hCO

developer guide

heidelpay payment platform

The complete solution for Ecommerce

Date: 15.11.2012

Version: 2.2

Heidelberger Payment GmbH ● Gaisbergstraße 11-13 ● 69115 Heidelberg ● Germany
Tel +49 (0) 6221 65170 10 ● Fax +49 (0) 6221 65170 12 ● info@heidelpay.de

History of changes

Version	Date	Author	Comment
1.0	18.09.2009	Schuhmann	Document created
1.1	10.01.2010	Schuhmann	Update of addresses
1.2	26.06.2012	Schuhmann	Update of Test Data
1.3	14.08.2012	Fredrich	Minor wording corrections
1.4	28.08.2012	Fredrich	Added new payment methods
1.5	10.09.2012	Fredrich	BillSAFE note added
1.6	11.09.2012	Fredrich	BillSAFE Channel added
1.7	10.10.2012	Fredrich	Updated minor points
1.8	17.10.2012	Fredrich	BillSAFE Flow added and Invoice example
1.9	17.10.2012	Fredrich	Updated minor points
2.0	18.10.2012	Fredrich	Updated minor points
2.1	15.11.2012	Fredrich	Updated minor points
2.2	30.11.2012	Fredrich	BillSAFE flow chart update

Document history

Table of content

Introduction **5**

1 *hCO*: Connecting shops made easy **6**

1.1 *hCO* – Page-flow 6

1.2 Control Flow 9

1.3 Examples 10

2 *hCO* Request **11**

2.1 *hCO* Asynchronous Response 12

2.2 Registration with *hCO* 13

3 *hCO* Customizing the user-interface **14**

3.2 Other Configuration Options 16

4 Special Integration Methods **19**

4.1 PayPal 19

4.2 Paysafecard and Cash-Ticket 21

4.3 Prepayment 22

4.4 MangirKart 23

4.5 BarPay 23

4.6 BillSAFE 24

5 FAQ **29**

A Links and accounts to *hIP* and *hCO* **31**

A Test Data (Credit Cards, Debit Cards, Bank Accounts) **33**

B Sample PHP-Script for calling hCO-page **35**

C Index **40**

List of pictures

Pic 1 *hCO* Page........ 7
Pic 2 Success-page........ 8
Pic 3 Failure-page........ 8
Pic 4 Control Flow of the *hCO*........ 9
Pic 5 Resulting hCO page example........ 15
Pic 6 Merchant Final Payment Confirmation Page........ 20
Pic 7 Paysafecard Preselection Page........ 22
Pic 8 Prepayment Portugal Result Page........ 23
Pic 9 Flow Chart BillSAFE........ 24
Pic 10 Flow Chart BillSAFE........ 25
Pic 11 BillSAFE - Invoice example........ 27
Pic 12 hIP - Finalize (FIN)........ 27

List of tables

Table 1 CSS Styles Parameters........ 16
Table 2 Transaction Parameters handled with BillSAFE........ 28
Table 3 *hIP* (TEST)........ 31
Table 4 *hCO* (TEST)........ 32
Table 5 *hIP* (LIVE)........ 32
Table 6 *hCO* (LIVE)........ 32

Introduction

This document is heidelpay's developer-guide concerning ***hCO*** (heidelpay CheckOut).

A general introduction to heidelpay's payment-system is available as "heidelpay – Integration Basics" – you should be familiar with that document before working with this documentation.

The content of this documentation is as following:

Chapter "*1 hCO: Connecting shops made easy*"gives you a general view on the behaviour and functionality of hCO

Chapter "*2 hCO Request*" explains in details important parameters that can be used with hCO.

Chapter "*3 hCO Customizing the user-interface*" describes the possibilities provided to adopt hCO's user-interface to your needs

Chapter "*4 Special Integration Methods*" Some payment-methods require distinct management and handling. Those payment-methods are described in this chapter.

In the Appendix *A* you can find a sample script for calling the hCO-Checkout-page.

ANNOTATION: This document is dedicated to how to submit transactions to heidelpay's payment-system using ***hCO.*** At some later point in time you may as well be interested in selecting transaction-data from the payment-system automatically. This can be done using heidelpay's "***XML-Query***" which is described in a separate document.

1 *hCO*: Connecting shops made easy

hCO is a relatively simple but very effective method to connect your shop to heidelpay's payment-system. It enables

- merchants within a short timeframe to process real transactions while
- still fulfilling all security precautions requested by credit-card-issuing banks and any kind of financial institutes
- heidelpay simply provides the according infrastructure and processes to be used by you – you do not have to implement them and get certified by issuing banks.

ANNOTATION: In the following chapters it is described how to submit transactions to heidelpay's payment-system. At some later point in time you may as well be interested in selecting data from the payment-system automatically. This can be done using heidelpay's "XML-Query" which is described in a separate document.

1.1 *hCO* – Page-flow

In general processing payments using ***hCO*** is an asynchronous process: you enter your request to heidelpay's system and inform us about the link heidelpay will deliver the result of the transaction

The page flow when using heidelpay CheckOut (***hCO***) is an asynchronous one which consists of visible and invisible pages that are executed on the merchant‟s server as well as on the payment server.

Despite the asynchronous nature of the whole process your customer will never be aware of the complexity and if conducted correctly he or she might not even be aware about the fact that he or she is leaving your page (this can be done by embedding the ***hCO*** into a frame within your shopping cart).

It is required that the response-page of your shop resides on a server which is accesible through standard ports as there are:

80 - in case of http

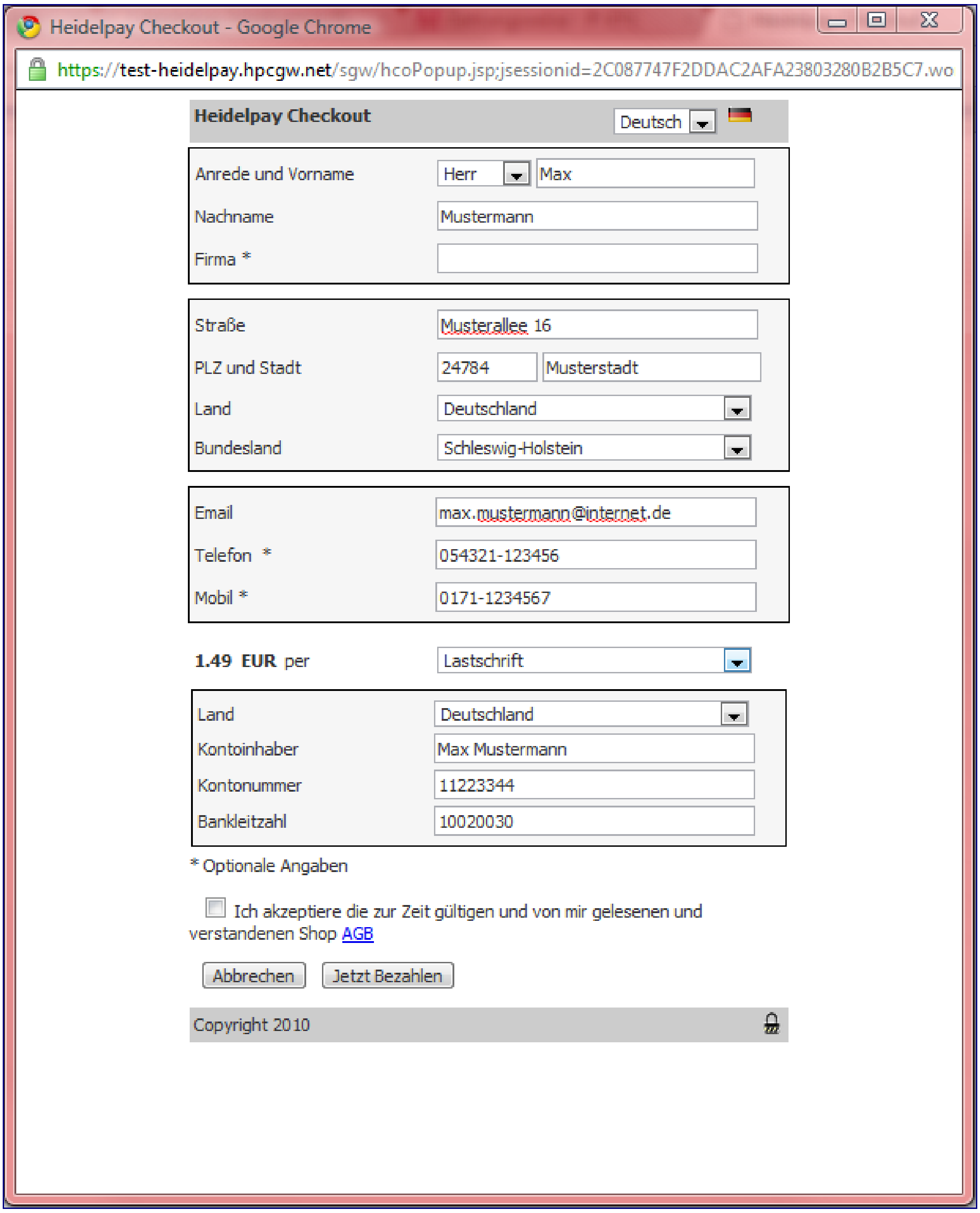
443 - in case of https

<u>IMPORTANT</u>: Be aware that you supply only one response-page to us. You have to analyse our response and show the "success" or "failure"-page respectively to your customer depending on the result received from us !! In fact your response-page has to let the payment-server know where to redirect the user's browser at the end of the payment-process !!

The user usually starts with a shopping cart at your system. You collect the data necessary to fulfill the order, for example the name, shipping address, contact info and of course the product the customeris about to buy.

At the end of that process the customer clicks a button to finish the order.

Upon clicking the "order now" button the shopping cart calls the ***hCO***-call-page on your system. This is a hidden page that evaluates the values already collected in the shopping cart and builds an according ***hCO*** request. This page then redirects the user browser to the ***hCO*** on the payment server. An example of an hCO-page is the following:



https://test-heidelpay.hpcgw.net/sgw/hcoPopup.jsp;jsessionid=2C087747F2DDAC2AFA23803280B2B5C7.wo

Heidelpay Checkout Deutsch

Anrede und Vorname	Herr · Max
Nachname	Mustermann
Firma *	

Straße	Musterallee 16
PLZ und Stadt	24784 · Musterstadt
Land	Deutschland
Bundesland	Schleswig-Holstein

Email	max.mustermann@internet.de
Telefon *	054321-123456
Mobil *	0171-1234567

1.49 EUR per Lastschrift

Land	Deutschland
Kontoinhaber	Max Mustermann
Kontonummer	11223344
Bankleitzahl	10020030

* Optionale Angaben

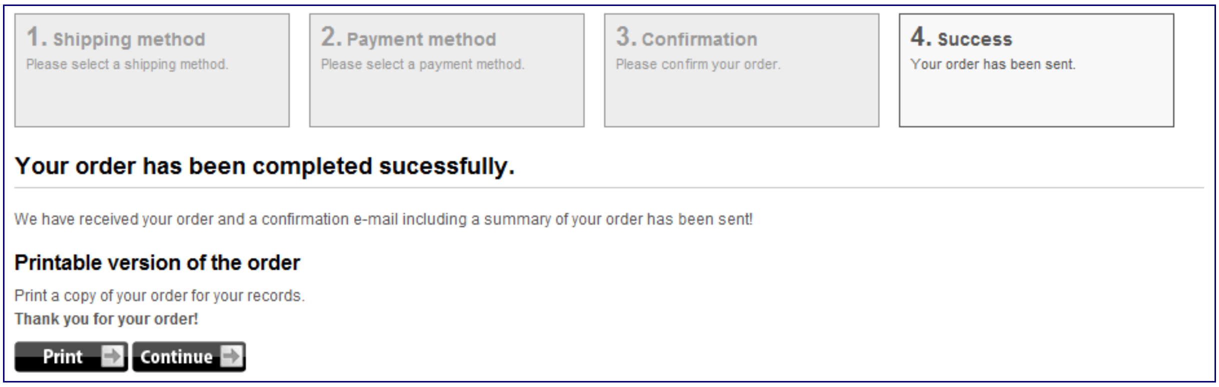
□ Ich akzeptiere die zur Zeit gültigen und von mir gelesenen und verstandenen Shop AGB

Copyright 2010

Pic 1 ***hCO* Page**

Depending on the values the shopping cart has collected already the user must enter some more data (especially of course credit card data or account information).

When the customer has keyed in everything that is necessary he clicks the "pay now" button and the payment server executes the payment request. After finishing the processing the payment server calls the (hidden) response-page on merchant server. The response page decides upon receiving the request which page to show next – a success-page when the transaction was successful (ACK) and a failure-page in case the transaction was not successful (NOK). A success-page might look like this:



1. Shipping method
Please select a shipping method.

2. Payment method
Please select a payment method.

3. Confirmation
Please confirm your order.

4. Success
Your order has been sent.

Your order has been completed sucessfully.

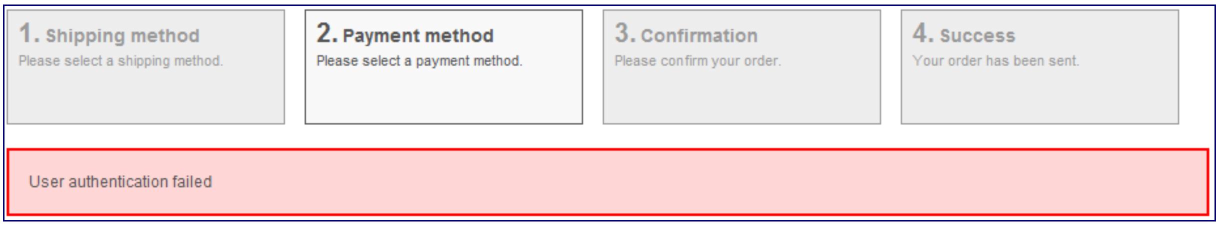
We have received your order and a confirmation e-mail including a summary of your order has been sent!

Printable version of the order

Print a copy of your order for your records.
Thank you for your order!

Pic 2 Success-page

If the payment request failed for some reason the shop must display an error-page like this:



1. Shipping method
Please select a shipping method.

2. Payment method
Please select a payment method.

3. Confirmation
Please confirm your order.

4. Success
Your order has been sent.

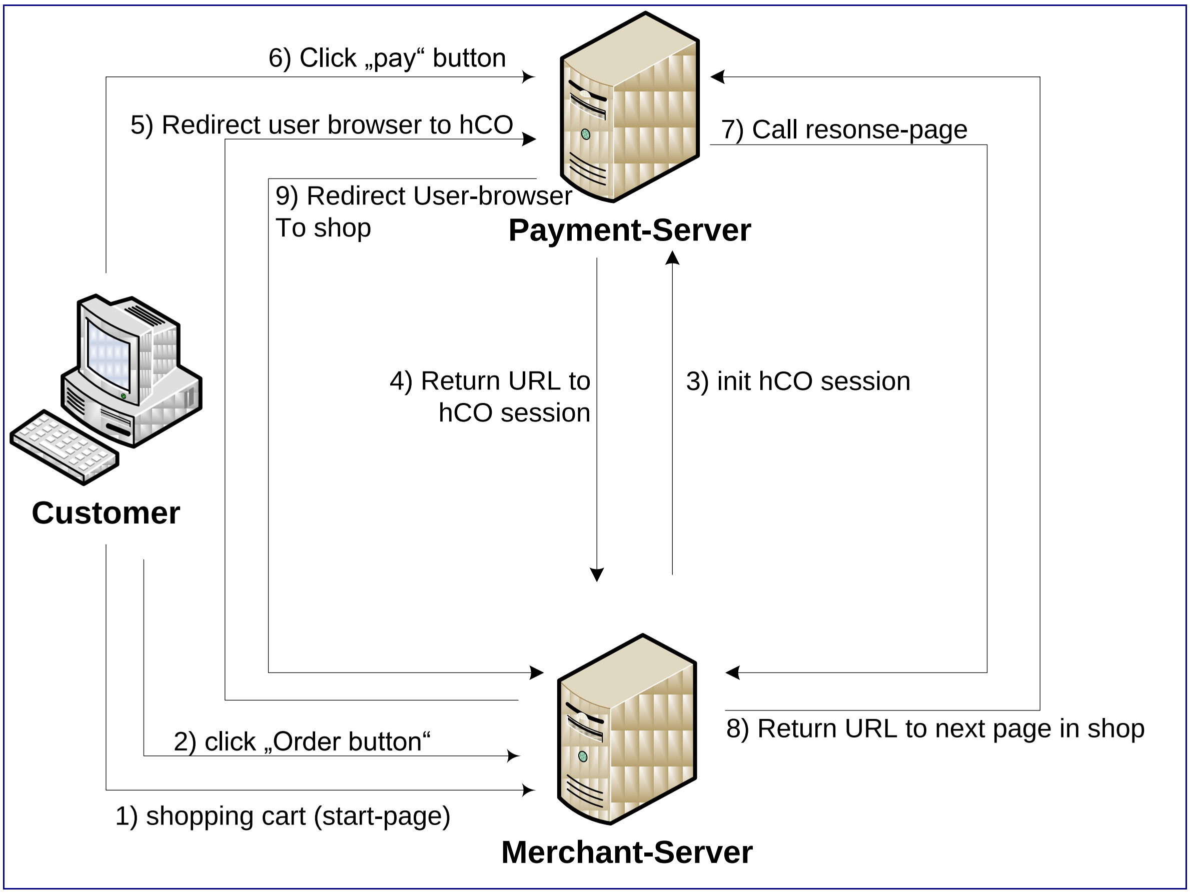
User authentication failed

Pic 3 Failure-page

IMPORTANT: You have to provide two separate result-pages: one for the success-case and for the failure-case which you show the customer accordingly after having analysed the response from heidelpay.

1.2 Control Flow

As explained in the chapter before the workflow for the heidelpay CheckOut (***hCO***) is highly asynchronous where both, the merchant server and the payment server, share the control over the customer browser. During that flow one server acts once as the serving party and the other server as the client before they change roles. The following figure shows which server acts in which role during that workflow.



Pic 4 Control Flow of the *hCO*

1) The user enters the merchant’s shopping cart (the merchant server is in control)

2) The user keys in his personal data (or whatever is required) and puts the products in the shopping-cart. At some point he clicks the "order now" button.

3) At this point the merchant-server calls the ***hCO***-page. This is a hidden page on the merchant-server which creates the initial ***hCO***-request from the data received from the shopping cart. The merchant server sends that request to the payment server at heidelpay.

> Please take care that you define a parameter for the URL of the response-page (which is the page on the merchant server that receives the results of the payment)

4) The payment server receives the initial request and creates a payment session (if all is ok). Then it sends a URL identifying the session back to the merchants server.

5) The merchant server redirects the user's browser to the payment server. Effectively the merchant-server cedes control to the payment server.

6) The user enters now the remaining data necessary to execute the payment (especially the credit card data or the bank account data) and clicks the "pay now" button.

7) The payment server now calls the response-page on the merchant-server and posts the result of the payment to the merchant server. The response-page now evaluates the post parameters.

8) As the result of evaluation of the response parameters the response-page MUST reply with a plain text URL that identifies the next page to be displayed in the user's browser. This page is hosted on the merchant-server.

9) The payment server redirects the user's browser to the URL it received from the previous request. Usually it is a success-page if the payment succeeded otherwise it is an error-page. Effectively the payment server returns control over the user's browser back to the merchant server.

1.3 Examples

The most complex parts within the workflow are the call-***hCO***-page and the response-page. To give you a quick start we have added sample code in the next section. Of course this code will not run out of the box as you have to supply different URLs and login data. Please read carefully the comments within the code.

Important Note:
The Response Page must be situated on a web server running on the default ports.

- HTTP port 80 or
- HTTPS port 443.

2 *hCO* Request

Besides the standard POST parameters that can, but do not have to be submitted, a few frontend specific parameters can be sent to the Web Payment Server.

These parameters are listed in the document “POSTTransactions”

For all transactions replace the following parameters with your test or live parameters:

- `SECURITY.SENDER`
- `USER.LOGIN`
- `USER.PWD`
- `TRANSACTION.CHANNEL`

A typical sample request for the heidelpay CheckOut contains parameters like the following:

```
REQUEST.VERSION=1.0
TRANSACTION.CHANNEL= 31HA07BC81A71E2A47DA94B6ADC524D8
IDENTIFICATION.TRANSACTIONID=1234567890
TRANSACTION.MODE=INTEGRATOR_TEST
PRESENTATION.AMOUNT=123.45
PRESENTATION.CURRENCY=EUR
SECURITY.SENDER= 31HA07BC8124AD82A9E96D9A35FAFD2A
USER.LOGIN= 31ha07bc8124ad82a9e96d486d19edaa
USER.PWD=password
FRONTEND.ENABLED=true
FRONTEND.POPUP=true
FRONTEND.MODE=DEFAULT
FRONTEND.RESPONSE_URL=
http://www.merchantshop.com/paymentResult?jsessionid=12343215413243214213
FRONTEND.LANGUAGE=DE
FRONTEND.LINK.1.KIND=TERMS
FRONTEND.LINK.1.LINK=https://www.merchantshop.com/agb.php
FRONTEND.LINK.1.AREA=EMBEDDED_CHECKBOX
FRONTEND.LINK.1.POPUP_WIDTH=320
FRONTEND.LINK.1.POPUP_HEIGHT=400
```

The Web Payment Server will respond the URL to redirect to this POST request and a validation code of the request.

The result typically looks like this:

```
FRONTEND.REDIRECT_URL=https://test.merchantshop.de/frontend/startFrontend.prc
?jsessionid=12345435643543435&P3.VALIDATION=ACK
```

The list of possible validation codes is available in the document “POST-Transactions”.

The most important parameters (and the most likely you need to change depending on your requirements) are:

FRONTEND.ENABLED Set this to true, in case you are using the ***hCO***. Setting this parameter to false means you are doing direct payment via the POST interface (check document “POST_Transactions” for details)

FRONTEND.POPUP Set this to true to let the ***hCO*** appear inside a popup window, or to false if you want to have it embedded.

PAYMENT.CODE By default, “CC.DB” (Credit Card Debit) will be used for the ***hCO***. In case you want to do a Preauthorization instead of a Debit the code has to be CC.PA. For a registration process use CC.RG. For a complete list of payment codes, check the document “POST_Transactions”.

FRONTEND.RESPONSE_URL The URL where the payment system shall post the payment response of the payment. This is NOT the URL where the end user’s browser will be redirected at the end! The redirect URL at the end must be replied to the call to this URL. This allows the shop to dynamically react on the result of the payment and make the decission where to redirect the user to at the end of the whole process.

In general, you can send any parameter you like (also not documented) and it will be returned to you at the end of the process as part of the payment response message.

2.1 *hCO* Asynchronous Response

The Web Payment Server (WPS) will communicate the payment result back to the merchant’s shop when the user has finished or cancelled the payment process. The target URL where the result is posted to is sent to the hCO in the initial request.

A typical response looks like this:

```
RESPONSE.VERSION = 1.0
TRANSACTION.MODE = INTEGRATOR_TEST
TRANSACTION.RESPONSE = SYNC
TRANSACTION.CHANNEL = 31HA07BC81A71E2A47DA94B6ADC524D8
IDENTIFICATION.TRANSACTIONID = MerchantAssignedID
IDENTIFICATION.UNIQUEID = 402880e5faf35d0700faf35d0cec0002
IDENTIFICATION.SHORTID = 1234.5678.9012
PAYMENT.CODE = CC.DB
PRESENTATION.AMOUNT = 123.45
PRESENTATION.CURRENCY = EUR
PRESENTATION.USAGE = Order Number 1234
CLEARING.AMOUNT = 123.45
CLEARING.CURRENCY = EUR
CLEARING.DESCRIPTOR = 1101.9571.9800 - Order Number 1234
PROCESSING.CODE = CC.DB.90.00
PROCESSING.TIMESTAMP = 2010-01-10 20:46:26
PROCESSING.RESULT = ACK
PROCESSING.STATUS = NEW
PROCESSING.STATUS.CODE = 90
PROCESSING.REASON = Successful System Entry
PROCESSING.REASON.CODE = 00
PROCESSING.RETURN = Request successfully processed in Integrator Test Mode
PROCESSING.RETURN.CODE = 000.100.110
FRONTEND.MODE = DEFAULT
FRONTEND.REQUEST.CANCELLED = false
```

ATTENTION: This is how the response is sent to the merchant-server. Because of the URL encoding of the response, the parameters you receive might be slightly changed. “.” Is converted to “_” e.g. PROCESSING.RESULT becomes PROCESSING_RESULT

The shop has to send a response-URL where to redirect the user back to the shop.

The payment system then redirects the end user’s browser to this URL. This URL must be replied as plain text to the request. The response of the merchant must not contain any other information than this URL:

- no HTML header
- no other text

JUST THE URL ONLY!

Typically it just looks like

```
http://myshop.com/scripts/continue_shopping;jsessionid=32443q54325432
```

In any case the payment server will try to redirect the user to the URL specified in the response.

Therefore this will lead to errors in the end user’s browser if your script contains errors (try to redirect to something that is not a URL)

If you want to receive the account information the user entered in the ***hCO***, send in the parameter “FRONTEND.RETURN_ACCOUNT=true” with your initial request. You will then receive the account parameters (check chapter “Account Group” in the document “POST_Transactions”) in the payment response. Of course you will receive the parameters according to PCI regulations in masked form.

2.2 Registration with *hCO*

If you want to use the ***hCO*** to register customer and account information only, just send in the payment code <Payment Method>.RG (for example CC.RG or DD.RG) with your initial request within the parameter PAYMENT.CODE.

If the user is registered in the system, you can debit his account by simply specifying the UniqueId (parameter IDENTIFICATION.UNIQUEID) you got back in the registration response in future payments. To be able to do so, store the IDENTIFICATION.UNIQUEID together with the user record in your database.

Payment can then be done without the ***hCO*** by simply using the POST transaction interface. You switch to direct payment mode by sending in FRONTEND.ENABLED=false.

The unique id of the registration needs to specified within the parameter ACCOUNT.REGISTRATION.

You can also reregister a customer by sending in the UniqueId of the registration with IDENTIFICATION.REFERENCEID and specifying the payment code <Payment Method>.RR (for example CC.RR or DD.RR with your initial request within the parameter PAYMENT.CODE.

Find more details about payment with the POST Transaction interface in the document “POST Transactions”.

3 *hCO* Customizing the user-interface

There are two parameters that allow you to change the Look & Feel and the behaviour of the ***hCO***:

- FRONTEND.CSS_PATH: to override default style sheet settings
- FRONTEND.JSCRIPT_PATH: allows to do initial setup or changes via Java Script

3.1.1 Add your own CSS

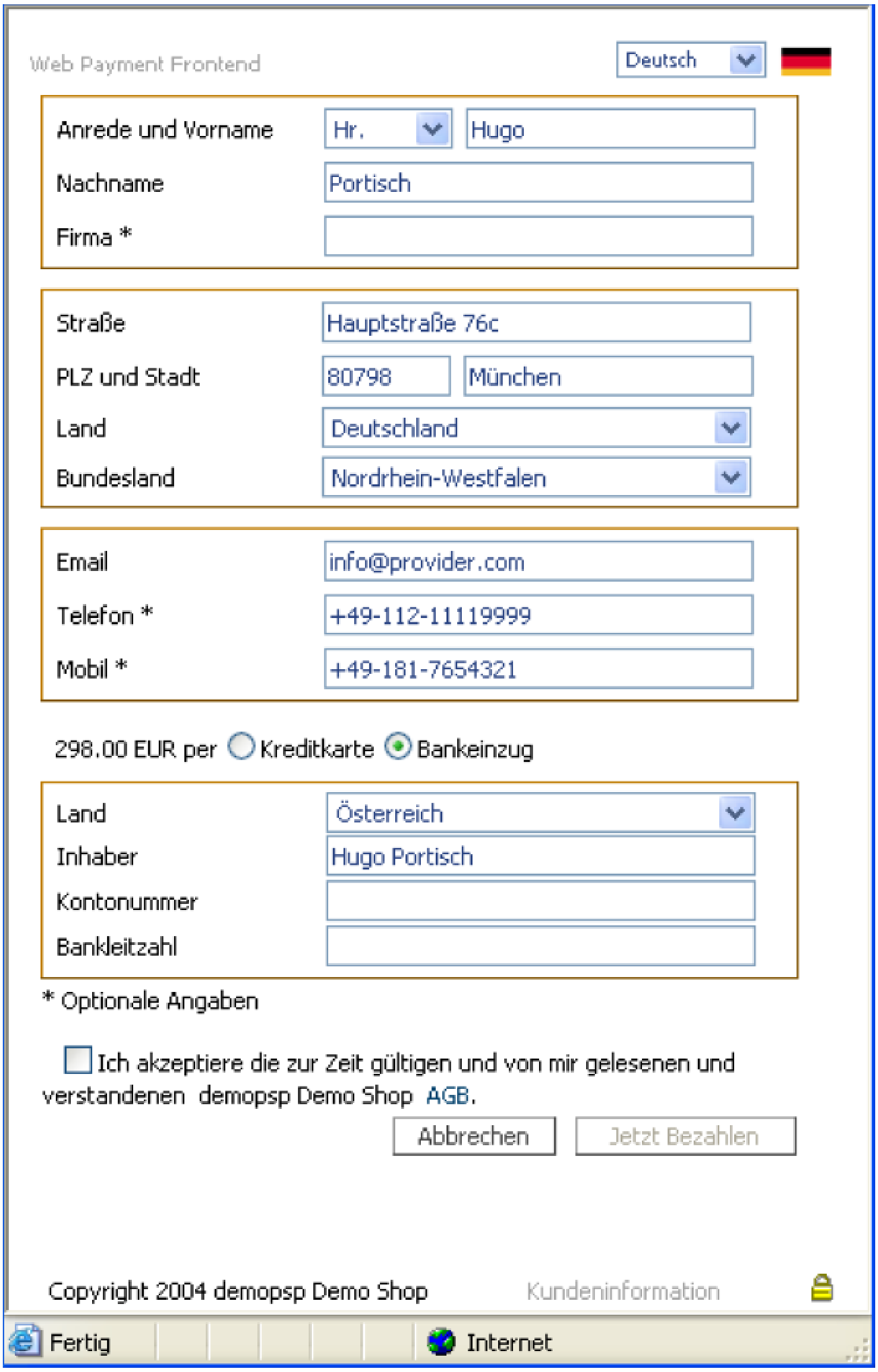
Using the parameter FRONTEND.CSS_PATH you can pass in a path to your own CSS:

Example: FRONTEND.CSS_PATH = http://www.merchant-page.com/merchant.css

The following CSS shows an example for the content of a custom CSS:

```
html, body {background-color: #ffffff;
scrollbar-face-color: #red;}
td {color: #000000; }
a:link { color: #red; text-decoration: none; }
a:visited {color: #orange; text-decoration: none; }
a:hover { color: #grey; text-decoration: underline; }
a:active { color: #orange; text-decoration: none; }
.bar {background-color: #ffffff;}
.frm_box { border-style:ridge; border-width:1; border-color:orange; font-
size:
13px; background-color: #ffffff; font-style:normal; font-
variant:normal; font-weight:normal; color:#000000; margin-
left:4px;margin-right:4px; padding-left:4px; padding-right:4px;
padding-top:1; padding-bottom:1; }
.tab {background-color:#ffffff; border-style: none;}
```

The resulting ***hCO*** payment page looks like this:



Anrede und Vorname	Hr. Hugo
Nachname	Portisch
Firma *	
Straße	Hauptstraße 76c
PLZ und Stadt	80798 München
Land	Deutschland
Bundesland	Nordrhein-Westfalen
Email	info@provider.com
Telefon *	+49-112-11119999
Mobil *	+49-181-7654321

298.00 EUR per ○ Kreditkarte ● Bankeinzug

Land	Österreich
Inhaber	Hugo Portisch
Kontonummer	
Bankleitzahl	

* Optionale Angaben

□ Ich akzeptiere die zur Zeit gültigen und von mir gelesenen und verstandenen demopsp Demo Shop AGB.

Pic 5 **Resulting hCO page example**

The following list gives an overview of all the available styles that can be changed:

CSS Styles	Description
html, body	Default settings for the page
td	Default settings for table cells
a (a:link, a:visited, a:hover, a.active, a.button, a.button:link, a.button.visited)	Default settings for links
.bar	Bar styles on top and bottom of payment page
input	Input fields
select	Select fields
.text_bold	Bold text
.text_small	Small text
.btn	Default input buttons

Table 1 CSS Styles Parameters

3.1.2 Change default behaviour with Custom CSS

The following example shows how to hide the full address block:

```
#addressBlock {display:none;}
```

In case you want to change the style of the CVV code row, this could look like this:

```
#cvdRow
{color:#303030;background-color:#F7F7F7; border-color:#DEDEDE; border-
style:solid solid solid none; border-width:1px 1px 1px 0pt; font-size:11px;
margin:0px 0px 3px; padding:3px; width:425px;}
```

3.2 Other Configuration Options

This chapter describes the most important and interesting configuration options.

Find a list of all possible parameters in the document “POST Integrator Transactions”.

3.2.1 Banners

You can optionally use the area on top or the bottom of the ***hCO*** as banner areas to display any link you like.

To do this, use the FRONTEND.BANNER parameters.

The following example shows how to display a URL in the top area with a height of 80px:

```
FRONTEND.BANNER.1.LINK=https://myserver.com/banner1.html
FRONTEND.BANNER.1.AREA=TOP
FRONTEND.BANNER.x.HEIGHT=80
```

3.2.2 Hide or Show Payment Methods and Types

You can optionally restrict the payment methods and payment subtypes an end user should see and select from in the ***hCO***.

Typically a merchant’s account is setup for a set of payment methods on the payment server. By default all these methods are selectable for a ***hCO*** user. If there has already been a pre-selection of the payment method on the merchant’s site, a merchant might want to prohibit the end user to change this in the ***hCO***. Therefore it is possible to restrict certain payment methods in the ***hCO***.

The following example shows how to show only German and Austrian direct debit to the ***hCO*** user.

```
FRONTEND.PM.DEFAULT_DISABLE_ALL=true
FRONTEND.PM.1.METHOD=DD
FRONTEND.PM.1.ENABLED=true
FRONTEND.PM.1.SUBTYPES=AT,DE
```

Another example shows how to display only all Credit Cards to the end user

```
FRONTEND.PM.DEFAULT_DISABLE_ALL=true
FRONTEND.PM.1.METHOD=CC
FRONTEND.PM.1.ENABLED=true
FRONTEND.PM.1.SUBTYPES=VISA,AMEX
```

If you do not specify any information with these parameters all payment methods configured on the payment server are selectable to the end users.

Another option is to make sure that certain subtypes are not displayed. The following example makes sure that all Direct Debit types but not Germany is displayed:

```
FRONTEND.PM.DEFAULT_DISABLE_ALL=true
FRONTEND.PM.1.METHOD=DD
FRONTEND.PM.1.ENABLED=true
FRONTEND.PM.1.SUBTYPES_DISABLED=DE
```

If you define a method with ENABLED=true, you must send in the subtypes you want to have available!

3.2.3 Links Configuration

External links can be integrated into the ***hCO*** to display things like Terms and Conditions, copyright info, external help and so on.

The links can either be displayed embedded right upon the “Pay” button as part of a checkbox (where the user accepts terms for instance) or in the status bar at the bottom (for help for instance)

The following example shows how to integrate terms and conditions into the page. The user has to check a checkbox to accept the terms before he can commit the payment transaction. The terms are opened in a popup with the size of 320x400 pixels.

```
FRONTEND.LINK.1.KIND=TERMS
FRONTEND.LINK.1.LINK=https://www.merchantshop.com/agb.php
FRONTEND.LINK.1.AREA=EMBEDDED_CHECKBOX
FRONTEND.LINK.1.POPUP_WIDTH=320
FRONTEND.LINK.1.POPUP_HEIGHT=400
```

3.2.4 *hCO* Size Configuration

Use the following two parameters to define the size of the ***hCO*** as it appears:

- `FRONTEND.FORM_WIDTH`
- `FRONTEND.HEIGHT`

By default the two parameters are in pixels, meaning a value of 100 results in 100px.

You can also send the sizes in % for instance by specifying % as well.

Example:

```
        FRONTEND.FORM_WIDTH = 100%
        FRONTEND.HEIGHT = 450
```

3.2.5 Change Button Appearance

The buttons on the payment page and on the confirmation page can be replaced by images located on the server of the merchant or anywhere else. By default the buttons are standard buttons. Their general appearance can be changed via CSS as well.

The following sample parameters change the buttons on the payment page to be images:

```
        FRONTEND.BUTTON.1.NAME=PAY
        FRONTEND.BUTTON.1.TYPE=IMAGE
        FRONTEND.BUTTON.1.LINK= https://www.merchantshop.com/images/pay.gif
        FRONTEND.BUTTON.2.NAME=CANCEL
        FRONTEND.BUTTON.2.TYPE=IMAGE
        FRONTEND.BUTTON.2.LINK= https://www.merchantshop.com/images/cancel.gif
```

3.2.6 Immediate Redirect after Payment / Registration

If you do not want the end user to see the payment result or cancellation page at the end, just set the redirect time to 0. This will lead to an immediate redirect back into the shop.

Example:

```
        FRONTEND.REDIRECT_TIME=0
```

4 Special Integration Methods

Some payment methods require special integration effort inside the merchant's shop.

A typical example is the PayPal Express Checkout payment method where the payment is finalized on the merchant's side depending on the shipping address selected on the PayPal page.

4.1 PayPal

hCO currently supports two PayPal Express Checkout workflows, PAYPAL and PAYPAL_CONTINUE. See the document "Asynchronous Workflows" for more details.

Please be aware that PayPal requires taking over the whole browser-window and the PayPal URL must appear in the browser navigation bar. This means that the ***hCO*** will open PayPal in the top window thus removing outer frames of the shop that may have embedded the ***hCO***.

Thus when the shopper returns to the shop ('thanks for paying page' or error-page) this will be not within the frame anymore where the ***hCO*** might have been originally embedded. When enabling PayPal as a payment method this site must cope with that.

4.1.1 PAYPAL or PAYPAL_CONTINUE?

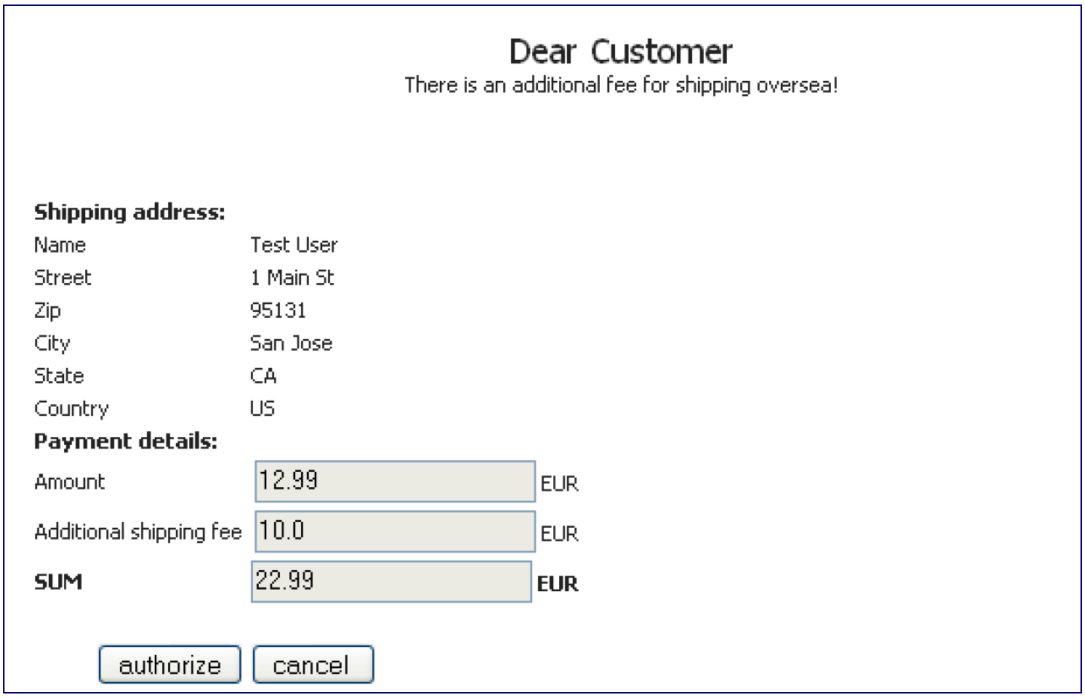
Depending on the required workflow you need to set PAYPAL or PAYPAL_CONTINUE as the brand in your initial request, e.g. ACCOUNT.BRAND=PAYPAL_CONTINUE.

For ACCOUNT.BRAND=PAYPAL there is nothing special to do, the PayPal workflow is fully embedded inside the ***hCO*** and the handling is the same as for any other payment method like Credit Card or Direct Debit.

For ACCOUNT.BRAND=PAYPAL_CONTINUE you must send in the payment code VA.PA in your initial request.

At the end of the ***hCO*** workflow you will need to show the shopper a summary page where you can change the amount of the transaction depending on the shippping address you receive from PayPal.

This page could typically look like this:



Dear Customer

There is an additional fee for shipping oversea!

Shipping address:

Name	Test User
Street	1 Main St
Zip	95131
City	San Jose
State	CA
Country	US

Payment details:

Amount	12.99	EUR
Additional shipping fee	10.0	EUR
SUM	22.99	**EUR**

Pic 6 **Merchant Final Payment Confirmation Page**

Note: The ADRESS.STATE field is mandatory for the following countries:

- USA (US)
- Canada (CA)
- (Mainland) China (CN)
- Japan (JP)
- Mexico (MX)
- Brasil (BR)

And must me have double-digits for the US following the rules of ISO3166-2:US (http://en.wikipedia.org/wiki/ISO_3166-2:US)

After this step you must finalize the payment by sending a synchronous transaction via XML or POST.

In case you want to pre-authorize the transaction only, send in a VA.PA referencing the initial VA.PA. To debit the PayPal account right away, capture the transaction by sending a VA.CP.

A sample XML request that finalizes a Preauthorization looks like this:

```
<Request version="1.0">
        <Header>
                <Security sender="31HA07BC8124AD82A9E96D9A35FAFD2A" />
        </Header>
        <Transaction mode="CONNECTOR_TEST"
channel="31HA07BC81A71E2A47DA94B6ADC524D8"
                response="SYNC">
                <Identification>
                <ReferenceID>ff80808116fc6ed00116fc6f0044011a</ReferenceID>
                </Identification>
                <User login="31ha07bc8124ad82a9e96d486d19edaa" pwd="password"
/>
                <Payment code="VA.PA">
```

```
                        <Presentation>
                              <Amount>15.99</Amount>
                              <Currency>EUR</Currency>
                        </Presentation>
                  </Payment>
            </Transaction>
      </Request>
```

A sample POST request that finalizes a Preauthorization looks like this:

```
REQUEST.VERSION=1.0
SECURITY.SENDER=31HA07BC8124AD82A9E96D9A35FAFD2A
TRANSACTION.CHANNEL=31HA07BC81A71E2A47DA94B6ADC524D8
TRANSACTION.MODE=CONNECTOR_TEST
USER.LOGIN=31ha07bc8124ad82a9e96d486d19edaa
USER.PWD=password
FRONTEND.ENABLED=false
IDENTIFICATION.REFERENCEID=ff80808116fc65120666fcdac2af004d
PAYMENT.CODE=VA.PA
PRESENTATION.AMOUNT=12.34
PRESENTATION.CURRENCY=EUR
```

4.1.2 Redirect to PayPal directly

In case you want to redirect to PayPal directly through ***hCO*** without showing any of the ***hCO*** pages, you need to disable all other configured methods but PayPal.

In case your configured payment methods are Credit Card with Mastercard and Visa and PayPal, you send in the following additional parameters when you call the ***hCO***:

```
FRONTEND.PM.DEFAULT_DISABLE_ALL=true
FRONTEND.PM.0.ENABLED=true
FRONTEND.PM.0.METHOD=VA
FRONTEND.PM.0.SUBTYPES=PAYPAL
```

4.2 Paysafecard and Cash-Ticket

4.2.1 Redirect to Paysafecard (Cash-Ticket) directly

In case you want to redirect to Paysafecard directly through the ***hCO*** without showing any of the ***hCO*** pages, you need to disable all other configured methods but Paysafecard (or Cash-Ticket).

In case your configured payment methods are Credit Card with Mastercard and Visa and Cash-Ticket, you send in the following additional parameters when you call the ***hCO***:

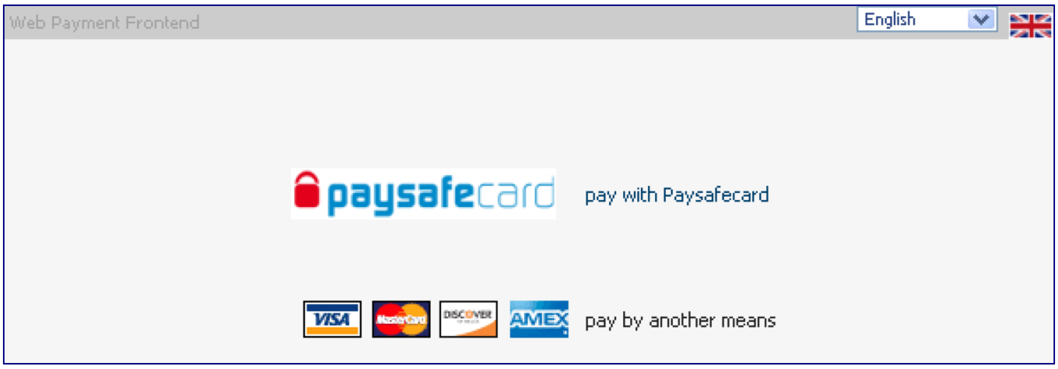
```
FRONTEND.PM.DEFAULT_DISABLE_ALL=true
FRONTEND.PM.1.METHOD=VA
FRONTEND.PM.1.ENABLED=true
FRONTEND.PM.1.SUBTYPES=PAYSAFECARD
PAYMENT.CODE=VA.PA
```

4.2.2 The Paysafecard (Cash-Ticket) pre-selection page

In case you have Paysafecard and other payment methods configured, a pre-selection page is displayed where the shopper is asked to select between Paysafecard and any other method.

If Paysafecard is not configured, the shopper will be redirected directly to the first ***hCO*** page.

The pre-selection page looks like this:



Pic 7 **Paysafecard Preselection Page**

4.3 Prepayment

Currently we only support special Prepayment pages for Portugal and Turkey (see MangirKart). Other countries are planned.

4.3.1 Redirect to the Prepayment page directly

In case you want to redirect to a pre-filled prepayment form directly through the ***hCO*** without showing any of the ***hCO*** pages, you need to disable all other configured methods but Prepayment for a certain country.

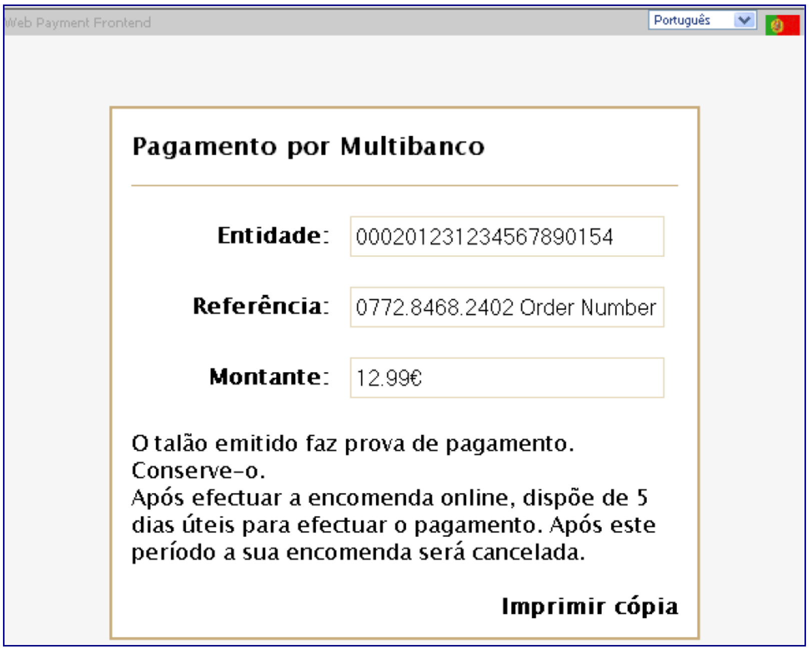
In case your configured payment methods are Credit Card with Mastercard and Visa and Prepayment Portugal, you send in the following additional parameters when you call the ***hCO***:

```
FRONTEND.PM.DEFAULT_DISABLE_ALL=true
FRONTEND.PM.1.METHOD=PP
FRONTEND.PM.1.ENABLED=true
FRONTEND.PM.1.SUBTYPES=PT
PAYMENT.CODE=PP.PA
ADDRESS.COUNTRY=PT
FRONTEND.LANGUAGE=PT
FRONTEND.REDIRECT_TIME=1
```

Attention: It is very important that the REDIRECT_TIME is set to a value greater than 0 for Prepayment! Otherwise the result page will not be displayed!

4.3.2 The Prepayment result page

On the Prepayment result page the user can open the Prepayment form and print it directly from within the browser. The result page looks like this:



Pagamento por Multibanco

Entidade: 000201231234567890154

Referência: 0772.8468.2402 Order Number

Montante: 12.99€

O talão emitido faz prova de pagamento. Conserve-o. Após efectuar a encomenda online, dispõe de 5 dias úteis para efectuar o pagamento. Após este período a sua encomenda será cancelada.

Imprimir cópia

Pic 8 Prepayment Portugal Result Page

4.4 MangirKart

As Payment Code it is necessary to pass over "PC.PA", also

ACCOUNT.BRAND=MANGIRKART as parameter.

The sequence is then similar to the online transfer process, the client will then after be transfered to the Mangirkart system. After that he will transition back, the system will pass over an 'receipt' transaction (see document Asychronous Transaction for more details).

In detail the shop system will receive after an PC.PA an PC.RC as response.

4.5 BarPay

As Payment Code it is necessary to pass over "PP.PA" as transaction typ, also

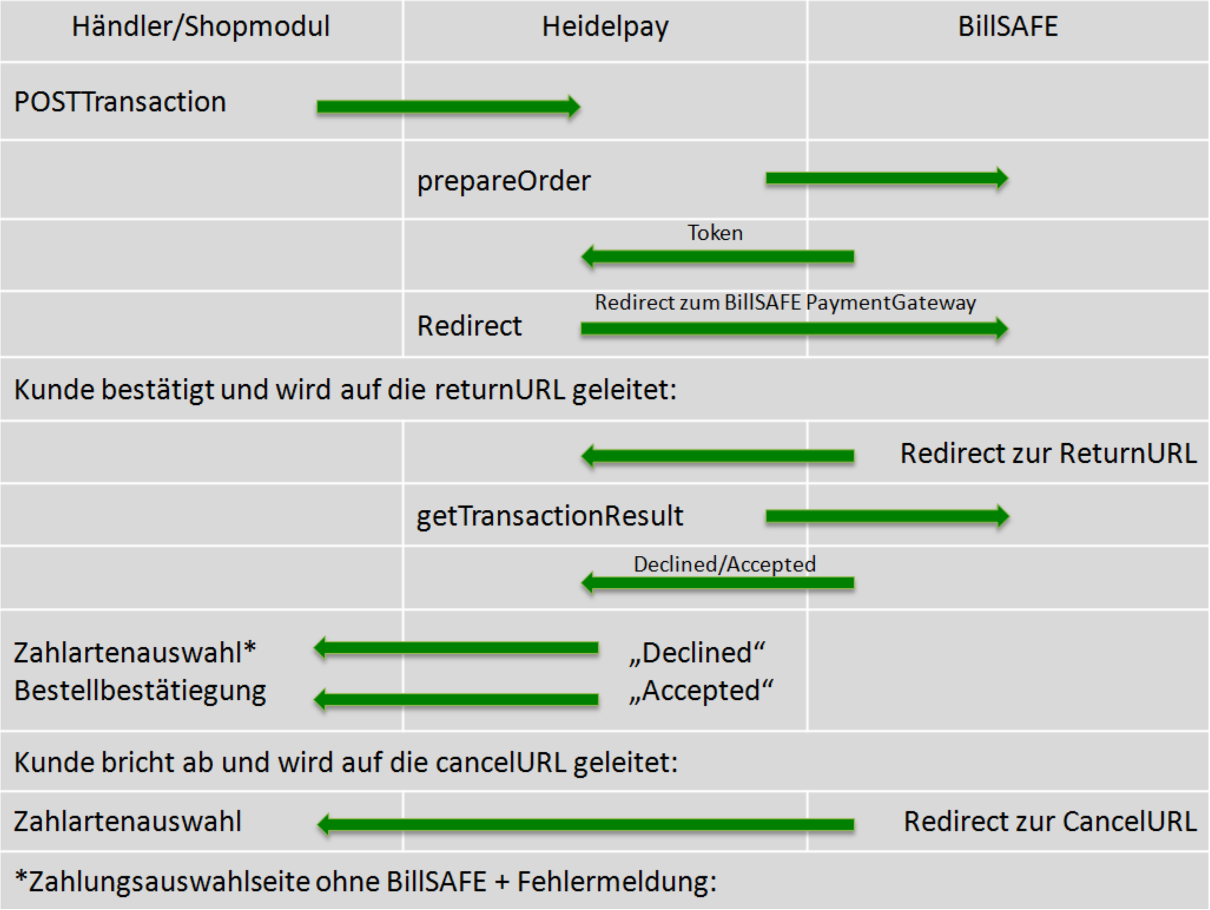
ACCOUNT.BRAND BARPAY (and no other account information) as parameter.

When the transaction "PP.PA" is successfully passed over, the system will receive an criterion "BARPAY_PAYCODE_URL" back. This will include the URL information to an PDF document, that the customer needs to print when I wants to perform his payment. This link information needs to pass over the shop system.

4.6 BillSAFE

Important Note: For integration purpose and test use only below test user for test transactions:

Privatkunde (B2C):
Firma: <leer>
Vorname: Paul
Nachname: Positiv
PLZ: 49084
Ort: Osnabrück



Händler/Shopmodul | Heidelpay | BillSAFE

POSTTransaction

prepareOrder

Token

Redirect

Redirect zum BillSAFE PaymentGateway

Kunde bestätigt und wird auf die returnURL geleitet:

Redirect zur ReturnURL

getTransactionResult

Declined/Accepted

Zahlartenauswahl*
Bestellbestätiegung

„Declined“
„Accepted“

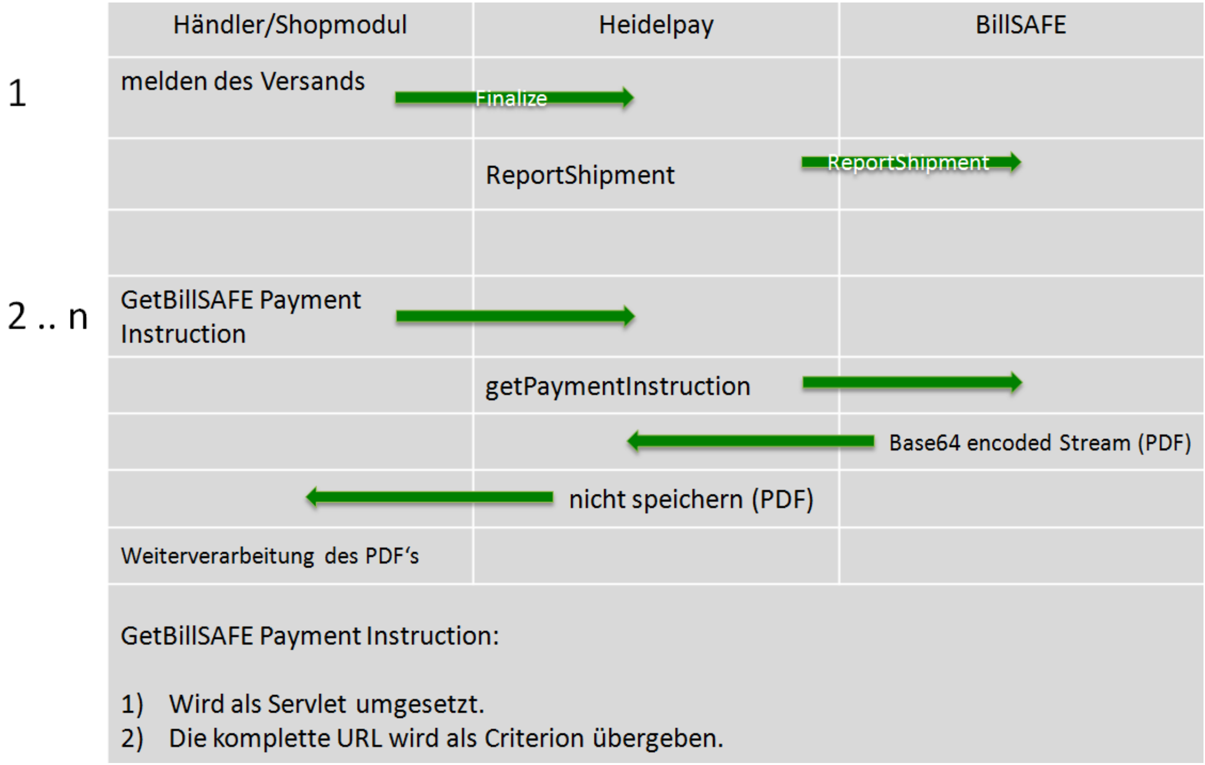
Kunde bricht ab und wird auf die cancelURL geleitet:

Zahlartenauswahl

Redirect zur CancelURL

*Zahlungsauswahlseite ohne BillSAFE + Fehlermeldung:

Pic 9 Flow Chart BillSAFE



GetBillSAFE Payment Instruction:

1) Wird als Servlet umgesetzt.
2) Die komplette URL wird als Criterion übergeben.

Pic 10 Flow Chart BillSAFE

As Payment Code it is necessary to pass over "IV.PA" also ACCOUNT.BRAND BILLSAFE (and no other account information) as parameter.

The "IV.PA" transaction is then in the meantime pending, while the customer will be forwarded to the website of BillSAFE and confirming the purchase in parallel. In the meantime will BillSAFE process the payment and set the "IV.PA" transaction to success.

After successful submission of a "IV.PA" transaction and also forwarded to the gateway of BillSAFE and filled correct all the fields out, as well as confirming the whole purchase basket, the store gets a response back from Heidelpay with Criterions, further included are:

```
<Criterion name="BILLSAFE_BANKCODE">26550105</Criterion>
<Criterion name="BILLSAFE_ACCOUNTNUMBER">231522</Criterion>
<Criterion name="BILLSAFE_BANKNAME">Sparkasse Osnabrück</Criterion> <Criterion
name="BILLSAFE_BIC">NOLADE22</Criterion>
<Criterion name="BILLSAFE_IBAN">DE87265501050000231522</Criterion>
<Criterion name="BILLSAFE_REFERENCE">BTN 10092225</Criterion> <Criterion
name="BILLSAFE_AMOUNT">36.3</Criterion>
<Criterion name="BILLSAFE_CURRENCY">EUR</Criterion>
<Criterion name="BILLSAFE_PERIOD">14</Criterion>
<Criterion name="BILLSAFE_NOTE">Bitte überweisen Sie den ausstehenden Betrag bis
zum 15.10.2012.</Criterion> <Criterion name="BILLSAFE_LEGALNOTE">Heidelberger
Payment GmbH
(www.heidelpay.de) hat die Forderung gegen Sie im Rahmen eines laufenden
Factoringvertrages an die PayPal (Europe) S.à r.l. et Cie, S.C.A.
```

```
abgetreten. Zahlungen mit schuldbefreiender Wirkung können nur an die PayPal
(Europe) S.à r.l. et Cie, S.C.A. geleistet werden.</Criterion>
```

Important Note: These upper fields are required for create the invoice and must be processed by the shop itself. It means the store needs to create the invoice. Please be aware the invoice needs to be paid from the client maximum after 14 days after shipping has been notified to BillSAFE, this is a fixed paying period from BillSAFE and can't be changed!

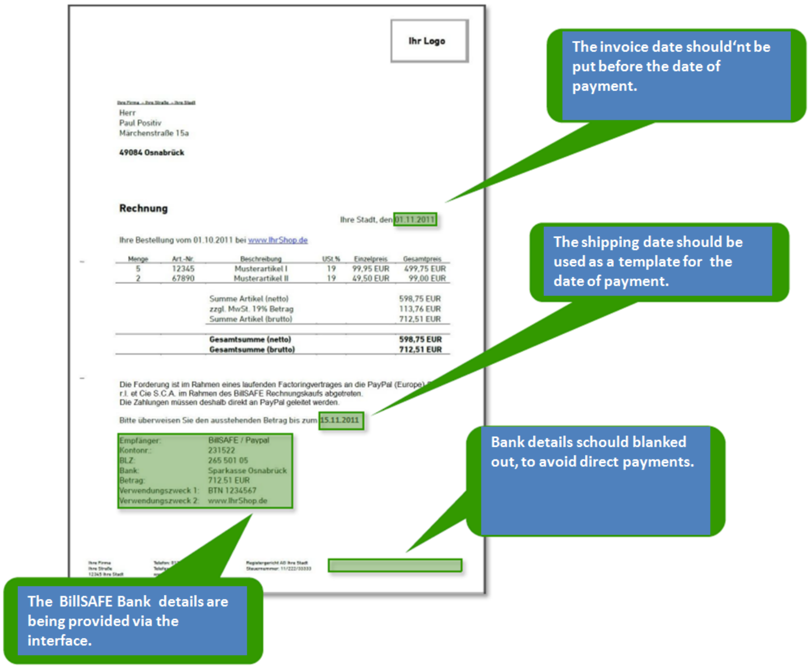
As example for the invoice you can use the below example:

```
-      Payment Name:"Payment on account"
-      Description :"Buy now and examine your purchases before you pay alone.
                    Integration of BillSAFE logos with information popup and
                    payment selection page with the following source code:"
```

```
<!--BillSAFE start-->
<noscript>
<a title="Ihre Vorteile" href="http://www.billsafe.de/special/payment-info"
target="_blank">
<img src="https://images.billsafe.de/image/image/id/191997712fbe"
style="border:0"/>
</a>
</noscript>
<a id="billsafeAdvantagesImageLink" title="Ihre Vorteile" href="#" style="display:
none;" onclick="openPopup();">
<img src="https://images.billsafe.de/image/image/id/191997712fbe"
style="border:0"/>
</a>
<script type="text/javascript" >
var link = document.getElementById('billsafeAdvantagesImageLink');
link.style.display = 'inline';
var openPopup = function() {
var myWindow = window.open('http://www.billsafe.de/special/payment-info',
'BillSAFE', 'width=520,height=600,left=300,top=100,scrollbars=yes');
myWindow.focus();
};
</script>
<!--BillSAFE end-->
```

Important Note: The marked grey field can be edited and configured by the shop owner. This parameter is pointing to the Logo and can be modified and adapted. Also be aware the shopping basket needs to be passed as Criterions "IV.PA" parameter.

See the attached specification for transferring Invoice items. It can be extended to the following values:



Ihr Logo

Ihre Firma - Ihre Straße - Ihre Stadt

Herr
Paul Positiv
Märchenstraße 15a

49084 Osnabrück

Rechnung

Ihre Stadt, den 01.11.2011

Ihre Bestellung vom 01.10.2011 bei www.IhrShop.de

Menge	Art.-Nr.	Beschreibung	USt.%	Einzelpreis	Gesamtpreis
5	12345	Musterartikel I	19	99,95 EUR	499,75 EUR
2	67890	Musterartikel II	19	49,50 EUR	99,00 EUR
		Summe Artikel (netto)			598,75 EUR
		zzgl. MwSt. 19% Betrag			113,76 EUR
		Summe Artikel (brutto)			712,51 EUR
		Gesamtsumme (netto)			**598,75 EUR**
		Gesamtsumme (brutto)			**712,51 EUR**

Die Forderung ist im Rahmen eines laufenden Factoringvertrages an die PayPal (Europe) S.à r.l. et Cie S.C.A. im Rahmen des BillSAFE Rechnungskaufs abgetreten.
Die Zahlungen müssen deshalb direkt an PayPal geleitet werden.

Bitte überweisen Sie den ausstehenden Betrag bis zum 15.11.2011

Empfänger:	BillSAFE / Paypal
Kontonr.:	231522
BLZ:	265 501 05
Bank:	Sparkasse Osnabrück
Betrag:	712.51 EUR
Verwendungszweck 1:	BTN 1234567
Verwendungszweck 2:	www.IhrShop.de

Ihre Firma
Ihre Straße
12345 Ihre Stadt

Registergericht AG Ihre Stadt
Steuernummer: 11/222/33333

The invoice date should'nt be put before the date of payment.

The shipping date should be used as a template for the date of payment.

Bank details schould blanked out, to avoid direct payments.

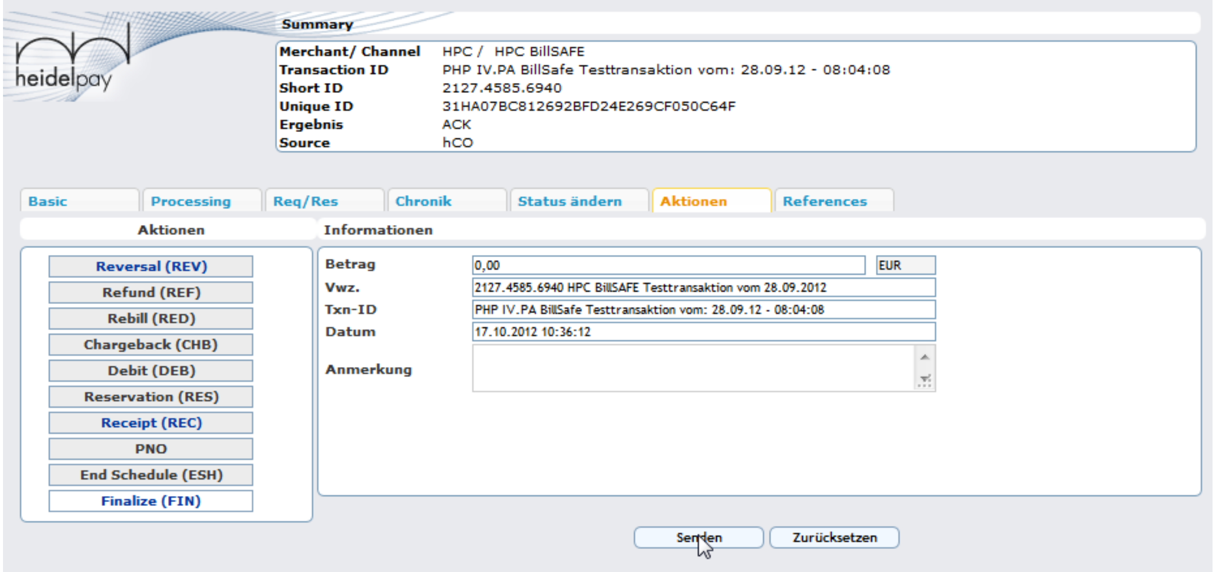
The BillSAFE Bank details are being provided via the interface.

Pic 11 BillSAFE - Invoice example

Important Note: The whole shopping basket is mandatory all time and needs to be passed over to BillSAFE, items not shipped needs to pass over with quantity value "0".

To finalize the purchase it's important to send a "shipping confirmation" to BillSAFE. To achieve this transaction you need to pass over the transaction type "FI" (represents FINALIZE) with ReferenceId (UniqueId) to the core system.

In case you have an merchandise planning and control system in use. You can finilize the purchase process in the hIP-System using the below mask under transaction details from the selected tab:



Summary

Merchant/ Channel	HPC / HPC BillSAFE
Transaction ID	PHP IV.PA BillSafe Testtransaktion vom: 28.09.12 - 08:04:08
Short ID	2127.4585.6940
Unique ID	31HA07BC812692BFD24E269CF050C64F
Ergebnis	ACK
Source	hCO

Informationen

Betrag	0,00 EUR
Vwz.	2127.4585.6940 HPC BillSAFE Testtransaktion vom 28.09.2012
Txn-ID	PHP IV.PA BillSafe Testtransaktion vom: 28.09.12 - 08:04:08
Datum	17.10.2012 10:36:12
Anmerkung	

Pic 12 hIP - Finalize (FIN)

4.6.1 Transaction Parameters handled with BillSAFE (all amounts are net amounts)

Name	Description	Annotation	Remark
POS_01.POSITION	Item number (e.g. "A 1.1")	AlphaNumeric 1..16	Mandatory
POS_01.QUANTITY	Number of units (e.g. "11")	Numeric 1..8	Mandatory
POS_01.UNIT	Unit (e.g. "unit")	AlphaNumeric 1..16	Mandatory
POS_01.AMOUNT_UNIT	Amount (net) per unit of the position in the smallest currency unit (eg, 3 EUR as "300")	Numeric 1..16	Mandatory
POS_01.AMOUNT	Total (net) position in the smallest currency unit (eg 33 EUR as "3300")	Numeric 1..16	Mandatory in case when POS_01 is used
POS_01.TEXT	Text position	AlphaNumeric 1..128	Mandatory in case when POS_01 is used
POS_01.COL1	Free field at item level for use by arrangement	AlphaNumeric 1..32	OPTIONAL
POS_01.COL2	Free field at item level for use by arrangement	AlphaNumeric 1..32	OPTIONAL
POS_01.COL3	Free field at item level for use by arrangement	AlphaNumeric 1..32	OPTIONAL
POS_01.COL4	Free field at item level for use by arrangement	AlphaNumeric 1..32	OPTIONAL
POS_01.ARTICLE_NUMBER	assigned by the Store item number, not to be confused with the position	AlphaNumeric 1..50	Mandatory
POS_01.PERCENT_VAT	Sales tax rate in percent (no VAT amount!) With a decimal point (eg, 19% as "19.00", 8.67% as "8.67")	#0.00 1..5,2	Mandatory
POS_01.ARTICLE_TYPE	"Goods": shipping items "Shipment": Returns "Voucher": coupon / discount (in the field POS_01.AMOUNT has a negative amount will be noted)	AlphaNumeric	OPTIONAL

Table 2 Transaction Parameters handled with BillSAFE

5 FAQ

Why not simply call the hCO URL from the browser and directly show the payment frontend?

There are several reasons why this is not working. First of all this process is very much security driven. The authentication service of the hCO only allows payment requests for a merchant that are sent from a specific IP or IP range, in this case the IP of the merchant’s shop server. Therefore nobody else can initiate a payment process on behalf of the merchant.

Secondly, unlike using the standard XML or POST integrator, this is an asynchronous payment method. This means, the payment result has to be communicated back to the merchant’s server. The merchant server needs to know if the payment was successful or not. Again this is happening via Server-to-Server communication. It would be very unsafe to somehow communicate this information back via the browser.

The payment window should have the same look and feel like my shop, what can I do?

The frontend of the hCO can be configured anytime. See chapter 3 of this document for more details.

Is there a demo shop available for other programming languages?

Currently the demo shop is fully developed in Java. There are other implementations planned as well, however, the document “Integration Packages” contains integration instructions for the XML and the POST integrator in all kind of programming languages. Since the POST integrator is the base of the hCO this document should allow you to integrate payment with any other language without any problems.

Do I need to send Account information for the *hCO* integration?

No. This is not necessary and not recommended. The account information (Credit Card or Bank Accounts) are entered by the end user in the ***hCO***. If you are able to send the account information as well, ***hCO*** is probably not the right solution for you, as you already got the payment data for the user. However, you can use the parameter PAYMENT.CODE to pre-select a payment method for the end user. If you prefer that the end user is paying with his bank account, send PAYMENT.CODE=DD.DB, if you prefer him paying with his credit card, send PAYMENT.CODE=CC.DB to the server.

How do I know if the payment transaction was successful or not?

hCO is based on an asynchronous process. This means, after sending your initial request to the server, you are loosing control of the process. However, after the end user has entered his payment data in the hCO, the payment server sends the result of the payment to the URL you initially specified in the parameter FRONTEND.. One of the parameters you receive as part of this response request is called PROCESSING.RESULT. If PROCESSING.RESULT is ACK, the transaction was successful, if it is NOK the transaction failed.

The end user has finished payment. Now how does the he get back to my page?

After payment is finished, the payment server sends the payment result to the URL you initially specified in the parameter FRONTEND.RESPONSE_URL. Simply response to this request with the URL you want the end user to be redirected to. You can write this URL as plain text to the output stream of your servlet or script. The payment server will read the response and redirect the user’s browser to this URL. This process gives you the chance to redirect the end user to different pages depending on the payment result.

I get a Javascript Error on the payment-finished page, what is the problem?

In this case it is very likely that the URL you returned as a response to the payment result message posted to your system is not valid.

The payment-finished page contains a Javascript-Snippet similar to the following:

```
function gotoMerchantSite()
{
parent.location.href = “http://myshop.com/thankyou.html”
}
```

The Url that is used in this method is the one we received back from you after posting the payment result to your response url. If this Url is not valid (or contains “) it is possible that you get a Javascript error.

BillSAFE: Did I need to pass over the whole shopping basket

Yes, all the shopping basket is Mandatory and needs to be passed over to BillSAFE all time.

A Links and accounts to *hIP* and *hCO*

TEST-system

The following link leads to heidelpay's TEST-system:

hIP	
Link	https://test-heidelpay.hpcgw.net/***hip***/
USER	heidelpay-test-agent
PASSWORD	password

Table 3 *hIP* (TEST)

The following data are available for testing the ***hCO*** on heidelpay's TEST-system:

hCO	
Link	https://test-heidelpay.hpcgw.net/sgw/gtw
heidelpay TEST-Merchant (3D-secure)	
LOGIN	31ha07bc8124ad82a9e96d486d19edaa
SENDER	31HA07BC8124AD82A9E96D9A35FAFD2A
PASSWORD	password
CHANNEL 1 (OT method „Sofortüberweisung“ and all credit cards, debit cards, direct debit, invoice, prepayment, PayPal)	31HA07BC81A71E2A47DA94B6ADC524D8
CHANNEL 2 (Giropay)	31HA07BC81A71E2A47DA662C5EDD1112
CHANNEL 3 (iDEAL)	31HA07BC81A71E2A47DA804F6CABDC59
CHANNEL 4 (BarPay)	31HA07BC81009F135218A33AE4A3F3FB
CHANNEL 5 (MangirKart)	31HA07BC81009F1352181DB01D6D904B
CHANNEL 6 (BillSAFE)	31HA07BC815DEA2098CCA5A7332203C6
heidelpay TEST-Merchant (no 3D-secure)	
LOGIN	31ha07bc810c91f086431f7471d042d6
SENDER	31HA07BC810C91F08643A5D477BDD7C0
PASSWORD	password
CHANNEL 1 (OT method „Sofortüberweisung“ and all credit cards, debit cards, direct debit, invoice, prepayment, PayPal)	31HA07BC810C91F086433734258F6628
CHANNEL 2 (Giropay)	31HA07BC810C91F086430EA18CE5E0BF
CHANNEL 3 (iDEAL)	31HA07BC810C91F0864355310BA6BD4C
CHANNEL 4 (BarPay)	31HA07BC81009F135218293AE82EA620
CHANNEL 5 (MangirKart)	31HA07BC813E25B9EFD23E973B929D02
CHANNEL 6 (BillSAFE)	31HA07BC815DEA2098CC2FDE7B444DB6

Content-Type	application/x-www-form-urlencoded;charset=UTF-8

Table 4 *hCO* (TEST)

5.1.1 LIVE-system

The following link leads to heidelpay's LIVE-system:

hIP	
Link	https://heidelpay.hpcgw.net/***hip***/
USER	*Supplied by heidelpay-service after registration*
PASSWORD	"

Table 5 *hIP* (LIVE)

The following is the link to ***hCO*** on LIVE-system:

hCO	
Link	https://heidelpay.hpcgw.net/sgw/gtw
Merchant data	*Supplied by heidelpay-service after registration*
LOGIN	"
SENDER	"
PASSWORD	"
CHANNEL	"
Content-Type	application/x-www-form-urlencoded;charset=**UTF-8**

Table 6 *hCO* (LIVE)

A Test Data (Credit Cards, Debit Cards, Bank Accounts)

Executing tests on the TEST-system can be done using any payment instrument, but heidelpay highly recommends using one of the listed below:

A.1 Credit Cards

Brand	Number	Annotation	Valid through	CVV
American Express	375000000000007		12 / 2015	1234
DISCOVER	6011587918359498		10 / 2015	123
Mastercard	5105105105105100		10 / 2015	123
	5453010000059543	3DSecure	10 / 2015	123
	5453010000059675	3DSecure enabled – authorization will fail	10 / 2015	123
VISA	4012888888881881		10 / 2015	123
	4111111111111111		10 / 2015	123
	4200000000000000	Not 3DSecure enabled	10 / 2015	123
	4711100000000000	3DSecure enabled	10 / 2015	123
	4012001037461114	3DSecure – authorization will fail	10 / 2015	123
VISA Electron	4012888888881881		10 / 2015	123

Table 6 Credit Cards (Test-numbers)

A.2 Debit Cards

Brand	Number	Valid Through	CVV
Carte Bleue	4111111111111111	10 / 2015	123
Maestro	6799851000000032	10 / 2015	123
SOLO	6334580500000000	10 / 2015	123

Table 7 Debit Cards (Test-numbers)

A.3 Direct Debits

Country	ISO	Institute	Account-no.
Belgium	BE	Bank Code Number (BLZ) 539	0075470-34
Denmark	DK	Bank Code Number (BLZ) 0040	0440116243
Germany	**DE**	Bank Code Number (BLZ) 37040044	5320130
		Bank Code Number (BLZ) 38050000	46581
		Bank Code Number (BLZ) 10000000	1234567890
Finland	FI	Bank Code Number (BLZ) 123456	785
France	FR	Bank Code Number (BLZ) 20041	010050500013M02606
Great Britain	GB	Bank Code Number (BLZ) 601613	31926819
Italy	IT	Bank Code Number (BLZ) -	X0542811101000000123456** Or B0501812100000000115000**
Netherlands	NL	Bank Code Number (BLZ) -	0417164300 Or

			0000012112 Or 0123456789
Norway	NO	Bank Code Number (BLZ) -	60033321115 Or 60031234568
Austria	AT	Bank Code Number (BLZ) 20151 Bank Code Number (BLZ) 20111 Bank Code Number (BLZ) 1232111	938044617 Or 28161647502 Or 65785423
Spain	ES	Bank Code Number (BLZ) -	21000418450200051332** Or 20382739996000057498**
Sweden	SE	Bank Code Number (BLZ) 5491	0000003
Switzerland	CH	Bank Code Number (BLZ) 100 Bank Code Number (BLZ) 4003	123456-1-123-11 Or 999999-99-999
Czech Republic	CZ	Bank Code Number (BLZ) 0800	19-2000145399
Hungary	HU	Bank Code Number (BLZ) 10012349	12345678-91234567

Table 8 Direct Debit Account-numbers (Test-accounts)

A.4 Online Transfer

Brand	Bank Code Number (BLZ)	Account-no.	U. PIN	U. TAN
Sofortüberweisung	88888888	123456	12345	123456
Giropay	12345679	0000000300	12345	123456

Table 9 Online Transfer Account-numbers (Test-accounts)

B Sample PHP-Script for calling hCO-page

The following script is an example using PHP for calling the hCO. You may have to change the field `'FRONTEND.RESPONSE_URL'` in order to have this working for your system:

```
<?php

//URL for the test system
$url = "https://test-heidelpay.hpcgw.net/sgw/gtw";
////////////////////////////////////
// Standard test data for test shop -- with 3D
//

$parameters['SECURITY.SENDER'] = "31HA07BC8124AD82A9E96D9A35FAFD2A";
$parameters['USER.LOGIN'] = "31ha07bc8124ad82a9e96d486d19edaa";
$parameters['USER.PWD'] = "password";
// Channel for CC, OT Sofort, DC, DD, PayPal
$parameters['TRANSACTION.CHANNEL'] = "31HA07BC81A71E2A47DA94B6ADC524D8";
// Channel für Giropay
//$parameters['TRANSACTION.CHANNEL'] = "31HA07BC81A71E2A47DA662C5EDD1112";
// Channel für iDeal
//$parameters['TRANSACTION.CHANNEL'] = "31HA07BC81A71E2A47DA804F6CABDC59";
////////////////////////////////////

////////////////////////////////////
// Standard test data for test shop -- without 3D
//
//$parameters['SECURITY.SENDER'] = "31HA07BC810C91F08643A5D477BDD7C0";
//$parameters['USER.LOGIN'] = "31ha07bc810c91f086431f7471d042d6";
//$parameters['USER.PWD'] = "password";
// Channel für CC, OT Sofort, DC, DD, PayPal
//$parameters['TRANSACTION.CHANNEL'] = "31HA07BC810C91F086433734258F6628";
// Channel für Giropay
//$parameters['TRANSACTION.CHANNEL'] = "31HA07BC810C91F086430EA18CE5E0BF";
// Channel für iDeal
//$parameters['TRANSACTION.CHANNEL'] = "31HA07BC810C91F0864355310BA6BD4C";
////////////////////////////////////

$parameters['ACCOUNT.HOLDER'] = "";
$parameters['ACCOUNT.NUMBER'] = "";
//$parameters['ACCOUNT.BRAND'] = "PAYPAL";
$parameters['ACCOUNT.BRAND'] = "";
```

```
$parameters['ACCOUNT.EXPIRY_MONTH'] = "";
$parameters['ACCOUNT.EXPIRY_YEAR'] = "";
$parameters['ACCOUNT.VERIFICATION'] = "";

//Payment Code -- Selection payment method and typ
//$parameters['PAYMENT.CODE'] = "DD.RG";
//$parameters['PAYMENT.CODE'] = "CC.RG";
$parameters['PAYMENT.CODE'] = "CC.DB";
//$parameters['PAYMENT.CODE'] = "OT.PA";
//$parameters['PAYMENT.CODE'] = "VA.DB";
$parameters['PRESENTATION.CURRENCY'] = "EUR";

//Response URL
$parameters['FRONTEND.RESPONSE_URL'] = "http://www.merchant.de/response-page.php";

//CSS- and/or Jscript-Datei
$parameters['FRONTEND.CSS_PATH']                                                  =
"http://127.0.0.1/Testskripte/Fuer_Kunden/onlycarddetails.css";
//$parameters['FRONTEND.JSCRIPT_PATH'] = "http://127.0.0.1/wpf/wpfui.js";

$parameters['PRESENTATION.AMOUNT'] = '15.20';
$parameters['IDENTIFICATION.TRANSACTIONID'] = 'merchants own ID (e.g. ordernumber -
customer number)';
$parameters['PRESENTATION.USAGE'] = 'USAGE (Verwendungszweck)';
$parameters['FRONTEND.MODE'] = "DEFAULT";
//$parameters['FRONTEND.MODE'] = "WPF_LIGHT";

// Modus selection
//$parameters['TRANSACTION.MODE'] = "LIVE";
$parameters['TRANSACTION.MODE'] = "INTEGRATOR_TEST";
//$parameters['TRANSACTION.MODE'] = "CONNECTOR_TEST";

$parameters['FRONTEND.ENABLED'] = "true";
$parameters['FRONTEND.POPUP'] = "false";
//$parameters['FRONTEND.SHOP_NAME'] = '';
$parameters['FRONTEND.REDIRECT_TIME'] = "0";

$parameters['FRONTEND.LANGUAGE_SELECTOR'] = "true";
$parameters['FRONTEND.LANGUAGE'] = "DE";

$parameters['REQUEST.VERSION'] = "1.0";
```

```
/*
$parameters['NAME.GIVEN'] = "";
$parameters['NAME.FAMILY'] = "";
*/


$parameters['NAME.GIVEN'] = "Hanz";
$parameters['NAME.FAMILY'] = "Mustermann";
$parameters['ADDRESS.STREET'] = "Musterstrasse 1";
$parameters['ADDRESS.ZIP'] = "12345";
$parameters['ADDRESS.CITY'] = "Demo City";
$parameters['ADDRESS.COUNTRY'] = "DE";
//$parameters['ADDRESS.STATE'] = "DE1";
// In case you want a list ...
//"DE1">Baden-Württemberg
//"DE2">Bayern
//"DE3">Berlin
//"DE4">Brandenburg
//"DE5">Bremen
//"DE6">Hamburg
//"DE7">Hessen
//"DE8">Mecklenburg-Vorpommern
//"DE9">Niedersachsen
//"DEA">Nordrhein-Westfalen
//"DEB">Rheinland-Pfalz
//"DEC">Saarland
//"DED">Sachsen
//"DEE">Sachsen-Anhalt
//"DEF">Schleswig-Holstein
//"DEG">Thüringen


$parameters['CONTACT.EMAIL'] = "test@test.de";


//building the postparameter string to send into the WPF


$result = '';
foreach ($parameters AS $key => $value)
$result .= strtoupper($key).'='.urlencode($value).'&';
$strPOST = stripslashes($result);


//echo $strPOST;


//open the request url for the Web Payment Frontend


$cpt = curl_init();
curl_setopt($cpt, CURLOPT_URL, $url);
curl_setopt($cpt, CURLOPT_SSL_VERIFYHOST, 2);
curl_setopt($cpt, CURLOPT_USERAGENT, "php heidelpaypost");
curl_setopt($cpt, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($cpt, CURLOPT_SSL_VERIFYPEER, FALSE);
curl_setopt($cpt, CURLOPT_POST, 1);
curl_setopt($cpt, CURLOPT_POSTFIELDS, $strPOST);
$curlresultURL = curl_exec($cpt);
$curlerror = curl_error($cpt);
$curlinfo = curl_getinfo($cpt);
curl_close($cpt);



// here you can get all variables returned from the heidelpay server (see post
integration transactions documentation for help)
//print $strPOST;
// parse results


$r_arr=explode("&",$curlresultURL);
foreach($r_arr AS $buf)
{
$temp=urldecode($buf);
$temp=split("=",$temp,2);
$postatt=$temp[0];
$postvar=$temp[1];
$returnvalue[$postatt]=$postvar;
//print "<br>var: $postatt - value: $postvar<br>";
}


$processingresult=$returnvalue['POST.VALIDATION'];


$redirectURL=$returnvalue['FRONTEND.REDIRECT_URL'];


// everything ok, redirect to the WPF,
if ($processingresult=="ACK")
{
       if (strstr($redirectURL,"http")) // redirect url is returned ==> everything
ok
{
              header("Location: $redirectURL");
}
       else // error-code is returned ... failure
{
              //header("Location: http://127.0.0.1/livesystem/error.php");
              print_r($returnvalue);
}
}// there is a connection-problem to the heidelpay server ... redirect to error
page (change the URL to YOUR error page)
       else
{
              // header("Location: http://127.0.0.1/livesystem/connection.php");
              print_r($returnvalue);
              //print_r($returnvalue);
}
?>
```

C Index

CONNECTOR_TEST.. 21
INTEGRATOR_TEST.................................... 11, 12
LIVE... 32
Page-flow 6