



# **Saferpay – Implementation Guide Programmers Manual**

Date: **May 2007**  
Version: **1.62**  
Status: **Final**

# TABLE OF CONTENTS

<b>1</b>	<b><u>INTRODUCTION</u></b>	<b>4</b>
<b>2</b>	<b><u>CONCEPTS</u></b>	<b>5</b>
2.1	DIGITAL OFFERS	5
2.2	VIRTUAL TERMINAL	5
2.3	SAFERPAY MERCHANT INTERFACE	5
2.4	SAFERPAY APPLICATION COMPONENT	5
2.5	TRANSACTION HANDLES	6
2.6	TRANSACTION BATCHES	6
2.7	RESERVATION TRANSACTION	6
2.8	PAYMENT TRANSACTION	6
2.9	ACCOUNT	6
2.10	USER LOGIN	7
2.11	MULTI-MERCHANT SYSTEMS	7
2.12	TRANSACTION PROCESS	8
<b>3</b>	<b><u>OPERATIONS</u></b>	<b>9</b>
3.1	THE CONFIGURATION PROCESS	9
3.1.1	CONFIGURATION STORE	9
3.1.2	MULTI-MERCHANT SYSTEMS	9
3.2	DIGITAL OFFER	9
3.2.1	PAYINIT ATTRIBUTES	9
3.2.2	ADDITIONAL PAYINIT ATTRIBUTES FOR STYLING	12
3.2.3	RESERVATION	13
3.3	PAYMENT CONFIRMATION	13
3.3.1	PAYCONFIRM ATTRIBUTES	13
3.4	FINANCIAL TRANSFER	14
3.4.1	PAYCOMPLETE ATTRIBUTES	14
3.4.2	EXTENDED PAYCOMPLETE OPERATIONS	15
3.4.2.1	Settlement Action	15
3.4.2.2	CloseBatch Action	15
3.4.2.3	Cancel Action	15
<b>4</b>	<b><u>IMPLEMENTATION OVERVIEW</u></b>	<b>16</b>
4.1	STATIC WEB PAGES	16
4.2	E-COMMERCE APPLICATIONS	16
4.2.1	MULTI-MERCHANT SYSTEMS	18
<b>5</b>	<b><u>PROGRAMMING INTERFACE</u></b>	<b>19</b>
5.1	TRANSACTION PROCESSING	19
5.1.1	CREATING THE DIGITAL OFFER	19
5.1.2	GETTING THE TRANSACTION HANDLE	19
5.1.3	PERFORMING THE FINANCIAL TRANSFER	19
5.2	ADMINISTRATION	20
5.3	IMESSAGEFACTORY INTERFACE	20
5.3.1	OPEN FUNCTION	20

---

5.3.2	CREATEPAYINIT FUNCTION	20
5.3.3	VERIFYPAYCONFIRM	20
5.3.4	CREATEPAYCOMPLETE	20
5.3.5	CREATEREQUEST	20
<b>5.4</b>	<b>IMESSAGEOBJECT INTERFACE</b>	<b>21</b>
5.4.1	SETATTRIBUTE FUNCTION	21
5.4.2	GETATTRIBUTE FUNCTION	21
5.4.3	GETURL FUNCTION	21
5.4.4	CAPTURE FUNCTION	21
5.4.5	EXECUTE FUNCTION	21
<b>5.5</b>	<b>ICONFIGURATIONSETUP INTERFACE</b>	<b>22</b>
5.5.1	SETCONFIGURATION FUNCTION	22
5.5.2	RECONFIGURE FUNCTION	22
<b>5.6</b>	<b>ERROR CODES</b>	<b>23</b>
<b>6</b>	<b>REFERENCE</b>	<b>24</b>

---

# 1 INTRODUCTION

With the evolution of the internet a wide range of payment systems has been invented. From the commercial point of view it is desirable for a merchant in the internet to support as many systems as possible. Unfortunately the integration of such a system into an existing or planned e-commerce application may impose unexpected costs, since each systems has its own API and concept.

Saferpay bundles these payment systems behind a single, generic interface to simplify the task of intergration. Once implemented in an e-commerce application, new payment systems may be activated without modification of the application.

Saferpay is browser-based. The payment operation is distributed over the merchant's web server, the client's browser and the Saferpay web server.

While the merchant defines the conditions of the payment transaction in a digital offer, he delegates the system specific dialogue and API calls to the Saferpay Virtual Terminal. The terminal returns the session back to the web server after the receipt has been displayed.

Transactions are divided in two phases: reservation and financial transfer. Reservations are mostly generated in an interactive session where the consumer selects a payment brand and enters some piece of identification and authorization which is obtained for the transaction. Financial transfer is always performed in a detached process in the background without being affected by network or system failures.

## 2 CONCEPTS

### 2.1 DIGITAL OFFERS

A digital offer is a piece of data digitally signed by the issuing merchant. The offer contains (estimated) transaction amount and description information.

Before a payment transaction can be started a digital offer must be created.

Static digital offers may be created interactively through the browser-based Saferpay Merchant Interface.

The Saferpay Application Component is used to create offers dynamically at the merchant's web server.

### 2.2 VIRTUAL TERMINAL

The Saferpay Virtual Terminal is a HTTPS-based web service hosted by Saferpay. A Virtual Terminal session is initiated by opening an URL that contains the internet address of the Virtual Terminal and a digital offer.

The Virtual Terminal returns a handle to the reservation – the transaction id – acquired during the session if authorization was successful.

### 2.3 SAFERPAY MERCHANT INTERFACE

This web application is a part of the Saferpay site and can be accessed by merchants.

Transaction processing operations may be performed through this interface. This includes creation of digital offer URLs as well as manual reservations and transaction settlements.

The Saferpay Merchant Interface is located at <http://www.saferpay.com/user>.

### 2.4 SAFERPAY APPLICATION COMPONENT

This component is installed on the merchant's web server. It creates and signs digital offers and verifies confirmation messages received from the Virtual Terminal.

The component supports following operations:

- publication and storage of certificates.
- message signing.
- creation of digital offer URLs.
- verification of message signatures.
- sending messages to the Saferpay server.

The component exposes a set of interfaces for transaction processing and administration. The interfaces can be accessed through Java, C/C++ or via the saferpay utility programm.

## 2.5 TRANSACTION HANDLES

Transaction handles are used to identify transactions throughout their lifetime. A transaction handle is created after successful authorization of the payment reservation and may be used to perform total or partial settlement of the reservation as well as total or partial refund of a settled transaction<sup>1</sup>.

Transaction operations are based on a transaction handle passed to the Saferpay server with additional processing parameters. A transaction handle consists of two strings, the ID unique transaction identifier and the TOKEN additional signed data.

The transaction handle may be persisted by the merchant application in order to postpone transaction processing to a later phase.

Active transaction handles are visible in the Saferpay Merchant Interface journal.

## 2.6 TRANSACTION BATCHES

A transaction batch groups a set of transactions together for further processing. The totals printed on the merchants bank statements usually reflect batch totals.

An account always has a current batch. New payment transactions are added to the current batch.

The current batch is closed either manually using the Saferpay Merchant Interface or automatically using the Saferpay Application Component. Closing the batch will trigger further transaction processing.

Payment transactions belonging to the current batch may be cancelled if this operation is supported by the payment media and its processor.

## 2.7 RESERVATION TRANSACTION

A reservation is the first step of a payment transaction. The consumer's account is checked for the requested amount and a transaction handle is returned by the processing institute. A reservation can be turned into a payment transaction within a limited period of time depending on the processor's statements.

## 2.8 PAYMENT TRANSACTION

The payment transaction triggers financial transfer between the consumer and the merchant bank accounts.

## 2.9 ACCOUNT

The existence of a merchant account is an important prerequisite for transaction processing with Saferpay. Each account is identified by a unique string, the ACCOUNTID.

The ACCOUNTID must be specified before digital offers and other payment transaction related operations can occur. It is issued by the Saferpay provider.

An account is assigned to exactly one merchant which has exclusive access to it.

---

<sup>1</sup> Partial settlement and refund may not be supported by all payment systems.

An account is uniquely identified by its ACCOUNTID.

## **2.10 USER LOGIN**

Before a merchant can interact with Saferpay he must identify himself with username and password. Credentials are also required to configure the Saferpay Application Component.

Each user is uniquely identified by his username. One or more User Logins are assigned to a merchant.

## **2.11 MULTI-MERCHANT SYSTEMS**

The Saferpay Application Component offers support for multi-merchant systems. More information about this topic can be found in the Implementation chapter of this document.

## 2.12 TRANSACTION PROCESS

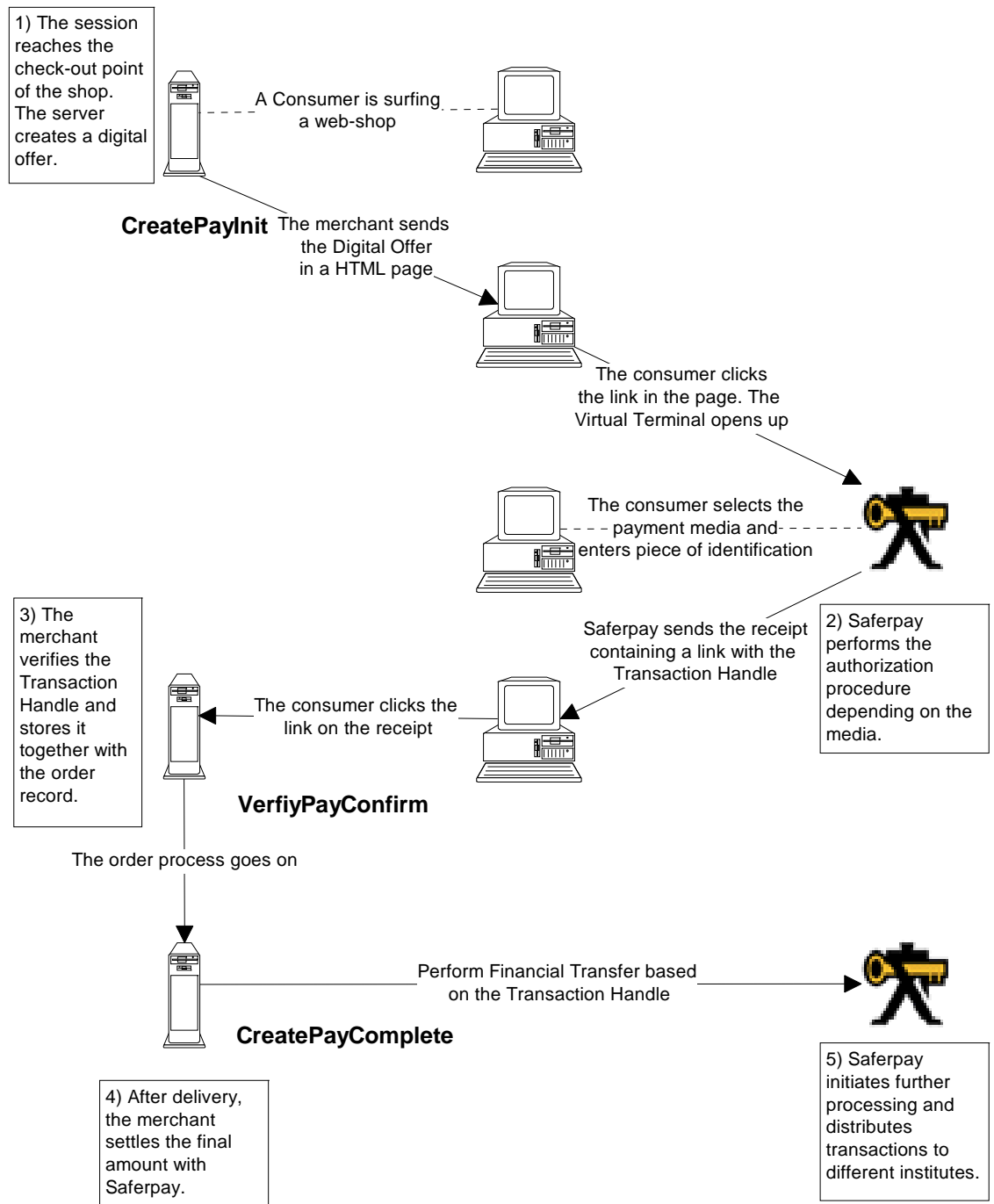


image 1 The basic transaction process

## 3 OPERATIONS

### 3.1 THE CONFIGURATION PROCESS

Before the Saferpay Application Component may be used to process transactions, an initial configuration procedure must be executed for each different merchant planning to make transactions on a system. This will install the Saferpay Server certificate in the configuration and publish the merchant certificate.

To renew the certificates in the system, the "Reconfigure" function is executed.

#### 3.1.1 CONFIGURATION STORE

Stores merchants certificates and keys in a file system directory which is identified by its path. By default Saferpay creates the configuration store in the components installation directory.

#### 3.1.2 MULTI-MERCHANT SYSTEMS

In a multi-merchant system multiple web shops for different merchants are running on the same web server. Since Saferpay requires the digital offers to be signed by the issuing merchant, a separate certificate store will be created for each merchant.

The web server application must specify which store will be used before starting a transaction or configuration operation.

### 3.2 DIGITAL OFFER

The digital offer is the starting point of the transaction, it contains the offered amount and offer description, packed and signed into a data string. Once created this piece of data may be transferred or cached until its expiration date has passed.

A digital offer is always signed by the issuing merchant.

It is created with the **CreatePayInit** function which creates the digital offer and packs it into an URL message called *PayInit* message.

#### 3.2.1 PAYINIT ATTRIBUTES

A set of attributes is defined for digital offers. All attribute values are strings and must be specified unless marked as optional.

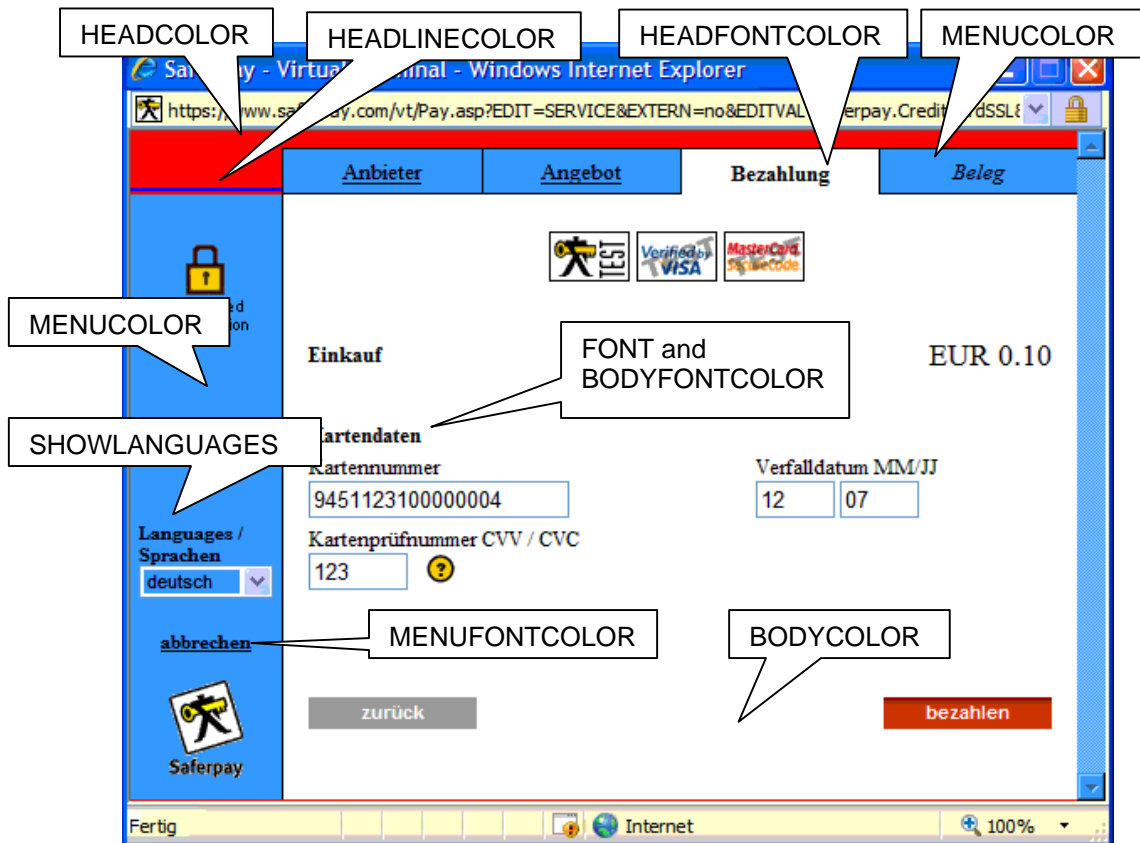
Attribute	Description
Account Information	
<b>ACCOUNTID</b>	Identifies the merchants account at the Saferpay System. An account may be associated with different payment systems. You will receive your account number together with the login

Attribute	Description
	information for Saferpay.
Payment Values	
<b>AMOUNT</b>	The amount to be reserved specified in minor currency unit. E.g. EUR 1.35 must be passed as 135.
<b>CURRENCY</b>	Specifies the transaction currency with a string ISO 4217 currency code, e.g. EUR, CHF etc.
<b>DESCRIPTION</b>	A textual description of the article or offer which will be displayed in the Saferpay Virtual Terminal.
<b>ORDERID</b>	<p><i>Optional</i></p> <p>The ORDERID is a string (max. 80 characters) which can be used by the merchant for identifying the transaction. Saferpay converts the ORDERID to the RetrievalNumber and passes it to the acquirer as your merchant reference number.</p> <p><b>The RetrievalNumber has a well defined format:</b></p> <ul style="list-style-type: none"> <li>- The length is limited to 12 characters.</li> <li>- Allowed characters: a-z A-Z 0-9 . ( + &amp; ! \$ * ) ; - / , % ? : ' = " and whitespace.</li> </ul> <p>This value is returned in the related PayConfirm message.</p>
Navigation	
<b>SUCCESSLINK</b>	<p>An URL identifying the web page to display after the reservation has been successfully completed.</p> <p>The confirmation message will be appended to this URL before it is sent to the merchant's web server.</p>
<b>FAILLINK</b>	This parameter indicates the page to display after a reservation attempt has failed.
<b>BACKLINK</b>	This page is displayed in the client's browser if the transaction is aborted.
<b>AUTOCLOSE</b>	<p><i>Optional</i></p> <p>Specifies the period of time in seconds to close the VT automatically; e.g. 10 closes the VT after 10 seconds and opens SUCCESSLINK.</p>
<b>NOTIFYURL</b>	<p><i>Optional</i></p> <p>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 GET parameters. Use VerifyPayConfirm to verify the content and it's digital signature.</p> <p><i>Note: NOTIFYURL does not support https addresses. Currently only http is supported.</i></p>
Processing Options	
<b>ALLOWCOLLECT</b>	<p>Must be set to "yes" or "no".</p> <p>Specified if multiple offers may be collected in the Virtual Terminal before a reservation is performed on the total amount. If this option is set to "yes", the DELIVERY option must be activated, otherwise the Virtual Terminal will deny the reservation request.</p>

Attribute	Description
<b>DELIVERY</b>	Must be set to "yes" or "no" If set to "yes" a input form for the customer delivery address appears during the Virtual Terminal session.
<b>CCCVC</b>	<i>Optional</i> If set to "yes" the CVC2/CVV2 input field is active.
<b>CCNAME</b>	<i>Optional</i> If set to "yes" the card holder name input field is active.
<b>NOTIFYADDRESS</b>	<i>Optional</i> Email address of the merchant. Saferpay sends a notification message after a successful reservation.
<b>USERNOTIFY</b>	<i>Optional</i> Email address of the customer. Saferpay sends a notification message after a successful purchase.
<b>LANGID</b>	<i>Optional</i> 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.
<b>Other Options</b>	
<b>DURATION</b>	<i>Optional</i> Specifies the duration of the payment link. The value of this parameter must be formatted as YYYYMMDDhhmmss. After the DURATION time the payment link will be declined.
<b>PROVIDERSET</b>	<i>Optional</i> 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 could be found here: <a href="http://www.saferpay.com/help/ProviderTable.asp">http://www.saferpay.com/help/ProviderTable.asp</a> .

### 3.2.2 ADDITIONAL PAYINIT ATTRIBUTES FOR STYLING

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



Attribute	Description
Style Attributes	
<b>BODYCOLOR</b>	<i>Optional</i> Specifies the color of the VT body in HTML format.
<b>HEADCOLOR</b>	<i>Optional</i> Specifies the color of the header of the VT header.
<b>HEADLINECOLOR</b>	<i>Optional</i> Specifies the color of the head-line.
<b>MENUCOLOR</b>	<i>Optional</i> Specifies the color of the menu bar background.
<b>BODYFONTCOLOR</b>	<i>Optional</i> Specifies the font color of the body area.
<b>HEADFONTCOLOR</b>	<i>Optional</i> Specifies the font color of the head.

Attribute	Description
<b>MENUFONTCOLOR</b>	<i>Optional</i> Specifies the font color of the menu.
<b>LINKCOLOR</b>	<i>Optional</i> Specifies the font color of the links of the body area.
<b>SHOWLANGUAGES</b>	<i>Optional</i> If set to "no" this option disables the language selector in the menu section of the VT.
<b>FONT</b>	<i>Optional</i> Defines the font-face used in the VT

### 3.2.3 RESERVATION

Once a consumer decides to purchase an item based on a digital offer, he or she transfers the *PayInit* message containing the digital offer to the Saferpay Virtual Terminal by some means and receives the payment form for the offer.

The Virtual Terminal verifies the merchant's signature on the *PayInit* message and runs the payment reservation dialog.

If the payment is successfully authorized by the media selected by the consumer during the Virtual Terminal session, the transaction handle is passed back to the merchant in the *PayConfirm* message.

The signature of the message is verified by the "VerifyPayConfirm" function.

## 3.3 PAYMENT CONFIRMATION

After a successful authorization the customer will be forwarded to the SUCCESSLINK. The SUCCESSLINK contains the two parameters DATA and SIGNATURE. DATA is a simple XML string and contains the authorization return values. SIGNATURE contains the digital signature which should be verified by the "VerifyPayConfirm" method.

### 3.3.1 PAYCONFIRM ATTRIBUTES

The following attributes are present in the *PayConfirm* message.

Attribute	Description
Transaction Handle	
<b>ID</b>	Unqiue identifier for the reservation. This attribute is limited to 80 alphanumeric characters.
<b>TOKEN</b>	Contains additional information for transaction processing.
Account Information	
<b>ACCOUNTID</b>	Identifies the merchants account at the saferpay system. Same as in the related <i>PayInit</i> message.
Payment Information	
<b>AMOUNT</b>	The reserved amount as specified in the <i>PayInit</i> message.

Attribute	Description
<b>CURRENCY</b>	The currency identifier specified with the related <i>PayInit</i> message.
<b>PROVIDERID</b>	Unique identifier of the payment provider.
<b>PROVIDERNAME</b>	Name of the payment provider.
<b>ORDERID</b>	<i>Optional</i> Merchant reference number for the transaction. This value is copied from the <i>PayInit</i> message if present.
<b>PAYMENTAPPLICATION</b>	Identifier of the payment application.
<b>ECI</b>	<i>Optional</i> Electronic Commerce Indicator: 0 = SSL-Transaction 1 = 3-D Secure transaction, fully authenticated (liability shift) 2 = 3-D Secure transaction, not authenticated (liability shift)

## 3.4 FINANCIAL TRANSFER

Once the merchant has received a valid transaction handle he may decide to perform financial transfer operation related to the reservation until the handle expires. The lifetime of the handle is 90 days by default.

A financial transfer operation is initiated by the "CreatePayComplete" method which returns a *PayComplete* message object.

Sending the *PayComplete* message to Saferpay be repeated if the operation fails due to network or system failure. Since safety and availability of financial transfer is crucial for proper operation, message queuing is the preferred transport for *PayComplete* messages.

### 3.4.1 PAYCOMPLETE ATTRIBUTES

The following attributes apply to the *PayComplete* message:

Attribute	Description
Transaction Handle	
<b>ID</b>	Unique Saferpay transaction identifier for the reservation.
<b>TOKEN</b>	Contains additional information for transaction processing. If not used it should be left empty.
Amount Adjustment	
<b>AMOUNT</b>	<i>Optional</i> Specifies the final settlement amount. The settled amount can be less or equal (default) to the authorized amount.
<b>ACTION</b>	<i>Optional</i> Specifies an extended action like settlement, close batch or cancellation. See below for more details.

Attribute	Description
<b>ACCOUNTID</b>	<i>Optional</i> Saferpay account id, required for the <b>CloseBatch</b> action only.

### 3.4.2 EXTENDED PAYCOMPLETE OPERATIONS

A set of extended operations is available depending on the media used to process the payment. The media may allow partial settlement and refund in which case the ACTION attribute may be specified with the message as one of the following values:

Operation	Description
<b>Settlement</b>	The indicated amount shall be captured on the system.
<b>CloseBatch</b>	Closes the current batch
<b>Cancel</b>	The indicated transaction shall be cancelled on the Saferpay System. This function will fail if the transaction has already been processed. Use the Refund action in this case.  Once a reservation or transaction has been successfully cancelled, it is removed from the system. No further operations will be allowed on the transaction handle.  One a CloseBatch has been performed on an account, settled transactions cannot be cancelled anymore.

#### 3.4.2.1 SETTLEMENT ACTION

Instructs the Saferpay System to turn the indicated reservation into a payment immediately. The payment will be exported to the processor after the next batch close.

To handle situations where the exact amount of the transaction is not known at time of reservation, it is possible to specify an amount less than the originally reserved amount .

The settlement action may be successfully sent multiple times for the same transaction. The final amount of the payment depends only on the amount specified with the first settlement message received by Saferpay.

#### 3.4.2.2 CLOSEBATCH ACTION

This action instructs Saferpay to close the batch of the indicated account. If no account is specified, this function fails.

#### 3.4.2.3 CANCEL ACTION

This function is used to cancel a previous reservation or payment.

The cancel action will fail if the transaction is no reservation or not part of the current batch. If the transaction being cancelled is a reservation, the transaction is removed from the system. If the transaction is a payment transaction, it will remain in the system and be marked as cancelled.

## 4 IMPLEMENTATION OVERVIEW

### 4.1 STATIC WEB PAGES

Paylnit URLs created with the Saferpay Merchant Interface can be embedded into static HTML pages. All information is entered into the web interface.

No programming skills are required for this solution.

Successful reservations are reported to the merchant by email. He can use the Saferpay Merchant Interface to settle the transaction.

### 4.2 E-COMMERCE APPLICATIONS

For completely automated integration of Saferpay into an e-commerce application two web server scripts and one batch process<sup>2</sup> are required to tie the components together.

These three programmes interact directly with the Saferpay Application Component which must be installed and configured on the server running the e-commerce application.

Saferpay supports a range of common programming environments for the web, including:

- Microsoft® Active Server Pages
- JAVA™ Servlets
- CGI
- Cold Fusion
- Perl Scripting
- PHP

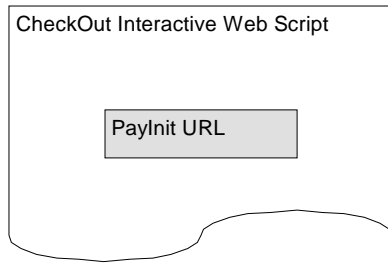
Samples and additional information may be obtained at <http://www.saferpay.com/help>.

A set of common automation interfaces is supported by the component:

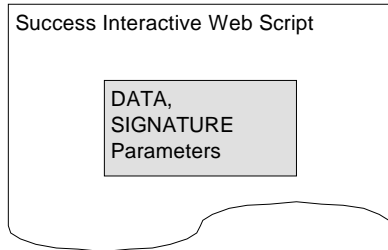
- JAVA™
- Microsoft® ActiveX™
- C/C++
- Command Shell

---

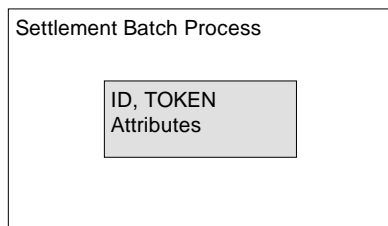
<sup>2</sup> Under Windows NT™ the batch process may be replaced by the Transaction Queue component.



1) The CheckOut script is executed by shop web server. It calls the Saferpay Application Component's CreatePayInit function to create a digital offer based on e.g. the price of the goods in the shopping basket. After setting the required attributes the web server call GetURL to retrieve to link to the Virtual Terminal and embeds it in the generate HTML code.



2) The Success script is executed when the PayConfirm reservation confirmation is received by the merchants web server. It calls VerifyPayConfirm to verify the message and persists the transaction handle.



3) The persisted transaction handle is used to settle the transaction in a background process. The batch calls the CreatePayComplete with the transaction handle, optionally adjusts transaction amount and calls Capture to perform financial transfer.

Image 3 Scripts running on the merchant's web server.

### 4.2.1 MULTI-MERCHANT SYSTEMS

To setup a multi-merchant system first define a root directory for the merchant configuration stores. A subdirectory will be created for each merchant.

Since configuration is stored in a directory tree, it may be backed-up, replicated etc.

## 5 PROGRAMMING INTERFACE

The Saferpay Application Component exposes a set of interfaces for transaction processing and administration.

The interfaces encapsulate the messages exchanged between the participants of an operation. Each message has a set of attributes associated with it.

### 5.1 TRANSACTION PROCESSING

Transaction processing functions are initiated through the **IMessageFactory** interface. It exposes the **CreatePayInit**, **VerifyPayConfirm** and **CreatePayComplete** functions. All of these functions return an **IMessageFactory** object which may be used to manipulate the message attributes and to execute the related operation.

#### 5.1.1 CREATING THE DIGITAL OFFER

A digital offer is created by the **CreatePayInit** function. The resulting message object must be filled with the transaction parameters before calling **GetURL**.

- 1) Create a **MessageFactory** object.
- 2) **Open** the requested configuration.
- 3) Call **CreatePayInit** to receive an empty message object.
- 4) Call **SetAttribute** on the message object to set the transaction parameters.
- 5) Call **GetURL** to receive an URL of the Virtual Terminal for this digital offer.

#### 5.1.2 GETTING THE TRANSACTION HANDLE

The confirmation message containing the transaction handle is parsed and verified by the **VerifyPayConfirm** function.

- 1) Create a **MessageFactory** object.
- 2) **Open** the requested configuration.
- 3) Call **VerifyPayConfirm** to receive the confirmation message object.
- 4) Call **GetAttribute** to retrieve ID and TOKEN attributes.
- 5) Store ID and TOKEN attributes.

#### 5.1.3 PERFORMING THE FINANCIAL TRANSFER

- 1) Create a **MessageFactory** object.
- 2) **Open** the requested configuration.
- 3) Call **CreatePayComplete** with the stored ID and TOKEN attributes.
- 4) Optionally adjust the transaction amount.
- 5) Call **Capture** on the resulting object.

## 5.2 ADMINISTRATION

For administration of multi-merchant systems it may be useful to run the configuration process automatically.

By calling the **Reconfigure** function of the **IConfigurationSetup** interface, the caller may install a new merchant configuration on one or more systems.

## 5.3 IMessageFactory INTERFACE

This interface is exposed by the MessageFactory object.

```
interface IMessageFactory
{
    void Open(String path);
    IMessageObject CreatePayInit();
    IMessageObject VerifyPayConfirm(String data, String signature);
    IMessageObject CreatePayComplete(String id, String token);
    IMessageObject CreateRequest( String msgType);
};
```

### 5.3.1 OPEN FUNCTION

The Open function is used to select the desired merchant configuration. If an empty string is specified the default configuration is selected.

Otherwise the path parameter must specify a file system directory containing a valid configuration store for the merchant requesting the operation. All subsequent function calls on that MessageFactory object will use the selected store.

The Open function must be called before any other function of the message factory object is called.

### 5.3.2 CREATEPAYINIT FUNCTION

Creates a new, blank digital offer. After all mandatory and optional requested attributes have been set, the PayInit URL can be retrieved by the GetURL function.

### 5.3.3 VERIFYPAYCONFIRM

Unpacks and verifies the message contained in the data argument. The data and signature parameters are received with the browser request before step 3 in Image 1.

### 5.3.4 CREATEPAYCOMPLETE

Creates a new PayComplete message. The ID and TOKEN attributes are set according to the id and token parameters of the function call.

### 5.3.5 CREATEREQUEST

Creates an empty request message object with the specified msgType.

## 5.4 IMESSAGEOBJECT INTERFACE

This interface is exposed by the message objects created by the message factory.

```
interface IMessageObject
{
    void SetAttribute(String name, String value);
    String GetAttribute(String name);
    String GetURL();
    void Capture();
};
```

### 5.4.1 SETATTRIBUTE FUNCTION

Sets the value of an attribute of the message.

Attribute names are case-sensitive.

### 5.4.2 GETATTRIBUTE FUNCTION

Retrieves the value of a message attribute. If the attribute is not present in the message, the function fails.

Attribute names are case-sensitive.

### 5.4.3 GETURL FUNCTION

Returns the PayInit URL associated with the message. The function will fail if the message object was not created using **CreatePayInit**.

### 5.4.4 CAPTURE FUNCTION

Performs a synchronous request-response operation to transfer the data in the message object to the Saferpay server.

This function will fail if the message object was not obtained from the message factory using the **CreatePayComplete** function.

### 5.4.5 EXECUTE FUNCTION

Executes a generic request. The data of the message object is transmitted to Saferpay. If a response is received, its data is unpacked and returned from the function call. Otherwise the function fails.

## 5.5 ICONFIGURATIONSETUP INTERFACE

```
interface IConfigurationSetup
{
    void SetConfiguration(String path);
    void Reconfigure(String serverUrl, String userName, String password);
}
```

### 5.5.1 SETCONFIGURATION FUNCTION

Defines the path for the configuration store used for subsequent operations.

### 5.5.2 RECONFIGURE FUNCTION

Creates or updates the configuration store identified with the SetConfiguration call. Specify <https://www.saferpay.com/user/setup.asp> for the serverUrl parameter.

## 5.6 ERROR CODES

The following table shows the possible return/exception codes returned by the Saferpay Application Component functions:

Code	Name	Description
Common Errors		
<b>0x80040201</b>	Context Missing	No configuration context has been specified. Call the Open function before calling any other function.
<b>0x80040202</b>	File Not Found	A file is missing in the configuration. Check if the component was properly configured on the system.
<b>0x80040203</b>	File Access Error	A security violation occurred during file access. Check permissions on configuration directories.
<b>0x80040206</b>	Invalid Path	An invalid path or URL was specified.
<b>0x80040207</b>	Invalid Option	An invalid Option value was specified.
<b>0x80040208</b>	Request Failed	A synchronous server request failed.
<b>0x80040209</b>	Cryptgraphic Error	An invalid key or key-identifier were specified in the message.
<b>0x8004020f</b>	No Configuration	No configuration is present in the path specified by Open.
<b>0x80040214</b>	Verify Failed	The verification of a signature failed because the signature was invalid.
HTTP Errors		
<b>0x80042019</b>	Invalid URL	An invalid URL was specified in a synchronous call to the component.
<b>0x80042021</b>	DNS Error	An error occurred while looking up a hostname. Most likely the reason for this problem will be a DNS configuration problem or missing proxy settings.
<b>0x800421xx</b>	HTTP Server Error	The HTTP Server returned a unsuccessful status code. The errorcode & 0xff will be the status code returned by the server.
Socket Errors		
<b>0x80043000</b>	Socket Base	Base for socket errors. The resulting error code is the base error number plus the socket error code.

## 6 REFERENCE

The documents listed below are integral part of this specification.

No	Description	Creator	Version	Date
1	<a href="http://www.saferpay.com/help">www.saferpay.com/help</a>			
2				
3				