Integration Basics

General information, hIP, hCO, test-data and links

heidelpay payment platform

The complete solution for Ecommerce

Date: 15.11.2012

Version: 1.9

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	17.04.2012	Schuhmann	Extending list of transaction types on page 9
1.3	08.08.2012	Fredrich	Updated all document
1.4	13.08.2012	Fredrich	Test Data updated
1.5	07.09.2012	Fredrich	Payments-methods and Transaction-type table updated
1.6	11.09.2012	Fredrich	BillSAFE Channel added
1.7	10.10.2012	Fredrich	Updated minor points
1.8	12.10.2012	Fredrich	Updated minor points
1.9	15.11.2012	Fredrich	Updated minor points

Document history

Table of content

List of pictures

List of tables

Introduction

This document serves as your source of information in all cases of questions around Heidelpay's payment-platform concerning ***hIP*** (heidelpay Intelligence platform) and ***hCO*** (heidelpay CheckOut).

The content of this documentation is as following:

Chapter "1 Overview"gives you a general view on the various systems provided by heidelpay, the process of connecting to the system, performing transactions and maintaining them.

Chapter "2" *hIP*: Maintaining transactions" explains the usage of ***hIP*** (heidelpay intelligence platform) which is heidelpay's administrative tool allowing to search and maintain transactions you have transmitted to the payment-platform.

Chapter "3 *hCO*: Connecting shops made easy" deals with details of connecting standard shop-solutions to the system using ***hCO*** (heidelpay Checkout). ***hCO*** is designed to cover almost every situation which may arise during connecting shops which are implemented using standard-software to our payment-solution. Examples of shop-solutions that easily fit to ***hCO*** are:

- XT Commerce
- OXID

Chapter „4 hIP –Virtual Terminal (v-Terminal)" explains the usage of heidelpay's "Virtual Terminal" (v-Terminal).v-Terminal gives you the capability to submit transactions to the payment-system in cases where customers place orders by telephone or are not able to use your website for some reason.

ANNOTATION: This document is dedicated to how to submit transactions to heidelpay's payment system using ***hCO*** and how to maintain transactions using heidelpay ***hIP.*** 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 Overview

This chapter gives you an overview on the systems provided by heidelpay to you. It also introduces you to some important notations and wordings used with heidelpay's payment platform as there are:

- TEST System and LIVE System
- Payment-methods
- Transaction mode / Integration mode
- Transaction-type

1.1 General purpose

The general purpose of heidelpay's system is the provision of a single interface to you as the merchant whichs:

- offers a huge number of various types of payment-methods used throughout Ecommerce
- processes payment-transactions with numerous financial institutes in a secure environment which allows full security of the data provided by you
- enables you to provide payment-methods which require enhanced security-precautions and infrastructure (e.g. processing of payments using credit-cards

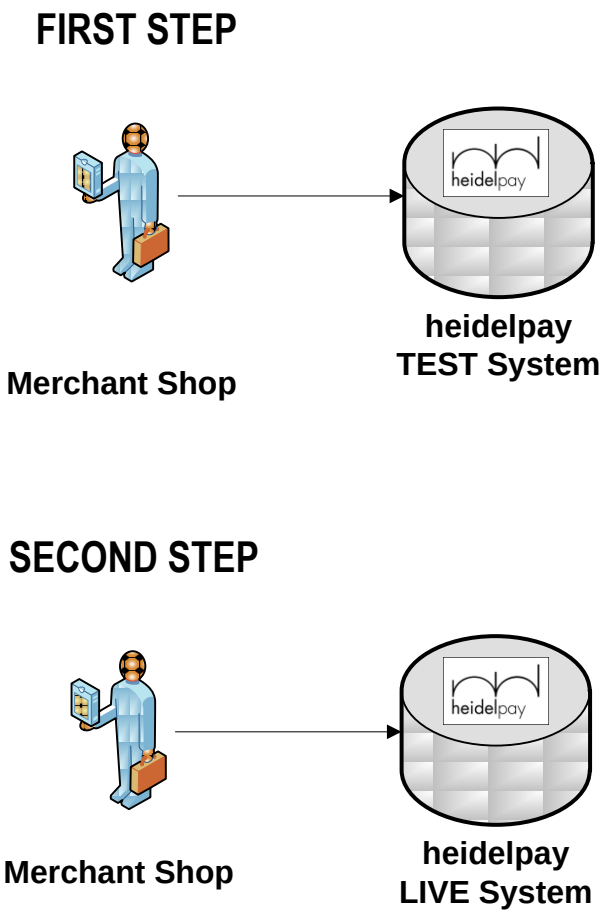
1.2 Two systems: TEST and LIVE System

Heidelpay's payment-system is provided to you in two separate installations:

- a **TEST System** and
- a **LIVE System**

The TEST-system is an exact copy of heidelpay's LIVE-system. The handling of payments makes up the difference:

- Transactions passed to the **TEST System** will never cause any movement of funds between the customer and the merchant, while
- Transactions passed to the **LIVE System** will cause movement of funds (see further information in chapter 1.5)



Pic 1 TEST and LIVE System

You may find the links to heidelpay's in section "A *Links and accounts to* ***hIP*** *and* ***hCO****"*

!! Annotation: it is HIGHLY recommended to follow the rule "test your shop against the test-system only" since testing against the production-system may cause unwanted transfers of funds !!

!! Annotation: although heidelpay's TEST-system undergoes the same high security precautions as the LIVE-system it is not recommended to use data of real payment-instruments on TEST-system – use test-data as provided in section "*B Test Data (Credit Cards, Debit* Cards, Bank Accounts)**"**

1.3 Payment-methods

A payment-method is any kind of mechanism which allows you as the merchant to retrieve money from your customers. The following list shows the payment-methods currently available at heidelpay:

Method	heidelpay naming	Availability	Description
Creditcard	CC		
Mastercard VISA American Express Diners		World-wide	Debits the customer's card and retrieves money from card-issuing institutes
Direct debits	DD	Numerous countries worldwide: DE, AT, ES, GB and many more (please contact sales at heidelpay for the complete list)	Perform debits on customer's bank-accounts and credit according amount to the merchant
Debit cards	DC		
Maestro		UK	
Solo		UK	
Postepay		Italy	
Carte Bleue		France	
Value accounts	VA		
PayPal		World-wide	
Online Transfers	OT		
Sofortüberweisung		Germany	
iDEAL		Netherlands	
Giropay		Germany	
EPS		Austria	
Payment Card	PC		
MangirKart		Turkey	
Invoice	IV	Available for numerous countries worldwide	The merchants sends out invoices and the payment-system informs about according records on the account-statements
BillSAFE		Germany	
Prepayment	PP	Available for numerous countries worldwide	
BarPay		Germany	

Table 1 List of payment-methods

The list of payment-methods provided by heidelpay is under continuos growth – please contact your sales-representative if you miss any method not listed here.

1.4 Transaction-types

While a "payment-method" denotes the method the customer uses to pay with the "transaction-type" denotes the activity you as the merchant place on the customer's payment-method

Example

The customer pays using the payment-method "credit-card" (CC) and the merchant debits (DEB) the customer's credit card. In heidelpay's system you will see a transaction type like this (the transaction is shown as "Credit", because it reflects the merchant's point of view).

Credit	Debit	Type	Meth.
5.80		DEB	CC

The following is a list of all transaction-types known within the payment-system:

Transaction-type	Description
DEB	**DEBIT:** debit the customer using the customer's payment-method and credit an according amount to the merchant's account. This is the usual "the customer pays the goods" activity
RED	**REDEBIT:** re-execute a former DEBIT using the same data
RES	**RESERVATION:** some payment-methods like credit-cards allow not to debit the customer immediately but to "reserve" the money. The debit is executed in a step later with reference to the RESERVATION. Such debit is called a "CAPTURE"
CAP	**CAPTURE:** Deduct the money from the customer using a formerly done RESERVATION
REC	**RECEIPT:** a receipt is initiated by the system when a credit is done to the merchant's account referencing a former transaction (usually using methods Invoice (IV), Prepayment (PP) and OT (Online-transfer)
REV	**REVERSAL:** the merchant can initiate a reversal of a former DEBIT, e.g. in case the goods or some of the goods can not be delivered; in the latter case a partial REVERSAL is possible as well. REVERSALs are only possible as long as no funds-transfers have been initiated so far. If the funds-transfer has been initiated a REFUND will be executed instead. The system keeps notice of the status of the transaction and decides automatically whether a REVERSAL is possible or a REFUND needs to be executed
REF	**REFUND:** similar to REVERSAL but invoking funds-transfer.
CRE	**CREDIT:** This allows to credit a defined amount to the consumer.
REG	**REGISTRATION:** the payment system enables merchants to register customer's payment-instruments within the system. This enables customers to use only enter the payment-instrument once and use it on future purchases
RRE	**RE REGISTRATION:** this type of transaction allows to change the data of an existing registration

	(payment data / address data)
CFR	**CONFIRM REGISTRATION:** used to confirm a registration as still valid
DRE	**DE REGISTRATION:** used to mark a registration as no longer valid
SCH	**SCHEDULE:** this transaction type allows the definition of transactions which occur periodically. That is e.g. monthly debits, debits which occur quarterly or yearly debits
CSH	**CHANGE SCHEDULE:** using this type of transaction schedules which were defined formerly can be changed (e.g. changing the execution cycle)
ESH	**END SCHEDULE:** this transaction type is used to end the periodically execution of an existing schedule
CHB	**CHARGE BACK:** this kind of transaction is generated when debiting the customer fails in a later stage of processing, e.g. the customer revokes the transaction at the issuer or at the bank
CBN	**CHARGE BACK NOTIFICATION:** in conjunction with chargebacks this type transaction may indicate an incoming chargeback which must not have been processed so far
CBR	**CHARGE BACK REVERSAL:** this is the revocation of a prior chargeback
RCL	**RECONCILIATION:** generated internally by the system when balancing accounts
FIN	**FINALIZE:** finalize transaction (example: report shipment to BillSAFE)

Table 2 Transaction-types

1.5 Transaction-mode and integration-mode

There is still a way to process transactions on the LIVE-system without causing funds-transfers: the difference is made up by the “transaction-mode” used by you when submitting transactions ti the system.

This „mode“ is assigned to merchants by heidelpay and it is assigned to transactions by the merchant when passing them in.

The mode is one of

- INTEGRATOR_TEST
- CONNECTOR_TEST
- LIVE

1.5.1 INTEGRATOR_TEST

Transactions using this mode are accepted by the system but the system will never contact external financial partners nor will it cause any movement of money between the merchant and the customer. The processing is executed completely locally

1.5.2 CONNECTOR_TEST

The mode CONNECTOR_TEST is a mode which is enhanced in comparison to INTEGRATOR_TEST: if provide by external systems heidelpay’s payment-platform wil conduct test-transactions at those systems and process the transaction accordingly.

1.5.3 LIVE

All transactions passed in in this mode are real transactions – they will cause movement of money transfers between the customer, the merchant and heidelpay.

1.5.4 Behaviour of the system depending on the mode

It is possible to submit test-transactions to the LIVE-system which will not cause customers to be debited.

Depending on the combination of „merchant-mode“ and „transaction-mode“ the system behave as described in the following table:

Merchant mode	Transaction mode	Question	Behaviour on TEST-System	Behaviour on LIVE-System
INTEGRATOR_TEST	LIVE	Can a merchant in mode INTEGRATOR_TEST submit LIVE – mode transactions?	A merchant in mode INTEGRATOR_TEST is not allowed to pass in LIVE-transactions. An error is generated and the transaction is denied	A merchant in mode INTEGRATOR_TEST is not allowed to submit transactions in mode LIVE An error is generated and the transaction is denied.
	CONNECTOR_TEST	Are payments debited to the customer?	A merchant in mode INTEGRATOR_TEST is not allowed to pass in LIVE-transactions. An error is generated and the transaction is denied	A merchant in mode INTEGRATOR_TEST is not allowed to submit transactions in mode LIVE An error is generated and the transaction is denied.
		Does the merchant pay transaction-fee ?	NO	**YES !!**
	INTEGRATOR_TEST	Are payments debited to the customer?	NO	NO
		Does the merchant pay transaction-fee ?	NO	**YES !!**
CONNECTOR_TEST	LIVE		A merchant in mode CONNECTOR_TEST is not allowed to submit transactions in mode LIVE An error is generated and the transaction is denied.	A merchant in mode CONNECTOR_TEST is not allowed to submit transactions in mode LIVE An error is generated and the transaction is denied

	CONNECTOR_TEST	Are payments debited to the customer?	NO	NO
		Does the merchant pay transaction-fee ?	NO	**YES**
	INTEGRATOR_TEST	Are payments debited to the customer?	NO	NO
		Does the merchant pay transaction-fee ?	NO	**YES**
LIVE	LIVE	Are payments debited to the customer?	NO	**YES**
		Does the merchant pay transaction-fee ?	NO	**YES**
	CONNECTOR_TEST	Are payments debited to the customer?	NO	NO
		Does the merchant pay transaction-fee ?	NO	**YES**
	INTEGRATOR_TEST	Are payments debited to the customer?	NO	NO
		Does the merchant pay transaction-fee ?	NO	**YES**

Table 3 System-behaviour depending on the integration-mode

2 *hIP*: Maintaining transactions

hIP is the interface provided by heidelpay to view and maintain all kind of transactions you have transmitted to heidelpay's payment-platform.

2.1 Links to *hIP*

The logon-page of ***hIP*** is accessible using the following links:

System	Link
TEST-System	https://test-heidelpay.hpcgw.net/***hip***/
LIVE-System	https://heidelpay.hpcgw.net/***hip***/

Table 4 Links to *hIP*

For the TEST-system you can use the following data for signing in:

Logging in into the TEST-system

You can enter heidelpay's TEST-system by using the following user and password:

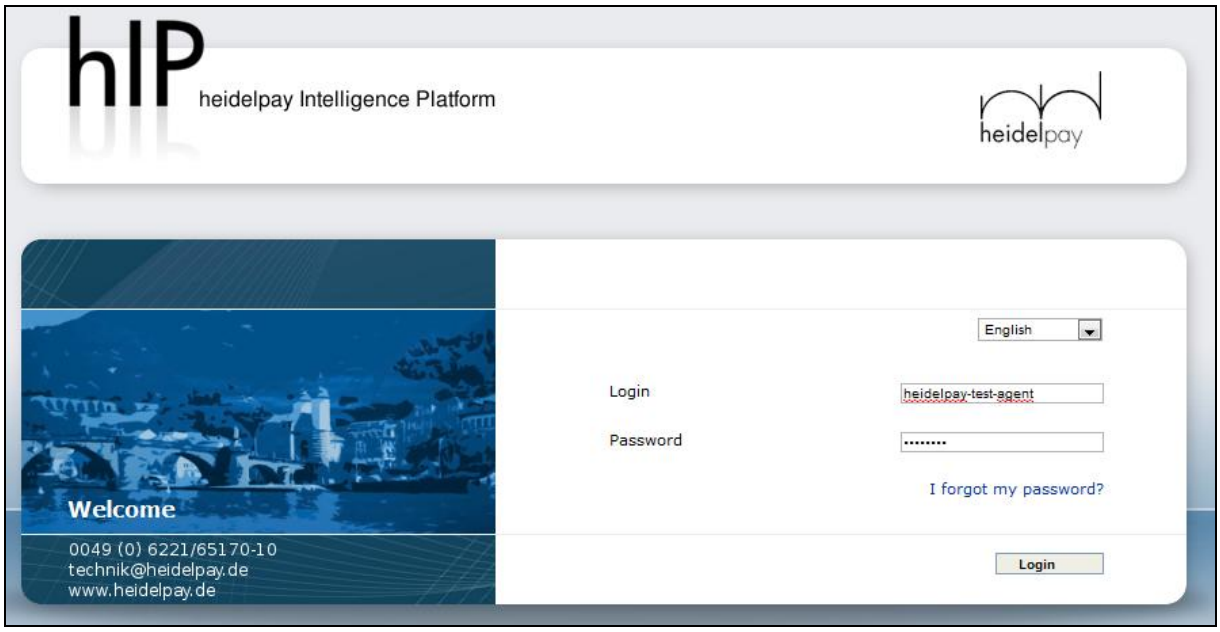
Username	Password
`heidelpay-test-agent`	`password`

Table 5 Login test-account

Annotation: Concerning heidelpay's LIVE-system you will receive according username and password from the support at heidelpay throughout the process of integration.

2.2 Logon-page

Selecting the login-link of ***hIP*** leads you to the following login-page



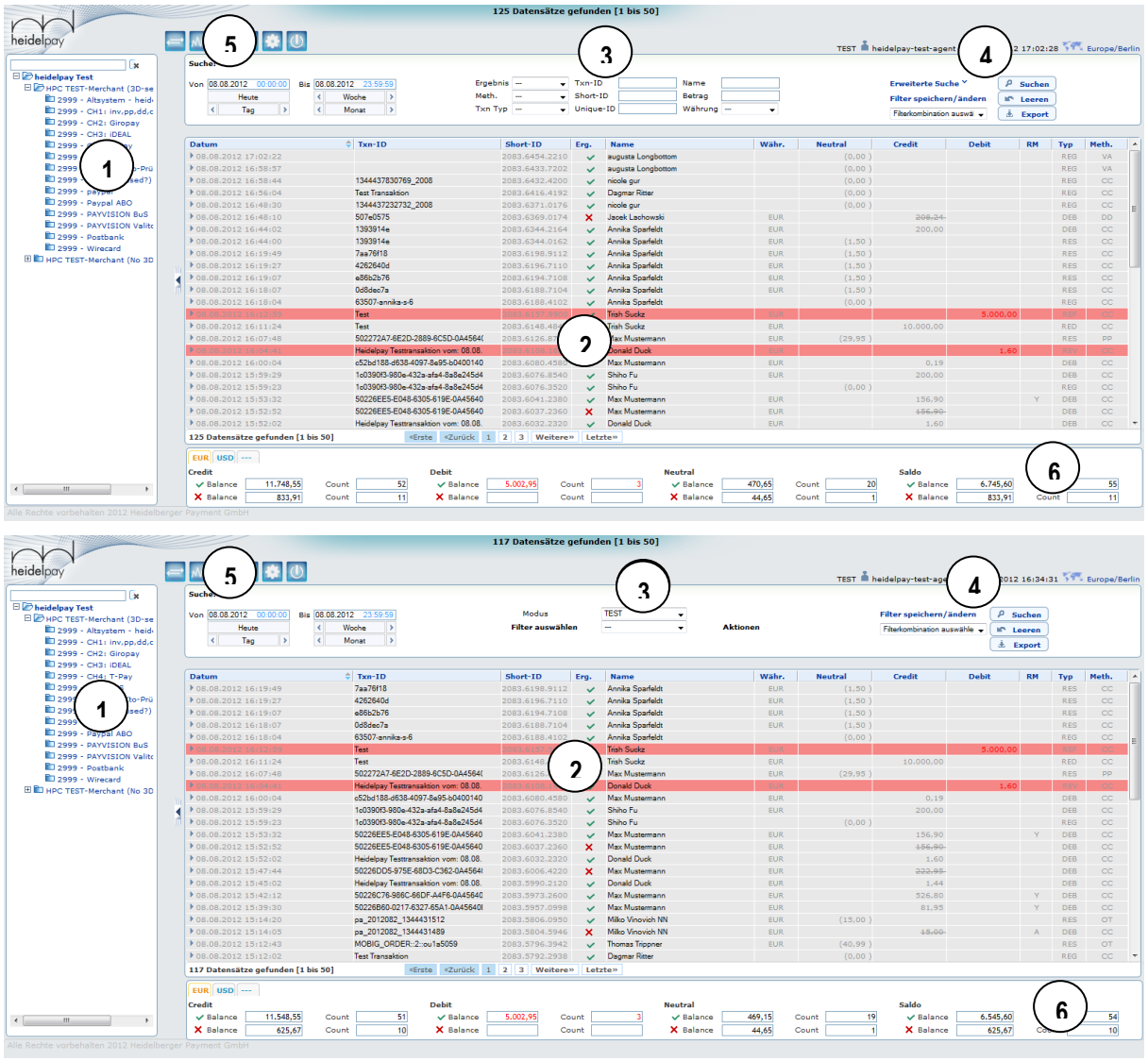
Pic 2 *hIP* Logon-page

hIP's logon-page offers the usual functionality of entering username and password. In case you have forgotten your password you can advice the system to send a new password using the link "I forgot my password".

Annotation: **Changing the password is not allowed and does not work on the account "heidelpay-test-agent" on heidelpay's TEST-System.**

2.3 Main page

The main page is shown after logging into the ***hIP***. You can configure and adapt the main page in 2 different views in the search section, that is the reason that in the following pictures are 2 versions here presented:



Pic 3 *hIP* Main page (note: these are 2 different views on point 3)

1 The tree on left hand side allows you to select a single shop in case you run multiple shops with our system

2 In the main area placed at the centre of the screen ***hIP*** displays your transactions. You may see the following details:

Date: the date the transaction entered our system

Txn-ID: the *Txn-ID* which the merchant (shop) has assigned to the transaction when passing it in into the system

Short-ID: the *Short- ID* the payment-system has assigned to your transaction

Result: whether the transaction was successful (ACK) or not successful (NOK). Moving the mouse over this field of a transaction shows you a tooltip containg more detailed information on the outcome of processing that transaction

Name: the name of the customer who initiated the transaction

Curreny: the currency of the transaction

Neutral / Credit / Debit: the amount of the transaction; the terms “credit” and “debit” denote the point of view as of the merchant’s account. A “neutral” transaction causes effectively no movement of finance.

Type: the type of the transaction (DEB, REV, REF). The distinct types are explained in detail in chapter “1.4 Transaction-types”

3D: denotes whether this transaction was processed using “3D-secure”

Method: the type of payment-method used with this transaction

3 The section on top allows you to perform searches on transactions using numerous criteria

4 This section allows you to create even more complex searches on transactions. These search-criterias can be stored and reused as “Filters”. This feature has shown to be very useful throughout your daily work with the system.

5 The tabs on the top lead to further screens of ***hIP***, as there are:

a. Volume: shows you an aggregated view on your transactions over time

b. Setup: allows to change your password

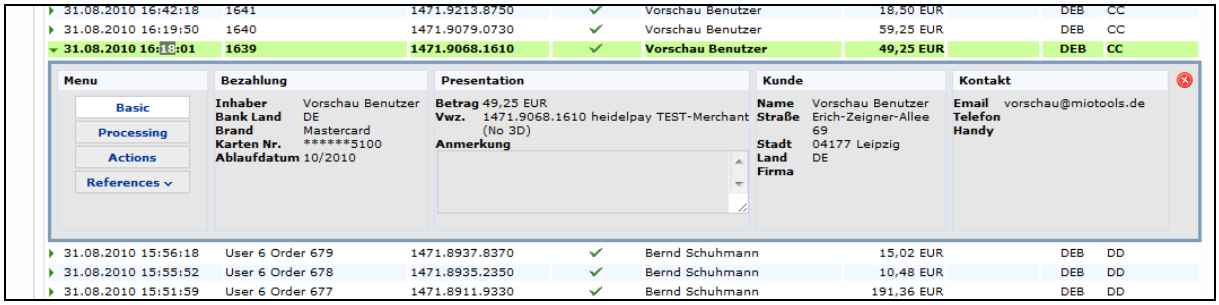
c. Files: allows you to upload and download files to the payment-system (this feature is optional and depends on the contract you have made with heidelpay)

6 The section at the bottom of the screen shows you aggregated sums of the transactions displayed

2.4 Details of transactions

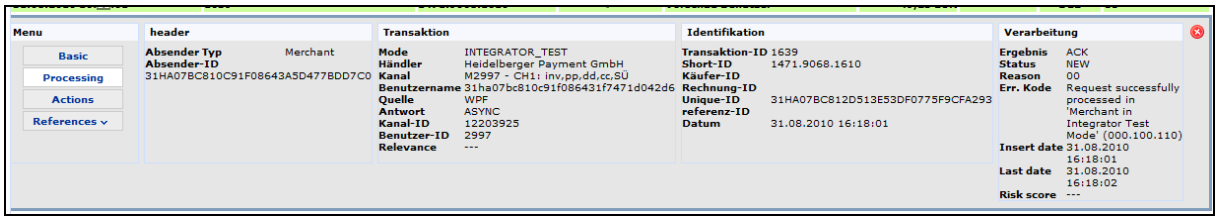
Performing a double-click on a single transaction opens a window displaying more details of the transaction chosen. Here you also have the possibility to perform actions on transactions.

The following pictures shows the “Basic” part of a transaction. You see the payment-instrument which was used by the customer, the information that will be presented on bank-receipts and on the customer’s account-statements and details of the customer’s address and contact-information.



Pic 4 *hIP* Details of transaction (Basic)

Choosing the button “Processing” shows you details of the transaction that where used and produced throughout processing in the system:



Pic 5 *hIP* Details of transaction (Processing)

Header: these are the information passed in as the "header" of the transaction

Transaction: information about the mode (LIVE, INTEGRATORTEST …) used with the transaction as well the channel and some more

Identification: there are multiple ways of identifying the transaction within in the system – these fields are listed here

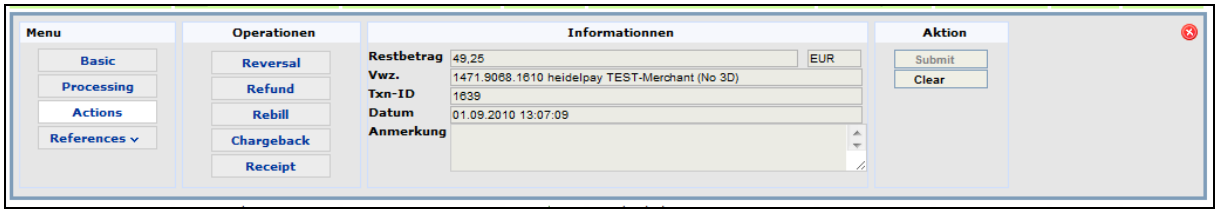
Transaction-ID as passed in by the merchant

Invoice-ID the invoice the merchant assigns with the transaction

Unique-ID an identifier that uniquely identifies the transaction

Processing: this section mainly deals with the result of the transaction at the end of processing. You see the reason of error (if the transaction was not successful) as well as the exact time the transaction arrived in the system and the end of processing

The button "Actions" allows you to perform various activities on a transaction:

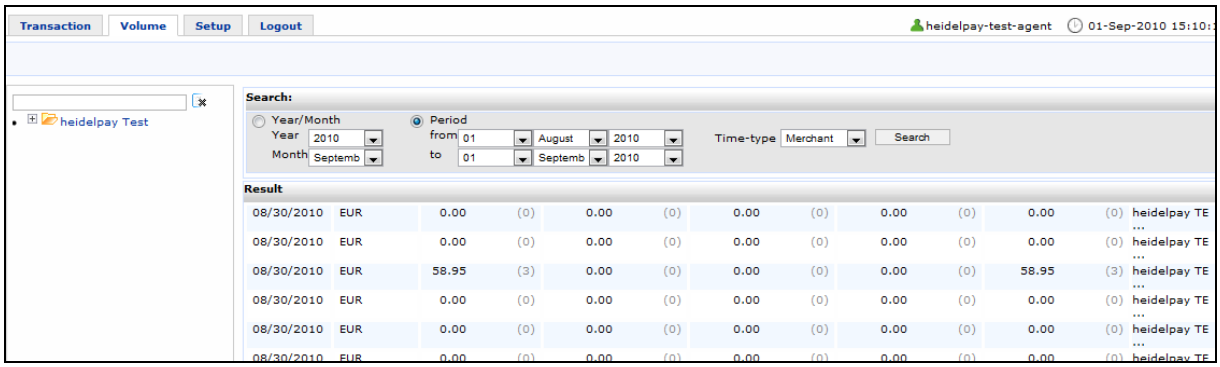


Pic 6 *hIP* Details of transaction (Actions)

- **Reversal**: a reversal can be performed as long as there is no cash-flow executed on the transaction. You may choose this when e.g. you agreed with the customer not to deliver the goods or to only deliver some of the goods. In the latter case you can perform a partial reversal of the transaction. Reversal is only possible as long as no transfer of funds has taken place yet
- **Refund**: in case you do not deliver the goods to the customer but cash-flow has already been initiated you need to refund an according amount (whole or partial). The system will initiate an according transaction and thereby debit your account and credit the customer's account. Refund is disabled when no money-flow has taken until yet
- **Rebill:** in case a prior debit of the transaction failed (for any reason) you can simply create another debit using the original data. ***hIP*** will create a new transaction using the data from the prior DEBIT and resubmit this transaction to the system
- **Chargeback:** if – for any reason – the customer declined debiting his account you can mark this transaction as "chargeback". The system creates a designated "chargeback-transaction" related to the transaction you have chosen. Processing of chargebacks depends heavily on the underlying payment-instrument that was used. Chargebacks may also be created automatically by the system when according information is processed. You will receive detailed information on this throughout the process of integration

2.5 Volume-tabulator

Another tab of interest is “Volume” which you find at top left of the screen. It shows you the volume and number of transactions processed per day within the timeframe you have chosen.



Pic 7 *hIP* Volume

The standard time-frame chosen is the current month. You see daily numbers of

- Payments, Refunds, Chargebacks, Reservation

grouped by:

- Date -> Currency -> Merchant -> Channel

3 *hCO*: Connecting shops made easy

This section is a small introduction to heidelpay's Check-Out (***hCO***). A complete documentation is available as "*heidelpay-Developer Guide-hCO*".

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.

3.1 *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

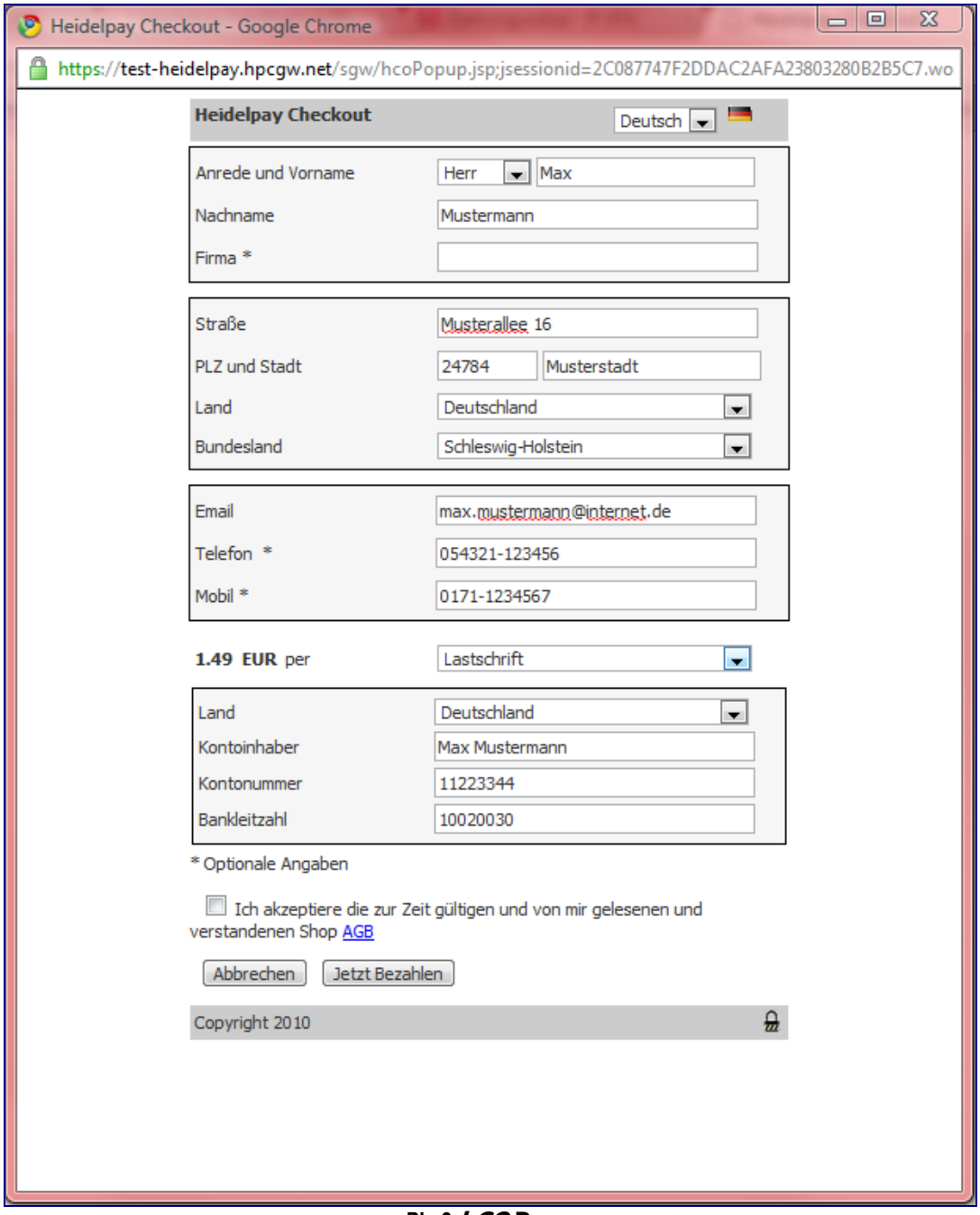
443 in case of https

<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.



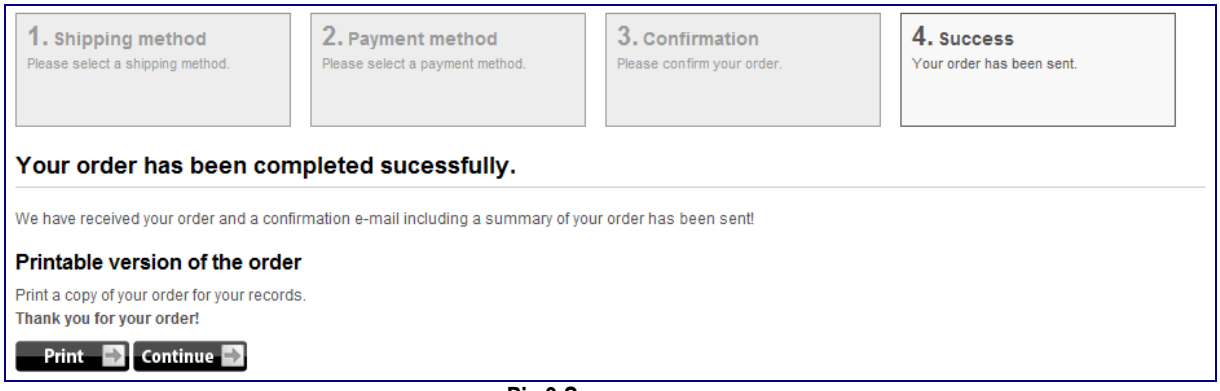
Pic 8 *hCO* Page

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



Pic 9 Success-page

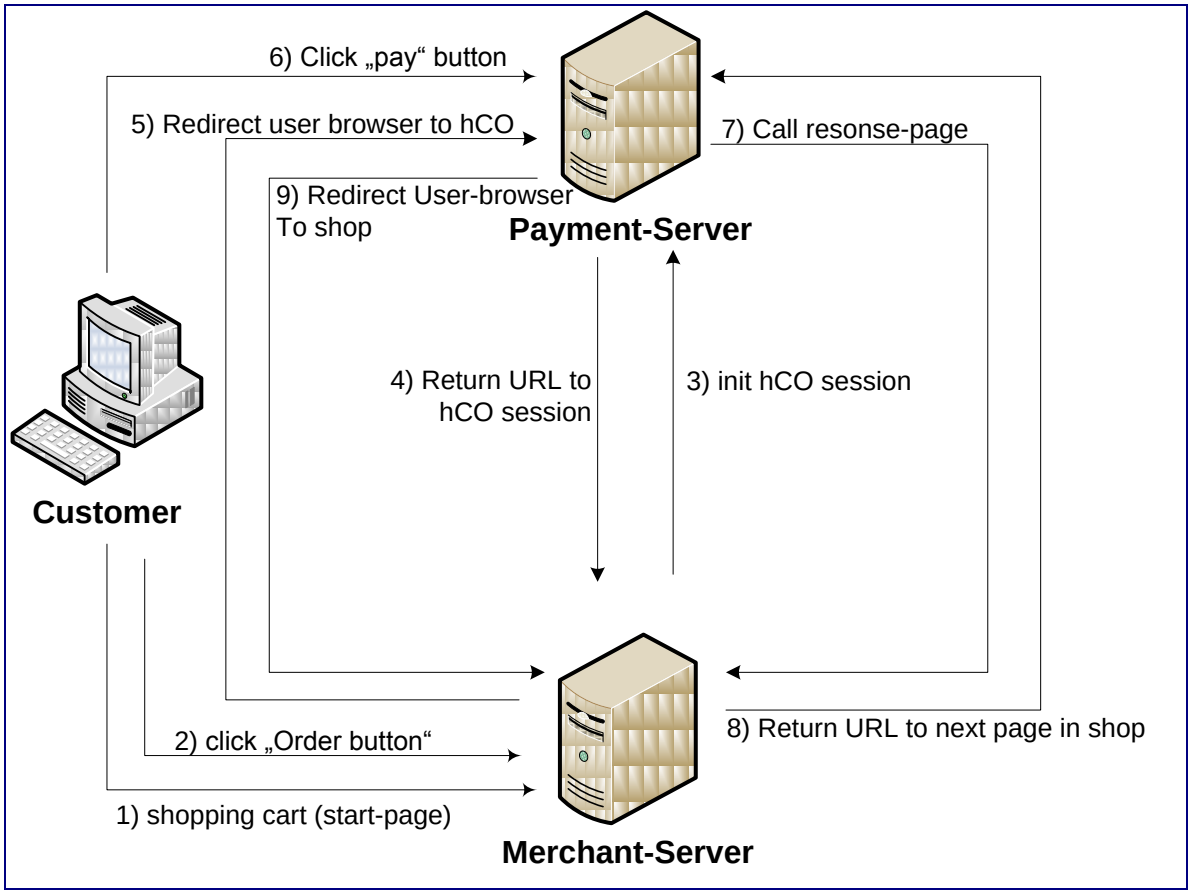
If the payment request failed for some reason the shop must display an error-page like this:

Pic 10 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.

3.2 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 11 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.

3.3 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 here. 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.

As mentioned before a complete documentation of hCO is available under "*heidelpay-Developer Guide-hCO*".

3.4 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 heidelpay CheckOut 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 hCO can be configured anytime. See chapter **Fehler! Verweisquelle konnte nicht gefunden werden.** of this document for more details.

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 heidelpay CheckOut, 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.

4 hIP –Virtual Terminal (v-Terminal)

4.1 Introduction

In case you have ordered it at heidelpay there is an opportunity for you to submit transactions manually to the payment-system by using heidelpay's "Virtual Terminal" (v-Terminal) which is accessible as another tabulator within the hIP.

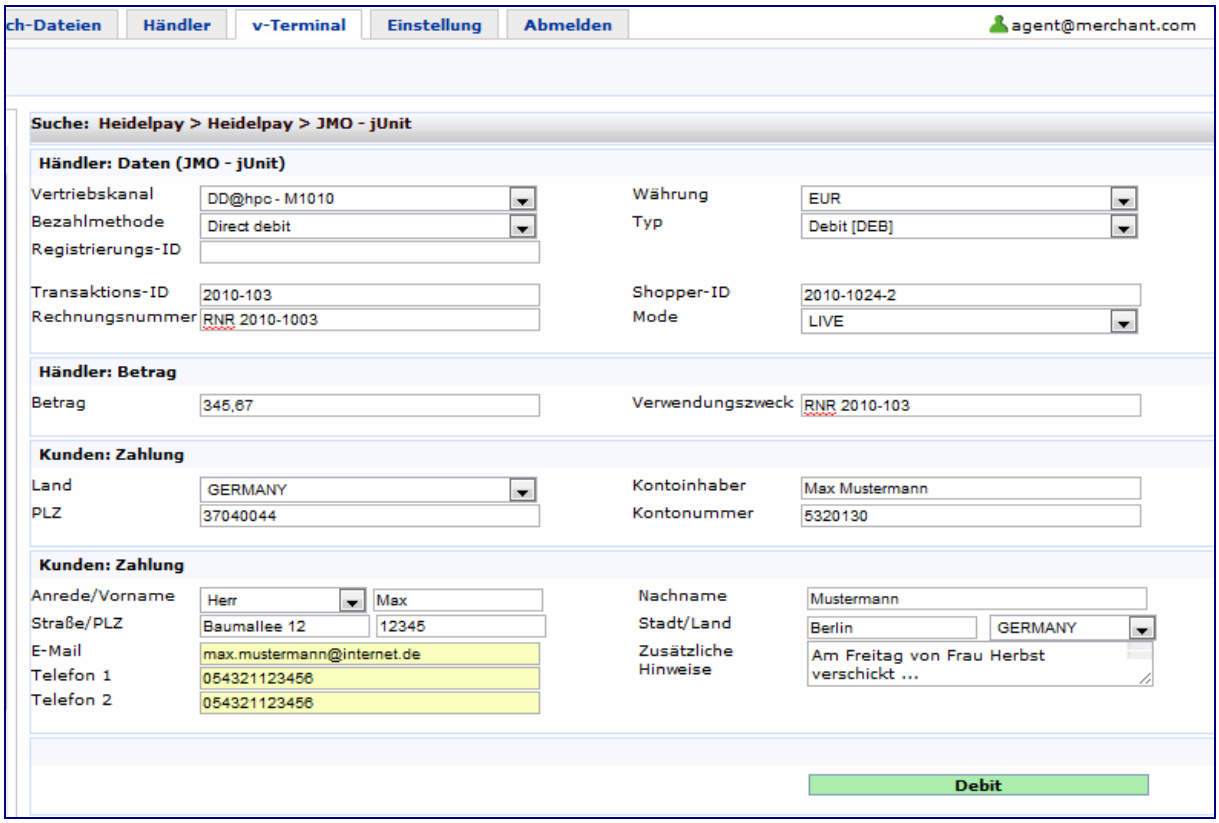
v-Terminal is the right tool when you need to process orders and transactions in cases where the customer places the order by telephone or the customer is not capable to use your website.

Using v-Terminal is straight-forward and mostly self-explaining:

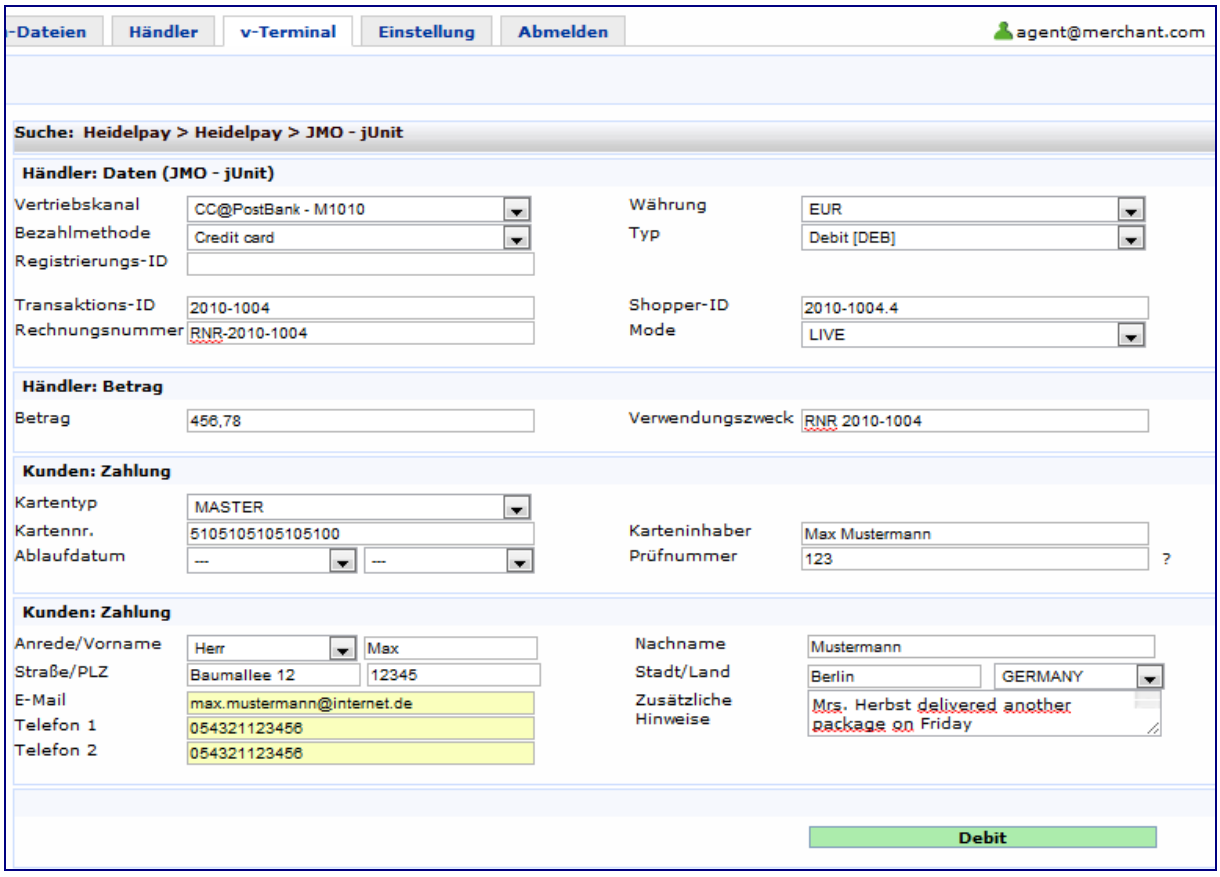
you can enter all data concerning your order and subsequently enter the data of the payment-instrument as provided to you by the customer.

4.2 Examples

The following pictures shows you examples of transactions using direct-debit an credit-cards:



Pic 12 v-Terminal Direct Debit



Pic 13 v-Terminal Credit card debit

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 6 *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 7 *hCO* (TEST)

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 8 *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 9 *hCO* (LIVE)

B 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:

B.1 Credit Cards

Brand	Number	Annotation	Valid through	CVV
American Express	375000000000001		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 10 Credit Cards (Test-numbers)

B.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 11 Debit Cards (Test-numbers)

B.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 12 Direct Debit Account-numbers (Test-accounts)

B.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 13 Online Transfer Account-numbers (Test-accounts)

B.5 BillSAFE Test data

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

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

B.6 PayPal

Important Note: For testing purpose with PayPal you need an Developer Account, you can yourself apply under https://developer.paypal.com/ for that.

C Index