# Advanced e-Commerce

Technical integration guide for e-Commerce – Version 3.4

PostFinance
SWISS POST

***e-payment.postfinance.ch***

# 1 INTRODUCTION

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

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

# 2 TEST ENVIRONMENT

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

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

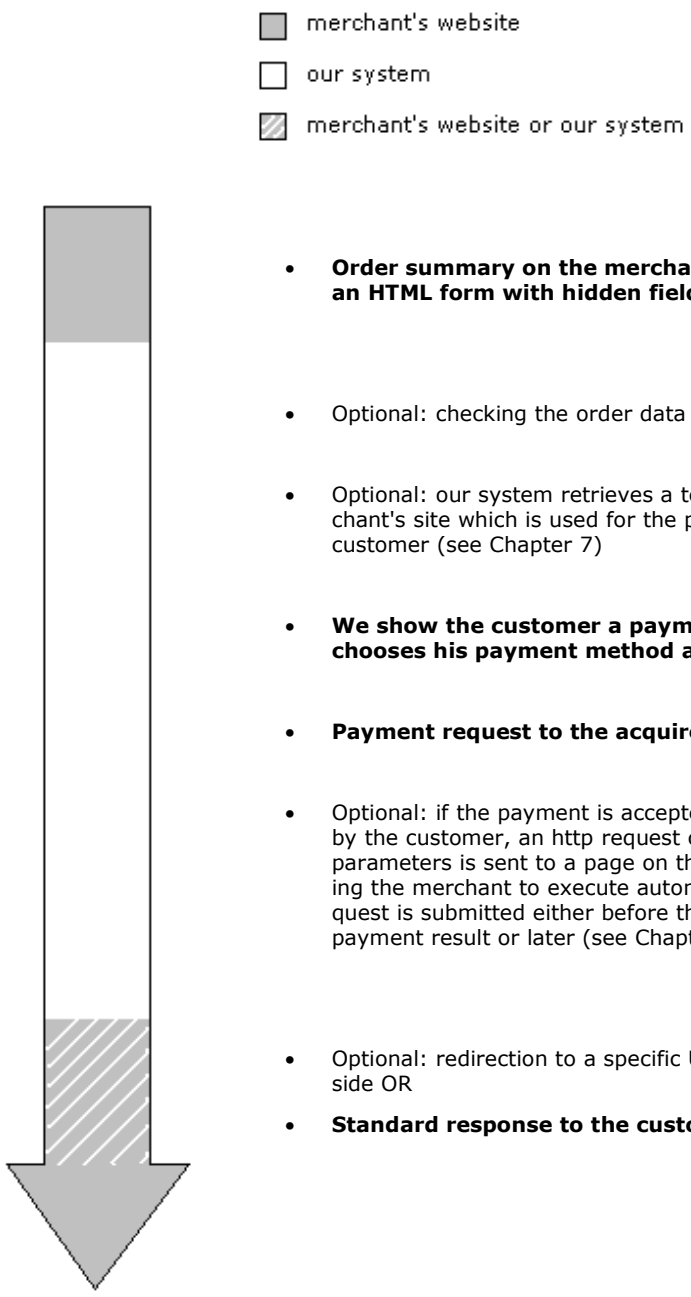
## 2.1 CONFIGURING YOUR TEST ACCOUNT

When you first log into your account, you will have to introduce your e-mail address and change the password. You will also be able to introduce/modify technical details of your test account and create new users. For details on how to create, access and configure your test account, please refer to the **Basic e-Commerce** documentation. The configuration of the technical details will be explained in the following chapters.

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

# 3 SALE PROCESS

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




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

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

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

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

# 4 GENERAL PAYMENT PARAMETERS

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

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

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

Additional payment methods (mainly credit cards) allow either online or offline transaction processing. (See Chapter 4.2)

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

The payment behavior depends on two parameters the merchant defines in items 8 and 9 of the Technical Information page of his administration module: the payment procedure and payment processing. These parameters are set for each account, meaning they apply to all transactions within the merchant's account.

## 4.1 PAYMENT PROCEDURE

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

Three possible payment procedures are available:

- Automatic authorization and data capture on request
- Automatic data capture after x days
- Direct sale

**Automatic authorization and data capture (payment) on request:** our system only requests an authorization when we receive a transaction request from the merchant. The money remains on the customer's account. To request the transfer of the reserved amount to his bank account, the merchant must call up his administration module and request the data capture (payment) for the specific transaction (please refer to the **Back Office User Guide**) or automate the data process by sending us the data captures via batch or via a server-to-server request (please refer to the **Batch** or **DirectLink** information).
The period for which an authorization is valid depends on your acquirer contract.

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

**Automatic data capture (payment) after X days:** the difference between this and the previous payment procedure is that our system requests the data capture automatically after x days.

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

**Direct sale:** the difference between this and the other payment procedures is that our system automatically requests the payment immediately after a successful authorization (VEN operation).

This procedure is often used for goods/services delivered online.

# 4.2 PAYMENT PROCESSING

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

## 4.2.1 ONLINE VERSUS OFFLINE

There are two ways of processing:

- Online (immediate)
- Offline (scheduled)

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

**Offline (Scheduled)**: we register the transaction and process it afterwards (max. 4 hours). This method is faster for the customer since we do not have to send the request to the acquirer immediately (can be used for goods/services that do not need to be delivered online).

## 4.2.2 OFFLINE AS AN OPTION

If you have chosen online processing but do not want to miss out on transactions if the online acquirer clearing system is temporarily unavailable, you can authorize offline processing in those specific circumstances. You can configure this option in items 6 of the Technical Information page.

If you set this option to 'yes' we will store the transactions arriving from your website during the unavailability of your acquirer and will process them offline as soon as the acquirer clearing system is back up again.

If you set this option to 'no', all online transactions will be declined if the acquirer clearing system is unavailable.

You can configure an offline status change notification in item 7 of the Technical Information page of your account. That way, you can be notified by email and/or http request when the status of a transaction changes offline in our system.

# 5 LINK BETWEEN THE MERCHANT'S WEBSITE AND OUR PAYMENT PAGE

## 5.1 ORDER FORM

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

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

### 5.1.1 FORM FIELDS

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

```
<form method="post" action="https://e-payment.postfinance.ch/ncol/XXXX/orderstandard.asp" id=form1 name=form1>
<!-- general parameters: see chapter 5.2 -->
<input type="hidden" name="PSPID" value="">
<input type="hidden" name="orderID" value="">
<input type="hidden" name="amount" value="">
<input type="hidden" name="currency" value="">
<input type="hidden" name="language" value="">
<!-- optional customer details, highly recommended for fraud prevention: see chapter 5.2 -->
<input type="hidden" name="CN" value="">
<input type="hidden" name="EMAIL" value="">
<input type="hidden" name="ownerZIP" value="">
<input type="hidden" name="owneraddress" value="">
<input type="hidden" name="ownercty" value="">
<input type="hidden" name="ownertown" value="">
<input type="hidden" name="ownertelno" value="">
<input type="hidden" name="COM" value="">
```

```
<!-- check before the payment: see chapter 6.2 -->
<input type="hidden" name="SHASign" value="">
<!-- layout information: see chapter 7.1 -->
<input type="hidden" name="TITLE" value="">
<input type="hidden" name="BGCOLOR" value="">
<input type="hidden" name="TXTCOLOR" value="">
<input type="hidden" name="TBLBGCOLOR" value="">
<input type="hidden" name="TBLTXTCOLOR" value="">
<input type="hidden" name="BUTTONBGCOLOR" value="">
<input type="hidden" name="BUTTONTXTCOLOR" value="">
<input type="hidden" name="LOGO" value="">
<input type="hidden" name="FONTTYPE" value="">
<!-- dynamic template page: see chapter 7.2 -->
<input type="hidden" name="TP" value="">
<!-- payment methods/page specifics: see chapter 9.1 -->
<input type="hidden" name="PM" value="">
<input type="hidden" name="BRAND" value="">
<input type="hidden" name="WIN3DS" value="">
<input type="hidden" name="PM list type" value="">
<input type="hidden" name="PMListType" value="">
<!-- link to your website: see chapter 8.1 -->
<input type="hidden" name="homeurl" value="">
<input type="hidden" name="catalogurl" value="">
<!-- post payment parameters: see chapter 8.2 -->
<input type="hidden" name="COMPLUS" value="">
<input type="hidden" name="PARAMPLUS" value="">
<!-- post payment parameters: see chapter 8.3 -->
<input type="hidden" name="PARAMVAR" value="">
<!-- post payment redirection: see chapter 8.2 -->
<input type="hidden" name="accepturl" value="">
<input type="hidden" name="declineurl" value="">
<input type="hidden" name="exceptionurl" value="">
<input type="hidden" name="cancelurl" value="">
<!-- optional operation field: see chapter 9.2 -->
<input type="hidden" name="operation" value="">
<!-- optional extra login field: see chapter 9.3 -->
<input type="hidden" name="USERID" value="">
<!-- Alias details: see Alias Management documentation -->
<input type="hidden" name="Alias" value="">
<input type="hidden" name="AliasUsage" value="">
```

```
<input type="hidden" name="AliasOperation" value="">
<input type="submit" value="" id=submit2 name=submit2>
</form>
```

An example (test page) representing the last page of a merchant's shopping basket, can be found at: https://e-payment.postfinance.ch/ncol/test/teststd.asp.

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

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

### 5.1.2 FORM ACTION

<form method="post" action="https://e-payment.postfinance.ch/ncol/XXXX/orderstandard.asp" id=form1 name=form1>

In the **TEST** environment the URL for the action will be https://e-payment.postfinance.ch/ncol/test/orderstandard.asp.

In the **PRODUCTION** environment the URL for the action will be https://e-payment.postfinance.ch/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://e-payment.postfinance.ch/ncol/prod/orderstandard.asp. If you forget to change the action of your form, once you start in production with real orders, your transactions will be sent to the test environment and will not be sent to the acquirers/banks.

## 5.2 GENERAL PARAMETERS AND OPTIONAL CUSTOMER DETAILS

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

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

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

### 5.2.1 HIDDEN FIELDS

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

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

```
<input type="hidden" name="amount" value="">
<input type="hidden" name="currency" value="">
<input type="hidden" name="language" value="">
<input type="hidden" name="CN" value="">
<input type="hidden" name="EMAIL" value="">
<input type="hidden" name="ownerZIP" value="">
<input type="hidden" name="owneraddress" value="">
<input type="hidden" name="ownercty" value="">
<input type="hidden" name="ownertown" value="">
<input type="hidden" name="ownertelno" value="">
<input type="hidden" name="COM" value="">
```

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

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

# 6 SECURITY: CHECK BEFORE THE PAYMENT

**Best practice:** SHA signature, chapter 6.2

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

## 6.1 REFERRER

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

### 6.1.1 CONFIGURATION

The merchant must fill out/is able to modifiy the referrer/URL of the page containing the order form with the hidden fields in item 2.2 of the Technical Information page in his account.

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

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

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

### 6.1.2 POSSIBLE ERRORS

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

### 6.1.3 LIMITATIONS

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

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

This data check is mandatory for configuring an account in "direct sale" or "Automatic payment (data capture) after x days" (payment procedure, see Chapter 4.1) mode. The data check is not mandatory if the merchant uses a 2-phase – reservation/manual payment request – payment procedure (since he can check the order data before sending the payment request); in the latter case, too, however, **we strongly advise the merchant to perform a data check before payment**.

Two data check possibilities are available: SHA Signature (Chapter 6.2) and an http request to an executable page (Chapter 6.3). We advise use of the SHA signature for the data check since it is faster and consumes fewer system resources than the other method.

## 6.2 SHA SIGNATURE

We recommend using the SHA signature as the data check method. This technique is based on the principle of the merchant's server generating a unique character string, hashed with the SHA1 algorithm, for each order. The result of this hash is then sent to us in the hidden fields of the merchant's order page. Our system reconstructs this signature to check the data integrity of the order information sent to us in the hidden fields. For further details about the SHA signature, please refer to Appendix 1.

## 6.3 HTTP REQUEST TO AN EXECUTABLE PAGE

If the merchant is unable to install the SHA signature, he can perform a data check using an HTTP request based on the XML format. We do not advocate this method as it has its flaws. If you do wish to use this method, please check the FAQ section on our website for further information. Merchants currently using an HTTP request based on the XML format are advised to switch to using an SHA signature.

## 6.4 IP ADDRESS CHECK

Item 2.1 of the Technical Information page only has to be completed if, in addition to his e-Commerce connection, there is a server-to-server connection with our system (i.e. requests on orderdirect.asp, maintenancedirect.asp, querydirect.asp, AFU_agree.asp).

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

# 7 LOOK & FEEL OF THE PAYMENT PAGE

**Best practice:** Static template, Chapter 7.1

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

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

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

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

## 7.1 PAYMENT PAGE LAYOUT (STATIC TEMPLATE)

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

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

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

| Field | Usage | Default value |
|---|---|---|
| TITLE | Title and header of the page | _ |
| BGCOLOR | Background color | white |
| TXTCOLOR | Text color | black |
| TBLBGCOLOR | Table background color | white |
| TBLTXTCOLOR | Table text color | black |
| BUTTONBGCOLOR | Button background color | _ |
| BUTTONTXTCOLOR | Button text color | black |
| FONTTYPE | Font family | Verdana |
| LOGO | URL/filename of the logo you want to display at the top of the payment page next to the title. The URL must be absolute (contain the full path), it cannot be relative.<br>The logo needs to be stored on a secure server (see Chapter 7.3). If you do not have a secure environment to store your image, you can send a JPG or GIF file (and your PSPID) to merchanthelp@postfinance.ch. Please contact our costumer care MerchantHelp, Tel. +41 (0)31 338 24 23, E-Mail: merchanthelp@postfinance.ch to activate the "Logo Hosting" option in your Account. If the logo is stored on our servers, you only need to enter the filename, not the whole URL. | _ |

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

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

# 7.2 TEMPLATE BASED PAGE LAYOUT (DYNAMIC TEMPLATE)

The dynamic template page is an advanced technique for customizing the design of the payment pages. Dynamic template usage is restricted to certain subscriptions. If you are interested in this option and it is not present in the options list of your subscription page in your account, please contact our costumer care Merchanthelp, Tel. +41(0)31 338 24 23, E-Mail: merchanthelp@postfinance.ch.

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

### 7.2.1 HIDDEN FIELDS

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

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

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

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

### 7.2.2 PAYMENT ZONE

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

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

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

***Examples***

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

https://e-payment.postfinance.ch/ncol/template_standard.htm

### 7.2.3 DYNAMIC BEHAVIOR

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

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

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

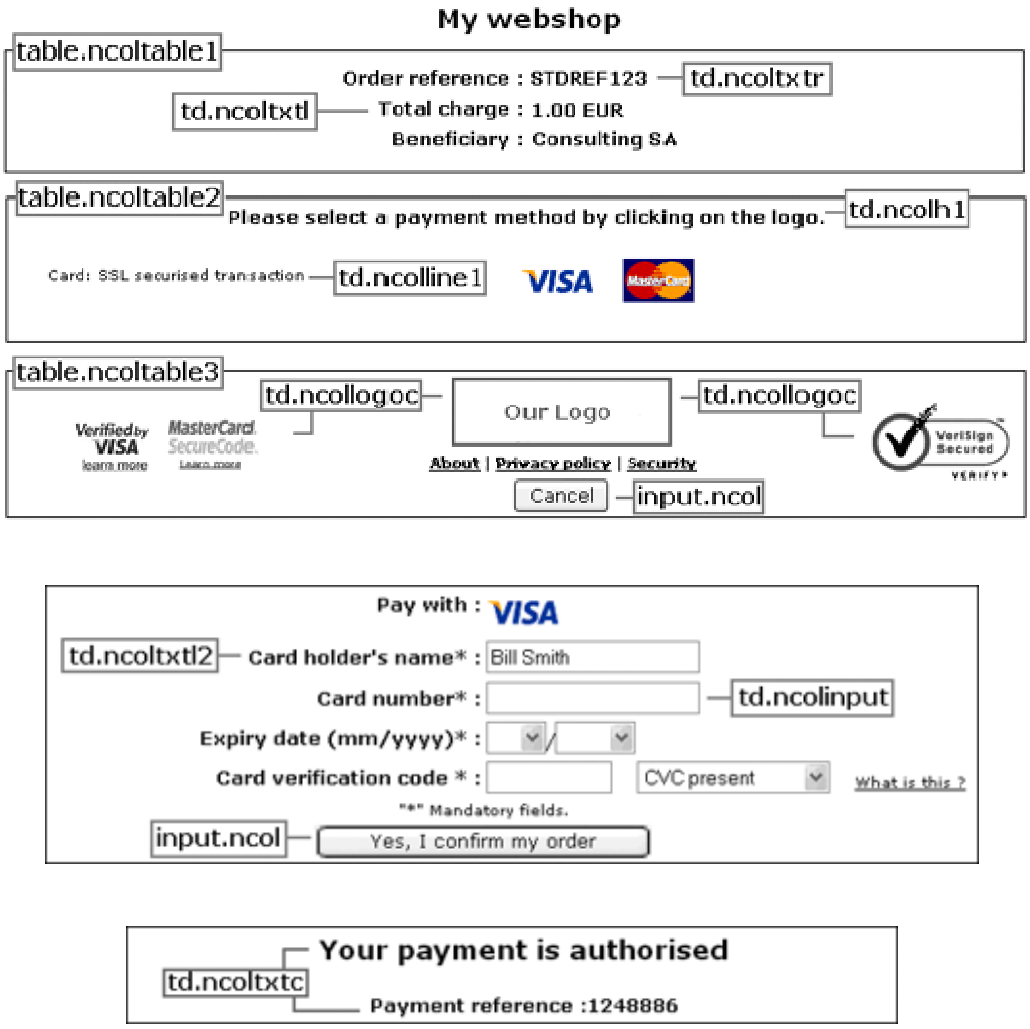
### 7.2.4 STYLE SHEETS

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

We have defined a class for the various types of tables and cells within our tables as well as a class for the submit buttons. Add the following block of code between the tags <head></head> and change the properties of those classes to fit to the look & feel of your site (see the example of the above mentioned template page):

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

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




### 7.2.5 PERFORMANCE

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

Please contact our costumer care Merchanthelp, Tel. +41 (0)31 338 24 23, E-Mail: merchanthelp@postfinance.ch, to change this timeout (HTTPTimeOut).

**Important**: This HTTPTimeOut field has an impact on both dynamic template requests and post sale requests (see Chapter 8.3). Consequently, if the merchant were to decide to change it to e.g. 15 seconds, the post sale 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 costumer care Merchanthelp for a solution if you have high transaction volumes.

# 7.3 SECURE ENVIRONMENT PADLOCK

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

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

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

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

1. Do not use frames for the payment pages: you can refresh the entire screen with a template page that looks as if you are using frames or allow the payment to be processed in a new window.
2. Do not link files to the template page (<link> tag) that you use for the payment page. Instead, use the <style> and <script> tags to include styles and scripts into the template page.
3. Make sure the images in your template are stored on a secure server (the template page can be on a non-secure server, however the images cannot be). We can offer hosting for those elements (see the image hosting options in your account).

# 8 TRANSACTION FEEDBACK TO THE CUSTOMER AND THE MERCHANT

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

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

## 8.1 DEFAULT REACTION

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


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

### 8.1.1 HIDDEN FIELDS

Following are the hidden fields used to transmit the URLs:

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

| Field | Usage |
|---|---|
| catalogurl | (Absolute) URL of your catalogue. When the transaction has been processed, your customer is requested to return to this URL via a button. |
| homeurl | (Absolute) URL of your home page. When the transaction has been processed, your customer is requested to return to this URL via a button. |

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

## 8.2 REDIRECTION DEPENDING ON THE PAYMENT RESULT

In the hidden fields of his ordering form, the merchant can send 4 URLs (accepturl, exceptionurl, cancelurl and declineurl) where our system redirects the customer at the end of the payment process:

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


### 8.2.1 HIDDEN FIELDS

The following hidden fields are used to transmit the URLs:

<input type="hidden" name="accepturl" value="">
<input type="hidden" name="declineurl" value="">
<input type="hidden" name="exceptionurl" value="">
<input type="hidden" name="cancelurl" value="">

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

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

### 8.2.2 BROWSER ALERT NOTIFICATION

When a customer returns from our secure payment pages to the merchant's website, he might get a browser alert, warning him that he is entering a non-secure environment (since he goes from an https:// environment to a http:// environment). When we detect a redirection to the merchant's website, we can display a message to the customer notifying him about the possibility of a warning (see first screenshot in Chapter 8.2), thereby avoiding undue concern about any browser alert. The merchant can activate this option in item 5.2 of the Technical Information page.

### 8.2.3 DATABASE UPDATE OPTION

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

The merchant can activate this option in item 4.5 of the Technical Information page.

**Please note that we additionally always activate the server-to-server postsale request (see Chapter 8, best practice) to avoid inconsistencies between orders and payments due to client interactions (e.g. closing the browser before receiving the authorisation confirmation).**

#### 8.2.3.1 FEEDBACK PARAMETERS

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

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

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

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

***Example***

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

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

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

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

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

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

***Example***

Following are the extra hidden fields sent by the merchant:

<input type="hidden" name="complus" value="12345678912345678912345678 9">

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

Resulting in redirection with feedback parameters:

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

### 8.2.3.2 SECURITY MEASURES

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

#### 8.2.3.3 COMBINATION WITH A POST SALE REQUEST (DEFAULT)

**The merchant is obliged to use – in addition to the feedback parameters sent to the accept-/exception-/cancel-/declineurl - a deferred/background post sale request as a fall back option for the redirection (see chapter 8.3).**

If the communication with the customer is interrupted, for instance when the customer exits his browser window before reaching the accept-, exception-, cancel- or declineurl, the merchant will not receive the redirection on the accept-, exception-, cancel- or declineurl. However, PostFinance enters a post sale URL in item 4.1 of the Technical Information page by default and sets item 4.2 to "Make this request in background and deferred", so the merchant will receive a deferred post sale request shortly after the transaction.

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

# 8.3 DIRECT FEEDBACK REQUESTS (POST SALE)

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

This process, called "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).

## 8.3.1 POST SALE URLS AND PARAMETERS

#### 8.3.1.1 POST SALE URLS

To automate your back-office tasks, you can define the URLs of two executable pages on your site in item 4.1 of the Technical Information page. One of these settings can be the URL where you receive the parameters in a request if the payment's status is accepted, pending or uncertain. The other can be the URL where you want to receive the parameters in a request when the transaction has been cancelled by the client or been declined too many times by the acquirer (i.e. more than the maximum permissible number of payment attempts as set in item 5.1 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 under item 7.2 of the Technical Information page. This is similar to a post sale URL with the difference that it is relevant for potential background processes. You can use the same URL here as the one set in item 4.1 of the Technical Information page, 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. If you want to receive these deferred HTTP requests, set item 7.1 in the Technical Information page to "for every offline status change" and item 7.2 to "by http request" or "by email and http request".

#### 8.3.1.2 VARIABLE POST-SALE URLS

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

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

The hidden field you have to use is the following:

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

| Field | Usage |
|---|---|
| PARAMVAR | The variable part to include in the URLs used for post sale and/or Cancel-Deny requests |

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

***Example***

Post sale 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 sale URL for the transaction:
https://www.yourwebsite.com/shop1/yourpage.asp

#### 8.3.1.3 FEEDBACK PARAMETERS

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

### 8.3.2 POST SALE REQUEST TYPE

In item 4.2 of the Technical Information of your account, you can choose the following request types for the post sale request:

- None: **The merchant is not allowed to set "none" since he is obliged to receive post sale requests (see chapter 8 "best practice" and chapter 8.2.3).**
- Make this request in background and deferred
- Make this request immediately after the payment and use the result to customize the response displayed to the customer (HTML code or redirection)

**Make this request in background and deferred**: the post sale request will be sent shortly after the end of the payment process. The post sale request will be a background task and cannot be used to send a personalized feedback to the customer on the merchant's website.

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

**Make this request** immediately after the payment and use the result to customize the response displayed to the customer **(HTML code or redirection)**: the post sale 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 an personalized response to the customer.

- The disadvantage of the online post sale process is that the merchant's system might be detrimentally affected if there are too many requests to his post sale page (e.g. high per minute transaction volume) – this could result in long response times before customers receive on screen feedback.
- If the merchant requires the online post sale so that he can use the result to tailor the response displayed to the customer, he can configure a fall-back option, should the online request on his post sale page fail. In item 4.2 of the Technical Information page, he can set "Authorize to defer the request if the online request has failed?" to YES. In this case we will retry the post sale request every 10 minutes up to a maximum of 10 times (in the background and deferred). This way the merchant does not miss out on the transaction feedback, should the online post sale request have failed as a result of e.g. temporary server problems at his end. The customer will be displayed the standard transaction feedback from our system (see Chapter 8.1).

### 8.3.3 EXAMPLE OF A POST-SALE EXECUTABLE PAGE ON THE MERCHANT'S SITE

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

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

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

***Example code***

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

Errormessage=""

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

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

```
End if

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

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


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

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

### 8.3.4 RESPONSE TO THE CUSTOMER

We use a possible reply of your post sale page to show a feedback (end of transaction page) to your customer. If your post sale 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 sale process to one of the 4 URLs you may have sent in the hidden fields (accepturl, exceptionurl, cancelurl and declineurl as described in Chapter 8.2).

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

### 8.3.5 POST SALE REQUEST TIMEOUT

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

If the merchant has activated the possibility to receive an offline post sale request in the event of the online post sale request failing, the timeout for the online post sale request in our system will be 10 seconds.

Our costumer care Merchanthelp would be happy to change this timeout (HTTPTimeOut) at the merchant's request.

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

## 8.4 SECURITY: CHECK ORIGIN OF THE REQUEST

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

### 8.4.1 IP ADDRESS CHECK (ONLY FOR POST SALE REQUESTS)

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

### 8.4.2 SHA-OUT SIGNATURE (FOR POST SALE REQUESTS AND REDIRECTIONS)

We strongly recommend that you use an SHA-out 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 1.

# 8.5 CONFIRMATION EMAILS

## 8.5.1 EMAILS TO THE MERCHANT

Our system can send you a payment confirmation email for each transaction (option to configure in item 4.3 of the Technical Information page). The email address we use is the payment confirmation email address in item 1.2 of the Technical Information page.

If you would also like to receive e-mails notifying transaction status changes, you can set item 7.1 in the Technical Information page to "for every offline status change" and item 7.2 to "email".

## 8.5.2 EMAILS TO THE CUSTOMER

Our system can send an automatic email to your customer notifying him of the transaction registration. This is a standard email whose contents cannot be changed. The "From" address used when sending the email is the payment confirmation e-mail address entered in item 1.2 of the Technical Information page. If you have entered more than one email address in this field, we will use the first of them.

You can activate this option in item 5.4 of the Technical Information page.

If you want us to send an email to your customer, you also have to send us his email address in the hidden fields:

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

| Field | Usage |
|---|---|
| EMAIL | Customer's email address |

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

# 9 OTHER OPTIONAL HIDDEN FIELDS

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

## 9.1 PAYMENT METHOD AND PAYMENT PAGE SPECIFICS

### 9.1.1 SHOWING SPECIFIC PAYMENT METHODS

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

The hidden fields are the following:

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

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

| Field | Usage |
|---|---|
| PM | Payment method |
| BRAND | Credit card brand |

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

***Examples***

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

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

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

* Hidden fields in case your customer has selected PostFinance e-finance on your site:

<input type="hidden" name="PM" value="PostFinance e-finance">

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

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

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

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

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

The hidden field is the following:

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

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

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

***Examples***

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

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

### 9.1.2 LAYOUT OF THE PAYMENT METHODS

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

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

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

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

### 9.1.3 3-D SECURE

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

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

The hidden field is the following:

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

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

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

## 9.2 OPERATION

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

You can send us a specific operation code for a transaction if you prefer not to use the same operation code as selected in item 9 of the "Technical Information" page in your account for that transaction.

The operation code you send us in the hidden fields will override the general operation code selected in item 9 of the "Technical Information" page in your account. You can send the operation code in the following hidden field:

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

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

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

**IMPORTANT**: In order for this parameter to be taken into account by our system, it needs to be included in the SHA signature calculation for the transaction. Please refer to Appendix 1 for more information on SHA1.

# 9.3 USER FIELD

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

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

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

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

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

Please refer to the online Parameter Cookbook for other fields.

# 10 APPENDIX 1: SHA1

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

## 10.1 SHA-IN SIGNATURE

This string is built by concatenating the value of the fields *orderID*, *Amount*, *Currency*, *PSPID, operation* from the order form of the merchant and an *additional string* defined by the merchant (we suggest to use a password/pass phrase of minimum 6 characters) in item 3.2 of the Technical Information page "Data checking before the payment, SHA1 Signature". If you asked PostFinance to configure the technical shop information the password/pass phrase will be at least 17 characters. Please note that these values are all case sensitive when compiled to form the string before the hash!

When you hash the string you have composed with the SHA1 algorithm, a 40 character hexadecimal digest will be returned. This outcome should be sent to our system in the hidden fields of your order, using the "SHASign" field.

Our system will concatenate the data received from the merchant's website, adding the password/pass phrase he has entered in item 3.2 of his Technical Information page. We will use the same SHA1 algorithm to hash the string and compare the outcome with the SHASign value the merchant sent in the hidden fields of his order. If the outcome is not identical, the order will be refused. This check ensures the accuracy of the order data.

You can test your SHASign on https://e-payment.postfinance.ch/ncol/test/testsha.asp.

***Example of a basic SHA signature***

orderID: 1234
amount: 15.00 -> 1500
currency: EUR
PSPID: MyPSPID
additional string: Mysecretsig

Complete string to be hashed: 12341500EURMyPSPIDMysecretsig
String after hashing: CC88E974F684C0804FD98BEA2FE403E9D11534BB

SHA1 signature in item 3.2 of the Technical Information -> Mysecretsig
SHASign in the hidden fields of your form->
CC88E974F684C0804FD98BEA2FE403E9D11534BB

***Example of a SHA signature including the operation code***

orderID: 1234
amount: 15.00 -> 1500
currency: EUR
PSPID: MyPSPID
operation: SAL
additional string: Mysecretsig

Complete string to be hashed: 12341500EURMyPSPIDSALMysecretsig
String after hashing: 9B9855F73E30088430C9297364087D0D257F1766

SHA1 signature in item 3.2 of the Technical Information -> Mysecretsig
SHASign in the hidden fields of your form->
9B9855F73E30088430C9297364087D0D257F1766

If the SHASign sent in the hidden HTML fields for the transaction does not match the SHASign which we derived using the details of the order and the additional string (password/pass phrase) entered in item 3.2 of the Technical Information page, you will receive the error message *"unknown order/1/s".*

If the "SHASign" field in the hidden HTML fields is empty but an additional string (password/pass phrase) has been entered in item 3.2 of the Technical Information page, indicating you want to use a SHA signature with each transaction, you will receive the error message *"unknown order/0/s".*

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

| **Field** | **Usage** |
|---|---|
| SHASign | Unique character string for order data validation. A string hashed with the SHA1 algorithm will always be 40 characters long |

# 10.2 SHA-OUT SIGNATURE

We compose the string by concatenating the values of the fields *orderID*, *currency*, *amount*, *PM*, *ACCEPTANCE*, *STATUS*, *CARDNO* (visible card number with xxx), *PAYID*, *NCERROR*, *BRAND*, from the post sale request to the merchant and an *additional string* defined by the merchant (password/pass phrase of minimum 6 characters) in item 4.4 of the Technical Information page.

The merchant's system has to concatenate the same data received in our system's request (please note that the values are all case sensitive when composed to form the string before the hash!) and add the password/pass phrase he has entered in item 4.4 of his Technical Information page. Use the SHA1 algorithm to hash the string and compare the outcome to the SHASign value we sent as parameters in the request. If the outcome is not identical, the feedback parameters might have been tampered with. This check ensures the accuracy of the parameter values sent in the request.

***Example of a basic SHA-out signature***

orderID: 12
currency: EUR
amount: 15
PM: CreditCard
ACCEPTANCE: 1234
STATUS: 9
CARDNO: xxxxxxxxxxxx1111
PAYID: 32100123
NCERROR: 0
BRAND: VISA
additional string: Mysecretsig

Complete string for hashing:
12EUR15CreditCard12349xxxxxxxxxxxx1111321001230VISAMysecretsig
String after hashing: 6DDD8C4538ACD0462837DB66F5EAB39C58086A29

SHA1 signature under item 4.4 of the Technical Information -> Mysecretsig
SHASign we send you in the request ->
6DDD8C4538ACD0462837DB66F5EAB39C58086A29

# 10.3 SHA1 MODULE

To be able to hash a string and send it to us, you must first install an SHA1 module on your server. If you work in a windows 2000/asp environment, you can download a DLL that includes a method to hash a string using SHA1 in the support > documentation page.

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

SHA1 modules can be found in the Internet, so you will not have any problem in finding a suitable one for your server. To help you find a SHA1 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

*.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

# 11 APPENDIX 2: TROUBLESHOOTING

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

*unknown order/1/r*

This error means that the referrer we detected is not a URL the merchant has entered in item 2.2 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 item 2.2.

*unknown order/0/r*

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

*unknown order/1/s*

You will receive this error message if the SHASign sent in the hidden HTML fields for the transaction does not match the SHASign calculated at our end using the details of the order and the additional string (password/pass phrase) entered in item 3.2 of the Technical Information page.

*unknown order/0/s*

You will receive this error message if the "SHASign" field in the hidden HTML fields is empty but an additional string (password/pass phrase) has been entered in item 3.2 of the Technical Information page, indicating you want to use a SHA signature with each transaction.

*PSPID not found or not active*

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

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

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

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

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

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

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

*not a valid currency : …*

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

*The currency is not accepted by the merchant*

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

*ERROR, PAYMENT METHOD NOT FOUND FOR: …*

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

# 12 APPENDIX 3: SHORT STATUS OVERVIEW

The following section contains a non-exhaustive list of statuses; for a full list please refer to:

https://e-payment.postfinance.ch/ncol/paymentinfos1.asp.

| Status | NCERROR | NCSTATUS | Explanation |
|---|---|---|---|
| 5 Authorized | 0 | 0 | The authorization has been accepted.<br>An authorization code is available in the field "ACCEPTANCE".<br>The status will be 5 if you have defined "automatic authorization and data capture on request" or "automatic data capture after x days" as payment procedure in item 9 of the Technical Information page in your account. |
| 9 Payment requested | 0 | 0 | The payment has been accepted.<br>An authorization code is available in the field "ACCEPTANCE".<br>The initial status of a transaction will be 9 if you have defined "direct sale" as payment procedure in item 9 of the Technical Information page in your account. |
| 0 Invalid or incomplete | 500.... | 5 | At least one of the payment data fields is invalid or missing. The NCERROR and NCERRORPLUS fields give an explanation of the error (list available at https://e-payment.postfinance.ch/ncol/paymentinfos1.asp). |
| 2 Authorization refused | 300.... | 3 | The authorization has been declined by the financial institution.<br>The customer can retry the authorization process after selecting another card or another payment method. |
| 51 Authorization waiting | 0 | 0 | The authorization will be processed offline.<br>This is the standard response if the merchant has chosen offline processing in his account configuration.<br>The status will be 51 in two cases:<br>• You have defined "offline" payment processing in item 8 of the Technical Information page in your account.<br>• You allow offline processing if the online acquirer clearing system is temporarily unavailable, in item 6 of the Technical Information page of your account. |
| 91 Payment processing | 0 | 0 | The data capture will be processed offline. |
| 52 Authorization not known<br>Or<br>92 Payment uncertain | 200... | 2 | A technical problem arose during the authorization/payment process, giving an unpredictable result.<br>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.<br>The customer should not retry the authorization process since the authorization/payment might already have been be accepted. |

| Status | NCERROR | NCSTATUS | Explanation |
|---|---|---|---|
| 93 Payment refused | 300.... | 3 | A technical problem arose. |

# 13 APPENDIX 4: E-COMMERCE VIA EMAIL

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

If the email is in HTML format you can use a form with hidden HTML fields to send us the necessary parameters in POST format.
If the email is in plain text format you can append the necessary parameters to the URL in GET format.
(e.g. https://e-payment.postfinance.ch/ncol/test/orderstandard.asp?PSPID=TEST123&orderID=order123&amount=12500&currency=EUR&SHASIGN=8DDF4795640EB9FE9B367315C48E47338129A4F5& …)
Please refer to Chapter 5 for more information.

**IMPORTANT**:

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

- You must leave the referrer/URL field in item 2.2 of the Technical Information page in your account empty in order to avoid "unknown order/1/r" errors.
- You must use an SHA-in signature as the data verification method for the order details. For further details about the SHA signature, please refer to Appendix 1.