



[www.ogone.com](http://www.ogone.com)

Payment service provider

Document II:  
**Ogone e-Commerce,  
integration in the  
merchant's WEB site**

# 1 Table of contents

---

1	TABLE OF CONTENTS .....	1
2	INTRODUCTION .....	3
3	STEP BY STEP BASIC INTEGRATION .....	4
3.1	OPEN A TEST ACCOUNT .....	4
3.2	CONFIGURE YOUR ACCOUNT .....	4
3.3	REQUEST ACTIVATION .....	6
3.4	MAKE YOUR FIRST TEST PAYMENT .....	6
3.5	ADD THE "ORDERING FORM" TO YOUR SHOPPING BASKET .....	7
3.6	SIMULATE A REJECTED PAYMENT IN TEST .....	7
3.7	TROUBLE SHOOTING .....	7
3.8	FURTHER INTEGRATION .....	8
3.9	HOW TO UPGRADE FROM A TEST ACCOUNT TO A PRODUCTION ACCOUNT .....	8
4	OPERATING PRINCIPLE .....	10
4.1	DESCRIPTION OF THE SALE PROCESS WITH OZONE E-COMMERCE FOR A BASIC INTEGRATION .....	10
4.2	DESCRIPTION OF A SALE WITH OZONE E-COMMERCE IN THE CASE OF AN ADVANCED INTEGRATION .....	13
5	INTEGRATION PRINCIPLES .....	15
5.1	STARTING THE PROCESS .....	16
5.2	SECURITY: CHECK BEFORE PAYMENT .....	18
5.2.1	http request to an executable page of the merchant's site .....	19
5.2.2	To send a signature with the order data .....	20
5.3	SECURITY: ENCRYPTION .....	20
5.4	SUBSEQUENT PROCESSING .....	20
5.5	LAYOUT TIP .....	21
5.6	PAYMENT (STEP 6) .....	24
5.6.1	Configuring the credit card payment method .....	24
5.7	POST-PAYMENT FEEDBACK (STEP 7) .....	24
5.7.1	Automating back-office tasks .....	25
5.7.2	Possible values of STATUS and NCERROR .....	26
5.7.3	Example of a post-sale executable page on the merchant's site ..	27
5.8	CUSTOMIZING THE END OF THE SALE PROCESS .....	28
5.8.1	Default behavior .....	28
5.8.2	Simple customization - Redirection depending on the payment result	29
5.8.3	Advanced customization .....	30

5.9	OFFLINE PROCESSING .....	32
6	APPENDIX 1 .....	33

## 2 Introduction

---

This document complements the Documents I a and I b: **General Information**. It explains the integration procedure of the Ogone e-Commerce Internet payment server provider in **Standard** mode (without the "Merchant Assistant" option).

The **Standard mode** is suitable for sites that already contain functions for delivery cost and VAT calculation, confirmation emails, addresses collection. Ogone e-Commerce intervenes when the exact amount to be paid is known. This mode of operation is referred to as "**Standard**" (without the "Merchant Assistant" option).

In this case, Ogone e-Commerce only asks, in secure mode, the payment particulars of the buyer (for example, credit card number) and requests the payment to the relevant acquirer.

For the description of the other products, the configuration of your account and the functionality of the administration site, please read the **General information** documents.

# 3 Step by step basic integration

---

This chapter explains, step by step, how to integrate quickly Ogone e-Commerce into your e-Commerce site in **Standard** mode. This integration will be a simple one allowing your customers to pay from your Internet site.

To take full advantage of the Ogone e-Commerce functionality (check the accuracy of the order data, automate post-sale processing...), see the following chapters.

If you are not sure the Standard mode is the most appropriate for you, please read the Document Ia: **General documentation** or go to <http://www.ogone.com> before to start.

It is recommended to do the integration first in our testing environment before upgrading to the production environment. Our testing environment is a site that works exactly like our production site, except that no transaction request is sent to the credit card acquirer, and we don't bill you to use it. So you can freely make test payments, change the configuration of your account, fine-tune the integration of our payment system into your e-Commerce application before you start to process real payments. More information about the testing procedure and the testing environment are available in the Document Ib: **General Information, advanced functions**.

## 3.1 Open a test account

With your browser, go to the following address: <http://www.ogone.com/>

- ⇒ Click on "Free trial"
- ⇒ Click on New test account
- ⇒ Choose Ogone e-Commerce (without the Merchant Assistant option).
- ⇒ Fill the form and click on "Register"

Your password will be sent to you at the email address you mentioned.

(For more information about the testing procedure, see the Document Ib: General information, advanced functions.)

## 3.2 Configure your account

Once you received your password, you can enter the management module to configure your test account.

- ⇒ With your browser, go to the following address: <http://www.ogone.com>
- ⇒ Click on Free Trial
- ⇒ Click on "Test login"
- ⇒ Enter your PSPID and your password and click on "Submit".

You will first complete your registration (click).

Fill the five forms

- ⇒ step 1 : confirm the information you entered earlier, click on "Submit", to modify it, click on "back". To go to the next step, click on next.
- ⇒ step 2 : choose your subscription type. In the test environment, the subscription is free, but the advanced functions of Ogone e-Commerce are not available with all the subscription type. Click on "Submit", click on next. For the list of our subscription rate see: <http://www.ogone.com> , option product>Ogone e-Commerce>Price.
- ⇒ step 3 : choose the currency you want to use for your payments (It must be supported by your acquirer). Click on "Submit". Click on "next".
- ⇒ step 4 : choose the languages you want to use with your buyers. You can enter the URL of a page where you explain your general sales conditions. A link to this URL will be placed on the payment pages shown to your customers. Click on "Submit", click on "next".
- ⇒ step 5 : enter the billing information and choose a charging method. As the subscription is free for a test account, this information is irrelevant. Nevertheless you need to fill the form, it doesn't matter if it is with fancy data. Click on "Submit". Click on "end".

Register with credit card acquirer (click)

You need to activate at least one credit card. We suggest you to activate VISA, as you can use their official credit card test number to perform test payments (4111111111111111)

- ⇒ Click on the "maintenance" button next to "credit card".
- ⇒ Choose how you want your credit card payments to be processed:

**Immediate:** the transaction request is sent to the acquirer as the customer is connected (appropriate for sales of goods delivered online)

**Scheduled:** we register the transaction and treat it later (max 4 hours). This method is faster for the customer and cheaper for the merchant (see : <http://www.ogone.com> , option **Product>Ogone e-Commerce>Price**)

- ⇒ Choose the type of transaction adapted to your e-Commerce:

**Direct sale:** the money is transferred directly to the merchant's account.

**automatic authorization and payment on request :** we request an authorisation (the money stay on your customer account, but is reserved (blocked) for you) . To get paid, you need to enter your administration site and request a payment (see the Document Ia: General Information).

**automatic payment after X days :** we request first an authorisation, then after X days, if you have not cancelled the authorisation, we request the payment. If X=0, we request the payment directly.

(Note: The automatic payment (data capture) is allowed only if you perform the check before payment (section 5.2 of this document)).

- ⇒ Click on "submit".
- ⇒ Select a credit card by clicking on the "Add" button next to it.
- ⇒ Choose your acquirer and click on "Add".
- ⇒ Click on the "maintenance" button next to your selected credit card.
- ⇒ Enter your UID (This is the identification number given to you by the credit card acquirer. In the test environment those data do not have to be real, so type 12345).
- ⇒ Select an acquirer.
- ⇒ Select "yes" for activation.
- ⇒ Click on "Submit".
- ⇒ Click on "End" to go back to the main page.
- ⇒ Click on "Technical information".

It is in this page that you have to enter the technical data needed to increase the security of the transaction, automate back office tasks, customize the end of the sale process... Keep the default configuration of the technical information page for your first test.

⇒ Click on "Submit".

Note: More information about the technical information is available by clicking on the "?" in the technical information page and in the following chapters of this documentation.

⇒ Click on "End".

### 3.3 Request activation

⇒ Click on "Ask Ogone e-Commerce administration to activate my account", You will see our general conditions. Confirm that you agree at the bottom of the page. (Click on "Ask Ogone e-Commerce administration to activate my account").

You will receive an email with the details of your account.

⇒ Forward this email to support@ogone.com to request the activation of your test account. For a test account, you don't have to sign the email and send it by fax or post or send the domiciliation form.

An email will confirm your activation.

### 3.4 Make your first test payment

With your browser, go to the following address: <https://secure.ogone.com/ncol/test/teststd.htm>

This page simulates your "shopping basket" application and will initiate a payment. Once your account is activated, you can use this form to perform payment and check that your account is correctly settled.

Fill it manually with your own data: your PSPID, an orderID, an amount multiplied by 100, the currency code you have chosen, the language code you have chosen. Leave the optional fields blank for the moment. (Their use is explained in the section 5.1 **Starting the sale process** of this document.)

⇒ Click on "Go !".

You should receive a page to select a payment method or a page asking you to confirm the payment.

For the credit card payment method, you can use a test VISA credit card number: 4111111111111111 (if you have activated this credit card).

⇒ Confirm the order.

You should see the message: "your order is accepted".

You will receive an email with the confirmation of your order.

You can look at your new payment in the administration module.

⇒ Go to: <http://www.ogone.com/>

⇒ Click on "Free trial".

⇒ Click on "Test login".

⇒ Login (enter your PSPID and your password).

⇒ Choose "Payments" on the left menu.

⇒ Enter your selection criteria and view your payments details.

For the first time, just select all the status checkboxes and leave the other fields at their default value. Click on list of payments. (The use of the **Payments** option is explained in the Document 1 a: General information)

## 3.5 Add the "ordering form" to your shopping basket

This procedure is explained in details in the section 5.1 of this document.

With your browser, go to the following address: <https://secure.ogone.com/ncol/test/teststd.htm>.

- ⇒ Copy and paste into your "shopping basket" page the html code of the ordering form (at the bottom of the page) as described in Section 5.1 of this document.
- ⇒ Replace the value of the fields with your own values (Those values must match the data you entered when you configured your account).

Some fields, like the orderID and the amount have to be assigned dynamically.

The action of the form must be: <https://secure.ogone.com/ncol/test/orderstandard.asp>

- ⇒ Make a test payment: click on the "submit" button of the ordering form of your shopping basket and follow the instructions on the screen.

Continue the payment process and check it in your administration module, as described in section 3.4.

## 3.6 Simulate a rejected payment in test

In the test environment, Ogone e-Commerce simulates the response from the acquirer. The result of a payment request (accepted, refused, uncertain) will depend on the amount of the transaction:

- from 0 to 9000, all the authorization requests are accepted,
- over 9000, all the authorization requests are refused,
- of 9999, the payment result is uncertain.

## 3.7 Trouble shooting

Problems might arise from the fact that the data we receive from your "ordering form" do not match the data you entered when configuring your account.

Most common errors:

**"PSPID not active"**:

Make sure your account is active.

Check the spelling of your PSPID (no spaces allowed).

**"Currency not accepted..."**:

The currency must be the one you choose for your catalogue.

**"Language not supported ..."**:

The language must be one of the languages you choose for your catalogue.

**"The card number doesn't match any card type supported by your merchant"**:

The credit card brand must be a brand you activated in your account. (We deduce the card brand from the first digits of the card number.)

**"Invalid decimals for the currency":**

The amount of an order must be multiplied by 100.

**"You must connect in secured mode ":**

Check the action URL of your "ordering form", it must start with https

**"This order has already been processed":**

We check that an order is not processed twice, so make sure that each payment has a different orderID. You can see the already used orderID by looking at the payments in the management module.

**"Timeout or double external reference":**

Same as above

**"Unknown order":**

The tests checking the accuracy of the payment data failed.

If you have entered your "ordering form" URL in the technical information page, point 2.2 (checking the payment's origin) and receive such an error, the payment request is coming from a different URL than the one you specified.

If you entered an XML page URL (point 3.1 of the technical information page), the result of that test was not satisfactory.

It is better to leave those fields blank for your first payments.

**The little padlock does not appear at the bottom of the browser**

If you use frames or if you add images that are not hosted on a secure server to your template page the little padlock will not appear even if the payment transaction is secured. Contact us to host your pictures (support@ogone.com).

We recommend you to use the "View source" option of your browser to check that the value of the hidden fields of your ordering form (in your shopping basket) seem correct.

## 3.8 Further integration

For the security checks, the customisation of the end of the payment, the automating of the back office tasks, see the next chapters.

For the offline processing, the transactions maintenance (payments, deletions...), see the Document I a and I b: **General Information**.

Attention, security checks are not the same in test and production environment, please read section 5.2 Check before payment.

## 3.9 How to upgrade from a Test account to a Production account

To open a real account, go to the following URL: <http://www.ogone.com> Click on **New account** in the top bar. Repeat the steps for the test account, except that:

- choose the type of subscription that fits your needs. (step 2 of the account registration)
- For the billing method, choose Credit card or direct debit, and fill carefully the relevant fields. (step 5)

Print, sign and fax us the confirmation email.

For the payment methods, enter the contractual parameters you received from your acquirer when you signed an online affiliation contract. Contact support@ogone.com, if you have any problem.

Your account will be activated as soon as all the administrative tasks have been completed.

In your ordering form replace the action:

<https://secure.ogone.com/ncol/test/orderstandard.asp>

with

<https://secure.ogone.com/ncol/prod/orderstandard.asp>

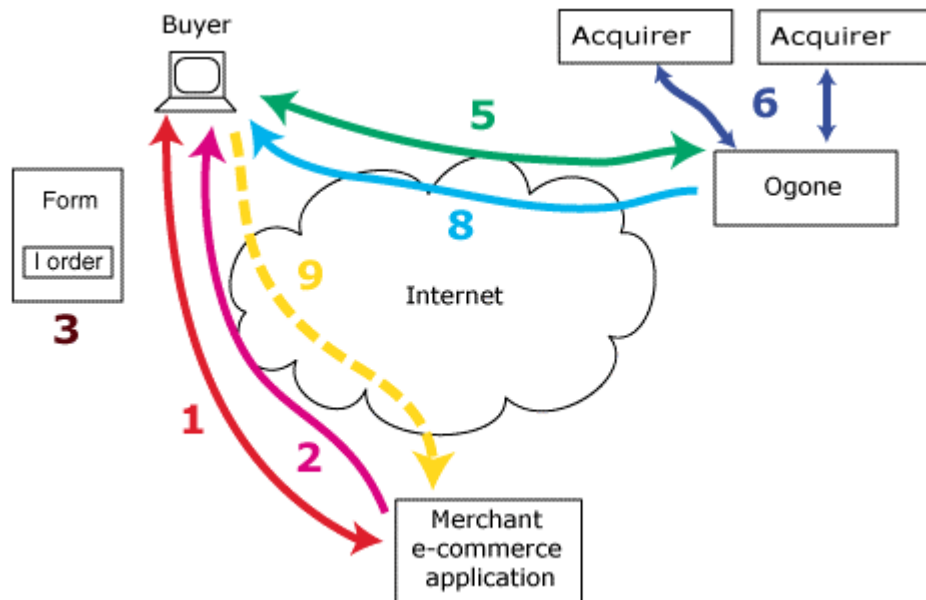
Replace your test PSPID with your production PSPID.

You are ready to process real payments.

Attention, security checks are not the same in test and production environment, please read 5.2 Check before payment.

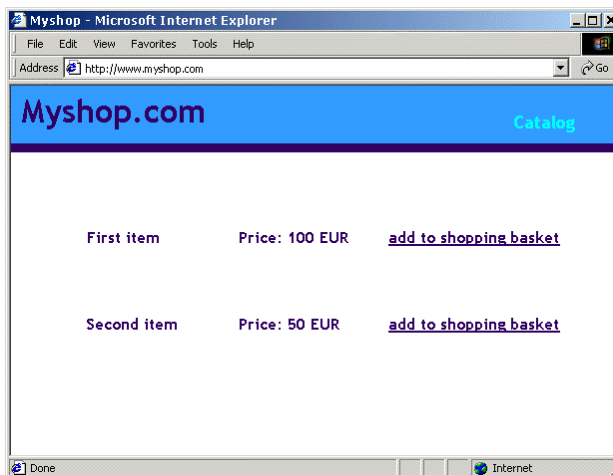
## 4 Operating principle

### 4.1 Description of the sale process with Ogone e-Commerce for a basic integration



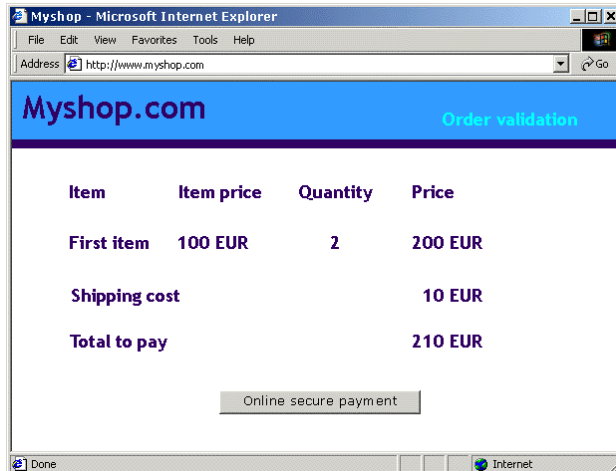
Schematically, the sale can be presented as follows:

On the merchant's site



**Step 1:** The buyer selects items from the merchant's catalog. He adds these to his virtual cart.

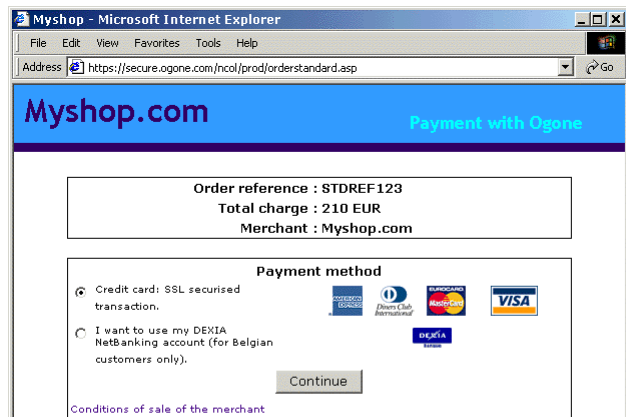
When he places his order, the buyer is invited to enter the parameters required for the administration of the order.



**Step 2:** The buyer receives a summary page giving the details of his order, his particulars and the exact amount to be paid (including shipping and VAT). He is invited to confirm this information so that he can pay for his order.

**Step 3:** This confirmation button is in fact the visible part of an "HTML form" that contains a certain number of hidden fields giving the payment data, and a submission action that automatically directs him in secure mode to a payment page of Ogone e-Commerce

### On the Ogone e-Commerce site



**Step 5:** Ogone e-Commerce presents a page to the buyer asking him to choose from the different payment methods selected by the merchant. If the payment is by credit card, the buyer is also asked to enter his card number, etc.

This can only be done if the buyer is connected in secure mode (HTTPS).

At any moment, the buyer may cancel the order.

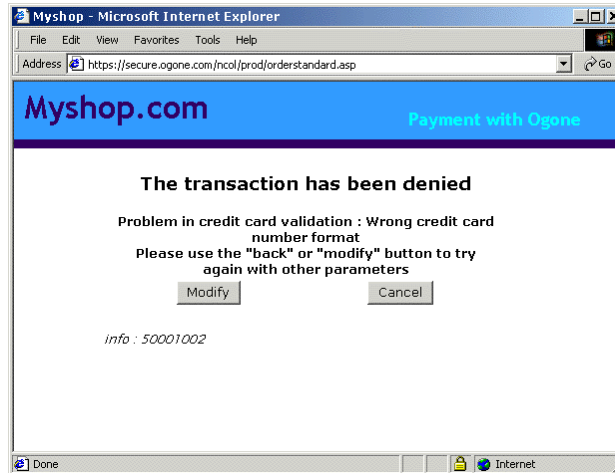


The merchant may customize the look of the pages shown to the buyer, by sending lay out info in the ordering form.

The buyer hardly notices that he has moved from one site to the other.

After choosing his payment method and, where appropriate, entering his particulars (for example, credit card number), the buyer confirms his payment request.

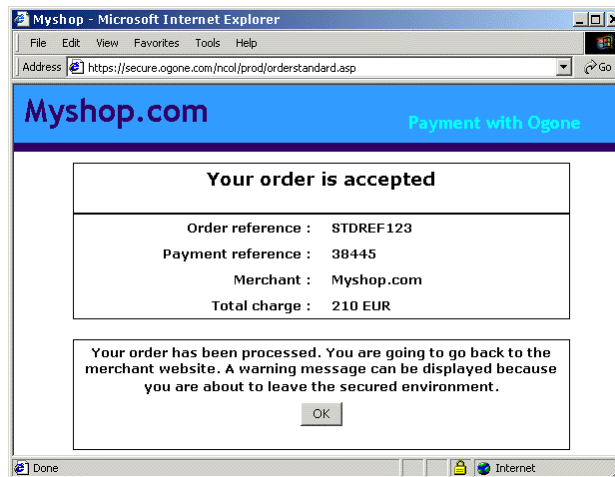
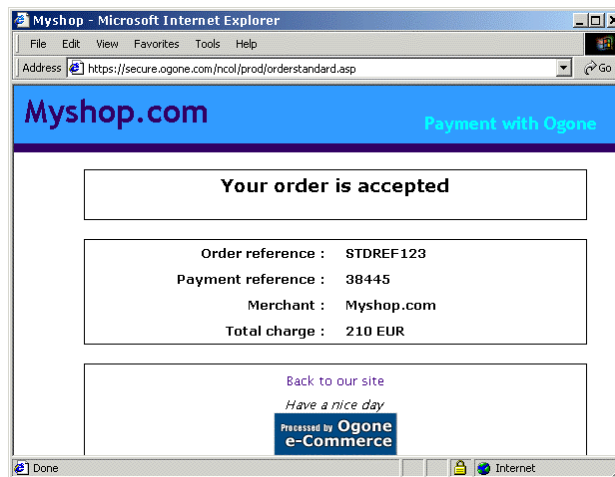
**Step 6:** The payment is requested from the relevant financial institution.



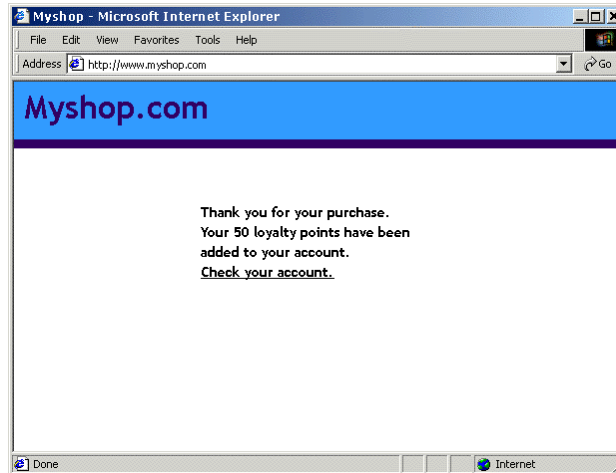
**Step 8:** Ogone e-Commerce sends to the buyer a page with the result of his payment.

If the payment is not accepted, the error is shown to the buyer in his language, and he is invited to use the "back" button of his browser to try using other parameters.

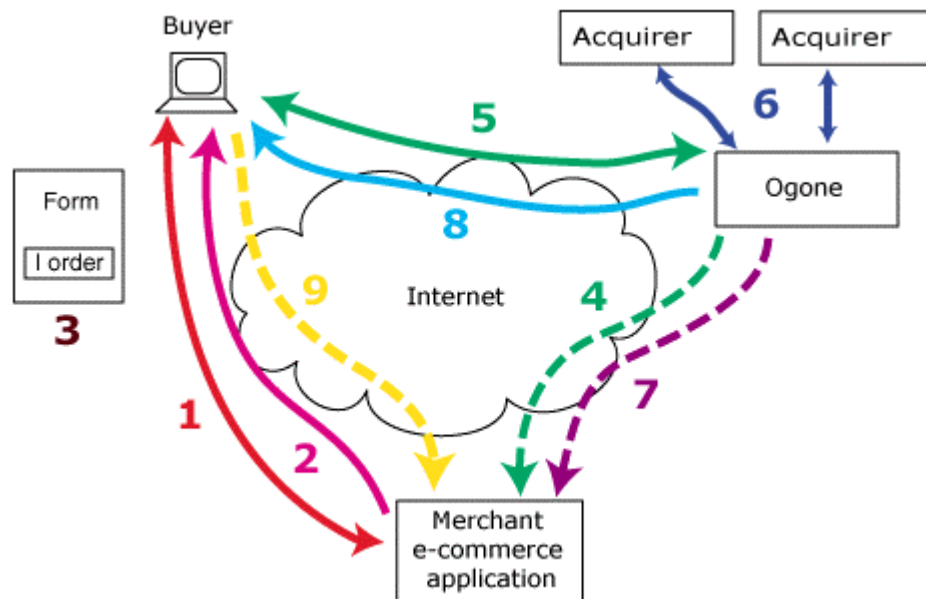
A confirmation email is also sent to the merchant



**Step 9:** (optional) To personalize the end of the payment process, the merchant may send into the ordering form, URL of several pages of his site where Ogone will redirect the buyer. This redirection might be done directly of after showing to the buyer an intermediary page with the result of his payment.



## 4.2 Description of a sale with Ogone e-commerce in the case of an advanced integration



When integrating further the Ogone e-Commerce into his application the merchant may:

- get a greater control on the look of the payment pages,
- check the accuracy of the order data,
- update automatically the payment status into his back-office application.

Notice: Some functions are not available for all the subscription type.

On the merchant's site

**Step 1:** Selecting items and placing an order: same as the basic integration

**Step 2:** Order summary: same as the basic integration

**Step 3:** Ordering form: same as the basic integration

On Ogone web site

**Step 4:** requests to the merchant's site before the payment

- Dynamic "template" page: Ogone e-Commerce retrieves on the merchant's site a template page used to build the pages shown to the buyer. This gives a lot of freedom to the merchant to control the lay out of the payment pages.
- Checking the payment data: Ogone e-Commerce makes an http request to an executable page on the merchant's site in order to retrieve the payment data in a XML format. The merchant may instead send a signature with the ordering form. If the merchant doesn't implement one of those check, he will not be allowed to make direct sales or automatic payment confirmation. See 5.2

**Step 5:** Dialog between the buyer and Ogone e-Commerce: same as the basic integration

**Step 6:** Payment request to the acquirer: same as the basic integration

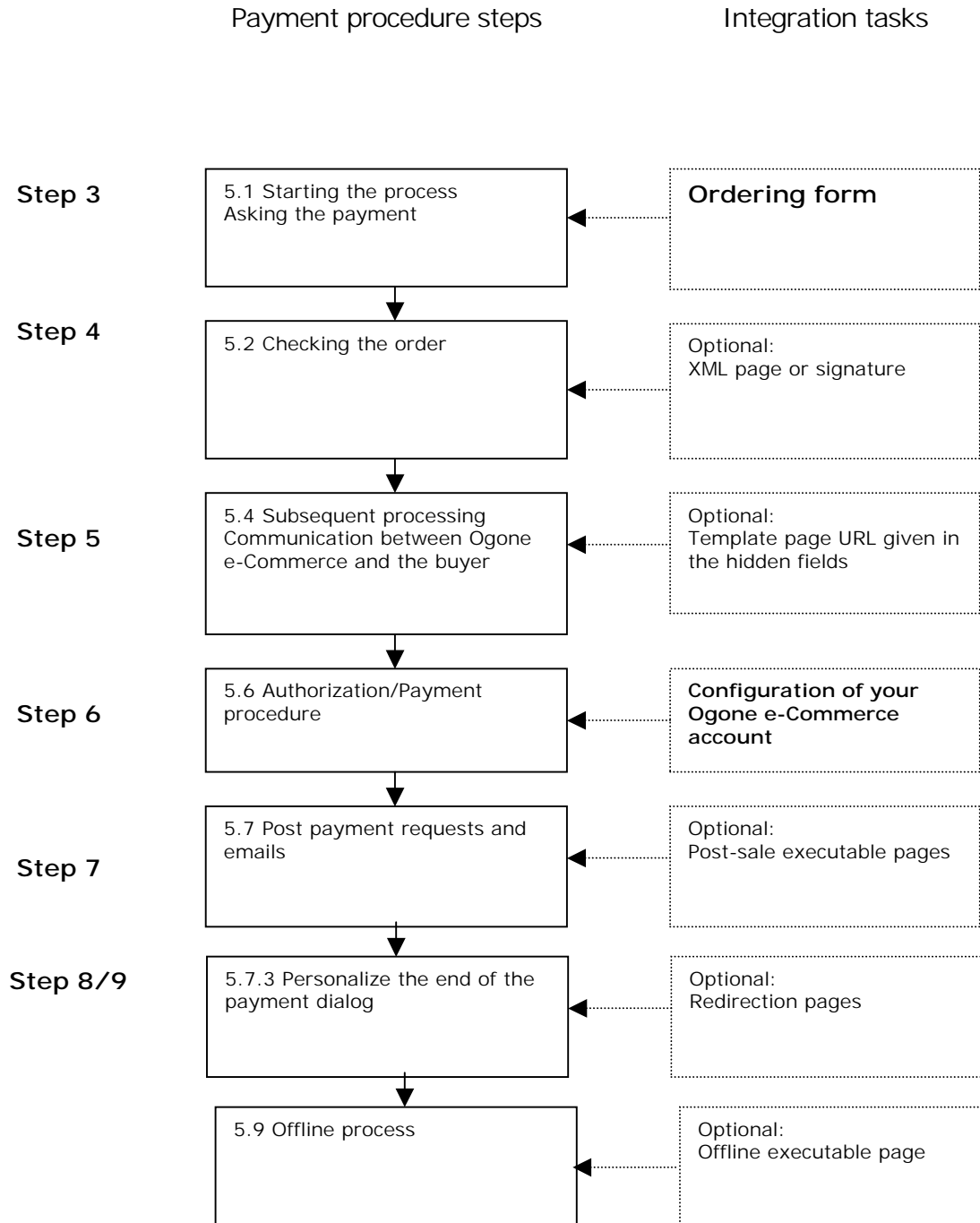
**Step 7:** Requests to the merchant's site after the payment:

- If the payment is accepted, an http request is made to the merchant site, on a page of which the URL is transmitted by the merchant at the time of his affiliation. To this URL are added, as parameters, the payment data. This allows the merchant to execute automatic processes such as the updating of the order status or the sending of an email to the buyer.
- If the payment is refused or cancelled by the buyer, Ogone e-Commerce makes a similar request to an other URL defined by the merchant when configuring his account.
- This request is done either before the buyer gets the payment result or later. In the first case, this page may generate a personalized response for the buyer.

**Step 8/9:** End of the dialog with the buyer:

The merchant may choose among the various solutions, the best way to personalize the response sent to his buyer. See 5.7.3

# 5 Integration principles



## 5.1 Starting the process

In the last page of the merchant site presented to the buyer (step 3), there must be a code block between the html tags "FORM" and "/FORM", this is the **ordering form**. To this form is attached an action which is the processing page of the payments on the Ogone e-Commerce site, as well as the hidden fields which give the order data.

```
<FORM METHOD="post" ACTION="https://secure.ogone.com/ncol/<test or
  prod>/<page for standard mode>">
  <INPUT TYPE="hidden" NAME="orderID" VALUE="99999">
  <INPUT TYPE="hidden" NAME="pspid" VALUE="XXXXX">
  <INPUT TYPE="hidden" NAME="RL" VALUE="ncol_2.0">
  <INPUT TYPE="hidden" NAME="currency" VALUE="EUR">
  <INPUT TYPE="hidden" NAME="language" VALUE="fr_FR">
  <INPUT TYPE="hidden" NAME="amount" VALUE="10000">
  <!--optional fields-->
  <!--layout fields-->
  <!--dynamic template page-->
  <INPUT TYPE="hidden" NAME="TP"
  VALUE="http://www.abssys.be/yyy/template9694.htm">
  <!--or static template page customisation -->
  <INPUT TYPE="hidden" NAME="TITLE" VALUE="The Title of my payment
  page">
  <INPUT TYPE="hidden" NAME="BGCOLOR" VALUE="#cccccc">
  <INPUT TYPE="hidden" NAME="TXTCOLOR" VALUE="black">
  <INPUT TYPE="hidden" NAME="TBLBGCOLOR" VALUE="yellow">
  <INPUT TYPE="hidden" NAME="TBLTXTCOLOR" VALUE="darkblue">
  <INPUT TYPE="hidden" NAME="BUTTONBGCOLOR" VALUE="orange">
  <INPUT TYPE="hidden" NAME="BUTTONTXTCOLOR" VALUE="#FFFFFF">
  <INPUT TYPE="hidden" NAME="LOGO" VALUE="MyLogo.gif">
  <INPUT type="hidden" NAME="FONTTYPE" VALUE="Arial">
  <!--other optional fields -->
  <INPUT TYPE="hidden" NAME="accepturl"
  VALUE="http://www.abssys.be/accept_order.asp">
  <INPUT TYPE="hidden" NAME="declineurl"
  VALUE="http://www.abssys.be/decline_order.asp">
  <INPUT TYPE="hidden" NAME="exceptionurl"
  VALUE="http://www.abssys.be/ exceptionurl_order.asp">
  <INPUT TYPE="hidden" NAME="cancelurl"
  VALUE="http://www.abssys.be/cancelurl_order.asp">
  <INPUT TYPE="hidden" NAME="SHASign" VALUE="The signature of your
  order">
  <INPUT TYPE="hidden" NAME="email" VALUE="dupont@blabla.be">
  <INPUT TYPE="hidden" NAME="COM" VALUE="3 telephone cards">
  <INPUT TYPE="hidden" NAME="CN" VALUE="Dupont Jacques">
  <INPUT TYPE="hidden" NAME="catalogurl"
  VALUE="http://www.abssys.be/catalog/">
  <INPUT TYPE="hidden" NAME="homeurl" VALUE="http://www.abssys.be">
  <!--optional payment method details-->
  <INPUT TYPE="hidden" NAME="PM" VALUE="CreditCard">
  <INPUT TYPE="hidden" NAME="BRAND" VALUE="VISA">
  <INPUT TYPE="hidden" NAME="ownercty" VALUE="NL">
  <INPUT TYPE="hidden" NAME="ownerZIP" VALUE="1040">
  <INPUT TYPE="hidden" NAME="owneraddress" VALUE="83, Main street">
  <INPUT TYPE="hidden" NAME="complus" VALUE="123456789">
  <INPUT TYPE="hidden" NAME="paramplus"
  VALUE="sessionID=12345&shopperID=45678">
  <INPUT TYPE="hidden" NAME="paramvar" VALUE="YhbRtoiUtThyrft">
  <INPUT TYPE="hidden" NAME="USERID" VALUE="ABCDE">

  <INPUT TYPE="submit" VALUE="Make the payment">
</FORM>
```

The exact URL of the action is communicated to the ISP or WEB designer at the time of developments and integration tests (See Document I b: **General information**, Testing Procedures).

The value of *orderID* is the number of the order: important! The system checks that a payment is not requested twice for the same order.

The value of *pspid* is the affiliation number of the merchant at Ogone e-Commerce.

The value of *RL* is the release version of the Ogone e-Commerce application.

The value of *currency* is the currency of the payment (code ISO alpha : EUR, USD, GBP, CHF...).

The value of *language* is the language of the buyer (languages available now: English – en\_US , French –fr\_FR, Dutch nl\_NL, Italian it\_IT, German de\_DE, Spanish es\_ES, Norwegian no\_NO).

The value of *amount* is the amount to be paid MULTIPLIED BY 100 ==> the format of the amount must not contain any decimals or other separator, but only figures:

100.00 --> 10000

100.25 --> 10025

1,250.32 --> 125032

Optional fields:

- Lay out

The value of *TP* is the URL of the dynamic template page that will serve as a skeleton to send the response to the buyer. The structure of this template page is explained later (section 5.4). If this parameter is not transmitted, Ogone e-Commerce will use a neutral page. This neutral can be personalized with the following fields:

*Title*: a text that appears at the top of the page.

*BGCOLOR*: background color, white by default

*TXTCOLOR*: text color, black by default

*TBLBGCOLOR*: background color of the tables where Ogone inserts the payment forms, white by default

*TBLTXTCOLOR*: text color, inside the tables, black by default

*BUTTONBGCOLOR*: background color of the submission button

*BUTTONTXTCOLOR*: text color of the submission button

*LOGO*: name of a .gif or .jpg picture placed at the top of the page. This picture must be sent to [support@ogone.com](mailto:support@ogone.com) before hand.

*FONTTYPE* font family, Verdana by default

- Post payment redirection

The value of *accepturl* is the URL of the page where to redirect the buyer when the payment is accepted (or waiting to be accepted).

The *declineurl* is similar to the *accepturl*. This URL is used when the acquirer rejects it more than the maximum of authorised tries.

The *exceptionurl* is used when the payment result is uncertain.

The *cancelurl* is used when the buyer cancelled the payment

Notice: Those redirections are explained in detail in the section 5.7.3 of this document. You should not use the redirection to trigger automatic back office tasks. If the communication with the buyer is interrupted (for instance the buyer clicking on the stop button of his browser) you will not receive the response. The section 5.7.3 of this document explains how to update your databases with the "post-sale" request. This "post-sale" request may be used to generate a response to the buyer in case of specific need (for instance to generate the key of a downloaded software).

The field SHASign contains a signature to validate the payment data. See 5.2

The *catalogurl* is the URL of your catalogue. When the payment is accepted, your buyer is invited to return to this URL.

The *homeurl* is the URL of your home page. When the payment is accepted, your buyer is invited to return to this URL.

The value of *COM* is a free description of the order.

The value of *CN* is the name of the buyer.

The value of *email* is the email address of the buyer.

*ownercty*: ISO code of the buyer's country (2 char.). If the Fraud detection option is activated, this parameter is used to show only the payment method relevant to the buyer's country.

*OwnerZIP* : ZIP code of the card owner.

*Owneraddress* Street/number/.. of the card owner.

These 2 fields are used by some acquirers to check the identity of the card owner. It doesn't prevent authorization or payment to be done but gives a warning to the merchant if check is not correct (more info at: [http://www.ogone.com/ncol/Infos\\_AAV\\_CVC1.htm](http://www.ogone.com/ncol/Infos_AAV_CVC1.htm)).

*Complus* and *Paramplus* are 2 values sent back to the merchant's website in the post-sale request (see chapter 5.7 )

*Paramvar* contains a variable part to include in the URL's used for Post-sale and/or Cancel-Deny requests. These URL's are configured in the technical information of the Ogone account, this variable part can be used for example to "adapt" the post-sale request in order to include session information, passing it as a part of the URL, and not as an additional parameter. It is the case for Intershop platforms or Servlets systems (see chapter 5.7 for the configuration of the post-sale and cancel-deny URL's and the use of this functionality)

UserID optional, only for merchants using the User Manager

*Optional Payment Method details* :If those fields are filled, the buyer is directed immediately to the relevant payment method page.

*PM*: Payment method :

CreditCard  
ING HomePay  
DEXIA NetBanking  
CBC Online  
KBC Online

*BRAND*: type of card

VISA  
American Express  
Diners Club  
Eurocard  
Mastercard  
Aurora

The text (value) of the submission button is obviously left up to the merchant, but the names of the fields must be absolutely respected.

## 5.2 Security: check before payment

Ogone e-Commerce checks the origin of the payment request (the URL of the page containing the "ordering form"). The merchant must configure this parameter in the management module, option **account / technical information point 2.2**.

This test is not foolproof, so, to ensure the accuracy of the order data, Ogone e-Commerce performs a data check before to process the payment.

The following control is not mandatory, but if the merchant doesn't use it, he will not be allowed to configure his account with automatic payment, he will have to use a 2 phases – reservation/manual payment request – procedure.

The merchant may choose 1 of the 2 following methods.

### 5.2.1 http request to an executable page of the merchant's site

Before the payment is actually processed, Ogone e-Commerce sends to the merchant's server the signal that the payment process has started and checks the payment data (step 4).

This works as follows:

The merchant enters an "XMLPAGE" parameter at the time of his affiliation. (Administration menu, option **Account, Technical information, point 3.1**)

As soon as an order is received, a request will be made on this URL, by adding, as parameters, the orderID: <https://www.abssys.be/getorder.asp?orderID=99999>.

In response to this request, the merchant's page must send back a very simple XML page containing the main payment data:

```
<DOCTYPE=....>
<orderID="99999"
amount="10000"
currency="EUR">
```

If the amount (multiplied by 100) is not filled in, the payment will be refused. The message "Order unknown..." will appear. If it is filled in, this is the amount that is taken as the actual amount of the payment.

Data from the "XML" page prevail over the data from the "ordering form".

Redirections are not allowed in this process.

Example of an ASP executable page sending back the payment data in XML format:

```
<%@ Language=VBScript %>
<%'*****
' * This is an ASP sample of pre-payment control dynamic page that can
' * be called before the payment by the Ogone Payment server.
' * This is delivered as is, without any guaranty, and should be installed and
' * adapted to merchant's specific e-Commerce application requirements
' * At least all the database names, userid, password, table names, field names
' * should be changed.
' * *****
orderid = request("orderid")
Set so_DB = Server.CreateObject("ADODB.Connection")
'so_DB.Open "mydatabase","userid","password"
so_DB.Open string ' string for connection
if (len(orderid) > 0) then
' If orderid is not defined as numeric, quotes should be added
SQLQuery = "SELECT TOTALAMOUNT, CURENCY, STATUS FROM PAYMENT"
SQLQuery = SQLQuery & " WHERE orderid = " & orderid
Set rs0_orders = so_DB.Execute(SQLQuery)

'*****
'* test if record exists and status is unpaid but completed :
'* adapt to your unpaid status and use strcmp function if status is not numeric
'*****

if rs0_orders.EOF then%>
<unknown orderid : <%=PAYID%>>
<%elseif (StrComp(rs0_orders("STATUS"),"not completed")=0) then%>
<uncorrect orderid : <%=orderid%>>
<%elseif (StrComp(rs0_orders("STATUS"),"paid")=0) then%>
<uncorrect orderid : <%=orderid%>>
<%else
amount=rs0_orders("TOTALAMOUNT") * 100%>
<orderid="<%=orderid%>"
amount="<%=amount%>"
currency="<%=rs0_orders("CURENCY")%>">
<%end if%>
<%else%>
'*****
'* Nor orderid received
'*****
<unknown orderid>
```

<%end if%>

## 5.2.2 To send a signature with the order data

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

This string is build by concatenating the value of the fields **OrderID**, **Amount**, **Currency**, **PSPID** from the ordering form and an additional string defined by the merchant in the **3.2. Data checking before the payment**, **SHA-1 Signature** field of the technical information page.

SHA-1(OrderID + Amount + Currency + PSPID + additional string)

If Ogone e-Commerce receives a signature in the ordering form, it checks that the order data match the signature.

In case of failure, the following message is shown "Order unknown..."

This control ensures the accuracy of the order data while avoiding a time-consuming http request.

## 5.3 Security: encryption

The URL used to connect the buyer to Ogone e-Commerce (the action of the ordering form: <https://secure.ogone.com/...> ) uses a secure protocol. So all the communication between Ogone e-Commerce and the buyer is securely encrypted.

However, the little padlock on the browser, showing to the buyer that the site is secured may not appear in some cases:

- Some elements included in the template page (e.g. a logo) are not located in secure sites.
- Some frames on the screen show pages that don't come from secure sites.

The rule is that, even if the payment processing communication is encrypted, the browser will recognize a secure connection only if **all the elements** on the screen, including images, sounds, etc. come from secure sites.

For merchants that do not have a secure site, we recommend the following procedures:

Stop usage of frames while payment begins. For example, the whole screen can be refreshed with a template page that looks like frames or the payment can be processed in a new window.

Don't link files to the template page (<link> tag). Use instead the <style>...</style> and <script>...</script> tags to include styles and scripts into the template page.

The template page can be in a non-secure site, but the images included in it, must be located in a secure site. Ogone e-Commerce can offer a housing for those elements (please contact us at [support@ogone.com](mailto:support@ogone.com) for conditions).

## 5.4 Subsequent processing

When Ogone e-Commerce asks the buyer for his credit card details, the buyer is on the secure Ogone server. In order to maintain the merchant's site look during all the payment process, there are two possible ways to customize the design of the payment process pages.

Static template page (simplest way to customize design)

With static template page, you can customize the design by simply adding some hidden fields in the payment form (see 5.1):

TITLE : title and header of the page  
 BGCOLOR : background color (e.g.: #FFFFFF or black,...), default value: white  
 TXTCOLOR : text color, default value: black  
 TBLBGCOLOR : table background color, default value: white  
 TBLTXTCOLOR : table text color, default value: black  
 BUTTONBGCOLOR: Button's background color  
 BUTTONTXTCOLOR: Button's text color, default value: black  
 FONTTYPE: the font family, default value: Verdana  
 LOGO : Filename of the logo you want to display at the top of the page. (Send this to [support@ogone.com](mailto:support@ogone.com) : JPG or GIF file)  
 Dynamic template page (advanced way to customize design, restricted for some subscriptions. Please check.)

The "Dynamic template" page is left for the merchant to complete. The only requirement is that it must contain the string "\$\$\$PAYMENT\_ZONE\$\$\$" indicating the location where Ogone e-Commerce can add its fields dynamically. It must therefore contain at least the following:

```
<html>
    $$$PAYMENT_ZONE$$$
</html>
```

This page can be static, common to all orders, or dynamic, generated according to the parameters of the order by the application of the merchant. To generate a dynamic page, the merchant can choose between creating a page specific to the order of which the URL is transmitted in the hidden fields (TP) or using a fixed URL but sending back a result depending on the order number. To do this, Ogone e-Commerce 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=..

The components included in the template page must have absolute URL's.

An example of template page is available at the following address:  
[http://www.ogone.com/ncol/template\\_standard.htm](http://www.ogone.com/ncol/template_standard.htm)

## 5.5 Layout tip

You can control the appearance of the payment forms by adding style to your template page. We have defined a class for the various types of tables and cells within those tables as well as a class for submit buttons. Add the following block of code between the tags <head></head> (see the example of template page):

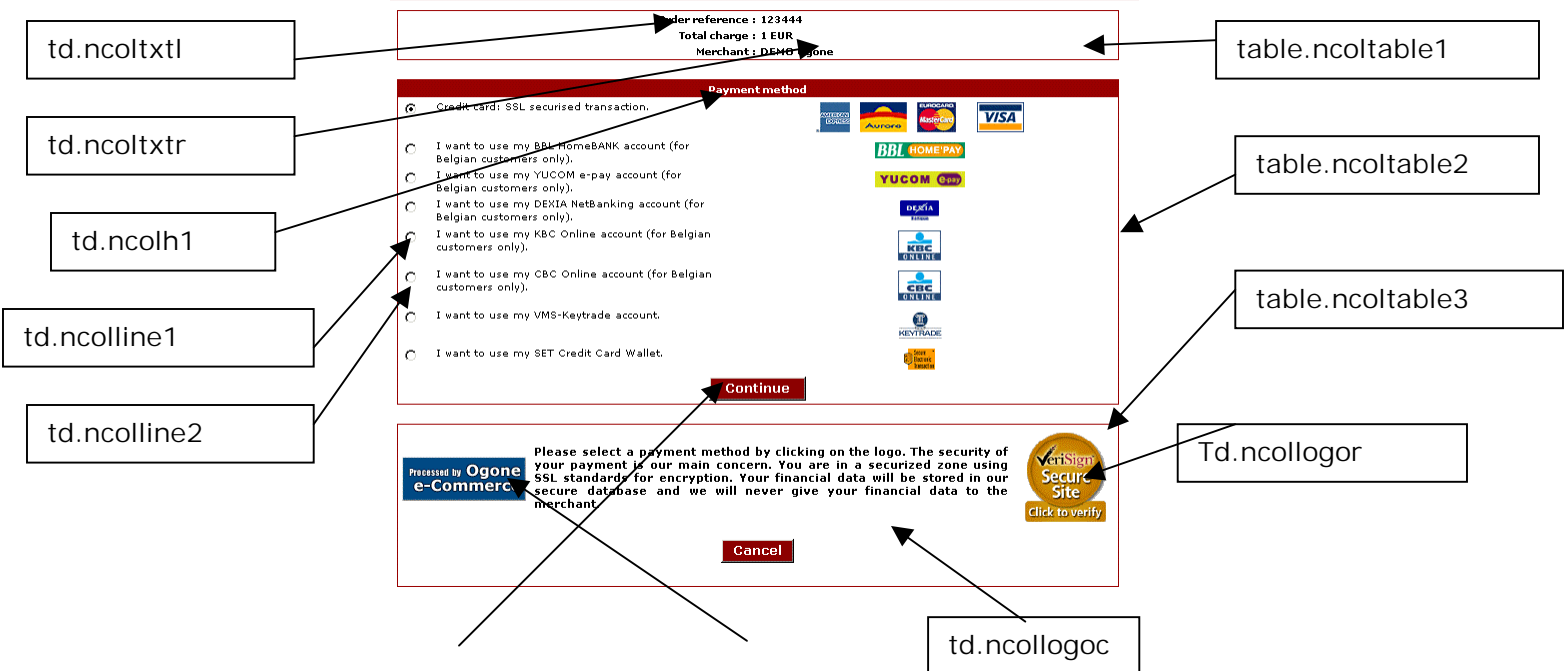
```
<style type="text/css">
    <!--
    td.ncolh1 { background-color : #006600; color : lightyellow; font-face : verdana}
    td.ncoltxtl { background-color : #ffffcc; color : black; text-align : right; font-weight :
bold}
```

```

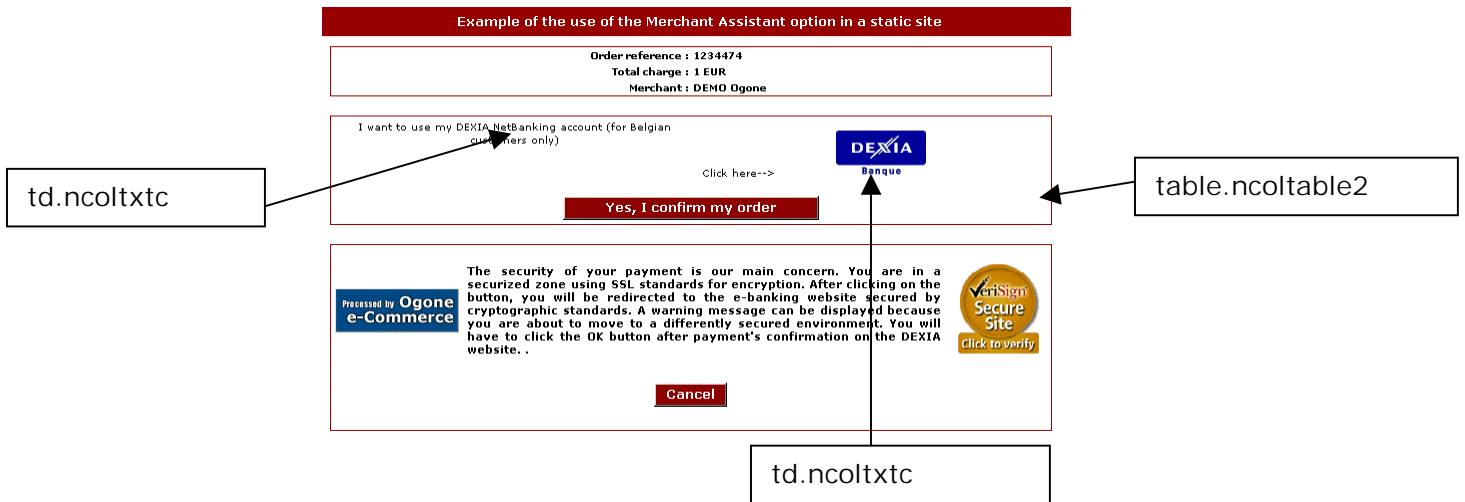
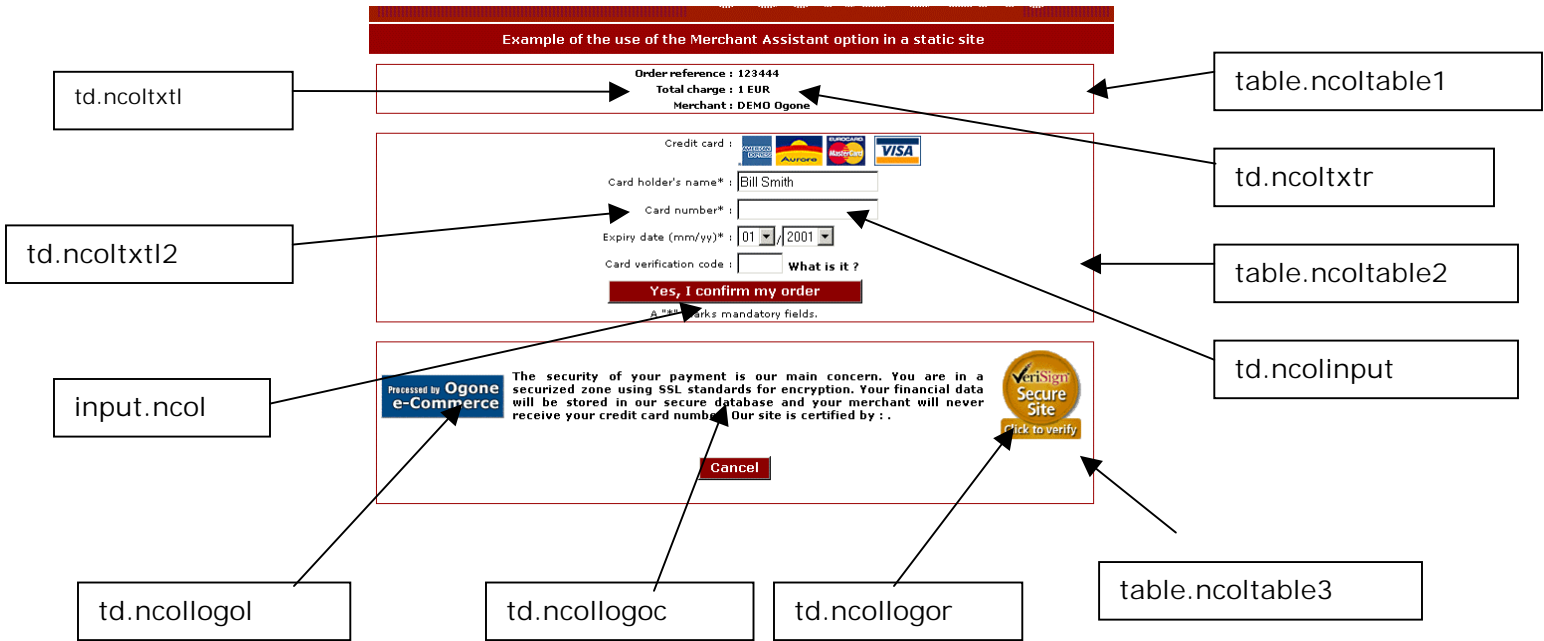
td.ncoltxtl2 {background-color : #ffffcc; color : black; text-align : right; font-weight :
bold}
td.ncoltxtr {background-color : #ffffcc; color : black; text-align : left; font-weight :
bold}
td.ncoltxtc {background-color : #ffffcc; color : black; text-align : center; font-weight :
bold}
td.ncoltxtmessage {background-color : #ffffcc; color : #ffffcc; text-align : left; 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.ncollogol {background-color : #ffffcc; color : black; text-align : right; font-weight :
bold}
td.ncollogor {background-color : #ffffcc; color : black; text-align : left; font-weight :
bold}
td.ncollogoc {background-color : #ffffcc; color : black; text-align : center; font-weight :
bold}
table.ncoltable1 { background-color: #ffffcc; }
table.ncoltable2 { background-color: #ffffcc; border: 2 solid green; }
table.ncoltable3 { background-color: #ffffcc; }
-->
</style>
    
```

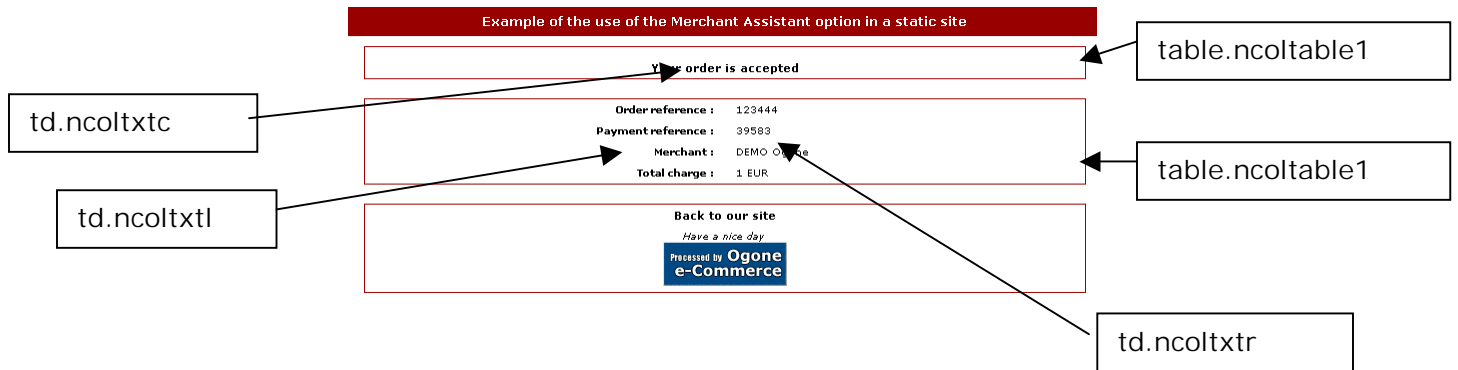
Fill the space between the brackets with your own layout instructions. You must follow the cascading style sheet syntax. We strongly advise you to test it with various browsers as the way they handle style may differ greatly.

Example of the use of the Merchant Assistant option in a static site



input.ncol      td.ncollogol





## 5.6 Payment (step 6)

When all the information is available, Ogone e-Commerce requests the payment or the authorization immediately from the acquirer, or simply confirms it and records it to process it later.

That behavior depends on two parameters that the merchant defines in the administration module (See Document I a: **General Documentation**).

### 5.6.1 Configuring the credit card payment method

In the administration module choose **payment method, credit card maintenance**.

Processing:

- Online (immediate), the payment is processed while the buyer is connected,
- Offline (scheduled) the payment request is recorded and treated later.

Payment procedure:

- direct sale,
- authorization and manual data capture,
- authorization and automatic data capture after x days.

## 5.7 Post-payment feedback (step 7)

This allows the e-commerce application of the merchant to automate the processes for status updating, email confirmation to the buyer, stock management and immediate downloading of intangible goods.

This step takes place:

- either as a background task, a few moment after the end of the payment process,
- or "online" between the moment when Ogone e-Commerce receives the answer from the acquirer and the moment when it notifies the buyer of the payment result.

In this last case, the payment process is longer for the buyer, but the merchant might send an individual response to the buyer, for instance show the key of a downloaded software.

The timing of this request is defined in the page **Account / technical information, point 4.2** of the management module.

An email is always sent to the merchant if the payment is authorized online. Otherwise, the email is sent later on, when payment is processed offline (see hereafter).

### 5.7.1 Automating back-office tasks

To automate your back office tasks, you can define in the **Technical information** (see **Ogone e-Commerce Administration Menu/Account/Technical information**) the URL of 2 executable pages of your site: a post-sale page and a cancel-deney page.

Once the sale process is complete, Ogone e-Commerce will make an http request to one of those pages, sending as parameter the following data:

```
http://www.abssys.be/post.asp?orderID=99999&currency=EUR&amount=10000&ACCEPTANCE=
.&STATUS=...&NCERROR=...&PAYID=...&PM=...&BRAND=...&CARDNO=...&CCCTY=...&I
PCTY=...&ECI=...&CVCCheck=...&AAVCheck=...&VC=...&IP=...&COMPLUS=...&[PARAMPL
US Fields]=... &SHASIGN=...
```

**orderID** is the order reference number of the merchant.

**currency** is the payment currency ISO code

**amount** is the payment amount. (It is the exact amount, not the amount multiplied by 100 as in the ordering form.)

**PAYID** is the Ogone e-Commerce reference number of the payment.

**ACCEPTANCE** : authorization number of the acquirer.

**STATUS** : numeric status of the payment.

(See <http://www.ogone.com/ncol/paymentinfos1.asp>)

**NCERROR** : gives the error code. For example 50001005.

(See <http://www.ogone.com/ncol/paymentinfos1.asp>)

**PM** : the payment method (CreditCard,...)

**BRAND** : the credit card brand (VISA, MasterCard..)

**CARDNO**: the credit card number masked (XXXXXXXXXXXX1111)

**CCCTY**: the credit card issuer country (only for merchants who activated the Fraud detection option)

**IPCTY**: the country of the buyer's IP address (only for merchants who activated the Fraud detection option)

**ECI**: Electronic Commerce Indicator, gives the status of the 3D-Secure authentication (only for merchants who activated the Fraud detection option) See the appendix for the values and meaning of ECI

**CVCCheck**: result of the CVC check, possible values NO, OK, KO.

**AAVCheck**: result of the automatic address verification, possible values: NO, OK, KO.

**VC**: indicates a virtual card

**IP**: IP address of the payment request.

**COMPLUS**: a parameter sent by the merchant in the payment form

**PARAMPLUS Fields**: a list of parameters/values can be submitted as hidden field in a PARAMPLUS field of the payment form. This PARAMPLUS field is not included in the POSTSALE request parameters but instead, the list of parameters/values submitted before the payment is parsed and resulting parameters are added to the Http request. For instance, if you provided a PARAMPLUS hidden field in the payment form with the following value sessionID=123&ShopperID=896, then, after payment, you won't receive any PARAMPLUS parameter but well both SessionID and ShopperID parameters with their relevant values.

**SHASIGN** : this parameter allows the merchant to check the integrity of the data. It works like the signature mechanism describe in 5.2.2. To use this, the merchant needs to enter a string in the Technical information page, point 4.3 SHA-1 Signature. Ogone creates a string by concatenating the value of the following parameters: orderID + currency + amount + PM + acceptance + status + cardno + alias + payid + NCERROR + BRAND +

stringEnteredInTheTechnicalInfoPage. This string is hashed with the SHA-1 algorithm and the resulting signature sent in the post ale request.

**PARAMVAR** : POSTSALE and CANCELDENY URL's may contain a variable part represented by <PARAMVAR>. When making the post-sale request, the variable part of the URL is replaced by the value of the "paramvar" field. For example, you configured the following URL for the post-sale request: "http://www.abssys.be/<PARAMVAR>/postsale.asp". In the "PARAMVAR" field of the payment form, you sent the value: " YhbRtoiUtThyrft". The URL used by Ogone for the http request will be: "http://www.abssys.be/YhbRtoiUtThyrft/postsale.asp".

This request can use the POST or GET method (configure in Ogone e-Commerce Administration Menu/Account/Technical information, point 1).

After too many rejections by the acquirer, or cancellation by the buyer, the request is made to the "cancel-deney" page (configure in **Ogone e-Commerce Administration Menu/Account/Technical information, 4.1**).

If the payment is authorized or accepted, but also if a temporary problem makes the authorization online impossible or unpredictable, the request is made to the "post-sale" page.

## 5.7.2 Possible values of STATUS and NCERROR

### 5.7.2.1 Postsale page :

**STATUS** = 5 (authorized) or 9 (Paid)

**NCERROR** = 0

Authorization or Payment is accepted

An authorization code is available in the field "ACCEPTANCE"

Status will be 5 if the payment method chosen by the buyer allows 2 phases payment processing and if the merchant has configured his account in order to make "automatic authorization" and "payment on request" or "payment after x days".

**STATUS** = 51 (waiting for authorization)

**NCERROR** = 0

The authorization will be processed offline

It is normal if the merchant has chosen offline processing in his account customisation.

It is due to a timeout if merchant has chosen online processing. In the case of online processing, the merchant can avoid this by configuring his account in order not to allow offline processing if online processing is not possible. See **management module / account / Technical information, point 6**.

**STATUS** = 52 or 92 (Authorization or payment uncertain)

**NCERROR** = 200...

A technical problem occurred during the authorization (payment) process giving an unpredictable result. The merchant must contact the acquirer helpdesk to know the exact status of the payment.

### 5.7.2.2 Cancel-deney page

**STATUS** = 0 (uncompleted or invalid)

**NCERROR** = 500....

More than 10 times, at least one of the payment data was not valid. The NCERROR field gives an explanation of the error (list available at <http://www.ogone.com/ncol/paymentinfos1.asp>).

**STATUS** = 2 (Authorization refused)

**NCERROR** = 300....

The authorization has been refused by the financial institution more than x times (x is a parameter defined in the technical information page, point 5)

The merchant can go to the payment financial history available in the administration module to look at the "payment ticket". When it is available, it can give some more details.

**STATUS** = 1 (uncompleted or invalid)

Cancelled by the buyer.

### 5.7.3 Example of a post-sale executable page on the merchant's site

Here is an example of a post-sale ASP page located on the merchant's server, which updates an order status and sends back an HTML page allowing the software to be downloaded. Of course, the merchant can use the development language he wants. This is just a sample page; we don't give any guarantee about the fact that this sample code works correctly:

```

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

Errormessage=""

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

If len(errormessage) = 0 then
    If (com_DB("montant")<>amount) or
        Strcomp(com_DB("dev"),curency) <> then
        Errormessage = "data don t match"
    End if
End if

If len(errormessage) = 0 then
    If (com_DB("montant")<>amount) or
        Strcomp(com_DB("dev"),curency) <> then
        Errormessage = "data don t match"
    End if
End if

If len(errormessage) = 0 then

```

```

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

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

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

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

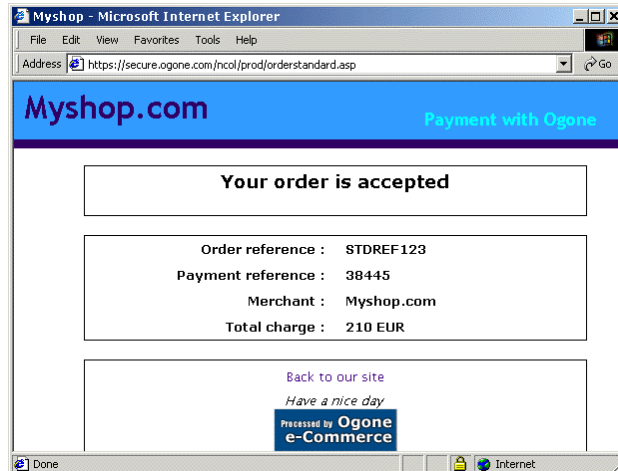
```

## 5.8 Customizing the end of the sale process

When the payment is accepted, or when the buyer cancelled the payment or when the acquirer refused the payment more than the maximum of allowed times, the behavior of Ogone e-Commerce depends of the parameters defined by the merchant. (Step 8/9)

### 5.8.1 Default behavior

When the merchant has specified nothing, Ogone e-Commerce sends to the buyer a standard message: "Your order is accepted" or "Your order is cancelled (refused)". This message is inserted into the template page. Ogone adds a link to the merchant's website and/or the merchant's catalog, using the URL sent with the ordering form or the URL mentioned in the management module.



## 5.8.2 Simple customization - Redirection depending on the payment result

In the ordering form, the merchant may send 4 URL (accepturl, exceptionurl, cancelurl and declineurl) where Ogone e-Commerce redirects the buyer at the end of the payment process.

When the payment is authorized or pending, (STATUS : 5, 51, 9 or 91) the accepturl is used.

When the payment is uncertain (STATUS: 52 or 92), the exceptionurl is used. If the exceptionurl field is empty, the buyer is redirected to the accepturl page.

When the payment (authorization) is refused (STATUS : 2), the declineurl is used.

When the buyer cancel the payment (STATUS : 1), the cancelurl is used. If the cancelurl field is empty, the declineurl is used instead.

For the list of possible status, see <http://www.ogone.com/ncol/paymentinfos1.asp>.

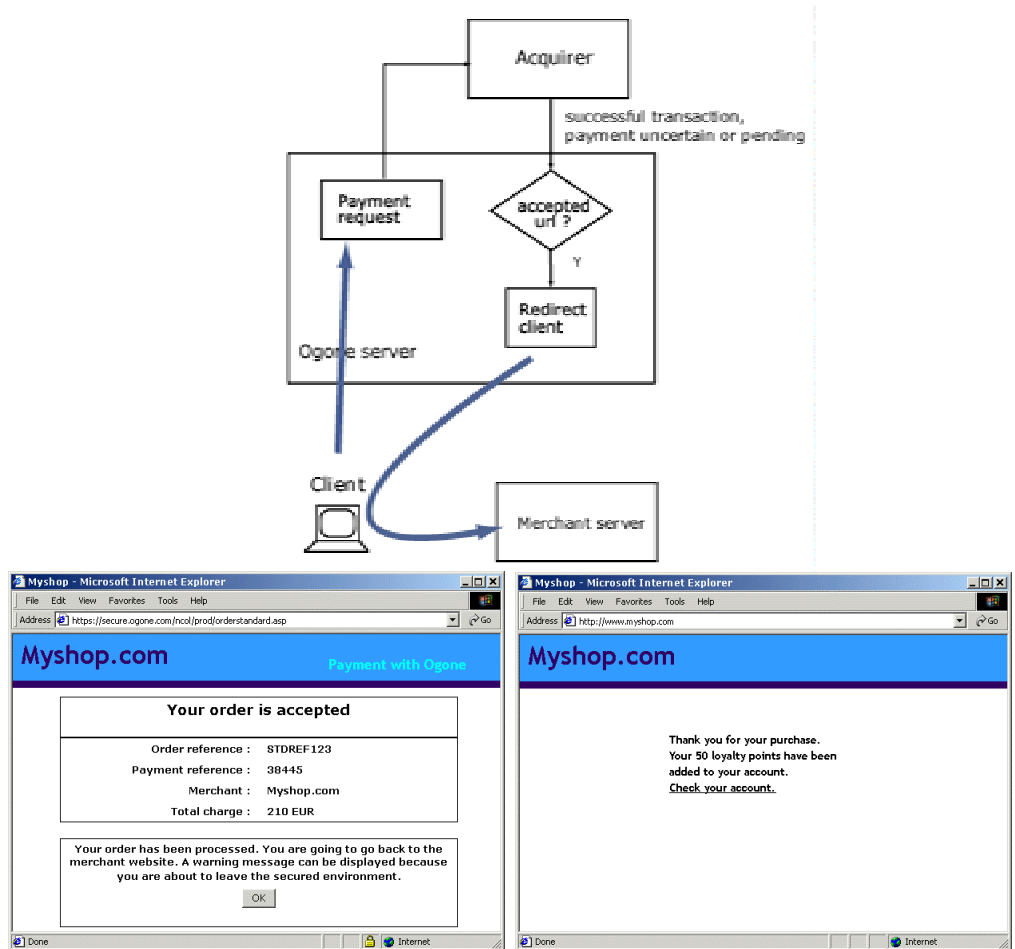
When the buyer goes from a secure site to a non-secure site, a message might appear saying that the information transmitted is not secure any more. This message might worry the buyer if the last screen contained sensitive data.

Ogone e-Commerce can show an intermediary page in order to reassure the buyer.

This parameter is defined in the management module **Account / Technical information, point 5**.

**Important remark:** A redirection process may always be interrupted by the buyer clicking on the stop button of his browser. Thus, it is not a reliable method to collect the result of a transaction.

**Example of the use of an "accepturl" to personalize the buyer response.**



### 5.8.3 Advanced customization

After the payment, Ogone e-Commerce makes an http request to the merchant's website, transmitting the payment data. (See section 5.7)

This process allows the merchant to update his databases but also to generate an individual response for the buyer.

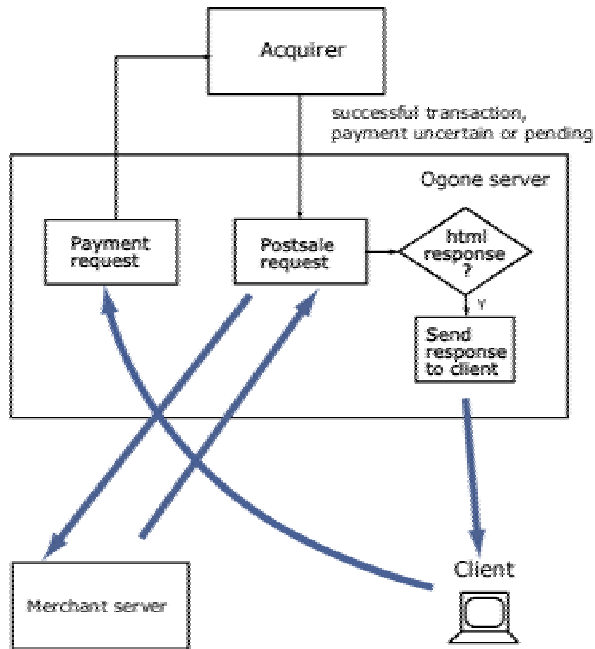
Several possible scenarios:

Ogone e-Commerce detects a redirection while making the post-payment request, and redirects the buyer.

Ogone e-Commerce gets a complete html page (starting with <html> and ending with</html>) in response to the request. Ogone e-Commerce sends "as is" this document. The components included in this page must have absolute URL's.

Ogone e-Commerce gets a few lines of text (no <html> tag) in response to the post-payment request. Ogone sends its standard response, with this text included.

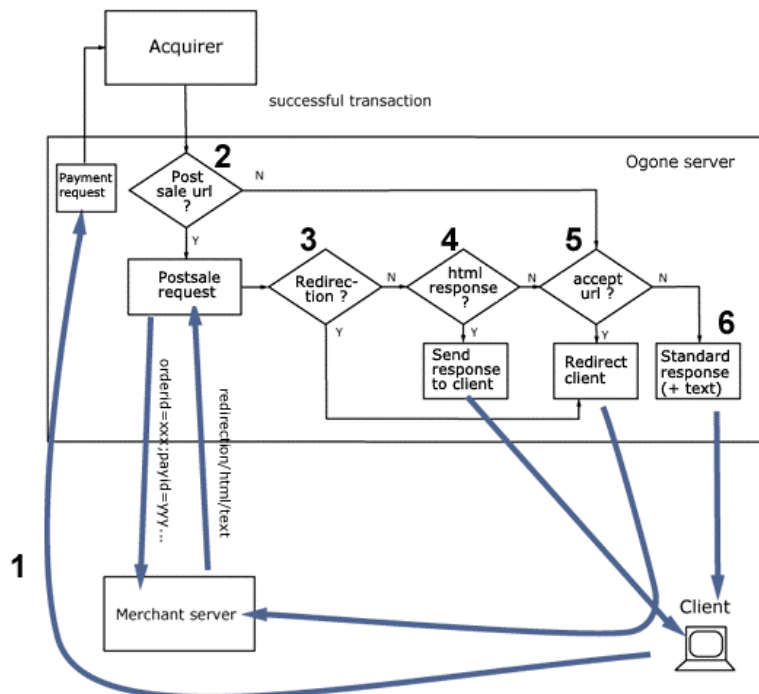
**Example of the use of a post-payment request to generate an individual response.**



Summary of the full process

The diagram below shows the full logical process of the end of a payment, when the payment is authorized or accepted and when there is an immediate post-payment request.

When the payment is cancelled, refused or uncertain the process is similar but the "cancelurl" / "declineurl" / "exceptionurl" and "cancellation/rejection" pages are used instead.



## 5.9 Offline processing

If the merchant has chosen the “scheduled mode” for authorization, the payment is processed offline. In this case, the merchant receives an email when the payment is accepted. He can also logon to the Ogone e-Commerce site to see the latest status of the payments.

If he wants to, he can also be warned of the offline authorization of a payment with an http post or get request. The mechanism is similar to the “post-payment” request mentioned here above, with a different URL but with the same parameters added to the URL (see Document I b: **General Information, advanced functions**).

# 6 Appendix 1

---

## Electronic Commerce Indicator

ECI value	Meaning
1	Manually keyed
2	Recurring payments
3	Installment payments
5	Cardholder identification successful
6	Merchant supports identification but not cardholder, liability shift rules apply
7	eCommerce with SSL encryption
8	eCommerce without any encryption
12	Merchant supports identification but not cardholder, liability shift rules apply (idem 6)
91	Cardholder identification FAILED !!!! (liability shift may apply, check with your acquirer)
92	Issuing bank ACS temporary unavailable, but transaction continued