction parameters" tab of the Technical Information page of his administration module: the default operation code, the default data capture (payment) procedure and the processing for individual transactions. These parameters are set for each account, meaning they apply to all transactions within the merchant's account.

4.1 Default operation code

IMPORTANT: The ability to work in two steps (authorization + data capture) depends on the payment methods you wish to use. (See the online **Payment Methods Processing/Procedure** overview).

Based on the two steps "authorization" and "data capture" the merchant can choose between two default operation codes in the "Global transaction parameters" tab, "Default operation code" section of the Technical Information page:

- **Authorisation**:

  Our system will only ask for an authorisation, in order to have the authorisation and data capture (payment request) steps performed separately at different times (the money remains on the customer's account until a data capture (payment request) has been performed).

- **Sale**:

  Our system automatically requests the payment (transfer of the amount) immediately after a successful authorisation. This procedure is often used for goods/services delivered online.

4.2 Default data capture (payment) procedure

IMPORTANT: The ability to work in two steps (authorization + data capture) depends on the payment methods you wish to use. (See the online **Payment Methods Processing/Procedure** overview).

If the merchant has "Authorisation" as the default operation code for his account or he included the "Authorisation" operation code in the transaction details, a data capture will have to be performed on the transaction to request the payment.

Three possible data capture (payment request) procedures are available:

- **Data capture by the merchant (manual or automatic)**:

  To request the transfer of the reserved amount to the merchant's bank account, the merchant must call up his administration module and request the data capture (payment) for the specific transaction (please refer to the Back Office User Guide).

  The merchant can also automate the data process by sending us the data captures via batch or via a server-to-server request (please refer to the Batch or DirectLink information).

  The period for which an authorization is valid depends on the merchant's acquirer contract.

  This procedure is often used if the merchant has to check his stocks before dispatching the ordered goods.

- **Automatic data capture by our system at the end of the day**:

  Our system requests the payment (data capture) automatically as from midnight, GMT+1 time.

- **Automatic data capture by our system after x days**:

  Our system requests the payment (data capture) automatically after x days (if the merchant hasn't cancelled the authorisation).

  The minimum number of days you can enter is "2" since "1" would lead the payment to be requested automatically as from midnight, i.e. an "Automatic data capture by our system at the end of the day".

  This procedure is often used for goods/services delivered within a specific time (24 hours, 48 hours, …).

4.3 Processing for individual transactions

IMPORTANT: the ability to work online or offline depends on the payment methods you wish to use. (See the online **Payment Methods Processing/Procedure** overview).

There are three ways of processing for individual transactions:

- **Always online (Immediate)**:

  The transaction request is sent to the acquirer immediately while the customer is connected (appropriate for goods/services delivered online).

  When the online acquirer clearing system is unavailable all online transactions will be declined.

- **Online but switch to offline in intervals when the online acquiring system is unavailable**:

  If the merchant wants online processing but does not want to miss out on transactions if the online acquirer clearing system is temporarily unavailable, he can authorize offline processing in those specific circumstances.

  We will store the transactions arriving from the merchant's website during the unavailability of his acquirer and will process them offline as soon as the acquirer clearing system is back up again. (Not suitable for services that are triggered online immediately after the transaction!)

- **Always offline (Scheduled)**:

  We register the transaction and process it afterwards (max. 4 hours). This method is slightly faster for the customer since we do not send the request to the acquirer immediately (can be used for goods/services that do not need to be delivered online). However, the customer will not immediately see the transaction/order result.

You can configure an offline status change notification in the "Transaction feedback" tab, "HTTP request for status changes" section of the Technical Information page of your account (for HTTP requests) or in the "Transaction e-mails" tab, "E-mails to the merchant" section of the Technical Information page (for e-mails). That way, you can be notified by e-mail and/or http request when the status of a transaction changes offline in our system.

5 Link between the Merchant's Website and our Payment Page

5.1 Order form

The link between the merchant's website and our e-commerce payment page has to be established on the last page of the shopping basket on the merchant's website, in other words: the last page of the merchant's site presented to the customer.

A form with hidden html fields containing the order data must be integrated into that last page. The action URL of the form will be our (e-commerce system's) payment processing page.

5.1.1 Form fields

The following section contains the block of code the merchant needs to paste in the last page of his shopping basket:

```
<form method="post" action="https://secure.ogone.com/ncol/XXXX/orderstandard.asp" id=form1 name=form1>
<!-- general parameters: see chapter 5.2 -->
<input type="hidden" name="PSPID" value="">
<input type="hidden" name="orderID" value="">
<input type="hidden" name="amount" value="">
<input type="hidden" name="currency" value="">
<input type="hidden" name="language" value="">
<!-- optional customer details, highly recommended for fraud prevention: see chapter 5.2 -->
<input type="hidden" name="CN" value="">
<input type="hidden" name="EMAIL" value="">
<input type="hidden" name="ownerZIP" value="">
<input type="hidden" name="owneraddress" value="">
<input type="hidden" name="ownercty" value="">
<input type="hidden" name="ownertown" value="">
<input type="hidden" name="ownertelno" value="">
<input type="hidden" name="COM" value="">
<!-- check before the payment: see chapter 6.2 -->
<input type="hidden" name="SHASign" value="">
<!-- layout information: see chapter 7.1 -->
<input type="hidden" name="TITLE" value="">
<input type="hidden" name="BGCOLOR" value="">
<input type="hidden" name="TXTCOLOR" value="">
```

```
<input type="hidden" name="TBLBGCOLOR" value="">
<input type="hidden" name="TBLTXTCOLOR" value="">
<input type="hidden" name="BUTTONBGCOLOR" value="">
<input type="hidden" name="BUTTONTXTCOLOR" value="">
<input type="hidden" name="LOGO" value="">
<input type="hidden" name="FONTTYPE" value="">
<!-- dynamic template page: see chapter 7.2 -->
<input type="hidden" name="TP" value="">
<!-- payment methods/page specifics: see chapter 9.1 -->
<input type="hidden" name="PM" value="">
<input type="hidden" name="BRAND" value="">
<input type="hidden" name="WIN3DS" value="">
<input type="hidden" name="PMLIST" value="">
<input type="hidden" name="PMListType" value="">
<!-- link to your website: see chapter 8.1 -->
<input type="hidden" name="homeurl" value="">
<input type="hidden" name="catalogurl" value="">
<!-- post payment parameters: see chapter 8.2 -->
<input type="hidden" name="COMPLUS" value="">
<input type="hidden" name="PARAMPLUS" value="">
<!-- post payment parameters: see chapter 8.3 -->
<input type="hidden" name="PARAMVAR" value="">
<!-- post payment redirection: see chapter 8.2 -->
<input type="hidden" name="accepturl" value="">
<input type="hidden" name="declineurl" value="">
<input type="hidden" name="exceptionurl" value="">
<input type="hidden" name="cancelurl" value="">
<!-- optional operation field: see chapter 9.2 -->
<input type="hidden" name="operation" value="">
<!-- optional extra login detail field: see chapter 9.3 -->
<input type="hidden" name="USERID" value="">
<!-- Alias details: see Alias Management documentation -->
<input type="hidden" name="Alias" value="">
<input type="hidden" name="AliasUsage" value="">
<input type="hidden" name="AliasOperation" value="">
<input type="submit" value="" id=submit2 name=submit2>
</form>
```

An example (test page) representing the last page of a merchant's shopping basket, can be found at: https://secure.ogone.com/ncol/test/teststd.asp.

The merchant can copy and paste the html code of the form at the bottom of this test page into his shopping basket page. The values in the fields need to be replaced by the merchant's account values.

Some fields, such as the orderID and amount, must be assigned dynamically.

5.1.2 Form action

<form method="post" action="https://secure.ogone.com/ncol/XXXX/orderstandard.asp" id=form1 name=form1>

In the TEST environment the URL for the action will be https://secure.ogone.com/ncol/test/orderstandard.asp.

IMPORTANT: When you switch to your PRODUCTION account you must replace "test" with "prod" so the action of the form will be https://secure.ogone.com/ncol/prod/orderstandard.asp. If you forget to change the action of your form, once you start in production with real orders, your transactions will be sent to the test environment and will not be sent to the acquirers/banks.

5.2 General parameters and optional customer details

The general parameters are the parameters that have to be sent with each transaction in order for us to be able to process it.

Although the mandatory parameters are the PSPID, orderID, amount, currency and language value, we nevertheless **strongly recommend you also send us some optional customer details such as the customer name, customer's e-mail, address, town, zip, country and telephone number since they can be useful tools for combating fraud.**

These optional customer details will also be stored with the transaction at our end and can be analyzed in your administration module when you look up the transaction details.

5.2.1 Hidden fields

The following hidden fields used to transmit the general parameters to our system:

<input type="hidden" name="PSPID" value="">

<input type="hidden" name="orderID" value="">

<input type="hidden" name="amount" value="">

<input type="hidden" name="currency" value="">

<input type="hidden" name="language" value="">

<input type="hidden" name="CN" value="">

<input type="hidden" name="EMAIL" value="">

<input type="hidden" name="ownerZIP" value="">

<input type="hidden" name="owneraddress" value="">

<input type="hidden" name="ownercty" value="">

<input type="hidden" name="ownertown" value="">

<input type="hidden" name="ownertelno" value="">

<input type="hidden" name="COM" value="">

Field	Usage
PSPID	Your affiliation name in our system
orderID	Your unique order number (merchant reference). The system checks that a payment has not been requested twice for the same order. The orderID has to be assigned dynamically.
amount	Amount to be paid MULTIPLIED BY 100 since the format of the amount must not contain any decimals or other separators. The amount must be assigned dynamically.
currency	ISO alpha order currency code, for example: EUR, USD, GBP, CHF, …
language	Language of the customer, for example: en_US, nl_NL, fr_FR, …
CN	Customer name. Will be pre-initialized (but still editable) in the cardholder name field of the credit card details.
EMAIL	Customer's e-mail address
owneraddress	Customer's street name and number
ownerZIP	Customer's ZIP code
ownertown	Customer's town/city name
ownercty	Customer's country
ownertelno	Customer's telephone number
COM	Order description

For further technical details about these fields, please refer to the online ***Parameter Cookbook***.

6 Security: Check before the Payment

Best practice: SHA signature, chapter 6.2

The "Best Practice" section at the top of a chapter indicates the most effective method for delivering the aspired outcome. This will help you optimize your e-Commerce integration.

6.1 Referrer

Our system checks the origin of the payment request, i.e. which URL the order comes from. This URL is called the referrer.

6.1.1 Configuration

The merchant must fill out the referrer/URL of the page containing the order form with the hidden fields in the URL field in the "Data and origin verification" tab, "Checks for e-Commerce" section of the Technical Information page in his account.

The URL(s) must always start with http:// or https://. You can enter the full URL or simply the domain name; the latter will result in all subdirectories and pages of that domain being accepted.

Several URLs can be entered, should the merchant have different domains, e.g. http://www.mysite.com;http://www.mysite.net;http://www.secure.mysite.com. The URLs must be semi-colon separated with no whitespaces before or after the semi-colon.

If you perform a test transaction from our test page, please remember to enter our site's URL as referrer, otherwise you will receive an error.

6.1.2 Possible errors

Some possible errors related to the referrer are *"unknown order/1/r"* and *"unknown order/0/r"*. Please refer to Appendix 2 for more information about these errors.

6.1.3 Limitations

We use the referrer to identify the origin of an order, but some browsers do not forward the referrer info which will result in an error.

However, the referrer check alone is not fool proof: although it checks the origin of an order, to ensure the integrity of the order data, the merchant needs to have our system perform a data check before processing the payment.

This data check is mandatory for configuring "Sale" as default operation code or an automatic data capture by our system as default data capture (payment) procedure (see Chapter 4). The data check is not mandatory if the merchant uses a 2-phase – reservation/manual payment request – payment procedure (since he can check the order data before sending the payment request); in the latter case, too, however, **we strongly advise the merchant to perform a data check before payment**.

We propose SHA-1 as data check method.

6.2 SHA-1-IN signature

We recommend using the SHA signature as the data check method. This technique is based on the principle of the merchant's server generating a unique character string, hashed with the SHA-1 algorithm, for each order. The result of this hash is then sent to us in the hidden fields of the merchant's order page. Our system reconstructs this signature to check the data integrity of the order information sent to us in the hidden fields. For further details about the SHA signature, please refer to Appendix 1.

6.3 IP address check

The IP address field in the "Data and origin verification" tab, "Checks for DirectLink and automatic Batch" section of the Technical Information page only has to be completed if, in addition to his e-Commerce connection, there is a server-to-server connection with our system (i.e. requests on orderdirect.asp, maintenancedirect.asp, querydirect.asp, AFU_agree.asp).

If not used, it can be left empty. (Please refer to the **DirectLink / Batch Advanced** documentation).

7 Look & Feel of the Payment Page

Best practice: Static template, Chapter 7.1

When our e-Commerce system requests the customer for his credit card details, the customer is on our secure server.

There are two types of information on the payment process page: static information (the merchant's logo for example) and payment details information (order reference, fields where the customer enters his card details, …).

The static information originates from our system's common layout or a specific merchant template page (as explained below). Our system adds the payment details dynamically for each transaction. The look & feel of these payment details may however be adapted by the merchant using html styles.

There are two ways to customize the payment process page design to maintain the look & feel of the merchant's site during the payment process: using a static or a dynamic template page.

7.1 Payment page layout (Static template)

The static template page is a common template on our side, but the merchant can change the look & feel of some elements on the payment page or add his logo by simply adding some hidden fields in the form he sends us (cf. Chapter 5):

The following hidden fields are used to transmit the look & feel parameters to our system:

```
<input type="hidden" name="TITLE" value="">
<input type="hidden" name="BGCOLOR" value="">
<input type="hidden" name="TXTCOLOR" value="">
<input type="hidden" name="TBLBGCOLOR" value="">
<input type="hidden" name="TBLTXTCOLOR" value="">
<input type="hidden" name="BUTTONBGCOLOR" value="">
<input type="hidden" name="BUTTONTXTCOLOR" value="">
<input type="hidden" name="LOGO" value="">
<input type="hidden" name="FONTTYPE" value="">
```

Field	Usage	Default value
TITLE	Title and header of the page	_
BGCOLOR	Background color	white
TXTCOLOR	Text color	black
TBLBGCOLOR	Table background color	white
TBLTXTCOLOR	Table text color	black
BUTTONBGCOLOR	Button background color	_
BUTTONTXTCOLOR	Button text color	black
FONTTYPE	Font family	Verdana
LOGO	URL/filename of the logo you want to display at the top of the payment page next to the title. The URL must be absolute (contain the full path), it cannot be relative. The logo needs to be stored on a secure server (see Chapter 7.3). If you do not have a secure environment to store your image, you can send a JPG or GIF file (and your PSPID) to support@ogone.com (only for production accounts since this is a paying option! Please activate the "Logo Hosting" option in your Account > Options page before sending us your logo). If the logo is stored on our servers, you only need to enter the filename, not the whole URL.	_

*For more technical details about these fields, please refer to the online **Parameter Cookbook**.*

The colors can be specified by their hexadecimal code (#FFFFFF) or their name (white). First check how the colours you want to use appear in different browsers.

7.1.1 iPhone static template

We have developed a specific template for iPhones on our platform. In order to use our static iPhone template, you need to transmit the URL of the iPhone template page using the following hidden field and value:

```
<input type="hidden" name="TP" value="PaymentPage_1_iPhone.htm">
```

IMPORTANT: We can only guarantee that our secure payment pages are iPhone compatible. We can't guarantee that all external pages accessible via our payment pages, e.g. third party or bank websites are iPhone compatible.

The data entry interface is especially designed for the small iPhone screen. The look & feel can be customized to the merchant's needs by simply adding some hidden fields in the form he sends us. The following hidden fields are used to transmit the look & feel parameters for the iPhone template to our system:

```
<input type="hidden" name="TITLE" value="">
<input type="hidden" name="BGCOLOR" value="">
```

<input type="hidden" name="TXTCOLOR" value="">

<input type="hidden" name="TBLBGCOLOR" value="">

<input type="hidden" name="TBLTXTCOLOR" value="">

<input type=”hidden” name=”HDTBLBGCOLOR” value=””>

<input type=”hidden” name=”HDTBLTXTCOLOR” value=””>

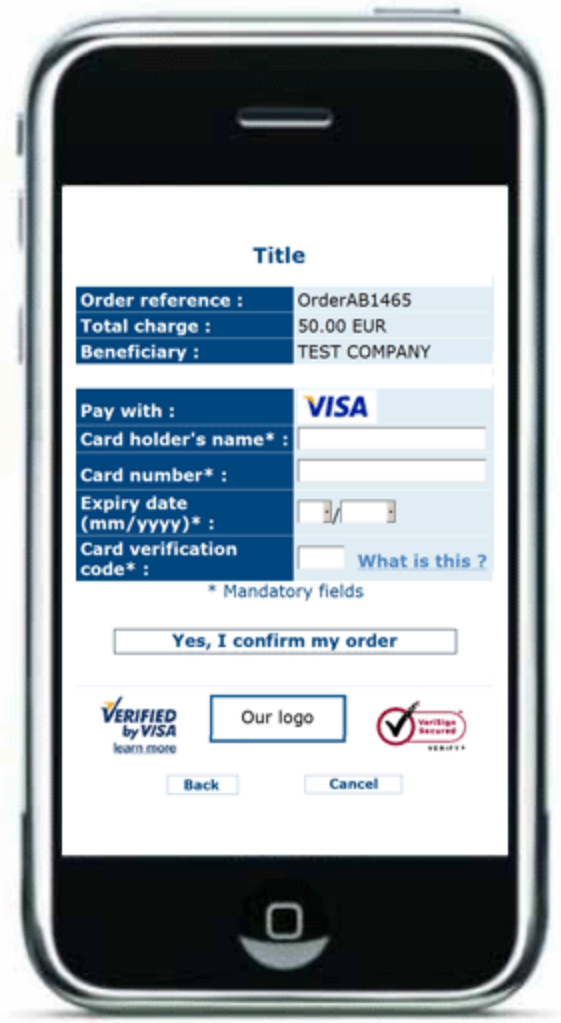
<input type=”hidden” name=”HDFONTTYPE” value=””>

<input type="hidden" name="BUTTONBGCOLOR" value="">

<input type="hidden" name="BUTTONTXTCOLOR" value="">

<input type="hidden" name="FONTTYPE" value="">

Field	Usage	Default value
TITLE	Title of the page	_
BGCOLOR	Background color	#FFFFFF
TXTCOLOR	Text color	#00467F
TBLBGCOLOR	Background color for the right columns	#E1EDF4
TBLTXTCOLOR	Text color for the right columns	#000000
HDTBLBGCOLOR	Background color for the left columns	#00467F
HDTBLTXTCOLOR	Text color for the left columns	#FFFFFF
HDFONTTYPE	Font family for the left columns	Verdana
BUTTONBGCOLOR	Button background color	#FFFFFF
BUTTONTXTCOLOR	Button text color	#00467F
FONTTYPE	Font family	Verdana



7.2 Template based page layout (Dynamic template)

The dynamic template page is an advanced technique for customizing the design of the payment pages. Dynamic template usage is restricted to certain subscriptions. If you are interested in this option and it is not present in the options list of your subscription page in your account, please contact our sales team.

When the merchant uses a dynamic template page, he fully designs his own template page, leaving just one area in that page to be completed by our system. The URL of the merchant's template page needs to be sent to us in the hidden fields for each transaction. Please bear in mind that using a dynamic template page involves an additional request from our system to look up your template page. This increases the time needed for the payment process.

7.2.1 Hidden fields

The following hidden field is used to transmit the URL of your template page:

<input type="hidden" name="TP" value="">

Field	Usage
TP	URL of the merchant's dynamic template page (the page must be hosted at the merchant's end). The URL must be absolute (contain the full path), it cannot be relative. Do not specify any ports in your URL, we only accept ports 443 and 80. Any component included in the template page must also have an absolute URL.

*For further technical details about this field, please refer to the online **Parameter Cookbook**.*

7.2.2 Payment zone

The dynamic template page can be designed completely to your liking. The only requirement is that it must contain the string "$$$PAYMENT ZONE$$$" indicating the location where our e-Commerce module can add its fields dynamically. It must therefore contain at least the following:

```
<html>
$$$PAYMENT ZONE$$$
</html>
```

IMPORTANT: do not use BASE tags, frames or FORM tags to encapsulate the $$$PAYMENT ZONE$$$ string.

Example

An example of a dynamic template page is available at the following address:

https://secure.ogone.com/ncol/template_standard.htm

7.2.3 Dynamic behavior

The same template page can be used for all orders, or it may be generated dynamically by the merchant's application according to the order parameters.

To generate the template page dynamically, the merchant can choose between creating a page specific to the order whose URL is transmitted in the hidden fields or using a fixed URL but returning a result derived from the order number. To allow this, our system adds the main payment data – including the merchant's order reference number (cf. Processing after payment) – when it retrieves the template page:

HTTP request = url_page_template ?orderID=...&amount=...&currency=…

7.2.4 Style sheets

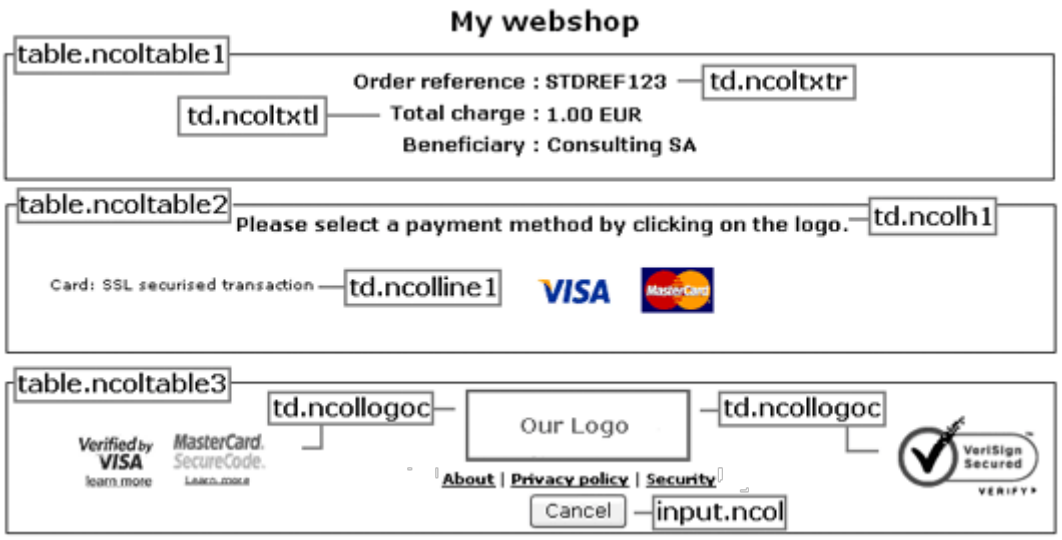
You can personalize the look & feel of your payment pages by adding style sheets to your template page.

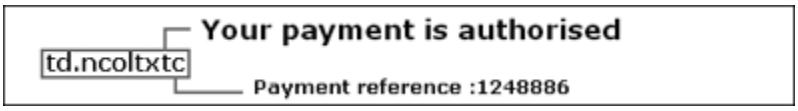
We have defined a class for the various types of tables and cells within our tables as well as a class for the submit buttons. Add the following block of code between the tags <head></head> and change the

properties of those classes to fit to the look & feel of your site (see the example of the above mentioned template page):

```
<style type="text/css">
<!--
td.ncolh1 {background-color : #006600; color : yellow; font-family : verdana}
td.ncoltxtl {background-color : #ffffcc; color : black; text-align : right; font-weight : bold}
td.ncoltxtl2 {background-color : #ffffcc; color : black; text-align : right; font-weight : bold}
td.ncoltxtr {background-color : #ffffcc; color : black; text-align : left; font-weight : bold}
td.ncoltxtc {background-color : #ffffcc; color : black; text-align : center; font-weight : bold}
td.ncolinput {background-color : #ffffcc; color : black}
td.ncolline1 {background-color : #ffffff; color : black}
td.ncolline2 {background-color : #ffffcc; color : black}
input.ncol {background-color : #006600; color : white}
td.ncollogoc {background-color : #ffffcc; color : black; text-align : center; font-weight : bold}
table.ncoltable1 { background-color: #ffffcc;   }
table.ncoltable2 { background-color: #ffffcc;  border-width : medium; border-color : green; }
table.ncoltable3 { background-color: #ffffcc;   }
-->
</style>
```

When you enter your own layout instructions, you must adhere to the cascading style sheet syntax. We strongly advise you to test it in various browsers as the way they handle style may differ enormously.





7.2.5 Performance

Our system is configured with a 5 second timeout for the request to retrieve the merchant's dynamic template page. If a timeout is flagged, we will use our static template instead.

We would be happy to change this timeout (HTTPTimeOut) at our end at the merchant's request (via a support ticket).

IMPORTANT: This HTTPTimeOut field has an impact on both dynamic template requests and post-payment feedback requests (see Chapter 8.3). Consequently, if the merchant were to decide to change it to e.g. 15 seconds, the feedback request timeout will also increase to 15 seconds.

For each order, our system performs a request to retrieve your dynamic template page. If you have high transaction volumes or you have a large template page (e.g. your dynamic template page contains a large number of images), these HTTP requests could take a long time. Please contact our sales department for a solution if you have high transaction volumes.

7.3 Secure environment padlock

The URL used to connect the customer to our platform uses a secure protocol (http**s**). All the communication between our e-Commerce platform and the customer is securely encrypted.

However, the small padlock on the browser – which indicates to the customer that the site is secure – may not be displayed if some elements (e.g. images) in the template page are not located on a secure server or if some frames on the screen show pages that do not originate from secure sites.

Even if the payment processing communication is encrypted, most browsers will not recognize a secure connection unless **all the elements** on the screen, including images, sounds, etc. come from secure sites.

For merchants that do not have a secure site, please bear in mind the following rules:

1. Do not use frames for the payment pages: you can refresh the entire screen with a template page that looks as if you are using frames or allow the payment to be processed in a new window.

2. Do not link files to the template page (<link> tag) that you use for the payment page. Instead, use the <style> and <script> tags to include styles and scripts into the template page.

3. Make sure the images in your template are stored on a secure server (the template page can be on a non-secure server, however the images cannot be). We can offer hosting for those elements (see the image hosting options in your account).

7.4 Cancel button

By default a "Cancel" button is available on our secure payment pages to allow the customer to cancel/interrupt his transaction. If you wish to hide the "Cancel" button, you can enable the corresponding checkbox in the "Payment page layout" tab of the "Technical Information Page" in your account.

8 Transaction Feedback to the Customer and the Merchant

Best practice: Redirection with parameters on the accept-/exception-/cancel-/declineurl with a deferred post-payment feedback request as backup, Chapter 8.2.

The feedback to the merchant and his customer – when the payment is accepted, the customer cancelled the payment or the acquirer declined the payment more than the maximum permissible number of times – depends on parameters defined by the merchant.

8.1 Default reaction

If the merchant has not specified a specific reaction, our system will display the customer the standard message: "Your payment is authorized" or "The transaction has been denied". This message is inserted into the template page.

In this page, we also add a link to the merchant's website and/or the merchant's catalog, using the URLs (homeurl and catalogurl) sent in the hidden fields of the order form. If the URLs are not specified in the hidden fields, our system will use the URL stated in the management module of your account (account > step 1).

8.1.1 Hidden fields

Following are the hidden fields used to transmit the URLs:

<input type="hidden" name="catalogurl" value="">

<input type="hidden" name="homeurl" value="">

Field	Usage
catalogurl	(Absolute) URL of your catalogue. When the transaction has been processed, your customer is requested to return to this URL via a button.
homeurl	(Absolute) URL of your home page. When the transaction has been processed, your customer is requested to return to this URL via a button. When you send the value "NONE" the button leading back to the merchant's site will be hidden.

*For further technical details about these fields, please refer to the online **Parameter Cookbook**.*

8.2 Redirection depending on the payment result

In the hidden fields of his ordering form, the merchant can send 4 URLs (accepturl, exceptionurl, cancelurl and declineurl) where our system redirects the customer at the end of the payment process. The merchant can also configure these URLs in the "Transaction feedback" tab, "HTTP redirection in the browser" section of the "Technical Information" page.

Example of the use of an "accepturl" to personalize the customer's response:

8.2.1 Hidden fields

The following hidden fields are used to transmit the URLs:

<input type="hidden" name="accepturl" value="">

<input type="hidden" name="declineurl" value="">

<input type="hidden" name="exceptionurl" value="">

<input type="hidden" name="cancelurl" value="">

Field	Usage
accepturl	URL of the web page to display to the customer when the payment has been authorized (status 5), stored (status 4), accepted (status 9) or is waiting to be accepted (pending, status 41, 51 or 91).
declineurl	URL of the web page to show the customer when the acquirer declines the authorization (status 2 or 93) more than the maximum permissible number of times.
exceptionurl	URL of the web page to display to the customer when the payment result is uncertain (status 52 or 92). If this field is empty the customer will be displayed the accepturl instead.
cancelurl	URL of the web page to display to the customer when he cancels the payment (status 1). If this field is empty the declineurl will be displayed to the customer instead.

*For further technical details about these fields, please refer to the online **Parameter Cookbook**.*

8.2.2 Browser alert notification

When a customer returns from our secure payment pages to the merchant's website, he might get a browser alert, warning him that he is entering a non-secure environment (since he goes from an https:// environment to a http:// environment). When we detect a redirection to the merchant's website, we can display a message to the customer notifying him about the possibility of a warning (see first screenshot in Chapter 8.2), thereby avoiding undue concern about any browser alert. The merchant can activate this option in the "Transaction feedback" tab, "HTTP redirection in the browser" section of the Technical Information page ("I want $genericname$ to display a short text to the customer on the secure payment page if a redirection to my website is detected immediately after the payment process").

8.2.3 Database update option

The merchant can use this redirection on the accept-/exception-/cancel-/declineurl to trigger automatic back office tasks such as database updates. When a payment is executed, we can send the transaction parameters on the merchant's accept-, exception-, cancel- or declineurl.

The merchant can activate this option in the "Transaction feedback" tab, "HTTP redirection in the browser" section of the Technical Information page ("I would like to receive transaction feedback parameters on the redirection URLs").

8.2.3.1 Feedback parameters

When a payment is executed, we can send the following parameter list on the merchant's accept-, exception-, cancel- or declineurl.

Parameter	Value
orderID	Your order reference
amount	Order amount (not multiplied by 100)
currency	Order currency
PM	Payment method
ACCEPTANCE	Acceptance code returned by acquirer
STATUS	Transaction status (see Appendix 3 for a short status overview)
CARDNO	Masked card number
PAYID	Payment reference in our system
NC ERROR	Error code
BRAND	Card brand (our system derives this from the card number)
ED	Expiry date
TRXDATE	Transaction date
CN	Cardholder/customer name
SHASIGN	SHA signature calculated by our system (if SHA-1-OUT configured)

*For further technical details about these fields, please refer to the online **Parameter Cookbook**.*

The list of feedback parameters can be longer for merchants who have activated certain options in their accounts, such as the Fraud Detection Module. Please refer to the respective option documentation for more information on extra feedback parameters linked to the option.

Example

https://www.yourwebsite.com/acceptpage.asp?orderID=ref12345&currency=EUR&amount=25
&PM=CreditCard&ACCEPTANCE=test123&STATUS=5&CARDNO=XXXXXXXXXXXX1111
&PAYID=1136745&NCERROR=0&BRAND=VISA&ED=0514&TRXDATE=12/25/08&CN=John Doe

The merchant can send us two extra parameters in the hidden fields of the order form, in order to retrieve them as feedback parameter after the payment. The following hidden fields are available:

<input type="hidden" name="complus" value="">

<input type="hidden" name="paramplus" value="">

Field	Usage
complus	Field for submitting a value you would like returned in the feedback request.
paramplus	Field for submitting some parameters and their values you would like returned in the feedback request. The field paramplus is not included in the feedback parameters as such; instead, the parameters/values you submit in this field will be parsed and the resulting parameters added to the http request.

*For further technical details about these fields, please refer to the online **Parameter Cookbook**.*

Example

Following are the extra hidden fields sent by the merchant:

<input type="hidden" name="complus" value="12345678912345678912345678 9">
<input type="hidden" name="paramplus" value="SessionID=126548354&ShopperID=73541312">

Resulting in redirection with feedback parameters:

https://www.yourwebsite.com/acceptpage.asp?[…standard.parameters…]
&COMPLUS=12345678912345678912345678 9&SessionID=126548354&ShopperID=73541312

8.2.3.2 Security measures

The redirection process is visible because it goes via the customer's browser. Consequently, the merchant must use an SHA signature (see Appendix 1) to verify the contents of the request and prevent customers tampering with the data in the URL field which could result in fraudulent database updates. If the merchant does not configure a SHA-1-OUT signature we will not send any parameters on his accept-, exception-, cancel- or declineurl.

8.2.3.3 Combination with a feedback request

The merchant can use a deferred/background feedback request as a fall back option for the redirection (see chapter 8.3).

If the communication with the customer is interrupted, for instance when the customer exits his browser window before reaching the accept-, exception-, cancel- or declineurl, the merchant will not

receive the redirection on the accept-, exception-, cancel- or declineurl. However, if the merchant enters a post-payment URL in the "Transaction feedback" tab, "Direct HTTP server-to-server request" section (URL fields) of the Technical Information page and sets the timing of the request to "Always deferred (not immediately after the payment)" he will receive a deferred feedback request shortly after the transaction.

For this to work, the merchant's post-payment page must be capable of accepting a request for an order that has already been processed. The merchant will receive this deferred feedback request in any case, even if the redirection on the accept-, exception-, cancel- or declineurl was successful. This second request can be ignored if the order status has already been updated in the merchant's database following the redirection on the accept-, exception-, cancel- or declineurl.

8.3 Direct feedback requests (Post-payment)

After the payment, our system can send an http request to a URL specified by the merchant, transmitting the transaction data.

This process allows the merchant to update his database with the order status etc. and trigger an "end of order" process (if this has not already been done after a redirection). It is also an alternative way of generating a personal response for the customer in case of specific needs (if this has not already been done via a redirection).

8.3.1 Post-payment URLs and parameters

8.3.1.1 Post-payment URLs

To automate your back-office tasks, you can define the URLs of two executable pages on your site in the "Transaction feedback" tab, "Direct HTTP server-to-server request" section (URL fields) of the Technical Information page. One of these settings can be the URL where you receive the parameters in a request if the payment's status is accepted, pending or uncertain. The other can be the URL where you want to receive the parameters in a request when the transaction has been cancelled by the client or been declined too many times by the acquirer (i.e. more than the maximum permissible number of payment attempts as set in the "Global transaction parameters" tab, "Payment retry" section of the Technical Information page). These two URLs may differ, but they may also be identical. You may also enter a URL for the first case but not for the second. Do not specify any ports in your URL; we only accept port 443 and port 80.

If you would also like to receive a deferred HTTP request in the case of a transaction status change, you can set an additional URL in the field in the "Transaction feedback" tab, "HTTP request for status changes" section of the Technical Information page (and select a timing of the request). This is similar to a post-payment URL with the difference that it is relevant for potential background processes. You can use the same URL here as the one set in the "Direct HTTP server-to-server request" section, but please bear in mind that there is no point in using it to generate a personal response for the customer in this (background) case.

8.3.1.2 Variable post-payment URLs

If you have a post-payment page configured in the Technical Information page in your account, but have several shops each connected to a specific directory for receiving the post-payment feedback, you can make a part of your post-payment URL variable.

This variable part can also be used to e.g. "adapt" the feedback request to include session information, passing it as a part of the URL rather than as an additional parameter. This is the case for Intershop platforms or Servlets systems.

The hidden field you have to use is the following:

<input type="hidden" name="PARAMVAR" value="">

Field	Usage
PARAMVAR	The variable part to include in the URLs used for feedback requests

*For further technical details about this field, please refer to the online **Parameter Cookbook**.*

Example

Post-payment URL in the merchant's Technical Information page:
https://www.yourwebsite.com/<PARAMVAR>/yourpage.asp

Following is the extra hidden field sent by the merchant:
<input type="hidden" name="PARAMVAR" value="shop1">

Resulting in the following Post-payment URL for the transaction:
https://www.yourwebsite.com/shop1/yourpage.asp

8.3.1.3 Feedback parameters

Our http request to your post-payment URL will contain the same feedback parameters as described in Chapter 8.2.3.1.

8.3.2 Timing of the feedback request

In the "Transaction feedback" tab, "Direct HTTP server-to-server request" section of the Technical Information of your account, you can choose the timing of the feedback request:

- **None**:

  In this case our system will not send any feedback request. This option allows you to disable your post-payment URLs in case of maintenance or problems on your server.

- **Always deferred (not immediately after the payment)**:

  The feedback request will be sent shortly after the end of the payment process. The feedback request will be a background task and cannot be used to send a personalized feedback to the customer on the merchant's website.

  If the merchant does not use his post-payment page to personalize a response for his customer, he can receive the feedback request in the background and deferred.

- **Always online (immediately after the payment to allow customisation of the response seen by the customer)**:

  The feedback request will be sent "online" sometime between our system's receipt of the acquirer's response and the time it notifies the customer of the payment result.

  In this case, the payment process takes longer for the customer, but the merchant can send a personalized response to the customer.

  The disadvantage of the online post-payment feedback process is that the merchant's system might be detrimentally affected if there are too many requests to his post-payment page (e.g. high per minute transaction volume) – this could result in long response times before customers receive on screen feedback.

- **Online but switch to a deferred request in intervals when the online requests fail**:

  This option allows merchants who require an online post-payment feedback (to tailor the response displayed to the customer) to have a fall-back option, should the online request on his post-payment page fail. In this case we will retry the feedback request every 10 minutes up to a maximum of 10 times (deferred). This way the merchant does not miss out on the transaction feedback, should the online post-payment feedback request have failed as a result of e.g. temporary server problems at his end. The customer will be displayed the standard transaction feedback from our system (see Chapter 8.1).

8.3.3 Example of a post-payment executable page on the merchant's site

The following is an example of a post-payment ASP page on the merchant's server. This script updates an order status and returns an HTML page.

The merchant is, of course, at liberty to use the development language of his choice.

REMARK: This is just sample code. We cannot guarantee that this code works correctly on your server. See Appendix 3 for a short explanation on the interpretation of the most important statuses.

```
<%'receiving order data
orderID = Request("orderID")
amount = Request("amount")
currency = Request("currency")
acceptance = Request("acceptance")
status = Request("status")
...
'retrieving order of the order status
Set object_DB=Server.CreateObject("ADODB.Connection")
object_DB.Open "MyDatabase"
SQLQuery="SELECT * FROM ORDERS WHERE ID='" & orderID & "'"
Set com_DB = so_DB.Execute(SQLQuery)

Errormessage=""

If com_DB.EOF then
        Errormessage = "unknown orderID"
End if

If len(errormessage) = 0 then
        If (com_DB("amount")<>amount) or Strcomp(com_DB("dev"),currency) <> then
                Errormessage = "data does not match"
        End if
End if

If len(errormessage) = 0 then
        If (len(acceptance) > 0) then
                ' processing of accepted orders
                ' Update DB, send email
                ...
                ...
            Response.write("<html>")
            Response.write("Thank you for your order....<br>")
            Response.write("You can now <a href=http://www.mysite.com/download ?ID=" & order
            & ">download your super software</a>")
            Response.write("</html>")
```

```
        elseif status = 51 then
                ' processing of offline authorized orders
                ' Update DB send specific email or wait for
                ' authorization to send it.
                ...
                ...
            Response.write("<html>")
            Response.write("You will be informed of the authorization of your order ...<br>")
            Response.write("</html>")


        else
                ' processing of uncertain authorizations
                ' Update DB, send specific email or wait for
                ' helpdesk support to send it.  ...
                ...
                ...
            Response.write("<html>")
            Response.write("You will be informed of the authorization of your order ...<br>")
            Response.write("</html>")

        end if
else
                ...
                ...
            Response.write("<html>")
            Response.write("An error occured ...<br>")
            Response.write("</html>")
end if%>
```

8.3.4 Response to the customer

We use a possible reply of your post-payment page to show a feedback (end of transaction page) to your customer. If your post-payment page replies with:

- An HTML page (containing an <html> tag)

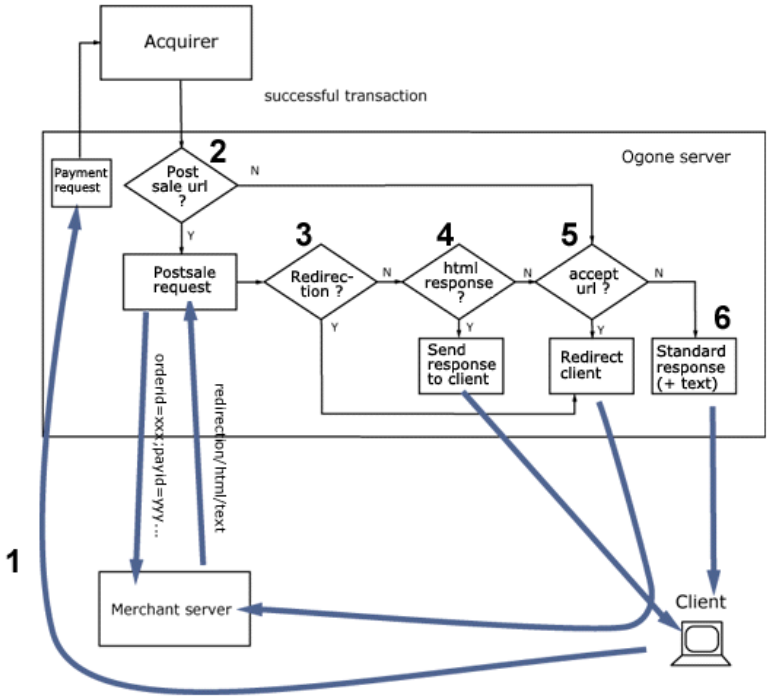
or

- A redirection (HTTP 302 Object Moved)

our system will send this HTML page "as is" to the client browser or perform the redirection, rather than redirecting your customer at the end of your post-payment feedback process to one of the 4 URLs you may have sent in the hidden fields (accepturl, exceptionurl, cancelurl and declineurl as described in Chapter 8.2).

Alternatively, if you use none of the above as feedback to your customer, you can have your post-payment page respond with a few lines of text (no <html> tag) which we will include in our standard response, or our system will just show the standard response (as described in Chapter 8.1 of this manual).

The diagram below shows the process at the end of a transaction, should the payment be authorized or accepted, with an online post-payment request. (When the payment is cancelled, declined or uncertain the process is similar but the "cancelurl" / "declineurl" / "exceptionurl" and "cancellation/rejection" pages are used instead).



8.3.5 Feedback request timeout

Our system is configured with a 20 second timeout for the online feedback request to the merchant. If a timeout is flagged for this online feedback request, the request will fail and the merchant will have to perform his back office tasks manually.

If the merchant has activated the possibility to receive an offline feedback request in the event of the online feedback request failing (Timing of request option: "Online but switch to a deferred request in intervals when the online requests fail"), the timeout for the online feedback request in our system will be 10 seconds.

We would be happy to change this timeout (HTTPTimeOut) at our end at the merchant's request (via a support ticket).

IMPORTANT: This HTTPTimeOut field has an impact on both feedback requests and dynamic template requests (see Chapter 7.2). Consequently, if the merchant were to decide to change it to e.g. 40 seconds, the dynamic template request timeout will also increase to 40 seconds.

8.4 Security: check origin of the request

If you receive a request with parameters from our system, you have two possibilities to verify that the request was truly sent from our system: an IP address check and an SHA signature.

8.4.1 IP address check (only for feedback requests)

You can configure our IP addresses in your firewall to be certain that the request is coming from one of our servers; alternatively, you can simply test the IP origin in your CGIs. The IP addresses are published in the FAQ section in your account. Please note that different ranges of possible IP addresses exist and that these IP addresses are subject to change!

8.4.2 SHA-1-OUT signature (for feedback requests and redirections)

We strongly recommend that you use an SHA signature to verify the contents of a request or redirection; this will e.g. prevent customers from tampering with the data in the URL field which could result in an incorrect database update. For further information about the SHA-1-OUT signature, please refer to Appendix 1.

8.5 Confirmation e-mails

8.5.1 E-mails to the merchant

Our system can send you a payment confirmation e-mail for each transaction (option to configure in the "Transaction e-mails" tab, "E-mails to the merchant" section of the Technical Information page).

You can also receive e-mails notifying transaction status changes.

8.5.2 E-mails to the customer

Our system can send an automatic e-mail to your customer notifying him of the transaction registration. This is a standard e-mail whose contents cannot be changed. The "From" address used when sending the e-mail is the address you entered in the "E-mail address(es) for transaction-related e-mails" field. If you entered more than one e-mail address in this field, we will use the first of them.

You can activate this option in the "Transaction e-mails" tab, "E-mails to the customer" section of the Technical Information page.

If you want us to send an e-mail to your customer, you also have to send us his e-mail address in the hidden field:

<input type="hidden" name="EMAIL" value="">

Field	Usage
EMAIL	Customer's e-mail address

*For further technical details about this field, please refer to the online **Parameter Cookbook**.*

9 Other Optional Hidden Fields

There are a number of other optional hidden fields the merchant can send us for specific purposes. This chapter provides an overview of these hidden fields and their usage.

9.1 Payment method and payment page specifics

9.1.1 Payment method selection at the merchant's side

9.1.1.1 Showing a specific payment method

When a customer is displayed our secure payment page, he will be shown an overview of the possible payment methods the merchant has activated in his account. If the customer is to select the payment method on the merchant's website instead of on our payment page, he can send us the payment method name and brand (only used when the payment method is "CreditCard") in the hidden fields, so we will only show this particular payment method on our payment page and will only allow payment by this payment method.

The hidden fields are the following:

<input type="hidden" name="PM" value="">

<input type="hidden" name="BRAND" value="">

Field	Usage
PM	Payment method
BRAND	Credit card brand

*For further technical details about these fields, please refer to the online **Parameter Cookbook**.*

Examples

* Hidden fields in case your customer has selected VISA on your site:

<input type="hidden" name="PM" value="CreditCard ">
<input type="hidden" name="BRAND" value="VISA">

* Hidden fields in case you only want your customer to pay by creditcard (for instance, if you also have other payment methods you don't wish to show):

<input type="hidden" name="PM" value="CreditCard ">
<input type="hidden" name="BRAND" value="">

* Hidden fields in case your customer has selected iDEAL on your site:

<input type="hidden" name="PM" value="iDEAL">
<input type="hidden" name="BRAND" value="">

9.1.1.2 Allowing the customer to choose another payment method: backurl

If the customer selects the payment method on the merchant's website, we will only show the selected payment method on our payment page.

If the payment with this payment method is unsuccessful and the customer would like to try using another payment method, he will not be presented with a list of the merchant's payment methods on our secure payment pages since the payment method selection took place on the merchant's website and not on our secure payment pages.

In this case the merchant can use the "backurl" to redirect the customer to a URL on the merchant's website where he can select another payment method. When the customer clicks the "Back" button on our secure payment page, after a declined authorization, or after having cancelled from a third party or bank website we redirect him to the URL the merchant entered as "backurl".

IMPORTANT: The back button described in this section is the back button in our secure payment pages, NOT the back button in your browser.

You can enter the "backurl" in the "Payment page layout" tab of the "Technical Information" page in your account, but can also send us a specific "backurl" in the hidden fields for a transaction if you prefer not to use the same "backurl" as entered in the "Payment page layout" tab of the "Technical Information" page in your account.

The "backurl" sent in the hidden fields will override the general "backurl" entered in the "Payment page layout" tab of the "Technical Information" page in your account. You can send the "backurl" in the following hidden field:

<input type="hidden" name="backurl" value="">

Field	Usage
backurl	URL of the web page to display to the customer when he clicks the "back" button on our secure payment page.

*For further technical details about this field, please refer to the online **Parameter Cookbook**.*

If the customer selects his payment method on our secure payment pages and not on the merchant's website, the "backurl" is not taken into account. When a customer clicks on the "Back" button on our secure payment page, he will simply be redirected to our secure payment method selection page containing a list of the merchant's payment methods.

9.1.2 Showing a specific list of payment methods

If the customer is to select the payment method from a specific list of payment methods on our payment page, the merchant can send us this list of payment methods in the hidden fields, so we will only show these specific payment methods on our payment page.

The hidden field is the following:

<input type="hidden" name="PMLIST" value="">

Field	Usage
PMLIST	List of selected payment methods and/or credit card brands. Separated by a ";" (semi-colon).

*For further technical details about these fields, please refer to the online **Parameter Cookbook**.*

Example

* Hidden field in case you only want your customer to choose between VISA and iDEAL on our payment page (e.g., if you also have other payment methods that you don't want to be displayed):

<input type="hidden" name="PMLIST" value="VISA;iDEAL">

9.1.3 Layout of the payment methods

You can arrange the layout/list of the payment methods on our payment page using the following hidden field:

<input type="hidden" name="PMListType" value="">

Field	Possible values
PMListType	The possible values are 0,1 and 2. 0: Horizontally grouped logos with the group name on the left (default value) 1: Horizontally grouped logos with no group names 2: Vertical list of logos with specific payment method or brand name

*For further technical details about this field, please refer to the online **Parameter Cookbook**.*

9.1.4 3-D secure

If you are working with 3-D Secure, you can choose how you want the identification page to be displayed to the customer by sending us an extra parameter in the hidden fields.

IMPORTANT: in certain cases your choice can be overridden by our system, based on brand scheme regulations or technical compliance.

The hidden field is the following:

<input type="hidden" name="WIN3DS" value="">

Field	Possible values
WIN3DS	"MAINW": to display the identification page in the main window (default value and **recommended by VISA/MasterCard**) "POPUP": to display the identification page in a POPUP window and return to main window at the end

*For further technical details about this field, please refer to the online **Parameter Cookbook**.*

9.2 Operation

IMPORTANT: The ability to work in two steps (authorization + data capture) depends on the payment methods you wish to use. (See the online **Payment Methods Processing/Procedure** overview).

You can send us a specific operation code for a transaction if you prefer not to use the same operation code as selected in the "Global transaction parameters" tab, "Default operation code" section of the "Technical Information" page in your account for that transaction.

The operation code you send us in the hidden fields will override the general operation code selected in the "Global transaction parameters" tab, "Default operation code" section of the "Technical Information" page in your account. You can send the operation code in the following hidden field:

<input type="hidden" name="operation" value="">

Field	Usage
operation	Operation code for the transaction. Possible values for new orders: ▪ **RES**: request for authorization ▪ **SAL**: request for sale (payment)

*For further technical details about this field, please refer to the online **Parameter Cookbook**.*

IMPORTANT: In order for this parameter to be taken into account by our system, it needs to be included in the SHA signature calculation for the transaction. Please refer to Appendix 1 for more information on SHA-1.

9.3 User field

If you have multiple users in your account and you want to register transactions associated with a specific user (e.g. for call center agents logging transactions via e-Commerce), you can send the UserID in the following hidden field:

<input type="hidden" name="USERID" value="">

Field	Usage
USERID	The username specified in the account's user management page

*For further technical details about this field, please refer to the online **Parameter Cookbook**.*

This field is just an informative field to add a UserID to a specific transaction. We do not perform any check at our end to establish e.g. if there have been password errors for this user. The only check we perform is to verify that the UserID is valid. If the UserID does not exist, we will replace it by the default UserID of the account (PSPID).

Please refer to the online Parameter Cookbook for other fields.

10 Appendix 1: SHA-1

For each order, the merchant's server generates a unique character string, hashed with the SHA-1 algorithm developed by NIST (see http://www.w3.org/TR/1998/REC-DSig-label/SHA1-1_0).

10.1 SHA-1-IN signature

This string is constructed by concatenating the values of the fields sent with the order (sorted alphabetically, in the format 'parameter=value'), separated by a passphrase. The passphrase is defined in the Merchant's *Technical information*, under the tab "Data and Origin Verification", section "Checks for eCommerce." For the full list of parameters to include in the SHA Digest, please refer to Appendix 6. Please note that these values are all case sensitive when compiled to form the string before the hash!

IMPORTANT

- all parameter names should be in UPPERCASE (to avoid any case confusion)
- Parameters that do not have a value should NOT be included in the string to hash

When you hash the string composed with the SHA algorithm, a hexadecimal Digest will be returned. The length of the SHA Digest is 40 characters for SHA-1, 64 for SHA-256 and 128 for SHA-512. This result should be sent to our system in your order request using the "SHASign" field.

Our system will recompose the SHA string based on the received parameters and compare the Merchant's Digest with our generated Digest. If the result is not identical, the order will be declined. This check ensures the accuracy and integrity of the order data.

You can test your SHASign at https://secure.ogone.com/ncol/test/testsha.asp

Example of a basic SHA-1-IN calculation

parameters (in alphabetical order)
amount: 15.00 -> 1500
currency: EUR
Operation: RES
orderID: 1234
PSPID: MyPSPID

SHA Passphrase (In technical info)
Mysecretsig1875!?

string to hash
AMOUNT=1500Mysecretsig1875!?CURRENCY=EURMysecretsig1875!?OPERATION=RESMysecretsig1875!?ORDERID=1234Mysecretsig1875!?PSPID=MyPSPIDMysecretsig1875!?

resulting Digest (SHA-1)
EB52902BCC4B50DC1250E5A7C1068ECF97751256

If the SHASign sent in the hidden HTML fields for the transaction does not match the SHASign which we derived using the details of the order and the additional string (password/pass phrase) entered in the SHA-1-IN Signature field in the "Data and origin verification" tab, "Checks for e-Commerce" section of the Technical Information page, you will receive the error message *"unknown order/1/s".*

If the "SHASign" field in the hidden HTML fields is empty but an additional string (password/pass phrase) has been entered in the SHA-1-IN Signature field in the "Data and origin verification" tab, "Checks for e-Commerce" section of the Technical Information page, indicating you want to use a SHA signature with each transaction, you will receive the error message *"unknown order/0/s".*

The following hidden field is used to transmit the SHA signature to our system:

Field	Usage
SHASign	Unique character string for order data validation. A string hashed with the SHA-1 algorithm will always be 40 characters long

10.2 SHA-1-OUT signature

This string is constructed by concatenating the values of the fields sent with the order (sorted alphabetically, in the format 'parameter=value'), separated by a passphrase. The passphrase is defined in the Merchant's *Technical information*, under the tab "Transaction Feedback", section "All transaction Submission modes." For the full list of parameters to include in the SHA Digest, please refer to Appendix 6. Please note that these values are all case sensitive.

Important: all parameter names should be in UPPERCASE (to avoid any case confusion)

In the same way we re-create the Digest to validate the Input of the transaction with the SHA-IN, you have to reconstruct the Hash, this time using your SHA-OUT passphrase and the parameters received from our system.

If the outcome is not identical, the request's parameters might have been tampered with. This check ensures the accuracy and integrity of the parameter values sent in the request.

Example of a basic SHA-1-OUT calculation

parameters (in alphabetical order)
ACCEPTANCE: 1234
amount: 15
BRAND: VISA
CARDNO: xxxxxxxxxxxx1111
currency: EUR
NCERROR: 0
orderID: 12
PAYID: 32100123
PM: CreditCard
STATUS: 9

SHA Passphrase (In technical info)
Mysecretsig1875!?

String to hash
ACCEPTANCE=1234Mysecretsig1875!?AMOUNT=1500Mysecretsig1875!?BRAND=VISAMysecretsig1875!?CARDNO=xxxxxxxxxxxx1111Mysecretsig1875!?CURRENCY=EURMysecretsig1875!?NCERROR=0Mysecretsig1875!?ORDERID=12Mysecretsig1875!?PAYID=32100123Mysecretsig1875!?PM=CreditCardMysecretsig1875!?STATUS=9Mysecretsig1875!?

Resulting Digest (SHA-1):
28B64901DF2528AD100609163BDF73E3EF92F3D4

10.3 SHA-1 module

To be able to hash a string and send it to us, you must first install an SHA-1 module on your server. If you work in a windows 2000/asp environment, you can download a DLL that includes a method to hash a string using SHA-1 in the support > documentation page.

Because there are many possible combinations of operating systems (version-numbers/patches) and programming languages we cannot be held responsible for any errors on your server during installation and/or processing.

SHA-1 modules can be found in the Internet, so you will not have any problem in finding a suitable one for your server. To help you find a SHA-1 module for your environment, we have compiled the following list of sites:

General info on SHA at W3.org:

http://www.w3.org/PICS/DSig/SHA1SHA-1_1_0.html

.NET/SHA1:

http://msdn2.microsoft.com/en-us/library/system.security.cryptographysha1.managed.aspx

PHP/SHA1:

http://www.php.net/manual/en/ref.mhash.php

11 Appendix 2: Troubleshooting

The following section contains a non-exhaustive list of possible errors:

unknown order/1/r

This error means that the referrer we detected is not a URL the merchant has entered in the URL field in the "Data and origin verification" tab, "Checks for e-Commerce" section of his Technical Information page. The merchant is sending us the form with the hidden fields containing the order information from a different page from the one(s) entered in the URL field in the "Data and origin verification" tab, "Checks for e-Commerce" section.

unknown order/0/r

This error means our server has not detected a referrer in the request we received. The merchant is sending us order details, but we do not know where they originated from. Please ensure that no methods are being used that blocking the referrer information (payment page in pop up, special web server configuration, customer's browser configuration, ...). If the customer's browser does not send the referrer information, we can bypass the referrer check if a SHASign is present and correct. (See Chapter 6.2)

unknown order/1/s

You will receive this error message if the SHASign sent in the hidden HTML fields for the transaction does not match the SHASign calculated at our end using the details of the order and the additional string (password/pass phrase) entered in the SHA-1-IN Signature field in the "Data and origin verification" tab, "Checks for e-Commerce" section of the Technical Information page.

unknown order/0/s

You will receive this error message if the "SHASign" field in the hidden HTML fields is empty but an additional string (password/pass phrase) has been entered in the SHA-1-IN Signature field in the "Data and origin verification" tab, "Checks for e-Commerce" section of the Technical Information page, indicating you want to use a SHA signature with each transaction.

PSPID not found or not active

This error means the value you have entered in the PSPID field does not exist in the respective environment (test or prod) or the account has not yet been activated.

no <parameter> (for instance: no PSPID)

This error means the value you sent for the obligatory <parameter> field is empty.

<parameter> too long (for instance: currency too long)

This error means the value in your <parameter> field exceeds the maximum length.

amount too long or not numeric: ... OR Amount not a number

This error means the amount you sent in the hidden fields either exceeds the maximum length or contains invalid characters such as '**.**' or '**,**' for instance.

not a valid currency : ...

This error means you have sent a transaction with a currency code that is incorrect or does not exist.

The currency is not accepted by the merchant

This error means you have sent a transaction in a currency that has not been registered in your account details.

ERROR, PAYMENT METHOD NOT FOUND FOR: ...

This error means the PM value you sent in your hidden fields does not match any of the payment methods you have selected in your account, or that the payment method has not been activated in your payment methods page.

12 Appendix 3: Short Status Overview

The following section contains a non-exhaustive list of statuses; for a full list please refer to: http://www.ogone.com/ncol/paymentinfos1.asp.

Status	NCERROR	NCSTATUS	Explanation
5 Authorized	0	0	The authorization has been accepted. An authorization code is available in the field "ACCEPTANCE". The status will be 5 if you have defined "Authorisation" as default operation code in the "Global transaction parameters" tab, "Default operation code" section of the Technical Information page in your account.
9 Payment requested	0	0	The payment has been accepted. An authorization code is available in the field "ACCEPTANCE". The initial status of a transaction will be 9 if you have defined "Sale" as default operation code in the "Global transaction parameters" tab, "Default operation code" section of the Technical Information page in your account.
0 Invalid or incomplete	500....	5	At least one of the payment data fields is invalid or missing. The NCERROR and NCERRORPLUS fields give an explanation of the error (list available at http://www.ogone.com/ncol/paymentinfos1.asp).
2 Authorization refused	300....	3	The authorization has been declined by the financial institution. The customer can retry the authorization process after selecting another card or another payment method.
51 Authorization waiting	0	0	The authorization will be processed offline. This is the standard response if the merchant has chosen offline processing in his account configuration. The status will be 51 in two cases: • You have defined "Always offline (Scheduled)" as payment processing in the "Global transaction parameters" tab, "Processing for individual transactions" section of the Technical Information page in your account. • When the online acquiring system is unavailable and you have defined "Online but switch to offline in intervals when the online acquiring system is unavailable" as payment processing in the "Global transaction parameters" tab, "Processing

			for individual transactions" section of the Technical Information page in your account.
91 Payment processing	0	0	The data capture will be processed offline.
52 Authorization not known Or 92 Payment uncertain	200...	2	A technical problem arose during the authorization/payment process, giving an unpredictable result. The merchant can contact the acquirer helpdesk to know the exact status of the payment or can wait until we have updated the status in our system. The customer should not retry the authorization process since the authorization/payment might already have been accepted.
93 Payment refused	300....	3	A technical problem arose.

13 Appendix 4: Special format Travel

You can send additional data for travel transactions if your acquirer is able to receive and process the data.

The hidden fields for travel data are the following:

<input type="hidden" name="DataType" value="">
<input type="hidden" name="AIAIRNAME" value="">
<input type="hidden" name="AITINUM" value="">
<input type="hidden" name="AITIDATE" value="">
<input type="hidden" name="AICONJTI" value="">
<input type="hidden" name="AIPASNAME" value="">
<input type="hidden" name=" AIEXTRAPASNAME1" value="">
<input type="hidden" name="AICHDET" value="">
<input type="hidden" name="AIAIRTAX" value="">
<input type="hidden" name="AIVATAMNT" value="">
<input type="hidden" name="AIVATAPPL" value="">
<input type="hidden" name="AITYPCH" value="">
<input type="hidden" name="AIEYCD" value="">
<input type="hidden" name="AIIRST" value="">
<input type="hidden" name="AIORCITY1" value="">
<input type="hidden" name="AIORCITYL1" value="">
<input type="hidden" name="AIDESTCITY1" value="">
<input type="hidden" name="AIDESTCITYL1" value="">
<input type="hidden" name="AISTOPOV1" value="">
<input type="hidden" name="AICARRIER1" value="">
<input type="hidden" name="AIBOOKIND1" value="">
<input type="hidden" name="AIFLNUM1" value="">
<input type="hidden" name="AIFLDATE1" value="">
<input type="hidden" name="AICLASS1" value="">

IMPORTANT: The detailed specifications for each field, especially "mandatory/optional", are only mentioned for information purposes and may differ slightly from one acquirer to the other. Also, not all acquirers accept all fields.

Name	Usage		Field details
DataType	"TRAVEL"	mandatory	TRAVEL
AIAIRNAME	Airline name.	optional	max.20

AITINUM	Ticket number Air+ defines this zone as follows: 3 digits for airline prefix (filled with 0's if ticket type <> BSP + 10 chars for ticket number). Other acquirers do not split this zone – it is just the ticket number.	mandatory	max.16
AITIDATE	Ticket issue date. The default value is the transaction date.	optional	MM/DD/YYYY or YYYYMMDD
AICONJTI	Conjunction ticket.	optional	max.3
AIPASNAME	Primary passenger name. The default value is the name of the credit card holder.	optional	max.49
AIEXTRAPASNAME1	Name of extra passenger for PNRs with more than one passenger. This field can be repeated up to 5 times (i.e. for 5 extra passengers), changing the digit at the end of the field name.	optional	max.49
AICHDET	Charge details. Free text description or reference.	optional	max.49
AIAIRTAX	Airport taxes.	optional	num *100 => no decimals
AIVATAMNT	VAT amount.	optional	num *100 => no decimals
AIVATAPPL	VAT applicable flag. Supported values: D: normal VAT applicable I: no VAT on the transaction	optional	max.1
AITYPCH	Type of charge.	optional	max.2
AIEYCD	Destination area code.	optional	max.3
AIIRST	Destination area code type.	optional	max.1

The following fields can be repeated n times, changing the digit at the end of the field name.

Field	Usage		Field details
AIORCITY1	Departure airport (short).	mandatory	max. 5
AIORCITYL1	Departure airport (long).	mandatory	max. 20
AIDESTCITY1	Arrival airport (short).	mandatory	max. 5
AIDESTCITYL1	Arrival airport (long).	mandatory	max. 20
AISTOPOV1	Stopover.	optional	Possible values: the capital letters O and X. O: the passenger is allowed to stop and stay. X: the passenger is not allowed to stay.
AICARRIER1	Carrier code.	mandatory	max. 4
AIBOOKIND1	Booking indicator.	optional	max. 2
AIFLNUM1	Flight number.	optional	max. 4
AIFLDATE1	Flight date.	optional	MM/DD/YY or YYYYMMDD
AICLASS1	Airline class.	optional	max. 15

14 Appendix 5: e-Commerce via e-mail

You can send your customers a payment request by e-mail, redirecting the customer to our secure payment page via a button or link in the e-mail.

If the e-mail is in HTML format you can use a form with hidden HTML fields to send us the necessary parameters in POST format.

If the e-mail is in plain text format you can append the necessary parameters to the URL in GET format. (e.g. https://secure.ogone.com/ncol/test/orderstandard.asp?PSPID=TESTSTD&orderID=order123

&amount=12500&currency=EUR&SHASIGN=8DDF4795640EB9FE9B367315C48E47338129A4F5& …)

Please refer to Chapter 5 for more information.

IMPORTANT:

For e-Commerce via e-mail to work, you must bear in mind the following verification related points before the payment:

- You must leave the referrer/URL field in the URL field in the "Data and origin verification" tab, "Checks for e-Commerce" section of the Technical Information page in your account empty in order to avoid "unknown order/1/r" errors.
- You must use an SHA signature as the data verification method for the order details. For further details about the SHA-1-IN, please refer to Appendix 1.

15 Appendix 6: List of Parameters to be included in SHA IN Calculation

15.1 SHA-IN

ACCEPTURL	ECOM_SHIPTO_POSTAL_STREET_LINE1	PARAMPLUS
ADDMATCH	ECOM_SHIPTO_POSTAL_STREET_LINE2	PARAMVAR
ADDRMATCH	ECOM_SHIPTO_POSTAL_STREET_NUMBER	PAYID
ALIAS	ECOM_SHIPTO_TELECOM_FAX_NUMBER	PAYMETHOD
ALIASOPERATION	ECOM_SHIPTO_TELECOM_PHONE_NUMBER	PM
ALIASUSAGE	ECOM_SHIPTO_TVA	PMLIST
ALLOWCORRECTION	ED	PMLISTPMLISTTYPE
AMOUNT	EMAIL	PMLISTTYPE
AMOUNTHTVA	EXCEPTIONURL	PMLISTTYPEPMLIST
AMOUNTTVA	EXCLPMLIST	PMTYPE
BACKURL	FIRSTCALL	POPUP
BGCOLOR	FLAG3D	POST
BRAND	FONTTYPE	PSPID
BRANDVISUAL	FORCECODE1	PSWD
BUTTONBGCOLOR	FORCECODE2	REF
BUTTONTXTCOLOR	FORCECODEHASH	REF_CUSTOMERID
CANCELURL	FORCETP	REF_CUSTOMERREF
CARDNO	GENERIC_BL	REFER
CATALOGURL	GIROPAY_ACCOUNT_NUMBER	REFID
CERTID	GIROPAY_BLZ	REFKIND
CHECK_AAV	GIROPAY_OWNER_NAME	REMOTE_ADDR
CIVILITY	GLOBORDERID	REQGENFIELDS
CN	GUID	RTIMEOUT
COM	HDFONTTYPE	RTIMEOUTREQUESTEDTIMEOUT
COMPLUS	HDTBLBGCOLOR	SCORINGCLIENT
COSTCENTER	HDTBLTXTCOLOR	SETT_BATCH
CREDITCODE	HEIGHTFRAME	SID
CUID	HOMEURL	TAAL
CURRENCY	HTTP_ACCEPT	TBLBGCOLOR
CVC	HTTP_USER_AGENT	TBLTXTCOLOR
DATA	INCLUDE_BIN	TID
DATATYPE	INCLUDE_COUNTRIES	TITLE
DATEIN	INVDATE	TOTALAMOUNT
DATEOUT	INVDISCOUNT	TP
DECLINEURL	INVLEVEL	TRACK2
DISCOUNTRATE	INVORDERID	TXTBADDR2
ECI	ISSUERID	TXTCOLOR
ECOM_BILLTO_POSTAL_CITY	LANGUAGE	TXTOKEN
ECOM_BILLTO_POSTAL_COUNTRYCODE	LEVEL1AUTHCPC	TXTOKENTXTOKENPAYPAL
ECOM_BILLTO_POSTAL_NAME_FIRST	LIMITCLIENTSCRIPTUSAGE	TYPE_COUNTRY
ECOM_BILLTO_POSTAL_NAME_LAST	LINE_REF	UCAF_AUTHENTICATION_DATA
ECOM_BILLTO_POSTAL_POSTALCODE	LIST_BIN	UCAF_PAYMENT_CARD_CVC2
ECOM_BILLTO_POSTAL_STREET_LINE1	LIST_COUNTRIES	UCAF_PAYMENT_CARD_EXPDATE_MONTH
ECOM_BILLTO_POSTAL_STREET_LINE2	LOGO	UCAF_PAYMENT_CARD_EXPDATE_YEAR
ECOM_BILLTO_POSTAL_STREET_NUMBER	MERCHANTID	UCAF_PAYMENT_CARD_NUMBER
ECOM_CONSUMERID	MODE	USERID
ECOM_CONSUMERORDERID	MTIME	USERTYPE
ECOM_CONSUMERUSERALIAS	MVER	VERSION
ECOM_PAYMENT_CARD_EXPDATE_MONTH	OPERATION	WBTU_MSISDN
ECOM_PAYMENT_CARD_EXPDATE_YEAR	OR_INVORDERID	WBTU_ORDERID
ECOM_PAYMENT_CARD_NAME	OR_ORDERID	WEIGHTUNIT
ECOM_PAYMENT_CARD_VERIFICATION	ORDERID	WIN3DS
ECOM_SHIPTO_COMPANY	ORIG	WITHROOT
ECOM_SHIPTO_DOB	OWNERADDRESS	
ECOM_SHIPTO_ONLINE_EMAIL	OWNERADDRESS2	
ECOM_SHIPTO_POSTAL_CITY	OWNERCTY	
ECOM_SHIPTO_POSTAL_COUNTRYCODE	OWNERTELNO	
ECOM_SHIPTO_POSTAL_NAME_FIRST	OWNERTOWN	
ECOM_SHIPTO_POSTAL_NAME_LAST	OWNERZIP	
ECOM_SHIPTO_POSTAL_POSTALCODE	PAIDAMOUNT	

15.2 SHA-OUT

AAVADDRESS
AAVCHECK
AAVZIP
ACCEPTANCE
ALIAS
AMOUNT
BRAND
CARDNO
CCCTY
CN
COMPLUS
CURRENCY
CVCCHECK
DCC_COMMPERCENTAGE
DCC_CONVAMOUNT
DCC_CONVCCY
DCC_EXCHRATE
DCC_EXCHRATESOURCE
DCC_EXCHRATETS
DCC_INDICATOR
DCC_MARGINPERCENTAGE
DCC_VALIDHOUS
DIGESTCARDNO
ECI
ED
ENCCARDNO
IP
IPCTY
NBREMAILUSAGE
NBRIPUSAGE
NBRIPUSAGE_ALLTX
NBRUSAGE
NCERROR
ORDERID
PAYID
PM
SCO_CATEGORY
SCORING
STATUS
TRXDATE
VC