developer guide

XML-Query

heidelpay payment platform

The complete solution for Ecommerce

Date: 28.11.2012

Version: 1.3

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	14.08.2012	Fredrich	Minor wording corrections
1.3	27.11.2012	Fredrich	Minor corrections (return code)

Document history

Table of content

1 General

This document describes the interface ***XML-Query*** which is designed for retrieving transaction data from heidelpay's payment-system. It has been designed in such a way that the retrieval and processing of the data can be fully automated and there is no need for manual intervention.

1.1 URL

In order to retrieve the requested data from the HeidelPay system an XML message has to be sent via HTTP-POST within a parameter called „load" to the following URL:

Test-URL: https://test-heidelpay.hpcgw.net/TransactionCore/xml
Live-URL: https://heidelpay.hpcgw.net/TransactionCore/xml

As a result of this request you will either get an XML stream including all of the transactions that fulfil your filter criteria.

1.2 Request Description

The Request version must be specified and is currently "1.0" always. The Header element is described in more detail in the XML Integrator (Transactions) documentation.
Query describes the mode for which you want to receive transaction data and for which entity. You can query for any defined entity that exists in your account. Please be aware that if you define for example the level to be CHANNEL the ID of the entity must be a channel ID.
The element Type describes which transaction types you might query from the gateway. At the moment six types are available for searches over a maximum period of one month, meaning the difference between "from" and "to" in the Period tag must be lower than or equal to one month:

- Chargeback (CB)
- Chargeback Reversal (CR)
- Confirmation (CF)
- Deregistration (DR)
- Receipt (RC)
- Reconcile (RL)

All other types can only be queried over a period of one day, meaning "from" and "to" in the Period tag must be the same.

IMPORTANT: The maximum number of transactions per query is limited depending on the types and period you are querying for:

- The maximum number of results per query in general is 10.000.
- In case you query for a longer period than one week or you do not specify a period at all the maximum number is 2.000.
- In case you query for a type none of CB, CR, CF, DR, RC, RL, the maximum number is 2.000.
- In case you search for "TransactionID", the maximum number is 200.

HeidelPay reserves the right to block the query interface for single merchants only in case the query API is misused. See chapter 4 for typical use cases and recommended queries.

1.3 Simple Request Example

```
<Request version="1.0">
	<Header>
		<Security sender="ff80808109c5bcc00109c5bce9f1003a"/>
	</Header>
	<Query entity="ff80808109c5bcc00109c5bce9f50056" level="CHANNEL"
	mode="INTEGRATOR_TEST" type="STANDARD">
		<User login="ff80808109c5bcc00109c5bce9f20092" pwd="geheim"/>
		<Period from="2006-03-04" to="2006-03-04"/>
		<Types>
			<Type code="RF"/>
			<Type code="PA"/>
			<Type code="RV"/>
		</Types>
	</Query>
</Request>
```

1.4 Simple Response Example 1

```
<Response version="1.0">
	<Result response='SYNC' type="LIST">
		<Transaction .../>
		<Transaction .../>
		<Transaction .../>
		...
	</Result>
</Response>
```

1.5 Simple Response Example 2

```
<Response>
	<Error>
		<Timestamp>2006-03-04 15:50:35</Timestamp>
		<Return code="100.200.101">invalid period, if searching for non-
		backchannel-types(CB, RC, RL, CF, DR) from and to must be on the same
		day</Return>
	</Error>
</Response>
```

2 Request Message

2.1 Header Group

The header group of the XML request holds transmission and security related information.

```
<Header>
      <Security sender="123a456b789c123d456e789f012g345”/>
</Header>
```

Value for Sender	Description
Alphanumeric 32	Each Server which sends requests to the system has an own sender unique ID. The sender UID is no logical business orientated subdivision like the channel ID, but refers to physical installations of software. Please provide here the value you have received from the customer support department.

2.2 Query Group

```
<Query mode="LIVE" level="CHANNEL" entity="678a456b789c123d456e789f012g432"
type="STANDARD" maxCount=”5”>
```

The Query tag has three attributes which determine the processing of the query.

Value for mode	Description
INTEGRATOR_TEST	Transaction is just send to the Integrator and not to the Validator (Risk Management) or Connector modules. Used to test compliance against the Integrator module
CONNECTOR_TEST	Transaction enters the Integrator module, accesses the Validator modules (Risk Management) and then goes to the Connector. The Connector operates in test mode
LIVE	Transaction enters the Integrator module, accesses the Validator modules (Risk Management) and then goes to the Connector. The Connector operates in live mode.
Value for level	**Description**
CHANNEL	Transactions are queried on channel level (id specified with entity is a channel id)
MERCHANT	Transactions are queried on merchant level (id specified with entity is a merchant id)
PSP	Transactions are queried on PSP level (id specified with entity is a PSP id)

Value for type	Description
STANDARD	This is the default type. No special behavior activated.

ACTIVE_TRANSACTIONS	Queries all transactions that match the given query parameters and that are active Transactions of their sessions.
LINKED_TRANSACTIONS	Queries all transactions that are linked directly or indirectly to one specific Transaction. For this type Id or ShortId of one transaction must be specified.
AVAILABLE_TRANSACTIONS	Queries all transactions that match the given query parameters and that are available transactions of their sessions. The difference between ACTIVE_TRANSACTIONS and AVAILABLE_TRANSACTIONS and applies to registrations only: active are only confirmed registrations, available are not confirmed registrations also.
ACTIVE_LINKED_TRANSACTIONS	Queries all transactions that are linked directly or indirectly to one specific Transaction and are active Transactions in their session. For this type Id or ShortId of one transaction must be specified.
AVAILABLE_LINKED_TRANSACTIONS	Queries all transactions that are linked directly or indirectly to one specific Transaction and are active Transactions in their session. For this type Id or ShortId of one transaction must be specified. The difference between ACTIVE_LINKED_TRANSACTIONS and AVAILABLE_LINKED_TRANSACTIONS applies to registrations only: active are only confirmed registrations, available are not confirmed registrations also.
Value for maxCount	**Description**
Number	Number of maximum returned transaction by the query. Especially helpful if you want to get the last 10 transaction only for an overview or similar to speedup your application.
Value for entity	**Description**
Alphanumeric 32	ID of the entity specified in level. The entity ID is a unique key for the identification of the unit which sends transactions into the system.

In case you want to query SCHEDULER generated transactions only (See Documents "XML_Transactions", Chapter Recurrence and "Recurrence Scenarios" for details) use the additional attribute "source":

```
<Query mode="LIVE" level="CHANNEL" entity="678a456b789c123d456e789f012g432"
type="STANDARD" source="SCHEDULER">
```

Value for source	Description
SCHEDULER	Retrieve only transactions that were system generated by the internal scheduler (recurrent payment)

2.3 User Group

The User tag group contains security related information. Check the document "Technical Quickstart" for information on how to retrieve this login data for your test account. Contact your account manager for the live account data.

```
<User login="421a456b789c123d456e789f012g098" pwd="56b789c123d456e789f"/>
```

Value for login	Description
Alphanumeric 32	The login is a unique ID for each human or system user. Each merchant or payment service provider can have several logins for system users and

	human users. It is not recommended to share one login between several human users.

Value for password	Description
Alphanumeric 32	A password which fits the login UID has to be provided. It is distributed together with the login UID

2.4 Identification Group

The identification group contains IDs which are used for the identification of the transaction:

- Unique ID
- Short ID
- TransactionID

```
<Identification>
      <UniqueID>12345678901234567890123456789012</UniqueID>
      <ShortID>1234.1234.1234</ShortID>
      <TransactionID>Merchant_Defined_ID_For_A_Transaction</ TransactionID >
</Identification>
```

2.4.1 Search for UniqueID or ShortID

This group allows the user to query for specific transactions if the ShortId or the UniqueId is known. Both ids are part of each XML Transaction Response. Only one of the two ids has to be specified. If either UniqueID or ShortID is present, Period and Types are optional.

Tag of Identification	Data Type	Length	Mandatory / Optional	Description
UniqueID	Alphanumeric	32	Conditional Mand.	Only ID where the uniqueness within the system is absolutely guaranteed. Has to be used for all automated matching and reference purposes
ShortID	Numeric / Dots	14	Conditional Mand.	ID which is used for manual entry and search purposes. The likelihood for uniqueness is very high, but not guaranteed.

2.4.2 Search for more than one UniqueID at the same time

This group allows the user to query for specific transactions the UniqueIds are known and return them all at once.
Period and Types are optional.

Tag of Identification	Data Type	Length	Mandatory / Optional	Description

UniqueIDs	Alphanumeric	32	Conditional Mand.	Contains a list of <ID> sub elements
ID	Alphanumeric	32	Conditional Mand.	Only ID where the uniqueness within the system is absolutely guaranteed. Has to be used for all automated matching and reference purposes

Example:

```
<Identification>
	<UniqueIDs>
		<ID>12345678901234567890123456789012</ID>
		<ID>ff8081811bfa5356011bfbe22a2800b4</ID>
	</UniqueIDs>
</Identification>
```

2.4.3 Search for TransactionID

This group also allows the user to query for specific transactions by the TransactionID. This id is the ID the merchant has assigned to a specific payment transaction.

Tag of Identification	Data Type	Length	Mandatory / Optional	Description
TransactionID	Alphanumeric	32	Optional	ID the merchant has assigned to a specific payment transaction and sent in with the original payment transaction.

IMPORTANT: This Id does not necessarily have to be unique when it is sent into the payment system with a payment transaction. Nevertheless, it should be, otherwise this query does not make too much sense. To prevent misuse of the query interface, this query allows you a maximum of retrieve 200 transactions per TransactionID.

2.5 TransactionType Group

The TransactionType group contains is an alternative and/or extension to the Methods group. It allows you to define what types if transactions your query result should contain.

```
<TransactionType>PAYMENT</TransactionType>
```

Value of TransactionType	Description
PAYMENT	Payment Transaction Types. Those are CB, CD, CR, DB, CP, PA, RB,RC, RF, RL and RV
REGISTER	Registration Transaction Types. Those are CF, DR, RG and RR
SCHEDULE	Scheduling Transaction Types. Those are DS, RS and SD
RISKMANAGEMENT	Risk Management Transaction Types. Those are EA, RI, 3D, SA, andIC. This list is always extending as new features are implemented.

2.6 Period Group

This tag group specifies the date period of the query. The "from" date is always extended with the time 00:00:00 and the "to" date with 23:59:59. All times are UTC only.

For the following payment types the "from" and "to" date can be any day:

- Chargeback (CB)
- Chargeback Reversal(CR)
- Confirmation (CF)
- Deregistration (DR)
- Receipt (RC)
- Reconcile (RL)

For all other payment types the "from" and "to" date have to be on the same day. This means only one day can be queried within one XML query request.

```
<Period from="2009-09-01" to="2009-09-15 12:00:00"/>
```

Value of "from"	Description
Date, format yyyy-MM-dd [HH:mm:ss]	Date from when the query starts.
Value of "to"	**Description**
Date, format yyyy-MM-dd [HH:mm:ss]	Date from when the query ends.

2.7 Methods Group

The Methods tag group contains information about the payment methods you want to retrieve.

```
<Methods>
	<Method code="CC"/>
	<Method code="DD"/>
</Methods>
```

Value for code	Description
Alpha(2)	A valid payment method. For a complete list of payment methods refer to the document "XML_Transactions"

2.8 Types Group

The Types tag group contains information about the payment types you want to retrieve.

```
<Types>
	<Type code="CB"/>
	<Type code="CF"/>
</Types>
```

Value for code	Description
Alpha(2)	A valid payment type. For a complete list of payment types refer to the document "XML_Transactions"

2.9 ProcessingResult Group

The ProcessingResult tag group contains information if you want to query for successful of failed transactions only. It is the same tag you would receive in an XML payment response as part of the Processing Tag Group (See document XML_Transactions)

```
<ProcessingResult>ACK</ProcessingResult>
```

Value for ProcessingResult	Description
ACK	Get successful Transactions only
NOK	Get failed (rejected) Transactions only

2.10 Account Group

The account group allows you to search for user account. This tag group is only needed if you search for registered user accounts.

```
<Account>
      <Id>joe.doe@email.net</Id >
      <Password>pwd12345</Password>
      <Brand>TESTWALLET</Brand>
</Identification>
```

Tag of Identification	Data Type	Length	Mandatory / Optional	Description
Id	Alphanumeric	128	Mandatory	Id of the user with the User Account
Brand	Alphanumeric	32	Mandatory	Brand of the User Account
Password	Alphanumeric	32	Mandatory	Password of the registered user

3 Response Message

The result to a query request message is – like the response to an XML transaction request – surrounded by a <Response> tag. The response tag contains either a Result Group or, in case of an invalid request, an Error Group.

3.1 Result Group

```
<Result response='SYNC' type="LIST">
```

Value for response	Description
SYNC	All requested transactions are part of this request and are nested inside the Result Group
Value for type	**Description**
LIST	Specifies that the transactions are provided as a list inside the Result group

3.2 Error Group

```
<Error>
      <Timestamp>2009-11-11 11:11:11</Timestamp>
      <Return code="100.100.101">invalid entity type</Return>
</Error>
```

Tag of error	Description
Timestamp	Date and Time of the request, format is "yyyy-MM-dd hh:mm:ss"
Return	Contains error information describing the occurred error as text
Value for code	**Description**
Alphanumeric, format is nnn.nnn.nn	Return code identifying the error that occurred

3.3 Transaction Group

Check the document "XML Transactions" for more details about the structure of the Transaction group.

4 Typical Example Requests and Responses

4.1 Chargebacks (CB)

For retrieving chargebacks for a certain period a request has to be send to the query engine at:

Test-URL: https://test-heidelpay.hpcgw.net/TransactionCore/xml
Live-URL: https://heidelpay.hpcgw.net/TransactionCore/xml

Typical Use Case: Automatically retrieve chargebacks from the payment system.
Depending on the acquirer or bank new Chargebacks usually appear once a day in the system. For details contact your account manager. Therefore it does not make sense to query the system for new Chargebacks every five minutes or the like. In case the query API is misused for something like this, merchant query access might be blocked.

```
<Request version="1.0">
	<Header>
		<Security sender="123a456b789c123d456e789f012g345"/>
	</Header>
	<Query mode="LIVE" level="CHANNEL"
	entity="678a456b789c123d456e789f012g432"
	type="STANDARD">
		<User login="peter" pwd="thewolf"/>
		<Period from="2009-09-01" to="2009-09-07"/>
		<Types>
			<Type code="CB"/>
		</Types>
	</Query>
</Request>
```

This request will retrieve the chargebacks for the channel identified by the id 678a456b789c123d456e789f012g432 for the week of the 1st of September to the 7th of September. Let's assume 2 chargebacks have been issued during that period. The result might look something like this:

```
<Response version="1.0">
  <Result response="SYNC" type="LIST">
    <Transaction mode="LIVE" response="SYNC"
      channel="678a456b789c123d456e789f012g432">
      <Identification>
        <TransactionID>4711</TransactionID>
        <UniqueID>h987i654j321k0981765m432n210o987
        </UniqueID>
        <ShortID>1234.5678.9876</ShortID>
        <ReferenceID>m123n456o789p876q543r210s123t456
        </ReferenceID>
      </Identification>
      <Processing code="DD.CB.00.00">
        <Timestamp>2009-09-02 14:58:07</Timestamp>
        <Result>ACK</Result>
        <Status code="00">SUCCESS</Status>
        <Reason code="40">Revocation or Dispute</Reason>
        <Return code="040.000.000">Transaction succeeded</Return>
      </Processing>
      <Account>
        <Number>******6581</Number>
```

```
        <Holder>Max Mustermann</Holder>
        <Bank>38050000</Bank>
        <Country>DE</Country>
      </Account>
      <Payment code="DD.CB">
        <Presentation>
          <Amount>1.00</Amount>
          <Currency>EUR</Currency>
        </Presentation>
        <Clearing>
          <Amount>1.00</Amount>
          <Currency>EUR</Currency>
          <Date>2009-09-02</Date>
        </Clearing>
      </Payment>
    </Transaction>
    <Transaction mode="LIVE" response="SYNC"
      channel="678a456b789c123d456e789f012g432">
      <Identification>
        <TransactionID>4712</TransactionID>
        <UniqueID>h987i654j321k0981765m43333222111
        </UniqueID>
        <ShortID>1234.5678.8191</ShortID>
        <ReferenceID>m123n456o789p876q543r21111222333
        </ReferenceID>
      </Identification>
      <Processing code="DD.CB.00.00">
        <Timestamp>2009-09-07 14:58:07</Timestamp>
        <Result>ACK</Result>
        <Status code="00">SUCCESS</Status>
        <Reason code="40">Revocation or Dispute</Reason>
        <Return code="040.000.000">Transaction succeeded</Return>
      </Processing>
      <Account>
        <Number>******6581</Number>
        <Holder>Max Mustermann </Holder>
        <Bank>38050000</Bank>
        <Country>DE</Country>
      </Account>
      <Payment code="DD.CB">
        <Presentation>
          <Amount>3.00</Amount>
          <Currency>EUR</Currency>
        </Presentation>
        <Clearing>
          <Amount>3.00</Amount>
          <Currency>EUR</Currency>
          <Date>2009-09-07</Date>
        </Clearing>
      </Payment>
    </Transaction>
  </Result>
</Response>
```

Please be aware that it only makes sense to execute this query once every week if you set the period like above.

4.2 Confirmations (CF)

Typical Use Case: Automatically retrieve confirmations from the payment system.

For processes where a Registration (RG) is not auto-confirmed (because of further risk management processes or because of a mandate based direct debit process) Confirmations are sent in any time after the Registration. Confirmations are for example the only way to see if a mandate based direct debit process has been registered by the client. Only if you receive a confirmation that you can match to a registration you can start sending real payment transactions into the gateway. Consequently you should query for confirmations frequently (e.g. once a day).

```
<Request version="1.0">
  <Header>
    <Security sender="123a456b789c123d456e789f012g345" />
  </Header>
  <Query mode="LIVE" level="CHANNEL" entity="678a456b789c123d456e789f012g432"
    type="STANDARD">
    <User login="peter" pwd="thewolf" />
    <Period from="2009-09-08" to="2009-09-08" />
    <Types>
      <Type code="CF" />
    </Types>
  </Query>
</Request>
```

This request will retrieve the confirmations for the channel identified by the id 678a456b789c123d456e789f012g432 for the 8th of September. Let's assume 3 confirmations have been received on that day. The result might look something like this:

```
<Response version="1.0">
  <Result response="SYNC" type="LIST">
    <Transaction mode="LIVE" response="SYNC"
      channel="678a456b789c123d456e789f012g432">
      <Identification>
        <TransactionID>reg 0815</TransactionID>
        <UniqueID>h987i654j321k0981765m432n210o987
        </UniqueID>
        <ShortID>1234.5678.9876</ShortID>
      </Identification>
      <Processing code="DD.CF.00.00">
        <Timestamp>2009-09-08 14:58:07</Timestamp>
        <Result>ACK</Result>
        <Status code="00">SUCCESS</Status>
        <Reason code="00">Successful Processing</Reason>
        <Return code="000.000.000">Transaction succeeded</Return>
      </Processing>
      <Payment code="DD.CF" />
      <Account id="678a456b789c123d456e789f0123333" />
    </Transaction>
    <Transaction mode="LIVE" response="SYNC"
      channel="678a456b789c123d456e789f012g432">
      <Identification>
        <TransactionID>reg 0816</TransactionID>
        <UniqueID>h987i654j321k0981765m432n2888999
        </UniqueID>
        <ShortID>1234.5678.0203</ShortID>
      </Identification>
      <Processing code="DD.CF.00.00">
        <Timestamp>2009-09-08 14:58:23</Timestamp>
        <Result>ACK</Result>
        <Status code="00">SUCCESS</Status>
        <Reason code="00">Successful Processing</Reason>
        <Return code="000.000.000">Transaction succeeded</Return>
      </Processing>
      <Payment code="DD.CF" />
```

```
      <Account id="678a456b789c123d456e789f0124545" />
    </Transaction>
    <Transaction mode="LIVE" response="SYNC"
      channel="678a456b789c123d456e789f012g432">
      <Identification>
        <TransactionID>reg 0817</TransactionID>
        <UniqueID>h987i654j321k0981765m432n2321321
        </UniqueID>
        <ShortID>1234.5678.0198</ShortID>
      </Identification>
      <Processing code="DD.CF.00.00">
        <Timestamp>2009-09-08 14:58:55</Timestamp>
        <Result>ACK</Result>
        <Status code="00">SUCCESS</Status>
        <Reason code="00">Successful Processing</Reason>
        <Return code="000.000.000">Transaction succeeded</Return>
      </Processing>
      <Payment code="DD.CF" />
      <Account id="678a456b789c123d456e789f07744411" />
    </Transaction>
  </Result>
</Response>
```

In order to match the confirmation you have to check for: Response/Transaction/Account/@id in your system.

4.3 Deregistration (DR)

Typical Use Case: Automatically retrieve deregistered registrations from the payment system.
Depending on the acquirer or bank new Chargebacks usually appear once a day in the system. For the deregistration process of an account can happen in two ways:

- The client uses your system to do so and your system triggers a deregistration request
- Also it can happen externally where the client advises his bank to reject transactions

For external mandates the HeidelPay system polls regularly for newly cancelled mandates on the banks side. If a mandate has been cancelled externally a deregistration transaction is created that can be retrieved via the query interface. All transactions that reference an deregistered account will be declined by the gateway.
Consequently you should regularly poll for new deregister (DR) events. Commonly a daily polling is appropriate.

```
<Request version="1.0">
  <Header>
    <Security sender="123a456b789c123d456e789f012g345" />
  </Header>
  <Query mode="LIVE" level="CHANNEL" entity="678a456b789c123d456e789f012g432"
    type="STANDARD">
    <User login="peter" pwd="thewolf" />
    <Period from="2009-09-08" to="2009-09-08" />
    <Types>
      <Type code="DR" />
    </Types>
  </Query>
</Request>
```

This request will retrieve the deregistrations for the channel identified by the id 678a456b789c123d456e789f012g432 for the 8th of September. Let's assume 1 confirmation has been received on that day. The result might look like this:

```
<Response version="1.0">
  <Result response="SYNC" type="LIST">
    <Transaction mode="LIVE" response="SYNC"
      channel="678a456b789c123d456e789f012g432">
      <Identification>
        <TransactionID>reg 0815</TransactionID>
        <UniqueID>h987i654j321k0981765m432n210o987</UniqueID>
        <ShortID>1234.5678.9876</ShortID>
      </Identification>
      <Processing code="DD.DR.00.00">
        <Timestamp>2009-09-08 14:58:07</Timestamp>
        <Result>ACK</Result>
        <Status code="00">SUCCESS</Status>
        <Reason code="00">Successful Processing</Reason>
        <Return code="000.000.000">Transaction succeeded</Return>
      </Processing>
      <Payment code="DD.DR" />
      <Account id="678a456b789c123d456e789f0123333" />
    </Transaction>
  </Result>
</Response>
```

In order to match deregistrations you have to check for: Response/Transaction/Account/@id in your system.

4.4 Receipts (RC)

Typical Use Case: Automatically retrieve if somebody has transferred money onto your bank account.
In order to check if a prepayment (PP) has been fulfilled you have to retrieve receipts. Again a request has to be sent to the query engine at:

Test-URL: https://test-heidelpay.hpcgw.net/TransactionCore/xml
Live-URL: https://heidelpay.hpcgw.net/TransactionCore/xml

```
<Request version="1.0">
  <Header>
    <Security sender="123a456b789c123d456e789f012g345" />
  </Header>
  <Query mode="LIVE" level="CHANNEL" entity="678a456b789c123d456e789f012g432"
    type="STANDARD">
    <User login="peter" pwd="thewolf" />
    <Period from="2009-09-01" to="2009-09-07" />
    <Types>
      <Type code="RC" />
    </Types>
  </Query>
</Request>
```

This request will retrieve the receipts for the channel identified by the id 678a456b789c123d456e789f012g432 for the week of the 1st of September to the 7th of September. Let's assume that one receipt has been issued during that period. The result might look like this:

```
<Response version="1.0">
  <Result response="SYNC" type="LIST">
    <Transaction mode="LIVE" response="SYNC"
      channel="678a456b789c123d456e789f012g432">
      <Identification>
        <TransactionID>4711</TransactionID>
        <UniqueID>h987i654j321k0981765m432n210o987
        </UniqueID>
        <ShortID>1234.5678.9876</ShortID>
        <ReferenceID>m123n456o789p876q543r210s123t456
        </ReferenceID>
      </Identification>
      <Processing code="DD.RC.00.00">
        <Timestamp>2009-09-02 14:58:07</Timestamp>
        <Result>ACK</Result>
        <Status code="00">SUCCESS</Status>
        <Reason code="40">Revocation or Dispute</Reason>
        <Return code="040.000.000">Transaction succeeded</Return>
      </Processing>
      <Account>
        <Number>******6581</Number>
        <Holder>Max Mustermann</Holder>
        <Bank>38050000</Bank>
        <Country>DE</Country>
      </Account>
      <Payment code="DD.RC">
        <Presentation>
          <Amount>1.00</Amount>
          <Currency>EUR</Currency>
        </Presentation>
        <Clearing>
          <Amount>1.00</Amount>
          <Currency>EUR</Currency>
          <Date>2009-09-02</Date>
        </Clearing>
      </Payment>
    </Transaction>
  </Result>
</Response>
```

4.5 Query Specific Transactions by Id

Typical Use Case: Automatically retrieve transactions from the payment system where you only have an Id

In order to retrieve a transaction where the UniqueId or ShortId is known, only one transaction can be queried. For example this can be particularly usefull to retrieve a Registration of a user to be able to show the user the (masked) payment data or customer data he entered the last time he paid in the shop.

Another possibility is to use the sent in <TransactionID> for the query. This is very helpful in case you either do not have the UniqueID or ShortID anymore or you never received it (e.g. Timeout). In this case you are able to query the state of the transaction.

```
<Request version="1.0">
  <Header>
    <Security sender="123a456b789c123d456e789f012g345" />
  </Header>
  <Query mode="LIVE" level="CHANNEL" entity="678a456b789c123d456e789f012g432"
    type="STANDARD">
    <User login="peter" pwd="thewolf" />
```

```
    <Period from="2007-01-01" to="2007-01-07" />
    <Identification>
      <ShortID>1234.1230.4422</ShortID>
    </Identification>
  </Query>
</Request>
```

Typical response message for a Receipt:

```
<Response version="1.0">
  <Result response="SYNC">
    <Transaction mode="LIVE" response="SYNC"
      channel="678a456b789c123d456e789f012g432">
      <Identification>
        <TransactionID>4711</TransactionID>
        <UniqueID>h987i654j321k0981765m432n210o987</UniqueID>
        <ShortID>1234.5678.9876</ShortID>
        <ReferenceID>m123n456o789p876q543r210s123t456</ReferenceID>
      </Identification>
      <Processing code="DD.RC.00.00">
        <Timestamp>2009-09-02 14:58:07</Timestamp>
        <Result>ACK</Result>
        <Status code="00">SUCCESS</Status>
        <Reason code="40">Revocation or Dispute</Reason>
        <Return code="040.000.000">Transaction succeeded</Return>
      </Processing>
      <Payment code="DD.RC">
        <Presentation>
          <Amount>1.00</Amount>
          <Currency>EUR</Currency>
        </Presentation>
        <Clearing>
          <Amount>1.00</Amount>
          <Currency>EUR</Currency>
          <Date>2009-09-02</Date>
        </Clearing>
      </Payment>
    </Transaction>
  </Result>
</Response>
```

Typical response message for a Debit:

```
<Response>
  <Result response="SYNC" count="1">
    <Transaction mode="INTEGRATOR_TEST" channel="ff80808112fc2f530112fc2f63e7000f"
      response="SYNC" source="XML">
      <Identification>
        <ShortID>9663.7985.4310</ShortID>
        <UniqueID>ff80808112f795b00112fc2f7859003f
        </UniqueID>
        <TransactionID>1174</TransactionID>
      </Identification>
      <Relevances />
      <Payment code="CC.PA">
        <Clearing>
          <Amount>73.50</Amount>
          <Currency>EUR</Currency>
          <Descriptor>9663.7985.4310 TPX_order# 222
            PSP_A/MER_A/DEFAULT</Descriptor>
          <FxRate>1.0</FxRate>
```

```
          <FxSource>INTERN</FxSource>
          <FxDate>2006-08-05 16:00:08</FxDate>
        </Clearing>
        <Presentation>
          <Amount>73.50</Amount>
          <Currency>EUR</Currency>
          <Usage>TPX_order# 222</Usage>
        </Presentation>
      </Payment>
      <Account>
        <Number>******1881</Number>
        <Holder>Max Mustermann</Holder>
        <Brand>VISA</Brand>
        <Expiry month="12" year="2014" />
      </Account>
      <Customer>
        <Name>
          <Family>kosel</Family>
          <Given>bobby</Given>
          <Company>kosel co.</Company>
          <Salutation>MR</Salutation>
          <Title></Title>
        </Name>
        <Contact>
          <Email>bob_kosel@googlemail.com</Email>
          <Ip>101.202.011.022</Ip>
          <Mobile>0049 199 6542123</Mobile>
          <Phone>0049 199 6542123</Phone>
        </Contact>
        <Address>
          <City>Frankfurt</City>
          <Country>DE</Country>
          <State>DE7</State>
          <Street>Hauptstrasse</Street>
          <Zip>61821</Zip>
        </Address>
      </Customer>
      <Processing code="CC.PA.90.00">
        <Timestamp>2006-08-05 16:00:08</Timestamp>
        <Result>ACK</Result>
        <Status code="90">NEW</Status>
        <Reason code="00">Successful Processing</Reason>
        <Return code="000.100.110">Request successfully processed in 'Merchant
          in
          Integrator Test Mode'</Return>
      </Processing>
      <RequestTimestamp>2006-08-05 16:00:08</RequestTimestamp>
      <Analysis />
    </Transaction>
  </Result>
</Response>
```

4.6 Query User Account (Login for User Accounts)

Typical Use Case: Automatically login a wallet user.
In order to retrieve a the User Account registration of a user by Id (Username) and Password, you need to specify the Account tag group with Id, Password and Brand.

```
<?xml version='1.0' encoding='utf-8'?>
```

```
<Request version='1.0'>
  <Query entity='ff8080811077e476011077e6419c00fd' level='CHANNEL'
mode='INTEGRATOR_TEST'
    type="ACTIVE_TRANSACTIONS">
    <User login='ff8080811077e476011077e6419900fa' pwd='demo' />
    <Account>
      <Id>test</Id>
      <Password>test</Password>
      <Brand>TESTUACCBRAND</Brand>
    </Account>
  </Query>
  <Header
    <Security sender='ff8080811077e476011077e6419800f8' type='MERCHANT' />
  </Header>
</Request>
```

Typical response message:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Response>
  <Result response="SYNC" count="1">
    <Transaction mode="INTEGRATOR_TEST" channel="ff8080811077e476011077e6419c00fd"
      response="SYNC" source="WPF">
      <Identification>
        <ShortID>2397.9030.7378</ShortID>
        <UniqueID>ff808081110d897f01110d8e84a10002
        </UniqueID>
        <TransactionID>demopsp</TransactionID>
      </Identification>
      <Payment code="UA.RG" />
      <Account>
        <Id>******test</Id>
        <Holder>student tester</Holder>
        <Brand>TESTUACCBRAND</Brand>
      </Account>
      <Customer>
        <Name>
          <Family>tester</Family>
          <Given>student</Given>
          <Company />
          <Salutation>MR</Salutation>
          <Title />
        </Name>
        <Contact>
          <Email>user@somewhere.org</Email>
          <Ip>123.123.123.123</Ip>
          <Mobile>+49-181-7654321</Mobile>
          <Phone>+49-112-11119999</Phone>
        </Contact>
        <Address>
          <City>Stadt</City>
          <Country>ZA</Country>
          <State />
          <Street>Street 3</Street>
          <Zip>80798</Zip>
        </Address>
        <Details>
          <Identity paper="IDCARD">xyz123</Identity>
        </Details>
      </Customer>
      <Processing code="UA.RG.90.00">
```

```
        <Timestamp>2007-03-01 13:51:55</Timestamp>
        <Result>ACK</Result>
        <Status code="90">NEW</Status>
        <Reason code="00">Successful Processing</Reason>
        <Return code="000.100.110">Request successfully processed in 'Merchant
          in
          Integrator Test Mode'</Return>
      </Processing>
      <RequestTimestamp>2007-03-01 13:51:54</RequestTimestamp>
      <Analysis>
        <Criterion name="jobtitle">manager</Criterion>
        <Criterion name="gender">male</Criterion>
        <Criterion name="age">20</Criterion>
      </Analysis>
    </Transaction>
  </Result>
</Response>
```

5 Error handling

If a query is rejected due to errors the Response will contain an Error-Tag.
The following example shows a request that is declined because the type is unknown (please be aware that codes and messages are subject to change):

```
<Request version="1.0">
  <Header>
    <Security sender="123a456b789c123d456e789f012g345" />
  </Header>
  <Query mode="LIVE" level="MERCHANT" entity="678a456b789c123d456e789f012g432"
type="STANDARD">
    <User login="peter" pwd="thewolf" />
    <Period from="2009-09-08 12:00:00" to="2009-09-08 13:00:00" />
    <Types type="LIST">
      <Type code="MM" />
    </Types>
  </Query>
</Request>
```

This request will yield an error message:

```
<Response version="1.0">
  <Error>
    <Timestamp>2009-11-11 11:11:11</Timestamp>
    <Return code="100.100.101">invalid entity type</Return>
  </Error>
</Response>
```

6 Request - DTD

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT Request (Header, Query)>
<!ATTLIST Request
version (1.0) #REQUIRED
>
<!ELEMENT Header (Security)>
<!ELEMENT Security EMPTY>
<!ATTLIST Security
sender CDATA #REQUIRED
>
<!ELEMENT Query (User, Period?, Identification?, Types?)>
<!ATTLIST Query
mode (LIVE | CONNECTOR_TEST | INTEGRATOR_TEST) #REQUIRED
level (CHANNEL | MERCHANT | PSP) #REQUIRED
entity CDATA #REQUIRED
>
<!ELEMENT User EMPTY>
<!ATTLIST User
login CDATA #REQUIRED
pwd CDATA #REQUIRED
>
<!ELEMENT Period EMPTY>
<!ATTLIST Period
from CDATA #REQUIRED
to CDATA #REQUIRED
>
<!ELEMENT Identification (UniqueID?, ShortID?)>
<!ELEMENT ShortID (#PCDATA)>
<!ELEMENT UniqueID (#PCDATA)>
<!ELEMENT Types (Type+)>
<!ELEMENT Type EMPTY>
<!ATTLIST Type
code CDATA #REQUIRED
>
```

7 Response-DTD

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT Response (Result | Error)>
<!ATTLIST Response
version CDATA #IMPLIED
>
<!ELEMENT Result (Transaction*)>
<!ATTLIST Result
response (SYNC) #IMPLIED
count CDATA #IMPLIED
type (LIST) #IMPLIED
>
<!-- for full definition of the Transaction element refer to the XML Transaction
documentation-->
<!ELEMENT Transaction (#PCDATA<!ATTLIST Transaction
response (SYNC | ASYNC) #IMPLIED
mode CDATA #IMPLIED
channel CDATA #IMPLIED
>
<!ELEMENT Error (Timestamp, Return)>
<!ELEMENT Timestamp (#PCDATA)>
<!ELEMENT Return (#PCDATA)>
<!ATTLIST Return
code CDATA #REQUIRED
>
```