

# e-Commerce Advanced

Technical integration guide for Ogone e-Commerce v.5.4.4

## Table of Contents

1	Introduction .....	5
2	Best Practices .....	6
3	Test Environment .....	7
3.1	Configuring your test account.....	7
4	Sales Process.....	8
5	General Payment Parameters.....	10
5.1	Default operation code .....	10
5.2	Default data capture (payment) procedure.....	10
5.3	Processing for individual transactions.....	11
6	Link between the merchant's website and our payment page .....	13
6.1	Order form .....	13
6.1.1	Form fields .....	13
6.1.2	Form action .....	15
6.2	General parameters and optional customer details.....	15
6.2.1	Hidden fields .....	15
7	Security: Check before the Payment.....	17
7.1	Referrer .....	17
7.1.1	Configuration .....	17
7.1.2	Possible errors .....	17
7.1.3	Limitations .....	17
7.2	SHA-IN signature.....	17
7.3	IP address check.....	17
8	Look and feel of the payment page.....	18
8.1	Payment page layout (Static template).....	18
8.1.1	iPhone static template .....	19
8.2	Template-based page layout (Dynamic template).....	21
8.2.1	Hidden fields .....	21
8.2.2	Payment zone .....	22

---

8.2.3	Dynamic behaviour .....	22
8.2.4	Style sheets .....	22
8.2.5	Performance .....	23
8.3	Template security control.....	24
8.4	Secure environment padlock.....	24
8.5	Cancel button.....	25
8.6	Payment page in iframe.....	25
<b>9</b>	<b>Transaction feedback to the customer and the merchant .....</b>	<b>26</b>
9.1	Default reaction .....	26
9.1.1	Hidden fields .....	26
9.2	Redirection depending on the payment result .....	27
9.2.1	Hidden fields .....	27
9.2.2	Browser alert notification .....	28
9.2.3	Database update option .....	28
9.2.3.1	Feedback parameters.....	28
9.2.3.2	Dynamic post-sale.....	30
9.2.3.3	Security measures.....	30
9.2.3.4	Combination with a feedback request.....	30
9.3	Direct feedback requests (Post-payment) .....	31
9.3.1	Post-payment URLs and parameters .....	31
9.3.1.1	Post-payment URLs.....	31
9.3.1.2	Variable post-payment URLs .....	31
9.3.1.3	Feedback parameters.....	32
9.3.2	Timing of the feedback request .....	32
9.3.3	Response to the customer .....	33
9.4	Feedback reinitiation.....	33
9.5	Security: check origin of the request.....	34
9.5.1	IP address check (only for feedback requests) .....	34
9.5.2	SHA-OUT signature (for feedback requests and redirections) .....	34
9.6	Confirmation e-mails.....	34
9.6.1	E-mails to the merchant .....	34
9.6.2	E-mails to the customer .....	34
<b>10</b>	<b>Other optional hidden fields .....</b>	<b>36</b>
10.1	Payment method and payment page specifics .....	36
10.1.1	Payment method selection on the merchant's side .....	36
10.1.1.1	Showing a specific payment method.....	36

10.1.1.2	Allowing the customer to choose another payment method: backurl.....	36
10.1.2	Showing a specific list of payment methods .....	37
10.1.3	Excluding specific payment methods .....	38
10.1.4	Layout of the payment methods .....	38
10.1.5	3-D secure .....	38
10.1.6	Split credit/debit cards .....	39
10.2	Operation .....	39
10.3	User field .....	40
10.4	Delivery & Invoicing Data .....	41
10.5	Order Details.....	42
10.6	Direct Debits.....	43
10.6.1	Direct Debits DE (ELV) .....	43
10.6.2	Direct Debits NL .....	44
11	Appendix: SHA .....	45
11.1	SHA-IN signature.....	45
11.2	SHA-OUT signature.....	46
11.3	SHA module .....	47
12	Appendix: UTF-8 .....	48
13	Appendix: Troubleshooting.....	49
14	Appendix: Short Status Overview .....	50
15	Appendix: Special Travel Format.....	52
16	Appendix: e-Commerce via e-mail.....	55
17	Appendix: List of Parameters to be included in SHA .....	56
	Calculations	
17.1	SHA-IN .....	56
17.2	SHA-OUT .....	62

# 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 Best Practices

Ogone has defined a set of best practices to guarantee an optimal integration of your website with the payment platform and in order to ensure the smooth processing of your transactions.

Ogone recommends these best practices to all merchants, but especially larger merchants who envisage an average transaction volume of over 1000 transactions per day and/or merchants who expect transaction peaks of more than 25 simultaneous transactions per minute.

If these criteria apply to you, we would kindly invite you to contact us and ask our technical support engineers to analyse whether the best practices were applied correctly and to give you feedback on how to further improve your integration with the payment platform.

Additionally, by informing Ogone of the launch of your e-Commerce website and/or specific promotion campaigns that will affect the traffic on your web shop, we shall be able to provide extra monitoring to ensure that everything runs smoothly.

These best practices can be found at the beginning of each chapter, where relevant.

## 3 Test Environment

We recommend you develop your integration in our test environment before going live in the production environment. Our test environment works identically to our production environment, except that 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.

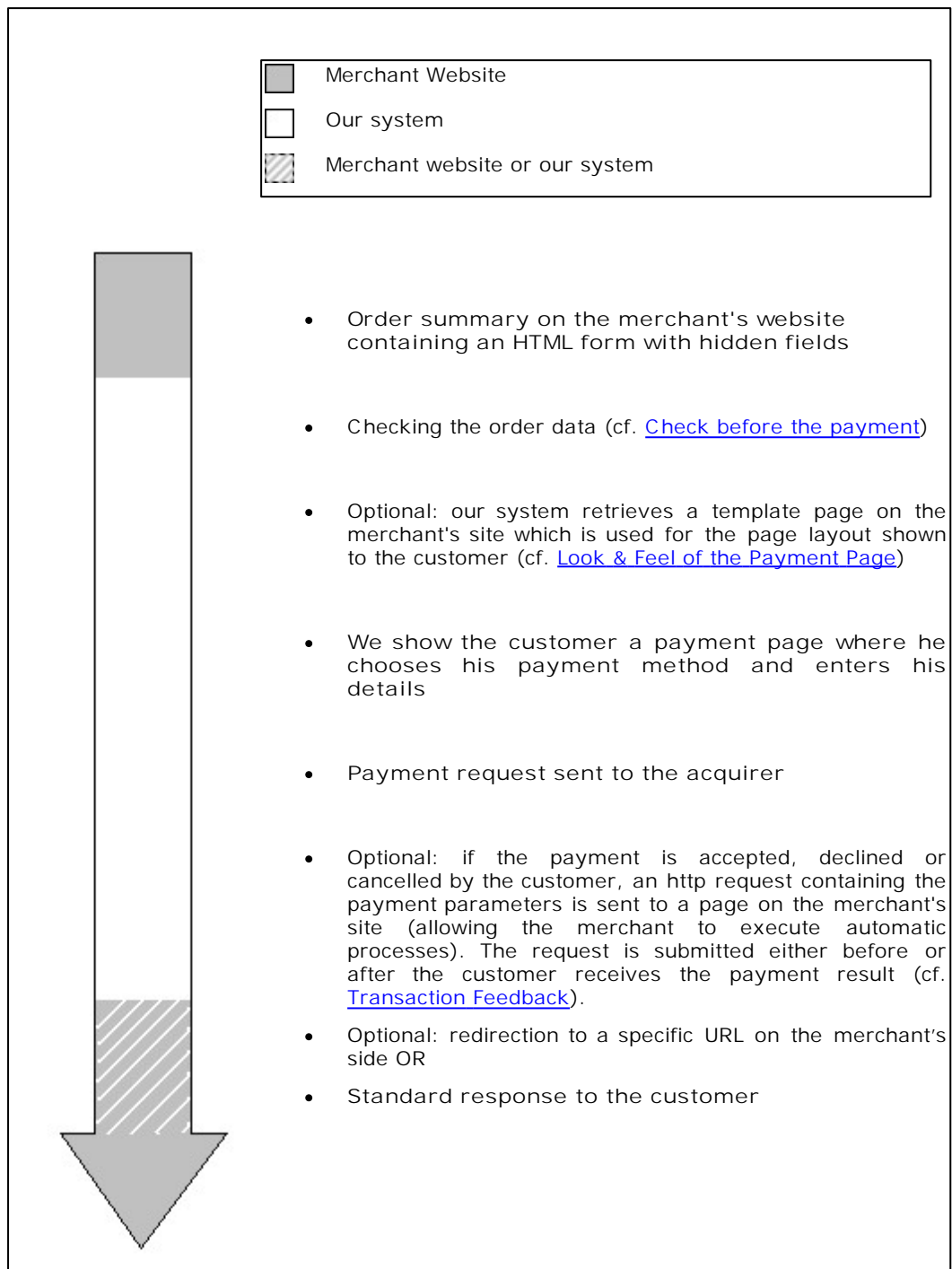
### 3.1 Configuring your test account

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 have to be configured on the Technical information page of your account. You can access the technical parameters via the link "Technical information" in your account menu.

## 4 Sales Process

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



The merchant can enhance his integration by securing the order data, personalising the payment pages, picking up feedback after a transaction and personalising the response to his customer.

This manual explains the advanced e-Commerce integration features, along with the optional steps to personalise the transaction flow and fine-tune the integration.



For a screenshot representation of a sales process following a basic e-Commerce integration, please refer to the Basic e-Commerce documentation.

## 5 General Payment Parameters

For some payment methods (mainly credit cards), transactions are performed in two stages: the authorisation and the data capture (payment request). (See [Default operation code](#) and [Default data capture \(payment\) procedure](#))

During the authorisation step, the transaction amount is either reserved on the customer's card or 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 from 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 [Processing for individual transactions](#))

The merchant can instruct our system to request the payment or authorisation 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 behaviour depends on the general parameters defined by the merchant in his back office, on the Technical information page > "Global transaction parameters" tab: 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.

### 5.1 Default operation code

#### Important

The ability to work in two stages (authorisation + data capture) depends on the payment methods you wish to use.

*More information about payment methods can be found in your Ogone account. Just log in and go to: Support > Integration & user manuals > Technical Guides > Payment Methods Processing/Procedures.*

Based on the two "Authorisation" and "Data capture" stages, the merchant can choose between two default operation codes in the "Global transaction parameters" tab, "Default operation code" section on the Technical information page:

- Authorisation

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

- Sale

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

### 5.2 Default data capture (payment) procedure

#### Important

The ability to work in two steps (authorisation + data capture) depends on the payment methods you wish to use.

*More information about payment methods can be found in your Ogone account. Just log in and go to: Support > Integration & user manuals > Technical Guides > Payment Methods Processing/Procedures.*

If the merchant has set "Authorisation" as the default operation code for his account or has included the "Authorisation" operation code in the transaction details, a data capture procedure will have to be performed to request the transaction 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 log in to his Ogone account 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 validity period an authorisation 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 frame (24 hours, 48 hours, etc.).

## 5.3 Processing for individual transactions

### Important

The ability to work online or offline depends on the payment methods you wish to use.

*More information about payment methods can be found in your Ogone account. Just log in and go to: Support > Integration & user manuals > Technical Guides > Payment Methods Processing/Procedures.*

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 authorise offline processing in these 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 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, as 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 your account, on the Technical information page, "Transaction feedback" tab, in the "HTTP request for status changes" section (for HTTP requests) or in the "Transaction e-mails" tab, in the "E-mails to the merchant" section (for e-mails). In this way, you can be notified by e-mail and/or http request when the status of a transaction changes to offline in our system.

## 6 Link between the merchant's website and our payment page

### 6.1 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 this last page. The form action URL will be our (e-Commerce system's) payment processing page.

#### 6.1.1 Form fields

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

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

<!-- general parameters: see General Payment Parameters -->
<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 General
parameters and optional customer details -->
<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="OWNERCITY" value="">
<input type="hidden" name="OWNERTOWN" value="">
<input type="hidden" name="OWNERTELNO" value="">
<input type="hidden" name="COM" value="">

<!-- check before the payment: see SHA-1N signature -->
<input type="hidden" name="SHASIGN" value="">

<!-- layout information: see Look & Feel of the Payment Page -->
<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="BUTTONXTCOLOR" value="">
<input type="hidden" name="LOGO" value="">
<input type="hidden" name="FONTTYPE" value="">
<!-- dynamic template page: see Look & Feel of the Payment Page -->
<input type="hidden" name="TP" value="">
<!-- payment methods/page specifics: see Payment method and payment page specifics -->
<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 Default reaction -->
<input type="hidden" name="HOMEURL" value="">
<input type="hidden" name="CATALOGURL" value="">
<!-- post payment parameters: see Redirection depending on the payment result -->
<input type="hidden" name="COMPLUS" value="">
<input type="hidden" name="PARAMPLUS" value="">
<!-- post payment parameters: see Direct feedback requests \(Post-payment\) -->
<input type="hidden" name="PARAMVAR" value="">
<!-- post payment redirection: see Redirection depending on the payment result -->
<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 Operation -->
<input type="hidden" name="OPERATION" value="">
<!-- optional extra login detail field: see User field -->
<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>

```

You can find an example (test page) illustrating the last page of a merchant's shopping basket 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.

### 6.1.2 Form action

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

In the production (PROD) environment, the URL for the action will be <https://secure.ogone.com/ncol/prod/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 thus will not be sent to the acquirers/banks.

## 6.2 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 (CN), customer's e-mail (EMAIL), address (OWNERADDRESS), town (OWNERTOWN), postcode (OWNERZIP), country (OWNERCTY) and telephone number (OWNERTELNO), as 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 analysed in your Ogone account when you look up the transaction details.

### 6.2.1 Hidden fields

The following hidden fields are 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

Field	Usage
	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. It will be pre-initialised (but still editable) in the cardholder name field of the credit card details.
EMAIL	Customer's email address
OWNERADDRESS	Customer's street name and number
OWNERZIP	Customer's postcode
OWNERTOWN	Customer's town/city name
OWNERCTY	Customer's country
OWNERTELNO	Customer's telephone number
COM	Order description

*More information about these fields can be found in your Ogone account. Just log in and go to: Support > Integration & user manuals > Technical guides > Parameter Cookbook.*



## 7 Security: Check before the Payment

Best practice: [SHA signature](#)

### 7.1 Referrer

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

#### 7.1.1 Configuration

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

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 separated by a semicolon with no spaces before or after the semicolon.

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

#### 7.1.2 Possible errors

Possible errors related to the referrer are *"unknown order/1/r"* and *"unknown order/0/r"*. Please refer to [Troubleshooting](#) for more information about these errors.

#### 7.1.3 Limitations

Although the referrer allows our system to identify the origin of an order, it does not guarantee the integrity of the data.

Therefore, our system requires the use of an SHA signature.

### 7.2 SHA-IN signature

This technique is based on the principle of the merchant's server generating a unique character string, hashed with the SHA 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: SHA](#).

### 7.3 IP address check

The IP address field in the Technical information page, "Data and origin verification" tab, "Checks for DirectLink and automatic Batch" section 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).

## 8 Look and feel of the payment page

Best practice: [Static template](#)

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 (e.g. the merchant's logo) and payment detail information (e.g. order reference, fields where the customer enters his card details, etc.).

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 and feel of these payment details may however be adapted by the merchant using html styles.

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

### 8.1 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.

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 colour	white
TXTCOLOR	Text colour	black
TBLBGCOLOR	Table background colour	white
TBLTXTCOLOR	Table text colour	black
BUTTONBGCOLOR	Button background colour	—
BUTTONTXTCOLOR	Button text colour	black
FONTTYPE	Font family	Verdana

Field	Usage	Default value
LOGO	<p>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 (i.e. contain the full path), it cannot be relative.</p> <p>The logo needs to be stored on a secure server (see <a href="#">Secure environment padlock</a>). If you do not have a secure environment to store your image, you can send a JPG, PNG or GIF file (and your PSPID) to support@ogone.com (only for production accounts, as this is a payment option (!) Please activate the "Logo Hosting" option in your Account &gt; Your options page before sending us your logo).</p> <p>If the logo is stored on our servers, the URL will be: <code>https://secure.ogone.com/images/merchant/[PSPID]/[image]</code></p>	—

*More information about these fields can be found in your Ogone account. Just log in and go to: Support > Integration & user manuals > Technical guides > Parameter Cookbook.*

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

### 8.1.1 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 cannot 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 customised 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="TBLBGOLOR" value="">
```

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

```
<input type="hidden" name="HDTBLBGOLOR" value="">
```

```
<input type="hidden" name="HDTBLTXTCOLOR" value="">
```

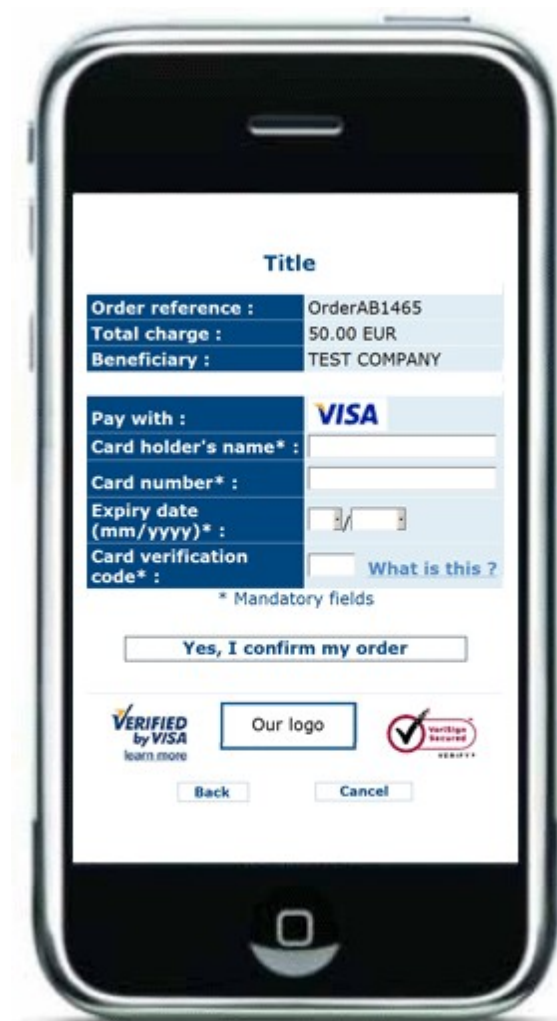
```

<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 colour	#FFFFFF
TXTCOLOR	Text colour	#00467F
TBLBGCOLOR	Background colour for the right columns	#E1EDF4
TBLTXTCOLOR	Text colour for the right columns	#000000
HDTBLBGCOLOR	Background colour for the left columns	#00467F
HDTBLTXTCOLOR	Text colour for the left columns	#FFFFFF
HDFONTTYPE	Font family for the left columns	Verdana
BUTTONBGCOLOR	Button background colour	#FFFFFF
BUTTONTXTCOLOR	Button text colour	#00467F
FONTTYPE	Font family	Verdana

*More information about these fields can be found in your Ogone account. Just log in and go to: Support > Integration & user manuals > Technical guides > Parameter Cookbook.*



## 8.2 Template-based page layout (Dynamic template)

The dynamic template page is an advanced technique for customising 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.

### 8.2.1 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

Field	Usage
	component included in the template page must also have an absolute URL.

*More information about these fields can be found in your Ogone account. Just log in and go to: Support > Integration & user manuals > Technical guides > Parameter Cookbook.*

### 8.2.2 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](https://secure.ogone.com/ncol/template_standard.htm)

### 8.2.3 Dynamic behaviour

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 specific page for 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=...

### 8.2.4 Style sheets

You can personalise 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. You will need to 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 (cf. the example of the aforementioned 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.ncoltxttr {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.

The first two screenshots show a payment form for VISA. The form includes fields for Card holder's name, Card number, Expiry date, and Card verification code. It also features a 'CVC present' dropdown and a 'What is this ?' link. A confirmation button 'Yes, I confirm my order' is at the bottom. The third screenshot shows a confirmation message: 'Your payment is authorised' with a 'Payment reference :1248886'.

### 8.2.5 Performance

Our system is configured with a 5-second timeout for the request to retrieve the merchant's dynamic template page.

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

If a timeout occurs, our system will use the merchant's static template instead.

If no static template is configured, our system will use the Ogone static template as a last resort.

#### Important

This HTTPTimeout field has an impact on both dynamic template requests and post-payment feedback requests (see [Direct feedback requests \(Post-payment\)](#)). 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 Team for a solution if you have high transaction volumes.

## 8.3 Template security control

To protect the merchant's customers against fraudulent activities, such as the manipulation of sensitive card data (card number, CVC), different security checks for the merchant template were made available.

In the merchant's Technical information page, "Global security parameters" tab, "Template" section, the following settings can be configured:

- **Enable Javascript check on template**  
The merchant can enable this feature to detect Javascript usage on the template page. If Javascript is detected, the template will be blocked and the default template will be used instead.
- **Allow usage of static template**  
If the merchant selects *Allow usage of static template*, it is mandatory to enter one or more values in the *Trusted static template name*. These values will then be used as input for a check that will compare it with the information received by Ogone during the payment process.
- **Allow usage of dynamic template**  
If the merchant selects *Allow usage of dynamic template*, it is mandatory to configure the *Trusted website host name hosting the dynamic template* field. This field can contain multiple web hosts, separated by semicolons, but they should all contain the full URL, e.g. `http://www.website.com/`. The sub-directories can be left out, so if the dynamic template is `http://www.website.com/templates/nl/template1.htm`, it suffices to configure `http://www.website.com` as trusted web host.

Additionally the merchant can also configure one or more fully trusted dynamic template urls, separated by semicolons, in the *Trusted dynamic template url* field.

If a dynamic template is submitted with a transaction, but dynamic templates are not allowed, then the template will be blocked and our system will use the static template instead.

If there is no static template configured, or if the static template is also not allowed, then the default Ogone template will be used.

#### Important

If a default static or dynamic template is configured in the merchant's account (requested beforehand to our Customer Care), then one of the 2 options (*Allow usage of static template* / *Allow usage of dynamic template*) must be enabled. The template URL should also be configured as a *trusted template*. If the *Trusted static template name* and *Trusted dynamic template url* input fields are left blank, by default all templates are trusted.

By default, *Enable JavaScript check on template* and *Allow usage of static template* are enabled. The *Trusted static template name* will be pre-configured with the merchant website host name.

## 8.4 Secure environment padlock

The URL used to connect the customer to our platform uses a secure protocol (https). 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 recognise a secure connection unless all the elements on the screen, including images, sounds, etc. come from secure sites.

For merchants who 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). We offer hosting for these elements (see the image hosting options in your account).

## 8.5 Cancel button

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

## 8.6 Payment page in iframe

Iframes allow the merchant to integrate an external web page (such as the payment page) on his website, while maintaining his own URL in the browser.

However, in the current context iframe also has very significant drawbacks:

- Since the URL is the merchant's URL, it could be a simple http (instead of an https) and the padlock icon may not appear in the browser. This could cause customers to doubt the security of the webshop.
- Some payment methods (like Giropay, Sofort, Bancontact/Mister Cash, PayPal, etc.) use redirections, which may give poor layout results and/or navigation misbehaviour.

As a result, Ogone discourages the use of iframe, and the usage thereof is the merchant's responsibility. Ogone strongly encourages the use of a Dynamic Template instead.

If you still wish to pursue the integration of iframe, we strongly recommend the following:

- Use iframe only on the payment method selection page (and beyond)
- Use pop-ups for external payment methods whenever possible, to ensure the visibility of third-party web applications.

## 9 Transaction feedback to the customer and the merchant

Best practice: Redirection with parameters on the accept-/exception-/cancel-/declineurl (cf. [Database update option](#)) with a deferred post-payment feedback request as a backup (cf. [Combination with a feedback request](#)) and a check of the SHA-OUT signature by the merchant (cf. [SHA-OUT signature \(for feedback requests and redirections\)](#))

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

### 9.1 Default reaction

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

HTTPS://OUR URL/order\_Agree.asp

**My webshop**

**Order reference : ORDER123**  
**Total charge : 49.00 EUR**  
**Beneficiary : Webshop**

**Authorised**

**Payment reference :15987181**

[About](#) | [Privacy policy](#) | [Security](#) | [Legal info](#)

[Back to merchant site](#)

In this page, we also add a link to the merchant's website and/or the merchant's catalogue, 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 URLs configured in your Ogone account's administrative details.

#### 9.1.1 Hidden fields

The following hidden fields are 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,

Field	Usage
	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.

*More information about these fields can be found in your Ogone account. Just log in and go to: Support > Integration & user manuals > Technical guides > Parameter Cookbook.*

## 9.2 Redirection depending on the payment result

In the hidden fields of the order form, the merchant can send four URLs (ACCEPTURL, EXCEPTIONURL, CANCELURL and DECLINEURL) to which our system will redirect 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 personalise the customer's response:*

HTTPS://OUR URL/order\_Agree.asp

**My webshop**

Order reference : STDREF451  
Total charge : 1.00 EUR  
Beneficiary : Webshop

**Your payment is authorised**

Payment reference :15988189

You will now be redirected to the merchant's website. A warning message can be displayed because you are about to leave the secured environment.

OK

YOUR URL

**Thank you for your purchase!**

Click here to go back to the shop

### 9.2.1 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 authorised (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 authorisation (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 see 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 customer will see the DECLINEURL instead.

*More information about these fields can be found in your Ogone account. Just log in and go to: Support > Integration & user manuals > Technical guides > Parameter Cookbook.*

## 9.2.2 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 is moving from an https:// environment to an http:// environment). When we detect a redirection to the merchant's website, we can display a message to notify the customer about the possibility of a risk (see first screenshot in Chapter [here](#)), thereby avoiding any undue concern about a 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 Ogone 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").

## 9.2.3 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").

### 9.2.3.1 Feedback parameters

When a payment is executed, we can send the following parameter list to the merchant's ACCEPTURL, EXCEPTIONURL, CANCELURL or DECLINEURL.

Parameter	Value
ORDERID	Your order reference

Parameter	Value
AMOUNT	Order amount (not multiplied by 100)
CURRENCY	Order currency
PM	Payment method
ACCEPTANCE	Acceptance code returned by acquirer
STATUS	Transaction status (see <a href="#">Appendix: Status overview</a> )
CARDNO	Masked card number
PAYID	Payment reference in our system
NCERROR	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-OUT configured)

*More information about these fields can be found in your Ogone account. Just log in and go to: Support > Integration & user manuals > Technical guides > 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 to be returned in the feedback request.
PARAMPLUS	Field for submitting some parameters and their values you would like to be 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.

*More information about these fields can be found in your Ogone account. Just log in and go to: Support > Integration & user manuals > Technical guides > Parameter Cookbook.*

#### *Example*

The following are the extra hidden fields sent by the merchant:

```
<input type="hidden" name="COMPLUS" value="123456789123456789123456789">  
<input type="hidden" name="PARAMPLUS"  
value="SessionID=126548354&ShopperID=73541312">
```

resulting in a redirection with the feedback parameters:

```
https://www.yourwebsite.com/acceptpage.asp?[...standard.parameters...]  
&COMPLUS=123456789123456789123456789&SessionID=126548354&ShopperID=73541312
```

#### 9.2.3.2 Dynamic post-sale

You may also choose which parameters are sent with the post-sale request. To do this, go to the "Transaction feedback" tab of your Technical information page, where you will see a list of "Available" and "Selected" fields. Only the "Selected" fields will be part of the post-sale request.

To add or remove parameters from the post-sale request, click on the parameter name and click the appropriate arrow to add/remove it from the list.

#### **Important**

If you add/remove parameters from this list, do not forget to update your SHA-OUT signature accordingly. Parameters that are not selected here will NOT be contained in the SHA-OUT calculation

#### 9.2.3.3 Security measures

The redirection process is visible because it is sent via the customer's browser. Consequently, the merchant must use an SHA signature (cf. [Appendix: SHA](#)) 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-OUT signature, we shall not send any parameters on his ACCEPT-, EXCEPTION-, CANCEL- or DECLINEURL.

#### 9.2.3.4 Combination with a feedback request

The merchant can use a deferred/background feedback request as a fall-back option for the redirection (see [Direct Feedback Request](#)).

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.

## 9.3 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, called a "post-sale request", 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).

### 9.3.1 Post-payment URLs and parameters

#### 9.3.1.1 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 to which the request parameters are sent if the payment's status is accepted, pending or uncertain. The other can be the URL to which the request parameters are sent when the transaction has been cancelled by the client or 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 the timing for 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.

#### 9.3.1.2 Variable post-payment URLs

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

The following hidden field should be used:

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

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

*More information about these fields can be found in your Ogone account. Just log in and go to: Support > Integration & user manuals > Technical guides > 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>

#### Important

Please do not use any special characters in the PARAMVAR field, as they will be URL encoded, and this could create invalid links

### 9.3.1.3 Feedback parameters

Our http request to your post-payment URL will contain the same feedback parameters as described in Chapter "[Feedback Parameters](#)"

## 9.3.2 Timing of the feedback request

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

- No request

In this case, our system will not send any feedback request. This option allows you to disable your post-payment URLs in the event 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 personalised feedback to the customer on the merchant's website.

If the merchant does not use his post-payment page to personalise 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 personalised 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 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 four times (deferred). In this way, the merchant does not miss out on the transaction feedback, should the online post-payment feedback request fail, e.g. as a result of temporary server problems at his end. The customer will be displayed the standard transaction feedback from our system (see [Default reaction](#)).



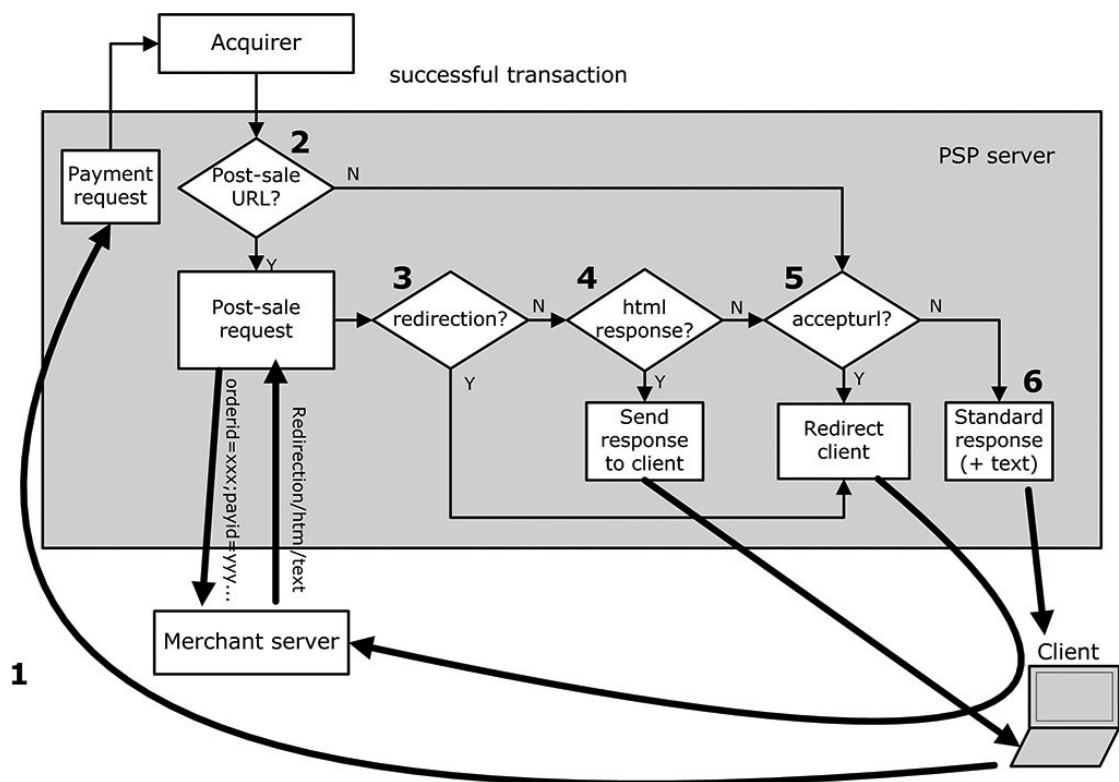
### 9.3.3 Response to the customer

We use a possible reply from 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 four URLs you may have sent in the hidden fields (ACCEPTURL, EXCEPTIONURL, CANCELURL and DECLINEURL as described here: [Redirection depending on the payment result](#)).

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 simply show the standard response (as described in: [Default reaction](#)).

The diagram below shows the process at the end of a transaction, should the payment be authorised 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).



## 9.4 Feedback reinitiation

In the event that a redirection/feedback request was not performed due to a blocking action by the customer on our secure payment pages (e.g. clicking the back button in the browser), we can reinitiate the post-payment request and/or redirection so your customer will be redirected to the page you want to display and your databases can be updated.

To enable this feature in your account, go to Configuration > Technical information > Transaction feedback > General, and tick the box "I would like Ogone to re-launch the "end of transaction" (post-payment request/redirection) process if required."

It is possible, however, that you will receive multiple post-payment requests for the same Order ID, since the redirection/feedback request will be resubmitted if the customer returns to our secure payment pages using the back button after having been redirected to your website.

Please ensure you configure your Post URL script to handle these "exceptions". For example, you

could configure your Post URL script to create a row in your database for each transaction status posted back, and/or generate an email to inform the merchant of an "exception" to the "expected" steps in the transaction process.

It is recommended that you do not overwrite the first transaction status message you receive with any subsequent messages you receive for the same Order ID - ideally you would store all responses for any order, and invoke a process whereby these can be investigated and dealt with appropriately.

If you do not select the box, when the customer clicks the back button to return to the secure payment pages, he will see a message indicating that the payment has already been processed.

## 9.5 Security: check origin of the request

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

### 9.5.1 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 our Firewall Configuration for the Transaction Platform Traffic configuration guide, available on the "Integration & user manuals" page in your account. Please note that different ranges of possible IP addresses exist and that these IP addresses are subject to change!

### 9.5.2 SHA-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-OUT signature, please refer to [Appendix: SHA](#).

## 9.6 Confirmation e-mails

### 9.6.1 E-mails to the merchant

Our system can send you a payment confirmation e-mail for each transaction (Technical information > "Transaction e-mails" tab, "E-mails to the merchant" section). You can enter several e-mail addresses, separated by a semicolon ";".

On the configuration screen, you may also choose to receive e-mails to be notified of transaction status changes.

### 9.6.2 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 sender ("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, the first one in the row will be used.

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

You can also choose to send e-mails to the customer when the transaction is confirmed (data capture) and when a transaction is refunded, by ticking the corresponding boxes. As the sender ("From") e-mail address for these e-mails, you can configure the "Support E-mail address to

include in transaction-related e-mails". If you don't enter an e-mail address here, we will use the first one entered in the "Support E-mail address to include in transaction-related e-mails" in the "E-mails to the merchant" section.

To be able to send confirmation e-mails to your customers, you must include the customer's e-mail address in the hidden field:

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

Field	Usage
EMAIL	Customer's e-mail address

*More information about these fields can be found in your Ogone account. Just log in and go to: Support > Integration & user manuals > Technical guides > Parameter Cookbook.*

## 10 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.

### 10.1 Payment method and payment page specifics

#### 10.1.1 Payment method selection on the merchant's side

##### 10.1.1.1 Showing a specific payment method

When a customer visits our secure payment page, he will be shown a list of the payment methods the merchant has activated in his account. If the customer has 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

*More information about these fields can be found in your Ogone account. Just log in and go to: Support > Integration & user manuals > Technical guides > Parameter Cookbook.*

##### Examples

- Hidden fields in the event that 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 the event that you only want your customer to pay by credit card (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 the event that your customer has selected iDEAL on your site:

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

##### 10.1.1.2 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, as 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 authorisation has been declined or after having cancelled from a third-party or bank website, we redirect him to the URL entered by the merchant 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" specified in the "Payment page layout" tab of the "Technical information" page in your account. However if you prefer not to use the same one, you can also send us a specific "BACKURL" in the hidden transaction fields.

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.

*More information about these fields can be found in your Ogone account. Just log in and go to: Support > Integration & user manuals > Technical guides > 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 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.

## 10.1.2 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 ";" (semicolon).

*More information about these fields can be found in your Ogone account. Just log in and go to: Support > Integration & user manuals > Technical guides > Parameter Cookbook.*

#### Example

\* The hidden field in the event that you only want your customer to choose between VISA and iDEAL on our payment page (i.e. if you also have other payment methods that you don't want to be displayed) will be:

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

### 10.1.3 Excluding specific payment methods

If the merchant does not wish to present a specific brand to the cardholder, he can use a hidden field to do so.

This is particularly useful for sub-brands, when a merchant wants to accept a brand (e.g. MasterCard) but not one of its sub-brands (e.g. Maestro)

The hidden field is the following:

```
<input type="hidden" name="EXCLPMLIST" value="">
```

Field	Usage
EXCLPMLIST	List of payment methods and/or credit card brands that should NOT be shown. Separated by a ";" (semicolon).

*More information about these fields can be found in your Ogone account. Just log in and go to: Support > Integration & user manuals > Technical guides > Parameter Cookbook.*

If a customer tries paying with a card linked to a payment method and/or (sub)brand you've excluded using the EXCLPMLIST parameter, the error message "Card number incorrect or incompatible" will be returned with the NCERRORPLUS return field. The customer will see a similar message on the payment page.

### 10.1.4 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	<p>The possible values are 0, 1 and 2:</p> <p>0: Horizontally grouped logos with the group name on the left (default value)</p> <p>1: Horizontally grouped logos with no group names</p> <p>2: Vertical list of logos with specific payment method or brand name</p>

*More information about these fields can be found in your Ogone account. Just log in and go to: Support > Integration & user manuals > Technical guides > Parameter Cookbook.*

### 10.1.5 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.

The hidden field is the following:

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

Field	Possible values
WIN3DS	<p>"MAINW": to display the identification page in the main window (default value)</p> <p>"POPUP": to display the identification page in a POPUP window and return to main window at the end</p>

*More information about these fields can be found in your Ogone account. Just log in and go to: Support > Integration & user manuals > Technical guides > Parameter Cookbook.*

#### Important

Please note that for some payment methods (e.g. Visa, MasterCard, JCB, etc.), the 'POPUP' value is not allowed and will be converted into 'MAINW' by the system. We recommend explicitly testing the behaviour of this field for every payment method.

### 10.1.6 Split credit/debit cards

The functionality to split VISA and MasterCard into a debit and a credit payment method allows you to offer them to your customers as two different payment methods (e.g. VISA Debit and VISA Credit), or you can decide only to accept one of both split brands.

To use the split of credit and debit cards via e-Commerce, you need to include the CREDITDEBIT parameter in the hidden fields that you send to the payment page.

Parameter	Format
CREDITDEBIT	<p>"C": credit card</p> <p>"D": debit card</p>

*This field has to be included in the SHA-IN calculation*

Related error: When the buyer selects the debit card method but next enters a credit card number, an error code will be returned: 'Wrong brand/Payment method was chosen'.

If the payment is successfully processed with the CREDITDEBIT parameter, the same parameter will also be returned in the post-sale feedback. However, whereas the submitted values are C or D, the return values are "CREDIT" or "DEBIT".

You will also find these return values in transaction overview via "View transactions" and "Financial history", and in reports you may download afterwards.

#### Configuration in your account

The split functionality can be activated and configured per payment method, in your Ogone account. Check our [Split Credit/Debit Cards guide](#) for more information.

## 10.2 Operation

#### Important

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

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

The operation code you send us in the hidden fields will override the default operation code

selected in the "Default operation code" section of the "Global transaction parameters" tab, on the "Technical information" page in your account. You can send the operation code with the following hidden field:

```
<input type="hidden" name="OPERATION" value="">
```

Field	Usage
OPERATION	<p>Operation code for the transaction.</p> <p>Possible values for <u>new orders</u>:</p> <ul style="list-style-type: none"> <li>▪ RES: request for authorisation</li> <li>▪ SAL: request for sale (payment)</li> </ul> <p>Optional:</p> <ul style="list-style-type: none"> <li>▪ PAU: Request for pre-authorisation</li> </ul> <p>In agreement with your acquirer you can use this operation code to temporarily reserve funds on a customer's card. This is a common practice in the travel and rental industry.</p> <p>PAU/pre-authorisation can currently only be used on MasterCard transactions and is supported by selected acquirers. This operation code cannot be set as the default in your Ogone account.</p> <p>Should you use PAU on transactions via acquirers or with card brands that don't support pre-authorisation, these transactions will not be blocked but processed as normal (RES) authorisations.</p>

*More information about these fields can be found in your Ogone account. Just log in and go to: Support > Integration & user manuals > Technical guides > 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: SHA](#) for more information on SHA.

## 10.3 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 centre 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

*More information about these fields can be found in your Ogone account. Just log in and go to: Support > Integration & user manuals > Technical guides > 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.*



## 10.4 Delivery & Invoicing Data

Some payment methods may require you to submit delivery information. You may do so by using the following fields:

Format: AN= Alphanumeric / N=Numeric, maximum allowed amount of characters

Field	Format	Usage
ORDERSHIPMETH	AN (25)	Delivery method
ORDERSHIPCOST	N	Delivery cost
ORDERSHIPTAXCODE	N	Delivery tax code
CUID	AN (50)	Social security number / company registration number
CIVILITY	AN (10)	Invoicing title (Mr., Mrs, Dr., etc.)
ECOM_BILLTO_POSTAL_NAME_FIRST	AN (35)	Invoicing first name
ECOM_BILLTO_POSTAL_NAME_LAST	AN (35)	Invoicing last name
ECOM_BILLTO_POSTAL_STREET_LINE1	AN (35)	Invoicing address
ECOM_BILLTO_POSTAL_STREET_NUMBER	AN (10)	Invoicing street number
ECOM_BILLTO_POSTAL_POSTALCODE	AN (10)	Invoicing postcode
ECOM_BILLTO_POSTAL_CITY	AN (40)	Invoicing city
ECOM_BILLTO_POSTAL_COUNTRYCODE	AN (2)	Invoicing country code (BE, FR, NL, DE, etc.)
ECOM_SHIPTO_POSTAL_NAME_PREFIX	AN (10)	Delivery civil status (Mr., Mrs, etc.)
ECOM_SHIPTO_POSTAL_NAME_FIRST	AN (35)	Delivery first name
ECOM_SHIPTO_POSTAL_NAME_LAST	AN (35)	Delivery last name
ECOM_SHIPTO_POSTAL_STATE	AN (2)	Delivery state (ISO code *)
ECOM_SHIPTO_POSTAL_STREET_LINE1	AN (35)	Delivery address
ECOM_SHIPTO_POSTAL_STREET_NUMBER	AN (10)	Delivery street number
ECOM_SHIPTO_POSTAL_POSTALCODE	AN (10)	Delivery postcode
ECOM_SHIPTO_POSTAL_CITY	AN (25)	Delivery city
ECOM_SHIPTO_POSTAL_COUNTRYCODE	AN (2)	Delivery country code
ECOM_SHIPTO_ONLINE_EMAIL	AN (50)	Delivery email address
ECOM_SHIPTO_DOB	yyyy-MM-dd	Date of birth

(\*ISO codes for US States can be found [here](#). Example: AL (Alabama), FL (Florida))

More information about these fields can be found in your Ogone account. Just log in and go to: Support > Integration & user manuals > Technical guides > Parameter Cookbook.

## 10.5 Order Details

Some payment methods may require you to submit detailed order information. You may do so using the following fields:

Format: AN= Alphanumeric / N=Numeric, maximum allowed amount of characters

Field	Description	Format
ITEMATTRIBUTES*xx*	Item attributes (size, colour etc.) (replace *xx* with a number to send multiple items: ITEMATTRIBUTES1, ITEMATTRIBUTES2, etc.)	AN, 50
ITEMCATEGORY*xx*	Item category (replace *xx* with a number to send multiple items: ITEMCATEGORY1, ITEMCATEGORY2, etc.)	AN, 50
ITEMCOMMENTS*xx*	Item comments (replace *xx* with a number to send multiple items: ITEMCOMMENTS1, ITEMCOMMENTS2, etc.)	AN, 255
ITEMDESC*xx*	Item description (replace *xx* with a number to send multiple items: ITEMDESC1, ITEMDESC2, etc.)	AN, 16
ITEMDISCOUNT*xx*	Item discount (replace *xx* with a number to send multiple items: ITEMDISCOUNT1, ITEMDISCOUNT2, etc.)	N, 10
ITEMID*xx*	Item identification (replace *xx* with a number to send multiple items: ITEMID1, ITEMID2, etc.)	AN, 15
ITEMNAME*xx*	Item name (replace *xx* with a number to send multiple items: ITEMNAME1, ITEMNAME2, etc.)	AN, 50
ITEMPRICE*xx*	Item price (replace *xx* with a number to send multiple items: ITEMPRICE1, ITEMPRICE2, etc.)	N, 50
ITEMQUANT*xx*	Item quantity (replace *xx* with a number to send multiple items: ITEMQUANT1, ITEMQUANT2, etc.)	N, 50
ITEMQUANTORIG*xx*	Maximum quantity per item (replace *xx* with a number to send multiple items: ITEMQUANTORIG1, ITEMQUANTORIG2, etc.)	AN, 50
ITEMUNITOFMEASURE*xx*	Item Unit of measure (replace *xx* with a number to send multiple items: ITEMUNITOFMEASURE1, ITEMUNITOFMEASURE2, etc.)	AN, 50
ITEMVAT*xx*	Item VAT amount (replace *xx* with a number to send multiple items: ITEMVAT1, ITEMVAT2, etc.)	N, 50
ITEMVATCODE*xx*	Item VAT code (replace *xx* with a number to send multiple items: ITEMVATCODE1, ITEMVATCODE2, etc.)	N, 50
ITEMWEIGHT*xx*	Item weight (replace *xx* with a number to send multiple items: ITEMWEIGHT1, ITEMWEIGHT2, etc.)	N, 10

*More information about these fields can be found in your Ogone account. Just log in and go to: Support > Integration & user manuals > Technical guides > Parameter Cookbook.*

Important: ascending order of line item fields for SHA calculation

Please take into account that as from 10 consecutive line item fields the order of the fields for the

SHA calculation may differ from what is expected on your system's side.

Where for the calculation of the SHA we normally stress that all fields should be sorted alphabetically, a "logical" ascending order is put in place for the line item fields.

Example of expected (ascending) order:

- ITEMNAME1, ITEMNAME2, ITEMNAME3, ITEMNAME4, ITEMNAME5, ITEMNAME6, ITEMNAME7, etc.

## 10.6 Direct Debits

### 10.6.1 Direct Debits DE (ELV)

The following fields can be sent along for Direct Debits DE (ELV) transactions. They will be displayed to the customer, in the Mandate text on the payment page.

Field	Description
CN	Cardholder name
Optional for SEPA (*) transactions (not applicable for Billpay and Wirecard):	
MANDATEID	Unique Mandate Reference Max. 35 AN characters, charset: "A-Z a-z 0-9 space /-?:()., '+'")

*More information about these fields can be found in your Ogone account. Just log in and go to: Support > Integration & user manuals > Technical guides > Parameter Cookbook.*

Note: These fields can be returned in the post-payment feedback (Dynamic e-commerce parameters) and need to be included in the SHA-IN (and optionally SHA-OUT) calculation.

#### Notes for SEPA (\*)

- If you don't provide the CN, this field will remain empty on the payment page.
- If you don't provide a MANDATEID, we will use the ORDERID. However, if the ORDERID is not compliant with the specifications of the MANDATEID, the PAYID will be used instead.
- If you haven't switched to the SEPA mode in the configuration page of Direct Debits DE / ELV, your acquirer will generate a MANDATEID.

The following fields will be presented to the buyer:

- IBAN
- Bankkonto
- BLZ

The buyer can choose whether to enter IBAN, or Konto+BLZ. If both are entered, the IBAN will be taken into account.

(\*SEPA: Single Euro Payments Area)

## 10.6.2 Direct Debits NL

The following fields should be sent along for Direct Debits NL transactions (only via Equens):

Field	Description
MANDATEID	<p>Unique Mandate Reference (for SEPA*)</p> <p>Max. 35 AN characters, charset: "A-Z a-z 0-9 /-?: () ., '+")</p> <p>No spaces allowed; cannot start or end with a forward slash "/" or contain two consecutive slashes.</p> <p><i>Only applicable for SEPA (*) transactions. If not provided, the ORDERID will be taken</i></p>
SIGNDATE	<p>Date mandate was signed by the buyer. Format: YYYYMMDD</p> <p><i>Only applicable for SEPA (*) transactions. If not provided, the transaction date will be taken.</i></p>
SEQUENCETYPE	<p>Possible values to indicate the Direct Debit transaction type (max. 4 alphanumeric characters):</p> <ul style="list-style-type: none"> <li>- "FRST": First collection of a series of Direct Debit instructions</li> <li>- "RCUR": Direct Debit instructions where the debtor's authorisation is used for regular Direct Debit transactions initiated by the creditor</li> <li>- "FNAL": Last collection of a series of Direct Debit instructions (afterwards same MandateID can't be used anymore)</li> <li>- "OOFF": Direct Debit instruction where the debtor's authorisation is used to initiate one single Direct Debit transaction</li> </ul> <p><i>Only applicable for SEPA (*) transactions.</i></p> <p><i>If not provided, the transaction will be considered as a "one-off" (OOFF is applied)</i></p> <p><i>If "FRST", "RCUR" or "FNAL" is sent, you must submit values with the MANDATEID and SIGNDATE</i></p>

*More information about these fields can be found in your Ogone account. Just log in and go to: Support > Integration & user manuals > Technical guides > Parameter Cookbook.*

Note: These fields can be returned in the post-payment feedback (Dynamic e-commerce parameters) and need to be included in the SHA-IN (and optionally SHA-OUT) calculation.

(\* Single Euro Payments Area)

The following fields will be presented to the buyer on the payment page:

- IBAN

## 11 Appendix: SHA

For each order, the merchant's server generates a unique character string, hashed with the SHA-1 algorithm developed by NIST (see [here](#)).

### Important

Our system requires the use of an SHA signature!

### 11.1 SHA-IN signature

This string is constructed by concatenating the values of the fields sent with the order (sorted alphabetically, in the format 'parameter=value'), followed by a passphrase. The passphrase is defined in the merchant's Technical information page, under the tab "Data and origin verification", section "Checks for e-Commerce." Please note that these values are all case sensitive when compiled to form the string before the hash!

### Important

- All parameters that you send (and that appear in the list in List of parameters to be included in SHA IN calculation), will be included in the string to be hashed.
- All parameter names should be in UPPERCASE (to avoid any case confusion).
- All parameters need to be arranged alphabetically
- Some sorting algorithms place special characters in front of the first letter of the alphabet, while others place them at the end. If in doubt, please respect the order as displayed in the SHA list.
- Parameters that do not have a value should NOT be included in the string to hash
- When you choose to transfer your test account to production via the link in the account menu, a random SHA-IN passphrase will be automatically configured in your production account.
- For extra safety, we request that you to use different SHA passphrases in test and production. Please note that if they are found to be identical, your TEST passphrase will be changed by our system (you will of course be notified).

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 [here](#).

*Example of a SHA-1-IN calculation with only basic parameters*

Parameters (in alphabetical order)

AMOUNT: 15.00 -> 1500

CURRENCY: EUR

LANGUAGE: en\_US

ORDERID: 1234

PSPID: MyPSPID

SHA-IN passphrase (in Technical information)

Mysecretsig1875!?

String to hash

AMOUNT=1500Mysecretsig1875!?!CURRENCY=EURMysecretsig1875!?!?

LANGUAGE=en\_USMysecretsig1875!?!ORDERID=1234Mysecretsig1875!?!?

PSPID=MyPSPIDMysecretsig1875!?!?

Resulting Digest (SHA-1)

*F4CC376CD7A834D997B91598FA747825A238BE0A*

If the SHASIGN sent in the hidden HTML fields of the transaction doesn't match the SHASIGN constructed at our end with the details of the order and the additional string (password/passphrase) entered in the SHA-IN passphrase field in the "Data and origin verification" tab, in the "Checks for e-Commerce" section of the Technical information page, you will receive the error message *"unknown order/1/s"*.

If nothing is sent in the "SHASIGN" field in the hidden HTML fields, even though an additional string (password/passphrase) has been entered in the SHA-IN passphrase field in the "Data and origin verification" tab, "Checks for e-Commerce" section of the Technical information page – indicating you want to use an SHA signature with each transaction – you will receive the error message *"unknown order/0/s"*.

Following is the hidden field 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.

*More information about these fields can be found in your Ogone account. Just log in and go to: Support > Integration & user manuals > Technical guides > Parameter Cookbook.*

## 11.2 SHA-OUT signature

This string is constructed by concatenating the values of the fields sent with the order (sorted alphabetically, in the format 'PARAMETER=value'), followed by a passphrase. The passphrase is defined in the merchant's *Technical information* > "Transaction feedback" tab > "All transaction Submission modes" section. For the full list of parameters to include in the SHA Digest, please refer to [Appendix: SHA Parameters](#). Please note that these values are all case sensitive.

### Important

- All sent parameters (that appear in the list in [Appendix: List of Parameters to be included in SHA Calculations](#)), will be included in the string to hash.
- All parameters need to be sorted alphabetically
- Parameters that do not have a value should NOT be included in the string to hash
- Even though some parameters are (partially) returned in lower case by our system, for the SHA-OUT calculation each parameter must be put in upper case.
- When you choose to transfer your test account to production via the link in the back-office menu, a random SHA-OUT passphrase will be automatically configured in your production account.
- For extra safety, we request that you use different SHA passphrases for TEST and PROD. Please note that if they are found to be identical, your TEST passphrase will be changed by our system (you will of course be notified).

In the same way we recreate the Digest to validate the transaction input 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 guarantees the accuracy and integrity of the parameter values sent in the request.

*Example of a basic SHA-1-OUT calculation*

*Parameters (in alphabetical order, as returned by Ogone):*

*ACCEPTANCE: 1234*

*amount: 15*

*BRAND: VISA*

*CARDNO: XXXXXXXXXXXXXXX1111*

```
currency: EUR
NCERROR: 0
orderID: 12
PAYID: 32100123
PM: CreditCard
STATUS: 9

SHA-OUT passphrase (in Technical info):
Mysecretsig1875!?
```

String to hash (with all parameters in uppercase:

```
ACCEPTANCE=1234Mysecretsig1875!?AMOUNT=15Mysecretsig1875!?
BRAND=VISAMysecretsig1875!?CARDNO=XXXXXXXXXXXX1111Mysecretsig1875!?
CURRENCY=EURMysecretsig1875!?NCERROR=0Mysecretsig1875!?
ORDERID=12Mysecretsig1875!?PAYID=32100123Mysecretsig1875!?
PM=CreditCardMysecretsig1875!?STATUS=9Mysecretsig1875!?
```

Resulting Digest (SHA-1):

```
209113288F93A9AB8E474EA78D899AFDBB874355
```

#### Character encoding for PostFinance payment methods

If you use UTF-8 character encoding for the integration of PostFinance Card and/or PostFinance E-finance, the transaction feedback will be returned ISO-8859-1 encoded.

## 11.3 SHA module

To be able to hash a string and send it to us, you must first install an Encryption module on your server.

SHA-1, SHA-256 and SHA-512 modules can be found on the internet, so you will not have any problem in finding a suitable one for your server. To help you find a 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/SHA1\\_1\\_0.html](http://www.w3.org/PICS/DSig/SHA1_1_0.html)

.NET/SHA1:

<http://msdn2.microsoft.com/en-us/library/system.security.cryptography.sha1managed.aspx>

PHP/SHA1:

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

## 12 Appendix: UTF-8

By default, Ogone uses ISO character encoding. However, our system supports the usage of UTF-8, if the merchant calls the appropriate pages.

For e-Commerce, the payment page would be [https://secure.ogone.com/ncol/test/orderstandard\\_utf8.asp](https://secure.ogone.com/ncol/test/orderstandard_utf8.asp)

### Important

- Our system cannot dynamically detect what character set the merchant is using. It is therefore the merchant's responsibility to call the appropriate page.
- If the merchant calls the UTF-8 payment page, all encoding for SHA will be done from a UTF-8 encoded string, for SHA-IN and for SHA-OUT. You can test your SHA integration with UTF-8 by calling the page [https://secure.ogone.com/ncol/test/testsha\\_utf8.asp](https://secure.ogone.com/ncol/test/testsha_utf8.asp)
- If the merchant is using dynamic templates, please ensure you declare the UTF-8 charset in the html header.

Please note that the usage of UTF-8 is mandatory for the following languages:

- Arabic
- Greek
- Hebrew
- Japanese
- Korean
- Russian
- Turkish



## 13 Appendix: 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 that 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 block the referrer information (payment page in pop up, special web server configuration, customer's browser configuration, etc.). 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 [SHA-IN signature](#)).

- *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/passphrase) entered in the SHA-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/passphrase) has been entered in the SHA-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 that the value you have entered in the PSPID field does not exist in the respective environment (test or production) or the account has not yet been activated.

- *no <parameter> (for instance: no PSPID)*

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

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

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

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

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

- *not a valid currency : ...*

This error means that 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 that 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 that 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.

## 14 Appendix: Short Status Overview

The following section contains a non-exhaustive list of statuses:

Status	NCERROR	NCSTATUS	Explanation
5 - Authorised	0	0	<p>The authorisation has been accepted.</p> <p>An authorisation code is available in the "ACCEPTANCE" field.</p> <p>The status will be 5 if you have defined "Authorisation" as the default operation code in the "Global transaction parameters" tab, in the "Default operation code" section of the Technical information page in your account.</p>
9 - Payment requested	0	0	<p>The payment has been accepted.</p> <p>An authorisation code is available in the field "ACCEPTANCE".</p> <p>The initial status of a transaction will be 9 if you have defined "Sale" as the default operation code in the "Global transaction parameters" tab, "Default operation code" section of the Technical information page in your account.</p>
0 - Invalid or incomplete	500....	5	<p>At least one of the payment data fields is invalid or missing. The NCERROR and NCERRORPLUS fields give an explanation of the error.</p>
2 - Authorization refused	300....	3	<p>The authorisation has been refused by the financial institution.</p> <p>The customer can retry the authorisation process after selecting another card or another payment method.</p>
51 - Authorisation waiting	0	0	<p>The authorisation will be processed offline.</p> <p>This is the standard response if the merchant has chosen offline processing in his account configuration.</p> <p>The status will be 51 in two cases:</p> <ul style="list-style-type: none"> <li>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.</li> <li>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.</li> </ul>
91 - Payment processing	0	0	<p>The data capture will be processed offline.</p>

Status	NCERROR	NCSTATUS	Explanation
52 - Authorisation not known  Or  92 - Payment uncertain	200...	2	A technical problem arose during the authorisation/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 authorisation process since the authorisation/payment might already have been accepted.
93 - Payment refused	300....	3	A technical problem arose.

*More information about statuses and error codes can be found in your Ogone account. Just log in and go to: Support > Integration & user manuals > User guides > List of the payment statuses and error codes.*

## 15 Appendix: Special Travel Format

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="AIEXPASNAME1" 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 the "mandatory/optional" fields, are only mentioned for information purposes and may differ slightly from one acquirer to another. Also, not all acquirers accept all fields.

Format: AN= Alphanumeric / N=Numeric, maximum allowed amount of characters

Field	Usage		Format
DATATYPE	"TRAVEL"	mandatory	TRAVEL
AIAIRNAME	Airline name	optional	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	16
AITIDATE	Ticket issue date. The default value is the transaction date.	optional	MM/DD/YYYY or YYYYMMDD

Field	Usage		Format
AICONJTI	Conjunction ticket	optional	3
AIPASNAME	Primary passenger name. The default value is the name of the credit cardholder.	optional	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	49
AICHDET	Charge details. Free text description or reference.	optional	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	1
AITYPCH	Type of charge	optional	2
AIEYCD	Destination area code	optional	3
AIIRST	Destination area code type	optional	1

*More information about these fields can be found in your Ogone account. Just log in and go to: Support > Integration & user manuals > Technical guides > Parameter Cookbook.*

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

Format: AN= Alphanumeric / N=Numeric, maximum allowed amount of characters

Field	Usage		Field details
AIORCITY1	Departure airport (short)	mandatory	3 (IATA)
AIORCITYL1	Departure airport (long)	mandatory	20
AIDESTCITY1	Arrival airport (short)	mandatory	3 (IATA)
AIDESTCITYL1	Arrival airport (long)	mandatory	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	4
AIBOOKIND1	Booking indicator	optional	2

Field	Usage		Field details
AIFLNUM1	Flight number	optional	4
AIFLDATE1	Flight date	optional	MM/DD/YY or YYYYMMDD
AICLASS1	Airline class	optional	15

*More information about these fields can be found in your Ogone account. Just log in and go to: Support > Integration & user manuals > Technical guides > Parameter Cookbook.*

Important: ascending order of consecutive travel fields for SHA calculation

Please take into account that as from 10 consecutive travel fields the order of the fields for the SHA calculation may differ from what is expected on your system's side.

Where for the calculation of the SHA we normally stress that all fields should be sorted alphabetically, a "logical" ascending order is put in place for the travel fields.

Example of expected (ascending) order:

AIFLNUM1, AIFLNUM2, AIFLNUM3, AIFLNUM4, AIFLNUM5, AIFLNUM6, AIFLNUM7, etc.

## 16 Appendix: 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& ...](https://secure.ogone.com/ncol/test/orderstandard.asp?PSPID=TESTSTD&OrderID=order123&amount=12500&currency=EUR&SHASIGN=8DDF4795640EB9FE9B367315C48E47338129A4F5&...))

Please refer to [Link between the Merchant's Website and our Payment Page](#) 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-IN, please refer to [Appendix: SHA](#).

## 17 Appendix: List of Parameters to be included in SHA Calculations

### 17.1 SHA-IN

ACCEPTANCE  
ACCEPTURL  
ADDMATCH  
ADDRMATCH  
AIACTIONNUMBER  
AIAGIATA  
AIAIRNAME  
AIAIRTAX  
AIBOOKIND\*XX\*  
AICARRIER\*XX\*  
AICHDET  
AICLASS\*XX\*  
AICONJTI  
AIDEPTCODE  
AIDESTCITY\*XX\*  
AIDESTCITYL\*XX\*  
AIEXPASNAME\*XX\*  
AIEYCD  
AIFLDATE\*XX\*  
AIFLNUM\*XX\*  
AIGLNUM  
AIINVOICE  
AIIRST  
AIORCITY\*XX\*  
AIORCITYL\*XX\*  
AIPASNAME  
AIPROJNUM  
AISTOPOV\*XX\*  
AITIDATE  
AITINUM  
AITINUML\*XX\*  
AITYPCH  
AIVATAMNT  
AIVATAPPL  
ALIAS  
ALIASOPERATION  
ALIASPERSISTEDAFTERUSE  
ALIASUSAGE  
ALLOWCORRECTION  
AMOUNT  
AMOUNT\*XX\*



AMOUNTHTVA  
AMOUNTTVA  
ARP\_TRN  
BACKURL  
BATCHID  
BGCOLOR  
BLVERNUM  
BIC  
BIN  
BRAND  
BRANDVISUAL  
BUTTONBGCOLOR  
BUTTONTXTCOLOR  
CANCELURL  
CARDNO  
CATALOGURL  
CAVV\_3D  
CAVVALGORITHM\_3D  
CERTID  
CHECK\_AAV  
CIVILITY  
CN  
COM  
COMPLUS  
CONVCCY  
COSTCENTER  
COSTCODE  
CREDITCODE  
CREDITDEBIT  
CUID  
CURRENCY  
CVC  
CVCFLAG  
DATA  
DATATYPE  
DATEIN  
DATEOUT  
DBXML  
DCC\_COMMPERC  
DCC\_CONVAMOUNT  
DCC\_CONVCCY  
DCC\_EXCHRATE  
DCC\_EXCHRATESTS  
DCC\_INDICATOR  
DCC\_MARGINPERC  
DCC\_REF  
DCC\_SOURCE  
DCC\_VALID

DECLINEURL  
DELIVERYDATE  
DEVICE  
DISCOUNTRATE  
DISPLAYMODE  
ECI  
ECI\_3D  
ECOM\_BILLTO\_COMPANY  
ECOM\_BILLTO\_POSTAL\_CITY  
ECOM\_BILLTO\_POSTAL\_COUNTRYCODE  
ECOM\_BILLTO\_POSTAL\_COUNTY  
ECOM\_BILLTO\_POSTAL\_NAME\_FIRST  
ECOM\_BILLTO\_POSTAL\_NAME\_LAST  
ECOM\_BILLTO\_POSTAL\_NAME\_PREFIX  
ECOM\_BILLTO\_POSTAL\_POSTALCODE  
ECOM\_BILLTO\_POSTAL\_STREET\_LINE1  
ECOM\_BILLTO\_POSTAL\_STREET\_LINE2  
ECOM\_BILLTO\_POSTAL\_STREET\_LINE3  
ECOM\_BILLTO\_POSTAL\_STREET\_NUMBER  
ECOM\_BILLTO\_TELECOM\_MOBILE\_NUMBER  
ECOM\_BILLTO\_TELECOM\_PHONE\_NUMBER  
ECOM\_CONSUMERID  
ECOM\_CONSUMER\_GENDER  
ECOM\_CONSUMEROGID  
ECOM\_CONSUMERORDERID  
ECOM\_CONSUMERUSERALIAS  
ECOM\_CONSUMERUSERPWD  
ECOM\_CONSUMERUSERID  
ECOM\_ESTIMATEDDELIVERYDATE  
ECOM\_ESTIMATEDDELIVERYDATE  
ECOM\_PAYMENT\_CARD\_EXPDATE\_MONTH  
ECOM\_PAYMENT\_CARD\_EXPDATE\_YEAR  
ECOM\_PAYMENT\_CARD\_NAME  
ECOM\_PAYMENT\_CARD\_VERIFICATION  
ECOM\_SHIPMETHOD  
ECOM\_SHIPMETHODDETAILS  
ECOM\_SHIPMETHODSPEED  
ECOM\_SHIPMETHODTYPE  
ECOM\_SHIPTO\_COMPANY  
ECOM\_SHIPTO\_DOB  
ECOM\_SHIPTO\_ONLINE\_EMAIL  
ECOM\_SHIPTO\_POSTAL\_CITY  
ECOM\_SHIPTO\_POSTAL\_COUNTRYCODE  
ECOM\_SHIPTO\_POSTAL\_COUNTY  
ECOM\_SHIPTO\_POSTAL\_NAME\_FIRST  
ECOM\_SHIPTO\_POSTAL\_NAME\_LAST  
ECOM\_SHIPTO\_POSTAL\_NAME\_PREFIX  
ECOM\_SHIPTO\_POSTAL\_POSTALCODE

ECOM\_SHIPTO\_POSTAL\_STATE  
ECOM\_SHIPTO\_POSTAL\_STREET\_LINE1  
ECOM\_SHIPTO\_POSTAL\_STREET\_LINE2  
ECOM\_SHIPTO\_POSTAL\_STREET\_NUMBER  
ECOM\_SHIPTO\_TELECOM\_FAX\_NUMBER  
ECOM\_SHIPTO\_TELECOM\_MOBILE\_NUMBER  
ECOM\_SHIPTO\_TELECOM\_PHONE\_NUMBER  
ECOM\_SHIPTO\_TVA  
ED  
EMAIL  
EXCEPTIONURL  
EXCLPMLIST  
EXECUTIONDATE\*XX\*  
FACEXCL\*XX\*  
FACTOTAL\*XX\*  
FIRSTCALL  
FLAG3D  
FONTTYPE  
FORCECODE1  
FORCECODE2  
FORCECODEHASH  
FORCEPROCESS  
FORCETP  
FP\_ACTIV  
GENERIC\_BL  
GIROPAY\_ACCOUNT\_NUMBER  
GIROPAY\_BLZ  
GIROPAY\_OWNER\_NAME  
GLOBORDERID  
GUID  
HDFONTTYPE  
HDTBLBGCOLOR  
HDTBLTXTCOLOR  
HEIGHTFRAME  
HOMEURL  
HTTP\_ACCEPT  
HTTP\_USER\_AGENT  
INCLUDE\_BIN  
INCLUDE\_COUNTRIES  
INITIAL\_REC\_TRN  
INVDATA  
INVDISCOUNT  
INVLEVEL  
INVORDERID  
ISSUERID  
IST\_MOBILE  
ITEM\_COUNT  
ITEMATTRIBUTES\*XX\*

ITEMCATEGORY\*XX\*  
ITEMCOMMENTS\*XX\*  
ITEMDESC\*XX\*  
ITEMDISCOUNT\*XX\*  
ITEMFDMPRODUCTCATEG\*XX\*  
ITEMID\*XX\*  
ITEMNAME\*XX\*  
ITEMPRICE\*XX\*  
ITEMQUANT\*XX\*  
ITEMQUANTORIG\*XX\*  
ITEMUNITOFMEASURE\*XX\*  
ITEMVAT\*XX\*  
ITEMVATCODE\*XX\*  
ITEMWEIGHT\*XX\*  
LANGUAGE  
LEVEL1AUTHPC  
LIDEXCL\*XX\*  
LIMITCLIENTSCRIPTUSAGE  
LINE\_REF  
LINE\_REF1  
LINE\_REF2  
LINE\_REF3  
LINE\_REF4  
LINE\_REF5  
LINE\_REF6  
LIST\_BIN  
LIST\_COUNTRIES  
LOGO  
MANDATEID  
MAXITEMQUANT\*XX\*  
MERCHANTID  
MODE  
MTIME  
MVER  
NETAMOUNT  
OPERATION  
ORDERID  
ORDERSHIPCOST  
ORDERSHIPMETH  
ORDERSHIPTAX  
ORDERSHIPTAXCODE  
ORIG  
OR\_INVORDERID  
OR\_ORDERID  
OWNERADDRESS  
OWNERADDRESS2  
OWNERCTY  
OWNERTELNO

OWNERTELNO2  
OWNERTOWN  
OWNERZIP  
PAIDAMOUNT  
PARAMPLUS  
PARAMVAR  
PAYID  
PAYMENTOCCURRENCE  
PAYMETHOD  
PM  
PMLIST  
PMLISTPMLISTTYPE  
PMLISTTYPE  
PMLISTTYPEPMLIST  
PMTYPE  
POPUP  
POST  
PSPID  
PSWD  
RECIPIENTACCOUNTNUMBER  
RECIPIENTDOB  
RECIPIENTLASTNAME  
RECIPIENTZIP  
REF  
REFER  
REFID  
REFKIND  
REF\_CUSTOMERID  
REF\_CUSTOMERREF  
REGISTERED  
REMOTE\_ADDR  
REQGENFIELDS  
RNPOFFERT  
RTIMEOUT  
RTIMEOUTREQUESTEDTIMEOUT  
SCORINGCLIENT  
SEQUENCETYPE  
SETT\_BATCH  
SID  
SIGNDATE  
STATUS\_3D  
SUBSCRIPTION\_ID  
SUB\_AM  
SUB\_AMOUNT  
SUB\_COM  
SUB\_COMMENT  
SUB\_CUR  
SUB\_ENDDATE

SUB\_ORDERID  
SUB\_PERIOD\_MOMENT  
SUB\_PERIOD\_MOMENT\_M  
SUB\_PERIOD\_MOMENT\_WW  
SUB\_PERIOD\_NUMBER  
SUB\_PERIOD\_NUMBER\_D  
SUB\_PERIOD\_NUMBER\_M  
SUB\_PERIOD\_NUMBER\_WW  
SUB\_PERIOD\_UNIT  
SUB\_STARTDATE  
SUB\_STATUS  
TAAL  
TAXINCLUDED\*XX\*  
TBLBGCOLOR  
TBLTXTCOLOR  
TID  
TITLE  
TOTALAMOUNT  
TP  
TRACK2  
TXTBADDR2  
TXTCOLOR  
TXTOKEN  
TXTOKENXTOKENPAYPAL  
TXSHIPPING  
TXSHIPPINGLOCATIONPROFILE  
TXURL  
TXVERIFIER  
TYPE\_COUNTRY  
UCAF\_AUTHENTICATION\_DATA  
UCAF\_PAYMENT\_CARD\_CVC2  
UCAF\_PAYMENT\_CARD\_EXPDATE\_MONTH  
UCAF\_PAYMENT\_CARD\_EXPDATE\_YEAR  
UCAF\_PAYMENT\_CARD\_NUMBER  
USERID  
USERTYPE  
VERSION  
WBTU\_MSISDN  
WBTU\_ORDERID  
WEIGHTUNIT  
WIN3DS  
WITHROOT

## 17.2 SHA-OUT

AAVADDRESS  
AAVCHECK  
AAVMAIL  
AAVNAME

AAVPHONE  
AAVZIP  
ACCEPTANCE  
ALIAS  
AMOUNT  
BIC  
BIN  
BRAND  
CARDNO  
CCCTY  
CN  
COLLECTOR\_BIC  
COLLECTOR\_IBAN  
COMPLUS  
CREATION\_STATUS  
CREDITDEBIT  
CURRENCY  
CVCHECK  
DCC\_COMMPERCENTAGE  
DCC\_CONVAMOUNT  
DCC\_CONVCCY  
DCC\_EXCHRATE  
DCC\_EXCHRATESOURCE  
DCC\_EXCHRATESTS  
DCC\_INDICATOR  
DCC\_MARGINPERCENTAGE  
DCC\_VALIDHOURS  
DEVICEID  
DIGESTCARDNO  
ECI  
ED  
EMAIL  
ENCCARDNO  
FXAMOUNT  
FXCURRENCY  
IP  
IPCTY  
MANDATEID  
MOBILEMODE  
NBREMAILUSAGE  
NBRIPIUSAGE  
NBRIPIUSAGE\_ALLTX  
NBRUSAGE  
NCERROR  
ORDERID  
PAYID  
PAYMENT\_REFERENCE  
PM

SCO\_CATEGORY  
SCORING  
SEQUENCETYPE  
SIGNDATE  
STATUS  
SUBBRAND  
SUBSCRIPTION\_ID  
TRXDATE  
VC