SIX CARD SOLUTIONS

Saferpay E-Commerce

Hosting Interface

Date:	**Mar 2009**
Version:	**1.5.9**
Status:	**Released**

TABLE OF CONTENTS

1 INTRODUCTION

This document describes the interfaces of the Saferpay certificate hosting server. Where the Saferpay Gateway is mentioned in the following, this refers to the certificate server.

The Saferpay Gateway is intended to be used if the Saferpay Client component cannot be installed or operated on the target system. This could be the case for some ISP (Internet Service Providers) which do not permit the installation of external software.

1.1 REQUIREMENTS

To be able to use the Saferpay Gateway, the merchant data must be set up on the Gateway. The activation and use of the hosting service is cost free but must be applied for explicitly for every account. To do so please send an e-mail with the AccountID and the request to activate the hosting service for this account to: for Switzerland saferpay.ch@telekurs.com, for the other countries service@saferpay.com.

The Saferpay certificates created for the merchant will be hosted on the Saferpay Gateway.

1.2 SUPPORTED FUNCTIONALITIES

The Saferpay Gateway can take on the following tasks:

- Creation of secured, digitally signed Saferpay URLs (CreatePayInit)
- Verification of digitally signed Saferpay responses (VerifyPayConfirm)
- Release of authorizations for book entry (PayComplete)
- The transfer of attributes on the Saferpay Gateway occurs by means of GET or POST

1.3 RESTRICTIONS

For security reasons, the following functions are only supported by the Saferpay Client component:

- Cancellation of authorizations or book entries,
- Controlling of the batch close,
- Use of the “Card Authorization” interface

1.4 SECURITY

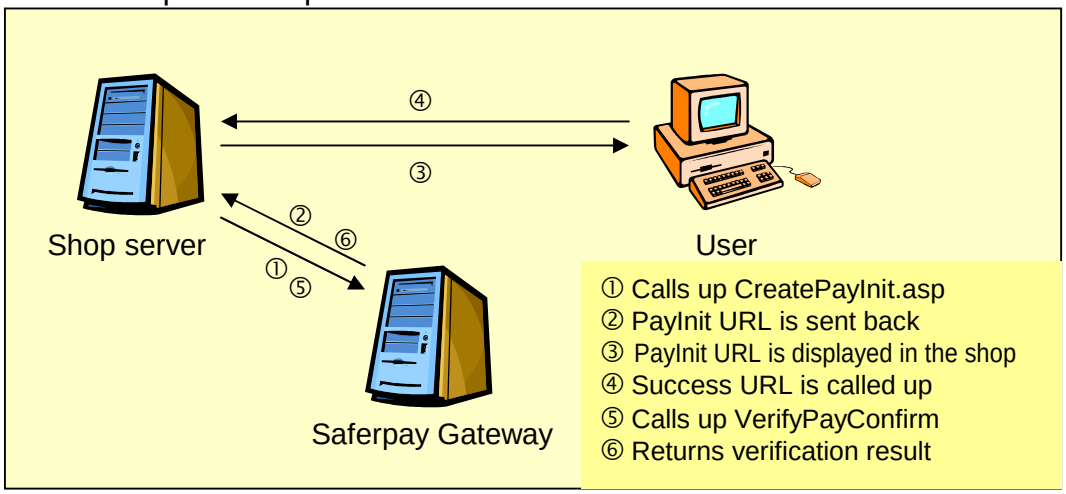
Saferpay stands for security in electronic payment traffic. In the process, Saferpay ensures that the data exchanged with the e-commerce system cannot be manipulated. Manipulations that could be made by savvy Internet users are recognized and reported to the e-commerce system.

The Saferpay Gateway enables the generation of a new Saferpay URL without having the Saferpay component or the Saferpay certificate installed on the shop server. This opens the way for other Internet users to create new Saferpay URLs in the merchant's name.

This in and of itself does not represent a security loophole since only sales made in the name of the merchant can be conducted. It is not possible to carry out credits or cancellations on the merchant account through this route.

2 SERVER-TO-SERVER INTERFACE

A server-to-server communication takes place in this scenario: The shop server sends inquires per http/GET to the hosting gateway, which in turn sends a response in “plain text”.



To create a payment link, the shop server sends the respective payment attributes such as amount, currency, description of sale and Saferpay account to the Hosting Gateway (1). The Gateway creates the payment link and sends it back as text output (2).

The payment link is displayed to the customer, for example, as a [Payment] button. Upon completion of the payment, the customer is steered back to the shop (4).

Once a payment has been successfully made, the retrieved payment data can be verified. The shop server sends the data to the Hosting Gateway for verification (5). The Gateway promptly returns the verification results (6).

The transfer of the attributes to the Saferpay Gateway occurs by means of GET or POST

2.1 ADVANTAGES

The creation of the payment link and the later evaluation of the response data takes place at the shop server, without the customer being aware of.

The PayConfirm response data is sent directly back to the Web shop and only then verified through the Saferpay Gateway.

The release of a payment occurs by the calling up of PayComplete.

2.2 GATEWAY ADDRESSES

The Saferpay Gateway is accessed through these Web addresses:

https://www.saferpay.com/hosting/CreatePayInit.asp

https://www.saferpay.com/hosting/VerifyPayConfirm.asp

https://www.saferpay.com/hosting/PayComplete.asp

2.3 PROCESS

A transaction basically proceeds according to the following schematic:

1) Generation of the Saferpay URL (CreatePayInit).
2) The authorization is handled by the customer under Saferpay.
3) As a response, SUCCESSLINK is called up with the results.
4) The response should be checked for plausibility/manipulation (VerifyPayConfirm).
5) The authorization must be entered into the books either manually by the Saferpay Backoffice or in an automated manner (PayComplete).

2.4 GENERATION OF THE PAYINIT URL (CREATEPAYINIT)

The e-commerce system transmits the Saferpay attributes to the Saferpay Gateway (per GET or POST). The Gateway creates a digitally-signed PayInit URL and immediately sends the result back in plain text. The thus created PayInit URL will be inserted on the Web site, as a "Pay" link, for example.

Example

In this example, the attributes are sent per GET to the Saferpay Gateway:

https://www.saferpay.com/hosting/CreatePayInit.asp?AMOUNT=…

The result is sent back in plain text without HTML tags:

https://www.saferpay.com/vt/Pay.asp?DATA=%3cIDP%20ACCOUNTID%3d%2299867%2d94913159%22%20ALLOWCOLLECT%3d%22no%22%20AMOUNT%3d%22100%22%20BACKLINK%3d%22%2e%22%20CURRENCY%3d%22DEM%22%20DELIVERY%3d%22no%22%20DESCRIPTION%3d%22Testkauf%20Warenkorb%22%20EXPIRATION%3d%2220010408%2012%3a13%3a50%22%20FAILLINK%3d%22%2e%22%20KEYID%3d%220%2d37217%2dea645c3f3f0911d583d70050da413f31%22%20MSGTYPE%3d%22PayInit%22%20SUCCESSLINK%3d%22%2e%22%20TOKEN%3d%22ea645c5d3f0911d583d70050da413f31%22%2f%3e&SIGNATURE=2f1ec1fa51002817941c22e98b9047422ba9ff8fce8b61dab8208a5aa8c82be7cda02ff8a66930481fc19b16d05e7bcedd2b0e5be98fecad3d48bd43916a502f

Should an error occur, then the response “ERROR” will be sent back along with an error description:

ERROR: Missing AMOUNT attribute

2.5 VERIFICATION OF THE PAYCONFIRM URL (VERIFYPAYCONFIRM)

The same schematic used for the creation of the PayInit URL can also be used with the PayConfirm URL. Upon calling up the SUCCESSLINK, the attributes DATA and SIGNATURE will be sent back to the e-commerce system.

For the verification of DATA, these two attributes are sent to the Saferpay Gateway and responded to with “OK” or “ERROR” as plain text.

Within DATA, all fields are XML-coded, e.g. PROVIDERID and PROVIDERNAME etc. (see Table under 4.1.2.).

2.5.1 VERIFYPAYCONFIRM EXAMPLE

In this example, the attributes DATA and SIGNATURE are sent by GET to the Saferpay Gateway:

https://www.saferpay.com/hosting/VerifyPayConfirm.asp?DATA=…

If the digital signature matches the values from DATA, then the positive verification is displayed with “OK” along with the Saferpay ID and TOKEN:

OK:ID=56a77rg243asfhmkq3r&TOKEN=%3e235462FA23C4FE4AF65…

Should an error occur, then the response “ERROR” will be sent back along with an error description:

ERROR: Possible manipulation

2.6 RELEASE FOR PAYMENT (PAYCOMPLETE)

The entering into the books of an authorization occurs through the calling up of the Gateway's PayComplete URL. The attributes ACCOUNTID and ID must be sent through GET or POST to the Saferpay Gateway.

All other attributes, such as ACTION or AMOUNT are ignored.

The execution status of PayComplete will be responded to by the Saferpay Gateway either as "OK" or "ERROR" in plain text.

Example

Calling up of PayComplete for the releasing of a book entry (GET):

https://www.saferpay.com/hosting/PayComplete.asp?ACCOUNTID=99867-94913159&ID=5sfhmkq3rg54345abcd234&spPassword=XAjc3Kna *

If the operation was carried out successfully, then "OK" is sent back in response.

OK

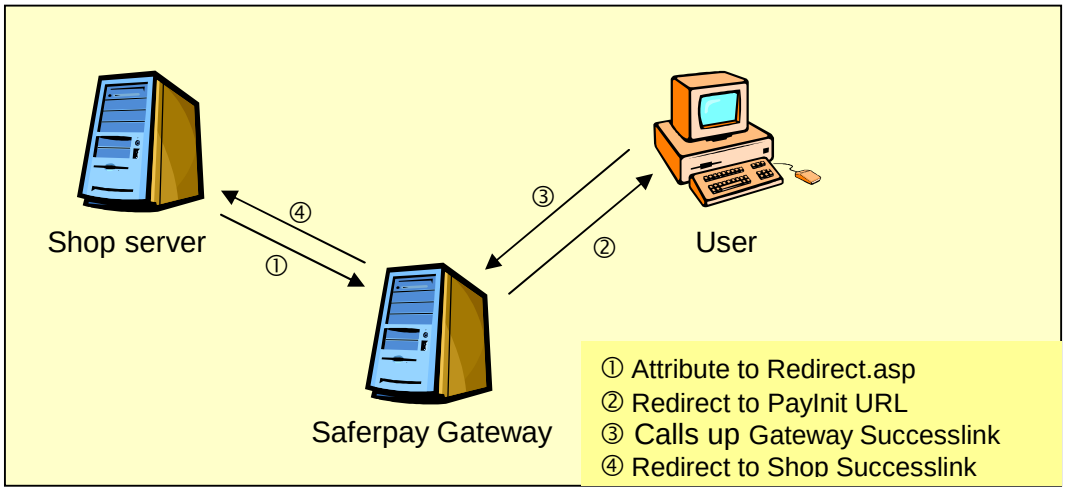
Should an error occur, then the response "ERROR" is sent back along with an error description:

ERROR: Error description

* The transmission of the parameter spPassword=XAjc3Kna is only needed when using the test account. For live accounts it is not to be used.

3 REDIRECT INTERFACE

If the Web server is unable to create the Saferpay URL per HTTP request or to verify the response data per HTTP request, then the Saferpay Gateway can be used by means of "Redirect".



3.1 DISADVANTAGES

The payment data and attributes are placed on the Web site and can thus be manipulated by Internet users. The PayConfirm message can also be altered or simulated.

3.2 GATEWAY ADDRESSES

The Saferpay Gateway is accessed through this Web address:

https://www.saferpay.com/hosting/Redirect.asp

3.3 PROCESS

A transaction using “Redirect” basically proceeds according to the following schematic:

Step 1: Calling up of “Redirect” in order to indirectly call up Saferpay

Step 2: The authorization is handled by the customer under Saferpay.

Step 3: As a response, SUCCESSLINK is called up with the result.

Step 4: The authorization must be entered into the books either manually by the Saferpay Back-office or in an automated manner (PayComplete).

3.4 REDIRECT FROM PAYINIT

The attributes needed for payment are transmitted via GET or POST to the Saferpay Gateway. The Gateway generates a Saferpay URL, which will be immediately called up via "Redirect".

In the attribut SUCCESSLINK, additional parameters can be inserted, such as a session ID.

3.5 REDIRECT FROM PAYCONFIRM

The Saferpay Server’s authorization response (PayConfirm) is primarily transmitted to the Saferpay Gateway. The Gateway checks the response for plausibility and manipulation (VerifyPayConfirm) and transmits the results to the original SUCCESSLINK. This occurs automatically by means of calling up “Redirect.asp”.

The result of the verification is displayed in the RESULT attribute, which contains either “0” for success or any other value for manipulation or error.

The relevant data-fields, which are delivered back by Saferpay by SUCCESSLINK, including ID, TOKEN, AMOUNT, CURRENCY, PROVIDERID, PROVIDERNAME, ACCOUNTID or other optional parameters as listed in in chapter 4.1.2. In case of error, these Saferpay attributes will not be sent back.

Transmission of the result attributes occurs via GET.

Example

In this example, the attributes are sent via GET to the Saferpay Gateway:

https://www.saferpay.com/hosting/Redirect.asp?AMOUNT=1095&…

The Internet user's browser will be instantly forwarded to the Saferpay site via "Redirect". The authorization takes place there using the preferred payment means.

In case of a positive authorization result, the success page will be called up on the Saferpay Gateway. Verification of the digital signature takes place there.

The result attributes are forwarded via GET to the SUCCESSLINK, whereby the positive result of the verification will displayed with "RESULT=0".

http://www.shop.de/kasse_ok.pl?Session=123&RESULT=0&ID=fasmm...

In case of an error the field "RESULT" will be sent back with a value that does not equal 0:

http://www.shop.de/kasse_ok.pl?Session=123&RESULT=1

4 PARAMETERS

Parameters are divided in parameters used in the payment link (PayInit) and parameters of the response to the SUCCESLINK.

4.1 PAYMENT LINK

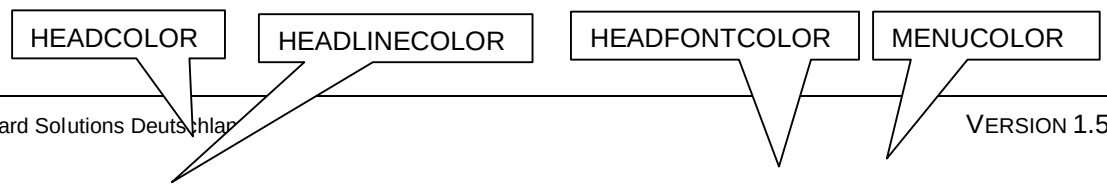
The following parameters can be used to create the payment link.

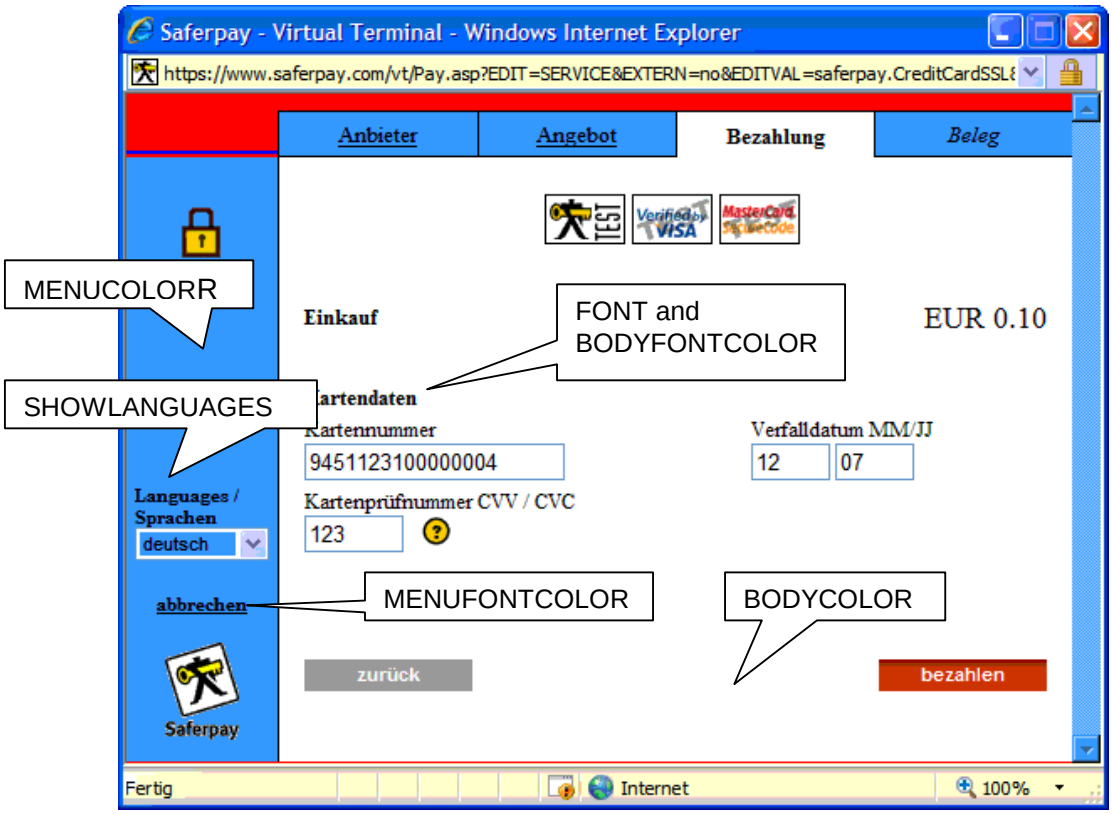
Parameter	Description
ACCOUNTID	The merchant's Saferpay account number Example: *99867-94913159*
AMOUNT	Payment amount in the smallest currency unit Example: *1295* stands for 12.95 EUR
CURRENCY	3-digit currency code Example: *EUR*, *USD*, *GBP*, *CHF*, etc.
DESCRIPTION	Sales description (URL encoded) Example: *This%20is%20a%20purchase*
CCCVC	Optional: Request the card verification number (CVV/CVC2) if set to "yes"
CCNAME	Optional: Request the cardholder or account holder if set to "yes"
ORDERID	Optional: Merchant reference number with a maximum of 12 digits Example: *2307020034*
SUCCESSLINK	Optional: Web site (shop URL), which is to be called

Parameter	Description
	up upon successful authorization. Only necessary if the link has not been placed permanently on the hosting server.
BACKLINK	Optional: Web site (shop URL), which is to be called up upon abort by the customer Only necessary if the link has not been placed permanently on the hosting server.
FAILLINK	Optional: Web site (shop URL) that is to be called up if the payment cannot be carried out. Only necessary if the link has not been placed permanently on the hosting server.
NOTIFYURL	Optional: Saferpay sends the result of the a successful authorization or payment directly to this URL (PayConfirm). The response data contains the DATA and SIGNATURE elements as POST parameters. Use VerifyPayConfirm to verify the content and it's digital signature.
DELIVERY	Must be set to "yes" or "no" If set to "yes" an input form for the customer delivery address appears during the Virtual Terminal session.
USERNOTIFY	Optional: Email address of the customer. Saferpay sends a notification message after a successful purchase.
NOTIFYADDRESS	Optional: Email address of the merchant. Saferpay sends a notification message after a successful purchase
AUTOCLOSE	Optional: Number of seconds (0 to n), after which the counter is to be routed automatically to SUCCESSLINK.
PROVIDERSET	Optional: Use this parameter to show the customer specific payment methods. PROVIDERSET must contain a comma delimited list of provider id's. A current list of provider id's can be found here: http://www.saferpay.com/help/ProviderTable.asp.
LANGID	Optional: Specifies the language for the Virtual Terminal session. Possible values are "en" (English), "de" (German), "fr" (French) and "it" (Italian). Per default the Virtual Terminal uses the browsers language setting to determine the dialog language. A recent list of language codes is available athttps://www.saferpay.com/vt/xml/language.xml.
SHOWLANGUAGES	Optional: If set to "no" this option disables the language selector in the menu section of the VT.

4.1.1 ADDITIONAL PAYINIT ATTRIBUTES

Use one or more of the following attributes in the PayInit request to define the new layout or color of the Saferpay VT. This graphic shows the usage of the additional Saferpay PayInit attributes for styling:





Styling Parameter	Description
BODYCOLOR	Optional: Specifies the color of the VT body in HTML format.
HEADCOLOR	Optional: Specifies the color of the header of the VT header.
HEADLINECOLOR	Optional: Specifies the color of the head-line.
MENUCOLOR	Optional: Specifies the color of the menu bar background.
BODYFONTCOLOR	Optional: Specifies the font color of the body area.
HEADFONTCOLOR	Optional: Specifies the font color of the head.
MENUFONTCOLOR	Optional: Specifies the font color of the menu.
FONT	Optional : Defines the font-face used in the VT

4.1.2 SUCCESSLINK

The response data is contained in the DATA parameter as an XML data record. The following parameters will be sent back:

Attribute	Description
ID	Saferpay transaction identification, max. 80 characters. Example: *QlU3CUbQQ8lISAAlQY7lAMW2E52A*
TOKEN	Optional: Reserved
PROVIDERID	Processor's internal identification number
PROVIDERNAME	Processor's name
AUTHCODE	The processor's authorization code
ORDERID	Optional: Contains the same merchant reference as in the inquiry message
CONTRACTNUMBER	Optional: The provider's contract number
ECI	Optional: *Electronic Commerce Indicator* 0 = SSL-Transaktion (keine Haftungsumkehr) 1 = 3-D Secure-Transaktion, voll authentifiziert (Haftungsumkehr) 2 = 3-D Secure-Transaktion, nicht authentifiziert (Haftungsumkehr)
CAVV	Optional: 3-D Secure *Cardholder Authentication Verification Value* (MasterCard UCAF-Wert)
XID	Optional: 3-D Secure *Transaction Identifier*
IPCOUNTRY	Optional: 2 digit countrycode origin of the IP Adress (ISO3166), Example: DE, CH. AT… If the country can't be identified the value „IX" is returned.
CCCOUNTRY	Optional: 2 digit countrycode origin of the Creditcard (ISO3166), Example: DE, CH. AT… If the country can't be identified the value „IX" is returned.

5 INTEGRATION AND PROGRAMMING

The following examples are configured for the ASP programming and should provide greater clarification of the integration of the Saferpay certificate server. Logically, the procedure also applies for other (object-oriented) programming languages.

5.1 REQUIREMENTS

5.1.1 PROCESSING OF PLAIN TEXT

"GetURL()" represents an important function that is used when calling up the Gateway sites CreatePayInit.asp, VerifyPayConfirm.asp, xmlCreatePayInit.asp and xmlVerifyPayConfirm.asp.

5.1.1.1 ASP EXAMPLE

This function sends back the (plain text) output of a Web site (strURL):

```
<% Function GetURL (strURL)
      ' Set oHTTP = Server.CreateObject ("MSXML2.ServerXMLHTTP")
      ' oHTTP.SetTimeouts 2000, 2000, 2000, 2000
      Set oHTTP = Server.CreateObject ("Microsoft.XMLHTTP")
      oHTTP.open "GET", strURL, false
      oHTTP.send
      GetURL = oHTTP.ResponseText
      Set oHTTP = nothing
   End function
%>
```

5.1.1.2 PHP EXAMPLE

The (plain text) output of a Web site (strURL) sends back the following line:

```
 $Text = join("", file(strURL));
```

5.1.1.3 VB.NET EXAMPLE

This function sends back the (plain text) output of a Web site (strURL):

```
Imports System.Net
Imports System.IO

Function GetURL(ByVal url As String) as string
  Dim request as HttpWebRequest
  Dim response as HttpWebResponse
  Dim Web stream as StreamReader
  request = WebRequest.Create(url)
  response = request.GetResponse()
  Web stream = New StreamReader(response.GetResponseStream())
  GetUrl = webstream.ReadToEnd()
  webstream.Close()
End function
```

5.1.2 CONTROLLING THE VIRTUAL SAFERPAY TERMINAL WINDOW

Embed the script “OpenSaferpayScript.js” in your Web pages. This script makes the Java function “OpenSaferpayTerminal()" available to you, with which the automatic opening and closing of the Saferpay payment window can be controlled:

```
<SCRIPT SRC="http://www.saferpay.com/OpenSaferpayScript.js"></SCRIPT>
```

5.2 PLAIN TEXT MESSAGES

5.2.1 CALLING UP CREATEPAYINIT.ASP

Creating the Saferpay URL is quite easy. In this example, the necessary transfer values are temporarily saved in the “Attribute” variables and transferred to the Saferpay Gateway.

```
<% Gateway = "https://www.saferpay.com/hosting/CreatePayInit.asp"

   Attribute = "AMOUNT=100&ACCOUNTID=99867-94913159&CURRENCY=DEM"
   Attribute = Attribute & "&DESCRIPTION=Saferpay test purchase"
%>
```

The function “GetURL()” is carried out to create the digital signature and the actual Saferpay URL through the Saferpay Gateway. The result is temporarily saved in the “SaferpayURL” variables:

```
<% SaferpayURL = GetURL (Gateway & "?" & Attribute) %>
```

The contents of “SaferpayURL” will be provided on the Web site according to its intended purpose. In this example, the virtual Saferpay terminal is opened in a new window as soon as the “To checkout…” button is clicked.

```
<input type="BUTTON" value="to checkout..."
 onClick="OpenSaferpayTerminal('<%=SaferpayURL%>',this,'BUTTON')" >
```

Processing of the authorization with Saferpay can now take place …

In this example, the attributes SUCCESSLINK, BACKLINK and FAILLINK were not provided. Therefore, the default entries from the respective Saferpay Gateway configuration are used for these attributes.

5.2.2 USE OF VERIFYPAYCONFIRM.ASP

Upon successful processing of the payment, the Web site provided with SUCCESSLINK will be called up.

In a manner similar to "CreatePayInit.asp", "VerifyPayConfirm.asp" is used to verify the digital signature sent back in SUCCESSLINK. Ascertained first are the DATA and SIGNATURE fields supplied by Saferpay via GET:

```
<% DATA      = Request.QueryString("DATA")
   SIGNATURE = Request.QueryString("SIGNATURE")
%>
```

Verification of the digital signature occurs by calling up the function "GetURL()". The result is temporarily saved in “Result”:

```
<% Gateway = "https://www.saferpay.com/hosting/VerifyPayConfirm.asp"

   Result = GetURL (Gateway & "?DATA=" & DATA & "&SIGNATURE=" &
SIGNATURE)
%>
```

Result contains either “OK:..” or “ERROR:…” Ascertained next is whether the verification of the digital signature could be correctly carried out:

```
<%
   arrayResult = split(Result, ":", 2, 1)
   Status = arrayResult(0)

   if Status = "OK" then
      ' positive verification, extract further attributes from status
      arrayStatus = split(arrayResult(1), "&", -1, 1)
      ...
   otherwise
      ' negative verification result
      ....
   end if
%>
```

Further processing will not be described in greater detail here.

5.2.3 CALLING UP PAYCOMPLETE.ASP

PayComplete.asp can be called up to enter an authorization as a book entry on the Saferpay server. The necessary attributes are:

```
<% Gateway = "https://www.saferpay.com/hosting/PayComplete.asp"

   Attribute = "ACCOUNTID=99867-94913159"
   Attribute = Attribute & "&ID=658mac33291245553ajkq3so"
   Attribute = Attribute & "&TOKEN=2AF5690C23BEF264865A648B5368C882121AA"
%>
```

The function “GetURL()” will be carried out to execute the function “PayComplete” through the Saferpay Gateway. The (plain text) result of the execution will be temporarily saved in the “Result” variables.

```
<% Result = GetURL(Gateway & "?" & Attribute) %>
```

The contents of “Result” must contain “OK” upon positive execution:

```
<% if Result = "OK" then
      ' PayComplete executed
      ...
   otherwise
      ' error during execution
      ...
   end if
%>
```

5.3 EMBEDDING IN HTML

The following scripts explain the calling up of the Saferpay Gateways for the creation of the Saferpay URL.

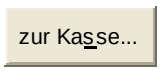
In these examples, SUCCESSLINK, BACKLINK and FAILLINK are not used. Instead, the contents of these links leading from Saferpay Gateway will take over the configuration data placed there.

In order for Saferpay to be opened as a pop-up, the function “OpenSaferpayTerminal()” must be embedded in the in the HTML page via <SCRIPT/>.

```
<script SRC="http://www.saferpay.com/OpenSaferpayScript.js"></script>
```

5.3.1 FORM (POST)

Example FORM: Saferpay opens in the same window

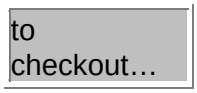



```
<HR>Example FORM: Saferpay opens in the same window<BR>

<form method="post"
       action="https://www.saferpay.com/hosting/Redirect.asp">
   <input type="hidden" name="ACCOUNTID" value="99867-94913159" >
   <input type="hidden" name="AMOUNT" value="100" >
   <input type="hidden" name="CURRENCY" value="EUR" >
   <input type="hidden" name="DESCRIPTION" value="Saferpay test purchase"
>
   <input type="submit" name="SUBMIT" value="To checkout...">
</form>
<HR>
```

5.3.2 LINK (GET)

Example LINK: Saferpay opens as a pop-up window

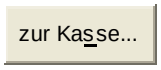



```
<HR> Example LINK: Saferpay opens its own new window<BR>

<TABLE borderColor=white borderColorDark=black bgColor=silver border=1>
  <TR><TD>
    <a href="https://www.saferpay.com/hosting/Redirect.asp?_
             AMOUNT=100&ACCOUNTID=99867-94913159&CURRENCY=EUR&_
             DESCRIPTION=Saferpay test purchase"
       onClick="OpenSaferpayTerminal(this.href, this, 'LINK')">
    To checkout...
    </a>
  </TD></TR>
</TABLE>
<HR>
```

5.3.3 BUTTON

Example BUTTON: Saferpay opens as a pop-up window



To checkout

```
<HR>Example BUTTON: Saferpay opens in its own new window<BR>

<input type="BUTTON"
onClick="OpenSaferpayTerminal(https://www.saferpay.com/hosting/Redirect.a
sp?AMOUNT=100&ACCOUNTID=99867-94913159&_
  CURRENCY=EUR&DESCRIPTION=Saferpay Gateway test purchase', this,
'BUTTON')"
  value="To checkout..." >
<HR>
```