

BORGUN

HEIMIR

USER GUIDE
VERSION 2.0,19.7.2011
Copyright © 2009 Borgun hf.

History

2007-09-18	PF	Added field lengths and data filed description
2009-05-04	BA	RRN search added to getTransactionList
2009-05-04	BA	Added disputed transaction chapter
2009-05-04	BA	Added settlement chapter
2009-08-15	BA	Added Card Information chapter
2009-08-28	BA	Added fraud transaction chapter
2010-05-06	BA	Clarification on minor issues
2011-01-15	BA	Numpayments/FirstPayments removed
2011-02-17	BA	Added getVirtualCard
2011-04-05	BA	RRN recommended value changed
2011-07-19	BA	Added FAQ, new template for look
2011-09-06	BA	Added getSettlementChargeback, Batch/Transaction changes
2011-11-30	BA	Added Partial Reversal

Table of Contents

About Heimir	3
Key features	3
Get authorizations.....	3
Cancel authorizations	3
Create new batches.....	3
Get batch authorizations.....	3
Get processing results	3
Get transaction lists	3
Send detail data.....	3
Introduction to batches	4
How does Heimir work?	4
Data types.....	5
Getting authorizations	6
Normal purchase transaction	6
Refund.....	6
Partial Reversal	6
Pre-authorizations.....	6
Getting authorizations – XML explanations	8
Getting authorizations – XML examples.....	11
Cancelling authorizations	13
Cancelling authorizations – XML explanations	13
Cancelling authorizations – XML examples.....	15
Virtual cardnumbers.....	16
Virtual cardnumbers – XML explanations.....	16
Cancelling authorizations – XML examples.....	16
Creating new batches	17
Batch numbers controlled by the Authorization Services.....	17
Batch numbers controlled by the merchant	17
Creating new batches – XML explanations	17
Creating new batches – XML examples	18
Getting processing results	19
Getting processing results – XML explanations.....	19
Getting processing results – XML examples	20
Getting transaction lists.....	21
Getting transaction lists – XML explanations.....	21
Getting transaction lists – XML examples.....	22
Getting Disputed Transactions	23
Getting disputed transactions.....	23
Getting disputed transaction list – XML examples	25
Getting Fraud Transactions.....	26
Getting fraud transactions.....	26
Getting fraud transaction list – XML examples	27
Getting Settlement Data	29
Getting settlement list	29
Getting a list of batches in settlement.....	29
Getting a list of transactions in batches. BatchDate and BatchNumber are unique for each batch.....	30
Getting a list of chargebacks in settlement.....	30
Request:	31
Response:	32
Request:	32
Response:	32
Sending detailed data	36
Sending detail data – XML explanations	36
Sending detail data – XML examples	40
Appendix	41
Action Codes	41
FAQ	43
Frequently Asked Questions.....	43

About Heimir

Heimir is a code name for a web service that allows a merchant to acquire authorizations for credit card transactions and to submit transactions for clearing and settlement.

Key features

Heimir's key features enable the user to do the following:

Get authorizations – allows the user to either make a *normal purchase transaction* or to pre-authorize it.

The difference is:

If a transaction is *pre-authorized*, the given amount is reserved from the cardholders account. The amount will not be charged to the cardholder until a financial transaction is submitted. The authorization code needs to be stored by the merchant and later used to send in the financial transaction. Financial transactions are submitted by calling `getAuthorization` with the `AuthCode` from a previous pre-authorize transaction.

With the *normal purchase transaction* the cardholders account is charged for the given amount. If the transaction was pre-authorized previously then the transaction must include the authorization code received from that transaction.

This feature also allows the user to make a refund transaction if he wants to refund an amount to the cardholders account.

Cancel authorizations – allows the user to cancel previously authorized transactions. Authorizations can be cancelled if the batch they belong to is still open. If it is closed then a refund transaction must be used.

Create new batches – allows the user to create a new batch. By doing so the user closes the currently open batch and opens a new one.

Get batch authorizations – allows the user to authorize many transactions as one batch.

Get processing results – allows the user to get results from previous batch job.

Get transaction lists – allows the user to get a transaction list from the authorization service listing on either batch or transaction level details about transactions processed through the service.

Send detail data – allows the user to supply detailed invoice after the financial transaction has been processed.

Introduction to batches

A batch is a unit that the acquirer uses to settle with the merchant. Its a collection of credit card transactions charged over a period of time, usually one day.

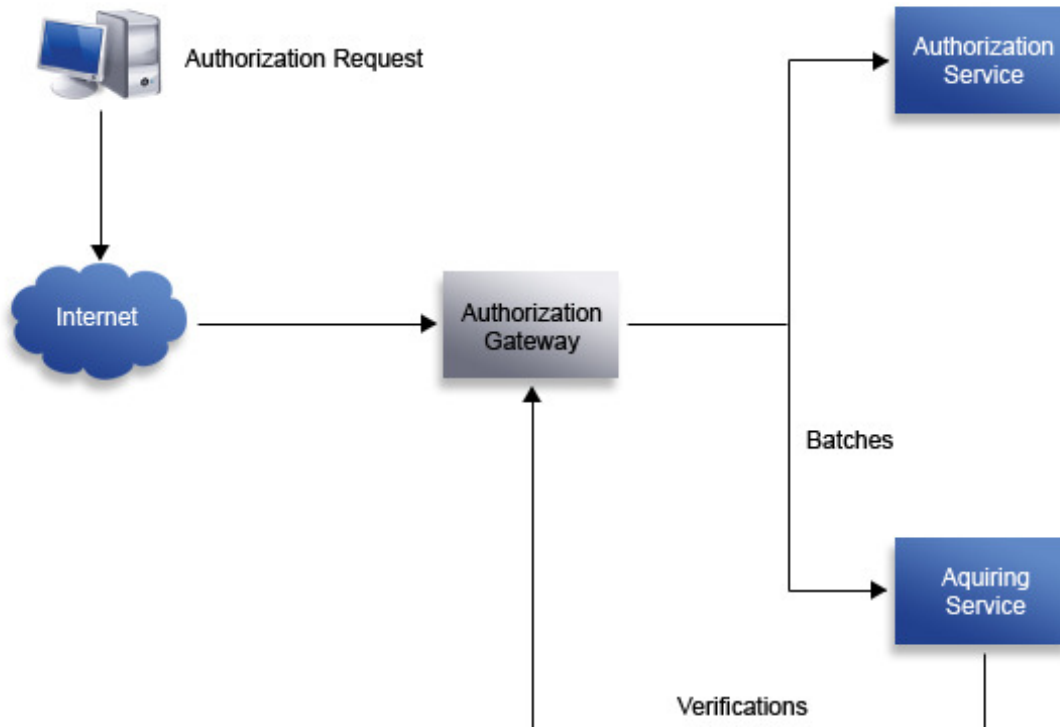
Each batch is identified by a batch number and this batch number must be unique per terminal within one year. The batch number can either be controlled by the merchant or the authorization services. This will be explained later.

How does Heimir work?

Heimir will expose the authorization services with the use of web services. The web services run securely over HTTPS.

All web services provided by Heimir will be password protected. The connecting merchant has to supply username and password each time the merchant calls a Heimir web service.

Once the merchant has connected to the web service he will be able to access the features mentioned above. Using these features the merchant has to send all necessary information on an XML format. This will be explained later. This XML information will then be sent through the web service and a response, also on XML format, will be given.



Model 1 – This diagram shows an example of Heimir’s process

Data Storage and Manipulation

The security of cardholder account data has become one of the biggest issues facing the payment card industry. Visa and MasterCard have therefore created the Payment Card Industry Data Security Standard (PCI DSS) – a set of industry-wide requirements and processes, which are supported by all international payment card systems.

PCI DSS is intended for any organization that stores, transmits or processes cardholder account and transaction data – merchants, acquiring banks and related service providers.

The PCI standard is available at <https://www.pcisecuritystandards.org> Please download and review the standard.

While all merchants are required to comply with the Payment Card Industry Data Security Standard, merchants that store, process or transmit card account data may also be required to validate compliance with their acquirer. Please note that for a merchant to be considered compliant, any Service Providers that store, process or transmit card account data on behalf of the merchant must also be compliant.

Data types

Field Form

A	Alphabetic
N	Numeric
AN	Alphanumeric
Z	ISO 7811 and ISO 7813 track 2 and track 3 data format

Field Length

10	Indicates length=10
LL..19	Indicates max length of 19
..17	Variable length up to 17 characters

Field Code

M	Mandatory
C	Conditional
O	Optional

Amounts

All amounts are represented with two digits after the decimal and an implied decimal point. Icelandic Krona is therefore represented in aurar. Dollars are in cents etc. For instance 100 USD is in TrAmount as “10000”.

Getting authorizations

Normal purchase transaction

This element is referred as TransType 1 in the XML sheet. If you want to make a normal purchase transaction you can either insert a variable in the AuthCode in the XML sheet or leave the field empty.

The difference is that if the AuthCode is empty Heimir will request a authorization from the issuer and if successful, Heimir will store the financial transaction in it's database and later submit the financial transaction to the acquirer automatically.

OR

If the transaction has already been authorized by the pre-authorizing element or by other channels the AuthCode will contain the auth code that was received during the pre-authorization process. This is sometimes referred to as submitting financial transactions. In this case Heimir will store the financial transaction in it's database and not request an authorization from the issuer to make sure the card account does not get reserved twice. Heimir will then later submit the transaction to the acquirer automatically.

In either case the transaction number and batch number are always returned from the web service. If Transaction and Batch variables are not set in input requests they are created and returned in the response.

Refund

This element is referred as TransType 3 in the XML sheet. This transaction type is used when merchant wants to refund an amount to customer. If the refund is successful the response to the merchant contains the AuthCode of the reversal, as well as Transaction and Batch numbers. In other case the response contains an error code.

Partial Reversal

This element is referred as TransType 4 in the XML sheet. This transaction type lowers the amount on a currently existing sale and advices the issuer to change the cardholders withdrawal limit by the same amount. The transaction type can only be used on a sale created on the same day and therefore in an open batch. The message must contain the original amount, and the new amount.

If the partial reversal is successful the response to the merchant contains the AuthCode of the reversal, as well as Transaction and Batch numbers. In other case the response contains an error code.

Pre-authorizations

This element is referred as TransType 5 in the XML sheet. This transaction type is used when the merchant wants to reserve an amount from the customer's credit card. An Authorization Request is then sent to the acquirer. If request was authorized the transaction is added to system database containing the new AuthCode. Transaction and batch numbers are not assigned to the transaction as the transaction is only a pre-authorization request. The response sent to the merchant therefore only contains AuthCode, not Transaction and Batch.

Getting authorizations – XML explanations

Table GA.1 – Field explanations for a sent XML

Field	Length	Form	Type	Explanation
Version	4	AN	Mandatory	Indicates the version of these specifications the client is using. Current version is "1000"
Processor	7	N	Mandatory	An ID issued by Borgun to the processor using this interface on behalf of the merchant
MerchantID	..8	AN	Mandatory	An Id in Heimir of the merchant requesting services
TerminalID	..8		Optional	A terminal id in Heimir for the merchant requesting services
TransType	1	N	Mandatory	1 = Normal purchase transaction 3 = Refund 4 = Partial Reversal 5 = Pre-Authorize
TrAmount	..12		Mandatory	Amount, note for example 100 USD is 10000 because the amount is represented in cents
NewAmount	..12		Mandatory	Only used by transtype 4. Contains the new amount to override the original sale amount.
TrCurrency	3	N	Mandatory	Transaction currency ISO 4217 352 => ISK 978 => EUR
DateAndTime	12	N	Mandatory	YYMMDDhhmmss. When repeated use original date and time
PAN	LL..19	N	Conditional	Card number – Used if a card number is entered manually. See also Track1/Track2. Card number can be a virtual card (see function getVirtualCard). If TransType is refund then PAN can be empty, PAN from a transaction of TransType=1 with the same DateAndTime + RRN will be used.
ExpDate	4	N	Conditional	Expiration date – YYMM not used if Track1/Track2
Track1	LL..6	AN	Conditional	Mandatory if magnetic stripe read and not Track2
Track2	LL--37	Z	Conditional	Mandatory if magnetic stripe read
RRN	12	AN	Mandatory	Retrieval Reference Number – Reference number for transaction. Must be exactly 12 characters . Recommended format is a fixed value followed by a sequence, for example ACME00012345. The last six letters must contain a numeric value.
AuthCode	..6	AN	Conditional	Authorization code of a previous pre-authorized transaction. Include this code to use the previously gotten authorization code
CVC2	3	N	Conditional	Used in cardholder not present situations such as internet sales. Code is printed on the back of the cardholder's card
SecurityLevelInd	1	AN	Conditional	Security Level Indicator – Used with Secure Code to indicate the security level used in

				<p>electronic transactions</p> <p>This property is mandatory in electronic commerce and is used to identify the level of security used when doing the transaction.</p> <p>0 = UCAF not supported by merchant 1 = UCAF supported by merchant but not provided by issuer 2 = UCAF is present</p> <p>Mapping from MPI is MPI(1) → 2 MPI(2,3,4) → 1 MPI(5,6,7,8) → 1</p>
UCAF	32	AN	<i>Conditional</i>	Universal Cardholder Authentication Field – Used with Secure Code
CAVV	28	AN	<i>Conditional</i>	Verified by Visa, Base64 encoded cardholder authentication verification value, f.ex AAACCRSQJ4lShoBncZAnAAAAA=
XID	28	AN	<i>Conditional</i>	Verified by Visa. The unique transaction action id identifier encode by Base64.

Table GA.2 – Field explanations for a received XML

Field	Type	Explanation
Version	<i>Mandatory</i>	Indicates the version of these specifications the client is using. Current version is “1000”
Processor	<i>Mandatory</i>	An ID issued by Borgun to the processor using this interface on behalf of the merchant
MerchantID	<i>Mandatory</i>	An Id in Heimir of the merchant requesting services
TerminalID	<i>Optional</i>	A terminal id in Heimir for the merchant requesting services
TransType	<i>Echo</i>	1 = Normal purchase transaction 3 = Refund 5 = Pre-Authorize
TrAmount	<i>Echo</i>	Amount, note for example 100 USD is 10000 because the amount is represented in cents
TrCurrency	<i>Echo</i>	Transaction currency ISO 4217 352 => ISK 978 => EUR
DateAndTime	<i>Echo</i>	YYMMDDhhmmss. When repeated use original date and time
PAN	<i>Conditional</i>	Card number – Used if a card number is entered manually. See also Track1/Track2
RRN	<i>Mandatory</i>	Retrieval Reference Number – Reference number for transaction.
Transaction	<i>Conditional</i>	Unique number of the transaction within batch.
Batch	<i>Conditional</i>	Unique number of the batch in use for this terminal.
CardAcclId	<i>Mandatory</i>	Card Acceptor Identification Code Code Identifying the card acceptor. This Id is issued by the acquirer
CardAccTerminal	<i>Optional</i>	Card Acceptor Terminal Identification Code

		Code Identifying the terminal the card acceptor is using
CardAccName	<i>Mandatory</i>	Card Acceptor Name and Location Name\Street\City\Postal Code\Region (two digit US state code or three digit ISO alpha country code if outside US)\Country Code as in ISO3166
AuthCode	<i>Mandatory</i>	Authorization code if the transaction was accepted
ActionCode	<i>Mandatory</i>	Reason code for response
StoreTerminal	<i>Mandatory</i>	A numeric code that identifies the store and terminal within each merchant chain. Format SSSSTTTT
CardType	<i>Mandatory</i>	If available this will indicate the name of the card type
Message	<i>Optional</i>	A message in comprehension with the Action Code

Getting authorizations – XML examples

GAXML.1 – An example of authorization request

```
<?xml version='1.0' encoding="utf-8"?>
<getAuthorization>
  <Version>1000</Version>
  <Processor>123</Processor>
  <MerchantID>2</MerchantID>
  <TerminalID>1</TerminalID>
  <TransType>5</TransType>
  <TrAmount>100000</TrAmount>
  <TrCurrency>352</TrCurrency>
  <DateAndTime>060216103700</DateAndTime>
  <PAN>5401711111116716</PAN>
  <ExpDate>0705</ExpDate>
  <RRN>TESTING00001</RRN>
  <CVC2>123</CVC2>
</getAuthorization>
```

GAXML.2 – An example of an authorization capture

```
<?xml version='1.0' encoding="utf-8"?>
<getAuthorization>
  <Version>1000</Version>
  <Processor>123</Processor>
  <MerchantID>2</MerchantID>
  <TerminalID>1</TerminalID>
  <TransType>1</TransType>
  <TrAmount>100000</TrAmount>
  <TrCurrency>352</TrCurrency>
  <DateAndTime>060216103700</DateAndTime>
  <Batch>11</Batch>
  <Transaction>75</Transaction>
  <RRN>TESTING00001</RRN>
  <AuthCode>060215012345</ AuthCode >
</getAuthorization>
```

GAXML.3 – An example of refund of sale

```
<?xml version='1.0' encoding="utf-8"?>
<getAuthorization>
  <Version>1000</Version>
  <Processor>123</Processor>
  <MerchantID>1</MerchantID>
  <TerminalID>1</TerminalID>
  <TransType>3</TransType>
  <TrAmount>100000</TrAmount>
  <TrCurrency>352</TrCurrency>
  <DateAndTime>060216103700</DateAndTime>
  <RRN>TESTING00001</RRN>
</getAuthorization>
```

GAXML.4 – An example of partial reversal of sale

```
<?xml version='1.0' encoding="utf-8"?>
<getAuthorization>
  <Version>1000</Version>
  <Processor>123</Processor>
  <MerchantID>1</MerchantID>
  <TerminalID>1</TerminalID>
  <TransType>4</TransType>
  <TrAmount>100000</TrAmount>
  <NewAmount>30000</NewAmount>
  <TrCurrency>352</TrCurrency>
  <DateAndTime>060216103700</DateAndTime>
  <RRN>TESTING00001</RRN>
</getAuthorization>
```

Cancelling authorizations

This operation cancels a previously authorized transaction. The previously authorized transaction is found by searching for the AuthCode of the transaction. If no AuthCode is present in request the most recent transaction that matches the criteria (i.e. the rest of the input request) is cancelled. Transactions authorized with TransType=1 can only be cancelled while the batch they belong to is still open. Transactions authorized with TransType=5 can always be cancelled as they don't belong to any batches.

The card number and amount must be the same as in the original transaction. In the event of an error such as timeout in communications the RRN number of the transaction in error must be included in the cancel process.

Authorizations can be cancelled if the batch they belong to is still open. If it is closed then a refund transaction must be used. Pre-Authorize transaction can be cancelled after the batches are closed since pre-authorized transactions do not belong to batches.

Cancelling authorizations – XML explanations

Table CA.1 – Field explanations for a sent XML

Field	Length	Form	Type	Explanation
Version	4	AN	<i>Mandatory</i>	Indicates the version of these specifications the client is using. Current version is "1000"
Processor	7	N	<i>Mandatory</i>	An ID issued by Borgun to the processor using this interface on behalf of the merchant
MerchantID	..8	AN	<i>Mandatory</i>	An Id in Heimir of the merchant requesting services
TerminalID	..8	AN	<i>Optional</i>	A terminal id in Heimir for the merchant requesting services
TransType	1	N	<i>Mandatory</i>	Same as the original transaction
TrAmount	..12	N	<i>Mandatory</i>	Same as the original transaction
TrCurrency	3	N	<i>Mandatory</i>	Same as the original transaction
DateAndTime	12	N	<i>Mandatory</i>	YYMMDDhhmmss. Same as the original transaction
PAN	LL..19	N	<i>Conditional</i>	Card number – Same as the original transaction. If PAN is empty then DateAndTime + RRN will be used to find PAN in original transaction.
Track1	LL..76	AN	<i>Conditional</i>	Same as the original transaction
Track2	LL..37	Z	<i>Conditional</i>	Same as the original transaction
RRN	12	AN	<i>Mandatory</i>	Retrieval Reference Number – Reference number for transaction. Same as the original transaction
AuthCode	..6	AN	<i>Conditional</i>	Same as the original transaction

Table CA.2 – Field explanations for a received XML

Field	Type	Explanation
Version	<i>Mandatory</i>	Indicates the version of these specifications the client is using. Current version is "1000"
Processor	<i>Mandatory</i>	An ID issued by Borgun to the processor using this interface on behalf of the merchant
MerchantID	<i>Mandatory</i>	An Id in Heimir of the merchant requesting services
TerminalID	<i>Optional</i>	A terminal id in Heimir for the merchant requesting services
TransType	<i>Echo</i>	Same as the original transaction
TrAmount	<i>Echo</i>	Same as the original transaction
TrCurrency	<i>Echo</i>	Same as the original transaction
DateAndTime	<i>Echo</i>	YYMMDDhhmmss. Same as the original transaction
PAN	<i>Conditional</i>	Card number – Same as the original transaction
RRN	<i>Mandatory</i>	Retrieval Reference Number – Reference number for transaction. Same as the original transaction
Transaction	<i>Conditional</i>	Number of the transaction. Same as the original transaction
Batch	<i>Conditional</i>	Number of the batch in use. Same as the original transaction
CardAcclId	<i>Mandatory</i>	Code Identifying the card acceptor. Same as the original transaction
CardAccTerminal	<i>Mandatory</i>	Same as the original transaction
CardAccName	<i>Mandatory</i>	Card Acceptor Name and Location Name\Street\City\Postal Code\Region (two digit US state code or three digit ISO alpha country code if outside US)\Country Code as in ISO3166
AuthCode	<i>Mandatory</i>	Authorization code if the transaction was accepted
ActionCode	<i>Mandatory</i>	Reason code for response
StoreTerminal	<i>Mandatory</i>	A numeric code that identifies the store and terminal within each merchant chain. Format SSSSTTTT
CardType	<i>Mandatory</i>	If available this will indicate the name of the card type
Message	<i>Optional</i>	A message in comprehension with the Action Code

Cancelling authorizations – XML examples

CAXML.1 – An example of sent XML

```
<?xml version='1.0' encoding="utf-8"?>
<cancelAuthorization>
  <Version>1000</Version>
  <Processor>123</Processor>
  <MerchantID>2</MerchantID>
  <TerminalId>1</TerminalId>
  <TransType>1</TransType>
  <TrAmount>000000100000</TrAmount>
  <TrCurrency>352</TrCurrency>
  <DateAndTime>060216103700</DateAndTime>
  <PAN>5401710000006716</PAN>
  <ExpDate>0705</ExpDate>
  <RRN>060215012345</RRN>
  <AuthCode>809285</AuthCode>
</cancelAuthorization>
```


Virtual cardnumbers

Credicard numbers need not be sent into getAuthorization, instead virtual cardnumbers can be used. This enhances security since it eliminates the need to store sensitive credit card information in local datasources.

Virtual cardnumbers – XML explanations

Table VA.1 – Field explanations for a sent XML

Field	Length	Form	Type	Explanation
MerchantContractNumber	7	AN	Mandatory	Contract number issued by Borgun.
PAN	16	N	Mandatory	Card number

Table VA.2 – Field explanations for a received XML

Field	Type	Explanation
VirtualCard	Mandatory	Cardnumber that can be used as reference to the real card number in authorization.

Cancelling authorizations – XML examples

VCXML.1 – An example of sent XML

```
<?xml version='1.0' encoding="utf-8"?>
<getVirtualCard>
  <MerchantContractNumber>7891234</MerchantContractNumber>
  <PAN>540122233334444</PAN>
</ getVirtualCard >
```

Creating new batches

New batch closes the currently open batch and opens a new one.

If the currently open batch has no transactions associated with it, the batch will be considered as being the new batch, thus it will not be closed nor will a new batch be opened.

Batch numbers controlled by the Authorization Services

This is the preferred way of handling batches. The merchant sends in authorization requests to the Authorization Services. The Authorization Services will add the batch and transaction numbers to the transaction as needed. When the merchant wants to close the batch the merchant can use the NewBatch function or the Authorization Services can take care of this daily at predefined time.

Batch numbers controlled by the merchant

The merchant adds the batch and transaction numbers to the authorization request. In order to start a new batch the merchant simply increments the batch number and starts the transaction number back at number 1. The batch number must be unique for particular terminal within the year and in ascending sequence. Transaction numbers must be unique within a batch number.

Creating new batches – XML explanations

Table CNB.1 – Field explanations for a sent XML

Field	Length	Form	Type	Explanation
Version	4	AN	<i>Mandatory</i>	Indicates the version of these specifications the client is using. Current version is "1000"
Processor	7	N	<i>Mandatory</i>	An ID issued by Borgun to the processor using this interface on behalf of the merchant
MerchantId	..8	AN	<i>Mandatory</i>	An Id in Heimir of the merchant requesting services
TerminalId	..8	AN	<i>Optional</i>	If not set then the batch for the default terminal will be closed. If set then the insert terminal will be close.
DateAndTime	12	AN	<i>Optional</i>	YYMMDDhhmmss. Only used for documentation purposes

Table CNB.2 – Field explanations for a received XML

Field	Type	Explanation
Version	<i>Mandatory</i>	Indicates the version of these specifications the client is using. Current version is "1000"
Processor	<i>Mandatory</i>	An ID issued by Borgun to the processor using this interface on behalf of the merchant
MerchantId	<i>Mandatory</i>	An Id in Heimir of the merchant requesting services
TerminalId	<i>Optional</i>	A terminal id in Heimir for the merchant requesting services
DateAndTime	<i>Echo</i>	YYMMDDhhmmss. Same as the request
NewBatch	<i>Conditional</i>	Number of the new batch in use
OldBatch	<i>Conditional</i>	Number of the batch just closed
CardAccTerminal	<i>Optional</i>	Terminal id
ActionCode	<i>Mandatory</i>	Reason code for response
StoreTerminal	<i>Mandatory</i>	If available this will indicate the name of the card type

Creating new batches – XML examples

CNBXML.1 – An example of a sent XML

```
<?xml version='1.0' encoding="utf-8"?>
<newBatch>
  <Version>1000</Version>
  <Processor>123</Processor>
  <MerchantId>2</MerchantId>
  <TerminalId>1</TerminalId>
  <DateAndTime>060216103700</DateAndTime>
</newBatch>
```

Getting processing results

Client application can get results from previous batch processing job. This function will make available report on transaction basis listing all transactions that have been processed and a result code.

The get processing results web service returns a list of transactions in a given batch. The web service only returns a transaction list for closed batches.

Getting processing results – XML explanations

Table PR.1 – Field explanations for a sent XML

Field	Type	Explanation
Version	<i>Mandatory</i>	Indicates the version of these specifications the client is using. Current version is "1000"
Processor	<i>Mandatory</i>	An ID issued by Borgun to the processor using this interface on behalf of the merchant
MerchantID	<i>Mandatory</i>	An Id in Heimir of the merchant requesting services
TerminalID	<i>Optional</i>	A terminal id in Heimir for the merchant requesting services
Batch	<i>Mandatory</i>	Number of the requested batch.

Tables PR.2 and PR.3 – Field explanations for a received XML

Explanations for the header of the XML output

Field	Type	Explanation
Version	<i>Mandatory</i>	Indicates the version of these specifications the client is using. Current version is "1000"
Processor	<i>Mandatory</i>	An ID issued by Borgun to the processor using this interface on behalf of the merchant
MerchantID	<i>Mandatory</i>	An Id in Heimir of the merchant requesting services
TerminalID	<i>Mandatory</i>	A terminal id in Heimir for the merchant requesting services
ActionCode	<i>Mandatory</i>	Reason code for response
Batch	<i>Mandatory</i>	Number of the batch in use. If empty then Heimir will assign a new batch number

Explanations for each transaction within the batch

Field	Type	Explanation
TransactionNumber	Mandatory	Number of the transaction
TrAmount	Mandatory	Amount, note for example 100 USD is 10000 because the amount is represented in cents
TrCurrency	Mandatory	Transaction currency ISO 4217 352 => ISK 978 => EUR
DateAndTime	Mandatory	YYMMDDhhmmss. When repeated use original date and time
PAN	Mandatory	Card number – Used if a card number is entered manually
RRN	Mandatory	Retrieval Reference Number – Reference number for transaction
AuthCode	Conditional	Authorization code of a previous pre-authorized transaction. Include this code to use the previously gotten authorization code
ActionCode	Mandatory	Reason code for response
CardType	Mandatory	If available this will indicate the name of the card type

Getting processing results – XML examples

PRXML.1 – An example of a sent XML

```
<?xml version="1.0" encoding="utf-8"?>
<RequestProcessingResults>
  <Version>1000</Version>
  <Processor>123</Processor>
  <MerchantID>2</MerchantID>
  <Batch>3</Batch>
</RequestProcessingResults>
```

Getting transaction lists

Client applications can request a transaction list from the authorization service listing on either batch or transaction level details about transactions processed through the service.

It is possible to specify a batch number in the request to get all transactions that belong to that given batch. It is also possible to get a list of transactions for a given time period.

Getting transaction lists – XML explanations

Table TL.1 – Field explanations for a sent XML

Field	Type	Explanation
Version	<i>Mandatory</i>	Indicates the version of these specifications the client is using. Current version is "1000"
Processor	<i>Mandatory</i>	An ID issued by Borgun to the processor using this interface on behalf of the merchant
MerchantId	<i>Mandatory</i>	An Id in Heimir of the merchant requesting services
TerminalId	<i>Optional</i>	A terminal id in Heimir for the merchant requesting services
BatchNumber	<i>Conditional</i>	Batch number. If the BatchNumber is omitted, the response will only include batchnumbers processed during the given time period. Otherwise a list of transactions for the given batchnumber is returned.
FromDate	<i>Conditional</i>	Format:YYYYMMDD. Mandatory field if BatchNumber is empty.
ToDate	<i>Conditional</i>	Format:YYYYMMDD. Only used if BatchNumber is empty. If this parameter is not set, current date will be used as ToDate.
RRN	<i>Conditional</i>	Searches all batches for transactions with the given RRN

Tables TL.2 and TL.3 – Field explanations for a received XML

Explanations for the header of the XML output

Field	Type	Explanation
Version	Mandatory	Indicates the version of these specifications the client is using. Current version is "1000"
Processor	Mandatory	An ID issued by Borgun to the processor using this interface on behalf of the merchant
MerchantID	Mandatory	An Id in Heimir of the merchant requesting services
TerminalID	Mandatory	A terminal id in Heimir for the merchant requesting services
ActionCode	Mandatory	Reason code for response

Explanations for each transaction within the batch

Field	Type	Explanation
TransType	Mandatory	1 = Normal purchase transaction 3 = Refund 5 = Pre-Authorize
TransactionNumber	Mandatory	Number of the transaction
BatchNumber	Mandatory	Number of the requested batch
TransactionDate	Mandatory	YYMMDDhhmmss. Date and time of the transaction
PAN	Mandatory	Card number – Used if a card number is entered manually
RRN	Mandatory	Retrieval Reference Number – Reference number for transaction
ActionCode	Mandatory	Reason code for response
AuthorizationCode	Mandatory	Authorization code of a previous pre-authorized transaction. Include this code to use the previously gotten authorization code
TrAmount	Mandatory	Amount, note for example 100 USD is 10000 because the amount is represented in cents
TrCurrency	Mandatory	Transaction currency ISO 4217 352 => ISK 978 => EUR
CardType	Mandatory	If available this will indicate the name of the card type

Getting transaction lists – XML examples

TLXML.1 – An example of a sent XML

```
<?xml version="1.0" encoding="utf-8"?>
<TransactionListRequest>
  <Version>1000</Version>
  <Processor>123</Processor>
  <MerchantId>2</MerchantId>
  <BatchNumber>3</BatchNumber>
</TransactionListRequest>
```

Getting Disputed Transactions

Client applications can request a list of disputed transactions.

All authorization transactions have a unique number named Acquiring Reference, for each transaction that has a chargeback there will exist a list (1 or more) of chargeback records detailing the history of the chargeback. They can be fetched by using the Acquiring Reference.

Dispute types returned are as follows

- **Chargeback**
Issuer to acquirer. Chargeback of the original transaction, short code is 'C'. If the chargeback is partial, that is only partial amount of the original transaction is used, then the short code is 'PC'
- **Retrieval Request / Information Retrieval**
Issuer to acquirer. Request for information regarding the original transaction, non financial impact, short code is 'RR'.
- **Chargeback Reversal**
Issuer to acquirer. Reverses the chargeback, short code is 'CR'.
- **Second Presentment / Re-Presentment**
Acquirer to Issuer. Negates the chargeback by sending the financial transaction again, initiated by merchant, short code is 'SP'.
If the re-presentment is partial, that is only partial amount of the original transaction is used, then the short code is 'PSP'
- **Second Chargeback / Arbitration Chargeback**
Issuer to acquirer. Chargeback in response to the re-presentment, short code is 'AC'.
If the arbitration chargeback is partial, that is only partial amount of the original transaction is used, then the short code is 'PAC'

The chargeback list describes the communication between acquirer and issuer during dispute, a path for a long running dispute might be creating the following four records

1. Retrieval Request / Information Retrieval: Issuer needs more information to determine if the original transaction should be accepted or a chargeback sent.
2. Chargeback: Issuer requests a chargeback from acquirer. Issuer can always back out of the chargeback by doing a chargeback reversal.
3. Second Presentment / Re-Presentment: Acquirer does not accept the chargeback and resends the transaction.
4. Issuer might then end with Second Chargeback(Issuer re-request for chargeback)

A rare occurrence that still happens once in a while is double registration of chargeback.

1. Chargeback: Issuer requests a chargeback from acquirer
2. Chargeback: A second chargeback from issuer to acquirer
3. Chargeback Reversal: Issuer reverses one chargeback for acquirer does a re-presentment.

Getting disputed transactions

Table DT.1 – Parameters for getDisputedTransactions ()

Field	Type	Explanation
Processor	Mandatory	An ID issued by Borgun to the processor using this interface on behalf of the merchant
MerchantContractNumber	Mandatory	Contract number issued by Borgun.
DateFrom	Mandatory	Returns all disputed transactions from date DateFrom. Date format YYYYMMDD.
DateTo	Mandatory	Returns all disputed transactions up to and including date DateFrom. Date format YYYYMMDD.

Table DT.2 – Field explanations for the status record contained in received XML

Field	Type	Explanation
ResultCode	Mandatory	Can have values 0 = "Success", 10 = "Information" (f.ex. No Data Found), 30 = "Error"
ResultText	Mandatory	Text description for ResultCode, f.ex. "Success", "Error", "No Data Found"
ErrorMessage	Mandatory	Detailed error description
MessageId	Mandatory	Error description code of ErrorMessage if applicable
ReferenceValue	Mandatory	Id of record within list where error occurred. Only returned if applicable
StackTrace	Mandatory	Stacktrace of error if applicable

Table DT.3 – Field explanations for a received XML

Field	Type	Explanation
MerchantNumber	Mandatory	Merchant contract number
ProcessingTime	Mandatory	Date and time of record processing at Borgun, yyyy-mm-ddThh:mm:ss
Date	Mandatory	Date of record creation, yyyy-mm-dd
Brand	Mandatory	Card brand type
Type	Mandatory	See table above
Amount	Mandatory	Current amount
Currency	Mandatory	Current currency
ReasonCode	Mandatory	Code describing reason for dispute
ReasonText	Mandatory	Text describing reason for dispute
AcquiringReferenceData	Mandatory	Acquirer reference
BatchDate	Mandatory	Batch date of transaction, yyyy-mm-dd
TerminalId	Mandatory	Transaction terminal id
RRN	Mandatory	Retrieval Reference Number – Reference number for transaction
CardNumber	Mandatory	Masked cardnumber in transaction
OriginalAmount	Mandatory	Amount in transaction
OriginalCurrency	Mandatory	Currency of transaction

OriginalDate	<i>Mandatory</i>	Date of transaction, yyyy-mm-dd
AuthorizationCode	<i>Mandatory</i>	Transaction authorization number

Getting disputed transaction list – XML examples

DTXML.1 – An example of a sent and received XML

```

<?xml version="1.0" encoding="utf-8"?>
<getDisputedTransactions>
<Processor>21</Processor>
<MerchantContractNumber>9275444</MerchantContractNumber>
<DateFrom>20101021</DateFrom>
<DateTo>20111025</DateTo>
</getDisputedTransactions>

<?xml version="1.0"?>
<DisputedTransactionResponse>
  <Status>
    <ResultCode>0</ResultCode>
    <ResultText>Success</ResultText>
  </Status>
  <DisputedTransactionList>
    <DisputedTransaction>
      <MerchantNumber>9275444</MerchantNumber>
      <ProcessingTime>2008-09-18T00:00:00</ProcessingTime>
      <Date>2008-09-18</Date>
      <Brand>MC</Brand>
      <Type>C</Type>
      <Amount>2000000</Amount>
      <Currency>352</Currency>
      <ReasonCode>4809</ReasonCode>
      <ReasonText>TRANSACTION NOT RECONCILED</ReasonText>
      <AcquiringReferenceData>5414838259002000123123</AcquiringReferenceData>
      <BatchDate>2008-06-23</BatchDate>
      <TerminalId>00010002</TerminalId>
      <RRN>111222</RRN>
      <CardNumber>002359*****3123</CardNumber>
      <OriginalAmount>20000</OriginalAmount>
      <OriginalCurrency>352</OriginalCurrency>
      <OriginalDate>2008-06-21</OriginalDate>
      <AuthorizationCode>0</AuthorizationCode>
    </DisputedTransaction>
  </DisputedTransactionList>
</DisputedTransactionResponse>

```

Getting Fraud Transactions

Client applications can request a list of fraud transactions.

Getting fraud transactions

Table FT.1 – Parameters for getFraudTransactions ()

Field	Type	Explanation
Processor	Mandatory	An ID issued by Borgun to the processor using this interface on behalf of the merchant
MerchantContractNumber	Mandatory	Contract number issued by Borgun.
DateFrom	Mandatory	Returns all fraud transactions from date DateFrom. Date format YYYYMMDD.
DateTo	Mandatory	Returns all disputed transactions up to and including date DateFrom. Date format YYYYMMDD.

Table FT.2 – Field explanations for the status record contained in received XML

Field	Type	Explanation
ResultCode	Mandatory	Can have values 0 = "Success", 10 = "Information" (f.ex. No Data Found), 30 = "Error"
ResultText	Mandatory	Text description for ResultCode, f.ex. "Success", "Error", "No Data Found"
ErrorMessage	Mandatory	Detailed error description
MessageId	Mandatory	Error description code of ErrorMessage if applicable
ReferenceValue	Mandatory	Id of record within list where error occurred. Only returned if applicable
StackTrace	Mandatory	Stacktrace of error if applicable

Table DT.3 – Field explanations for a received XML

Field	Type	Explanation
MerchantNumber	Mandatory	Merchant contract number
ProcessingTime	Mandatory	Date and time of record processing at Borgun, yyyy-mm-ddThh:mm:ss
Date	Mandatory	Date of record creation, yyyy-mm-dd
Brand	Mandatory	Card brand type
Type	Mandatory	Currently can only have the value 'F' (Fraud)
ReasonCode	Mandatory	Code describing reason for dispute
ReasonText	Mandatory	Text describing reason for dispute
AcquiringReferenceData	Mandatory	Acquirer reference to the original authorization transaction that this dispute record relates to. For each disputed transaction there exists 1 or more dispute record that all will have the same acquirer reference.

NotificationCode	<i>Mandatory</i>	Notification code 1 = Addition 2 = Addition of subsequent duplicate transaction 3 = Change 4 = Delete 5 = Reactivate
IssuerGeneratedAuthorization	<i>Mandatory</i>	VISA specific, Y = Issuer authorized transaction X = Transaction authorized but not by issuer N = Transaction not authorized
AccountSequenceNumber	<i>Mandatory</i>	VISA specific, identifies a transaction within the account number. A maximum of 1000 transactions on a specific account may be added to the Visa fraud master file. The member may also pre-assign a sequence number within the 4000–4999 range. account
FraudInvestigationStatus	<i>Mandatory</i>	VISA specific, member-assigned, may be 00 through 99 or AA through ZZ
CustomerReference	<i>Mandatory</i>	VISA specific
BatchDate	<i>Mandatory</i>	Batch date, yyyy-mm-dd
TerminalId	<i>Mandatory</i>	Transaction terminal id
RRN	<i>Mandatory</i>	Retrieval Reference Number – Reference number for transaction
CardNumber	<i>Mandatory</i>	Masked cardnumber in transaction
OriginalBatchId	<i>Mandatory</i>	VISA specific, original batch id
OriginalAmount	<i>Mandatory</i>	Amount in transaction
OriginalCurrency	<i>Mandatory</i>	Currency of transaction
OriginalDate	<i>Mandatory</i>	Date of transaction, yyyy-mm-dd
AuthorizationCode	<i>Mandatory</i>	Transaction authorization number

Getting fraud transaction list – XML examples

DTXML.1 – An example of a received XML

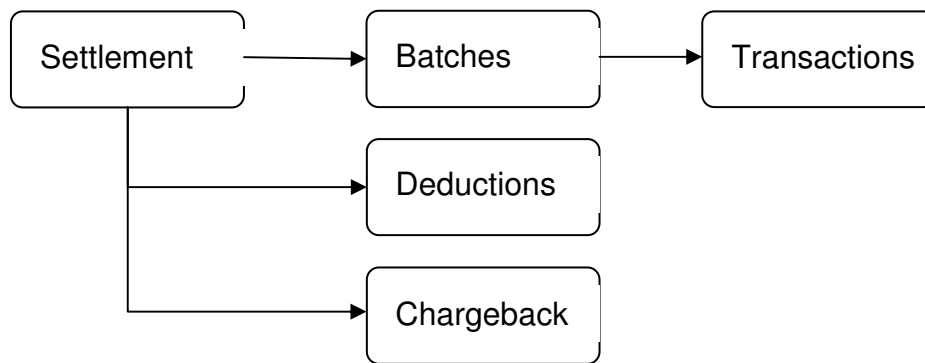
```
<?xml version="1.0"?>
<FraudTransactionResponse>
  <Status>
    <ResultCode>0</ResultCode>
    <ResultText>Success</ResultText>
  </Status>
  <FraudTransactionList>
    <FraudTransaction>
      <MerchantNumber>9577851</MerchantNumber>
      <ProcessingTime>2009-10-14T10:25:00.60831</ProcessingTime>
      <Date>2009-03-12</Date>
      <Brand>MC</Brand>
      <Type>F</Type>
      <ReasonCode>4</ReasonCode>
      <ReasonText>COUNTERFEIT</ReasonText>
      <AcquiringReferenceData>5414838196002000012345</AcquiringReferenceData>
      <NotificationCode>1</NotificationCode>
      <IssuerGeneratedAuthorization></IssuerGeneratedAuthorization>
      <AccountSequenceNumber></AccountSequenceNumber>
    </FraudTransaction>
  </FraudTransactionList>
</FraudTransactionResponse>
```

```
<FraudInvestigationStatus></FraudInvestigationStatus>  
<CustomerReference></CustomerReference>  
<BatchDate>2008-06-23</BatchDate>  
<TerminalId>00010002</TerminalId>  
<RRN></RRN>  
<CardNumber>398320*****2345</CardNumber>  
<OriginalBatchId>1</OriginalBatchId>  
<OriginalAmount>10000</OriginalAmount>  
<OriginalCurrency>352</OriginalCurrency>  
<OriginalDate>2008-06-21</OriginalDate>  
<AuthorizationCode>0</AuthorizationCode>  
</FraudTransaction>  
</FraudTransactionList>  
</FraudTransactionResponse>
```

Getting Settlement Data

Client applications can fetch settlement data.

Settlement can contain one or more batches, and each batch contains one or more transaction.



Settlement data can be fetched with the following calls

Getting settlement list

Table SD.1 – Parameters for getSettlement ()

Field	Type	Explanation
Processor	Mandatory	An ID issued by Borgun to the processor using this interface on behalf of the merchant
MerchantContractNumber	Mandatory	Contract number issued by Borgun.
DateFrom	Mandatory	Returns all settlements from date DateFrom. Date format YYYYMMDD.
DateTo	Mandatory	Returns all settlement up to and including DateTo. Date format YYYYMMDD.

Getting a list of batches in settlement

Table SD.2 – Parameters for getSettlementBatch()

Field	Type	Explanation
Processor	Mandatory	An ID issued by Borgun to the processor using this interface on behalf of the merchant
MerchantContractNumber	Mandatory	Contract number issued by Borgun.
SettlementNumber	Mandatory	Unique number of settlement

Getting a list of transactions in batches. BatchDate and BatchNumber are unique for each batch.

Table SD.3 – Parameters for getSettlementTransactions()

Field	Type	Explanation
Processor	Mandatory	An ID issued by Borgun to the processor using this interface on behalf of the merchant
MerchantContractNumber	Mandatory	Contract number issued by Borgun.
BatchNumber	Mandatory	Batch number
BatchDate	Mandatory	Batch date, format YYYYMMDD

Getting a list of chargebacks in settlement

Table SD.4 – Parameters for getSettlementChargeback()

Field	Type	Explanation
Processor	Mandatory	An ID issued by Borgun to the processor using this interface on behalf of the merchant
MerchantContractNumber	Mandatory	Contract number issued by Borgun.
SettlementNumber	Mandatory	Unique number of settlement

Table SD.4 – Field explanations for the status record contained in received XML

Field	Type	Explanation
ResultCode	Mandatory	Can have values 0 = "Success", 10 = "Information" (f.ex. No Data Found), 30 = "Error"
ResultText	Mandatory	Text description for ResultCode, f.ex. "Success", "Error", "No Data Found"
ErrorMessage	Mandatory	Detailed error description
MessageId	Mandatory	Error description code of ErrorMessage if applicable
ReferenceValue	Mandatory	Id of record within list where error occurred. Only returned if applicable
StackTrace	Mandatory	Stacktrace of error if applicable

Tables SD.5 – Settlement

Field	Type	Explanation
SettlementNumber	Mandatory	Settlement number
SettlementDate	Mandatory	Date of settlement
MastercardDomesticAmount	Mandatory	Settlement amount from domestic Mastercard transactions
MastercardInternationalAmount	Mandatory	Settlement amount from international Mastercard transactions

MastercardOtherAmount	<i>Mandatory</i>	Settlement amount from other Mastercard transactions
JcbAmount	<i>Mandatory</i>	Settlement amount from JCB transactions
VisaAmount	<i>Mandatory</i>	Settlement amount from VISA transactions
AmexAmount	<i>Mandatory</i>	Settlement amount from AMEX transactions
TotalAmount	<i>Mandatory</i>	Total amount in settlement
TransactionCountInSettlement	<i>Mandatory</i>	Number of transactions in settlement
SettlementCurrencyCode	<i>Mandatory</i>	Currency code of settlement
DeductionAmount	<i>Mandatory</i>	Deduction amount of settlement amount
RefundAmount	<i>Mandatory</i>	Refund amount in settlement

Tables SD.6 – Settlement deductions

Field	Type	Explanation
SettlementNumber	<i>Mandatory</i>	Settlement number
Code	<i>Mandatory</i>	Code describing deduction type
CodeDescription	<i>Mandatory</i>	Text describing deduction type
Amount	<i>Mandatory</i>	Deduction amount
DeductionCurrencyCode	<i>Mandatory</i>	Currency code of deduction

SDXML.5 and SDXML6 – An example of a received XML with settlement and deduction

Request:

```
<?xml version="1.0" encoding="utf-8"?>
<getSettlements>
  <Processor>1</Processor>
  <MerchantContractNumber>9999999</MerchantContractNumber>
  <DateFrom>20110129</DateFrom>
  <DateTo>20110930</DateTo>
</getSettlements>
```


Response:

```

<?xml version="1.0"?>
<SettlementResponse>
  <Status>
    <ResultCode>0</ResultCode>
    <ResultText>Success</ResultText>
  </Status>
  <Settlement>
    <SettlementNumber>2727</SettlementNumber>
    <SettlementDate>2008-01-02T00:00:00</SettlementDate>
    <MastercardDomesticAmount>0</MastercardDomesticAmount>
    <MastercardInternationalAmount>0.0</MastercardInternationalAmount>
    <MastercardOtherAmount>0</MastercardOtherAmount>
    <JcbAmount>0</JcbAmount>
    <VisaAmount>0</VisaAmount>
    <AmexAmount>0</AmexAmount>
    <TotalAmount>2899007000</TotalAmount>
    <TransactionCountInSettlement>5485</TransactionCountInSettlement>
    <SettlementCurrencyCode>ISK</SettlementCurrencyCode>
    <DeductionAmount>-28087900</DeductionAmount>
    <RefundAmount>0</RefundAmount>
    <SettlementDeductions>
      <SettlementNumber>2727</SettlementNumber>
      <Code></Code>
      <Amount>-246416.00</Amount>
      <DeductionCurrencyCode>ISK</DeductionCurrencyCode>
    </SettlementDeductions>
  </Settlement>
</SettlementResponse>

```

SD.7 – List of batches in settlement

Field	Type	Explanation
BatchNumber	Mandatory	Batch number
BatchDate	Mandatory	Date of batch
SettlementDate	Mandatory	Settlement date
SettlementNumber	Mandatory	Settlement number
TransactionCountInBatch	Mandatory	Number of transactions in batch
TransactionSumInBatch	Mandatory	Amount in transactions in batch
BatchCurrencyCode	Mandatory	Currency code of amounts

SDXML.7 – Example of received batchlist**Request:**

```

<?xml version="1.0" encoding="utf-8"?>
<getSettlementBatch>
  <Processor>1</Processor>
  <MerchantContractNumber>9999999</MerchantContractNumber>
  <SettlementNumber>4245</SettlementNumber>
</getSettlementBatch>

```

Response:

```
<?xml version="1.0"?>
<SettlementBatchResponse>
  <Status>
    <ResultCode>0</ResultCode>
    <ResultText>Success</ResultText>
  </Status>
  <SettlementBatch>
    <BatchNumber>116757</BatchNumber>
    <BatchDate>2007-12-05T00:00:00</BatchDate>
    <SettlementDate>2008-01-02T00:00:00</SettlementDate>
    <SettlementNumber>4245</SettlementNumber>
    <TransactionCountInBatch>9</TransactionCountInBatch>
    <TransactionSumInBatch>1840500</TransactionSumInBatch>
    <BatchCurrencyCode>ISK</BatchCurrencyCode>
  </SettlementBatch>
</SettlementBatchResponse>
```

SD.8 – Transactions in batches

Field	Type	Explanation
TransactionDate	Mandatory	Date of transaction
Amount	Mandatory	Transaction amount
CurrencyCode	Mandatory	Currency code of amounts in transaction
SlipNumber	Mandatory	Slip number
RRN	Mandatory	Retrieval reference number
CardNumber	Mandatory	Card number
CardType	Mandatory	Card type
OriginalAmount	Mandatory	Original amount on slip
OriginalAmountCurrencyCode	Mandatory	Amount in currency code
BatchNumber	Mandatory	Number of batch containing transaction
SettlementDate	Mandatory	Date of settlement
SettlementNumber	Mandatory	Number of settlement containing transaction

SDXML.8 – An XML example of a received transaction list

Request:

```
<?xml version="1.0" encoding="utf-8"?>
<getSettlementTransactions>
  <Processor>1</Processor>
  <MerchantContractNumber>9999999</MerchantContractNumber>
  <BatchNumber>101166</BatchNumber>
  <BatchDate>20110105</BatchDate>
</getSettlementTransactions>
```

Response:

```
<?xml version="1.0"?>
<SettlementTransactionResponse>
  <Status>
    <ResultCode>0</ResultCode>
```

```

    <ResultText>Success</ResultText>
  </Status>
  <SettlementTransaction>
    <TransactionDate>2011-01-05T00:00:00</TransactionDate>
    <Amount>68700</Amount>
    <CurrencyCode>ISK</CurrencyCode>
    <Slipnumber>9372</Slipnumber>
    <RRN>ORD456789372</RRN>
    <Cardnumber>**** * 3102 </Cardnumber>
    <Cardtype>MD</Cardtype>
    <OriginalAmount>68700</OriginalAmount>
    <OriginalAmountCurrencyCode>ISK</OriginalAmountCurrencyCode>
    <BatchNumber>101166</BatchNumber>
    <SettlementDate>2008-01-02T00:00:00</SettlementDate>
    <SettlementNumber>2727</SettlementNumber>
  </SettlementTransaction>
</SettlementTransactionResponse>

```

SD.9 – Chargeback transactions in settlements

Field	Type	Explanation
TransactionDate	Mandatory	Date of transaction
Amount	Mandatory	Transaction amount
CurrencyCode	Mandatory	Currency code of amounts in transaction
SlipNumber	Mandatory	Slip number
RRN	Mandatory	Retrieval reference number
CardNumber	Mandatory	Card number
CardType	Mandatory	Card type
OriginalAmount	Mandatory	Original amount on slip
OriginalAmountCurrencyCode	Mandatory	Amount in currency code
BatchNumber	Mandatory	Number of batch containing transaction
SettlementDate	Mandatory	Date of settlement
SettlementNumber	Mandatory	Number of settlement containing transaction
AcquirerReference	Mandatory	Acquirer reference for chargeback. Unique number for the authorization transaction. Can be used to reference a list of dispute records related to the chargeback.
ChargebackDate	Mandatory	Date of chargeback registration
ChargebackAmount	Mandatory	Amount of chargeback

SDXML.9 – An XML example of a received chargeback list

Request:

```

<?xml version="1.0" encoding="utf-8"?>
<getSettlementChargeback>
  <Processor>1</Processor>
  <MerchantContractNumber>9999999</MerchantContractNumber>
  <SettlementNumber>7566</SettlementNumber>
</getSettlementChargeback>

```

Response:

```
<?xml version="1.0"?>
<SettlementChargebackResponse>
  <Status>
    <ResultCode>0</ResultCode>
    <ResultText>OK</ResultText>
  </Status>
  <SettlementChargeback>
    <SettlementDate>2011-08-10T00:00:00</SettlementDate>
    <SettlementNumber>7566</SettlementNumber>
    <BatchNumber>162</BatchNumber>
    <Amount>-34.9500000000000000</Amount>
    <CurrencyCode>USD</CurrencyCode>
    <Cardnumber>4060 42** **** 8831</Cardnumber>
    <Cardtype>04</Cardtype>
    <TransactionDate>2011-07-08T00:00:00</TransactionDate>
    <OriginalAmountCurrencyCode>USD</OriginalAmountCurrencyCode>
    <OriginalAmount>-34.95</OriginalAmount>
    <Slipnumber>777003</Slipnumber>
    <RRN>AA1234567890</RRN>
    <AcquirerReference>5414811112222000025737</AcquirerReference>
    <ChargebackDate>2011-08-03T00:00:00</ChargebackDate>
    <ChargebackAmount>187</ChargebackAmount> </SettlementChargeback>
  </SettlementChargebackResponse>
```

Sending detailed data

This operation is used to supply detailed invoice after the financial transaction has been processed. In order to match the detail transaction to the financial transaction some parameters must be supplied.

Sending detail data – XML explanations

Tables SDD.1, SDD.2 and SDD.3 – Field explanations for a sent XML

The input is split into a header and detail. Following the header is a detailed description of the invoice in XML format. The invoice itself is split into a header record (A4) and detail records (A6) as follows:

Main header

Field	Type	Explanation
Version	Mandatory	Indicates the version of these specifications the client is using. Current version is "1000"
Processor	Mandatory	An Id issued by Borgun to the processor using this interface on behalf of the merchant.
MerchantId	Mandatory	An Id in Heimir of the merchant requesting services
TerminalId	Optional	A terminal id in Heimir for the merchant requesting services.
TransType	Mandatory	Same as in the original transaction
TrAmount	Mandatory	Same as in the original transaction
TrCurrency	Mandatory	Same as in the original transaction
DateAndTime	Mandatory	Same as in the original transaction
PAN	Mandatory	Same as in the original reply
ExpDate	Mandatory	Expiration date – YYMM
RRN	Mandatory	Same as in the original reply
Transaction	Optional	Same as in the original reply
Batch	Optional	Same as in the original reply
AuthCode	Conditional	Same as in the original reply

Invoice header (A4):

Field	Type	Explanation
Version	Mandatory	Indicates the version of these specifications the client is using. Current version is "1000".
MessageNumber	Mandatory	Message number (sequence number of line within file)
InvoiceNumber	Mandatory	Invoice number
InvoiceDate	Mandatory	Invoice date (yyyymmdd)
Reference	Optional	Reference number
MerchantSSN	Mandatory	Merchant social security number
MerchantVATNumber	Mandatory	Merchant VAT number
Amount1	Mandatory	Total (A6 Amount)
Discount	Optional	Total (A6*DiscountAmount) Discount
AmountWithDiscount	Optional	Total (A6*AmountWithDiscount) Amount with discount
Vgju	Optional	Total (A6*Vgju) (zero filled)
Amount3	Optional	Total (A6*Amount3) (same as AmountWithDiscount)
TotalVAT	Mandatory	Total (A6*VATAmount) VAT
AmountWithVAT	Mandatory	Total (A6*AmountWithVAT) amount with VAT
InvoiceAmount	Mandatory	Total invoice
Rounding	Optional	Round off amount in "aurar"
RikiskaupFee	Optional	Total Rikiskaup fee
GLAccount	Optional	General ledger account number
RecordNumber	Mandatory	Record number within invoice (0001 for header)
NumberOfRecords	Mandatory	Number of records in invoice (A4+number of A6)
SlipNumber	Optional	Slip number
Comments	Optional	Comments
OrderNumber	Optional	Order number
OrderDate	Optional	Order date
DeliveryDate	Optional	Delivery date
DeliveryAddress	Optional	Delivery address

Invoice detail (A6):

Field	Type	Explanation
Version	<i>Mandatory</i>	Indicates the version of these specifications the client is using. Current version is "1000".
MessageNumber	<i>Mandatory</i>	Message number (sequence number of line within file)
ProductCode	<i>Mandatory</i>	Product code
ItemDescription	<i>Mandatory</i>	Item description
UNSPSCcode	<i>Optional</i>	UNSPSC code
UNSPSCdescription	<i>Optional</i>	UNSPSC description
CommodityCode	<i>Optional</i>	Commodity code
ItemQuantity	<i>Mandatory</i>	Item quantity
UnitOfMeasurement	<i>Optional</i>	Item unit of measure (ltr, km, kg...)
GrossValue	<i>Mandatory</i>	Unit price without VAT
Amount	<i>Mandatory</i>	Amount (ItemQuantity * GrossValue)
DiscountPercent	<i>Optional</i>	Discount percent (%)
DiscountAmount	<i>Optional</i>	Discount amount
AmountWithDiscount	<i>Optional</i>	Amount with discount (Amount – DiscountAmount)
Vgjp	<i>Optional</i>	"00000"
Vgju	<i>Optional</i>	"000000000000"
Amount3	<i>Optional</i>	AmountWithDiscount
VATPercent	<i>Mandatory</i>	VAT %
VATAmount	<i>Mandatory</i>	VAT amount
AmountWithVAT	<i>Mandatory</i>	Amount with VAT (Amount3 + VATAmount)
RikiskaupFeePercent	<i>Optional</i>	Rikiskaup fee (%)
RikiskaupFee	<i>Optional</i>	Rikiskaup fee amount
GLAccount	<i>Optional</i>	General ledger account number
RecordNumber	<i>Mandatory</i>	Record number within invoice (0002-xxxx)
NumberOfRecords	<i>Mandatory</i>	Number of records in invoice (A4+number of A6)
SlipNumber	<i>Optional</i>	Slip number
Ref1	<i>Optional</i>	Reference 1
Ref2	<i>Optional</i>	Reference 2
Ref3	<i>Optional</i>	Reference 3
Ref4	<i>Optional</i>	Reference 4

Table SDD.4 – Field explanations for a received XML

Field	Type	Explanation
Version	Mandatory	Same value as in the request message
Processor	Mandatory	Same value as in the request message
MerchantId	Mandatory	Same value as in the request message
TerminalId	Mandatory	Same value as in the request message
TransType	Mandatory	Same value as in the request message
TrAmount	Mandatory	Same value as in the request message
TrCurrency	Mandatory	Same value as in the request message
DateAndTime	Mandatory	Same value as in the request message
PAN	Mandatory	Same value as in the request message
ExpDate	Mandatory	Same value as in the request message
RRN	Mandatory	Same value as in the request message
Transaction	Mandatory	Same value as in the request message
Batch	Mandatory	Same value as in the request message
CardAccID	Mandatory	Code Identifying the card acceptor. This Id is issued by the acquirer
CardAccTerminal	Mandatory	Code Identifying the terminal the card acceptor is using
AuthCode	Mandatory	Same value as in the request message
ActionCode	Mandatory	Reason code for response

Sending detail data – XML examples

SDDXML.1 – An example of a sent XML

```
<?xml version="1.0" encoding="utf-8"?>
<DetailData>
  <Version>1000</Version>
  <Processor>123</Processor>
  <MerchantId>2</MerchantId>
  <TransType>1</TransType>
  <TrAmount>100000</TrAmount>
  <TrCurrency>352</TrCurrency>
  <DateAndTime>060216103700</DateAndTime>
  <PAN>5401123456783234</PAN>
  <ExpDate>0506</ExpDate>
  <RRN>404</RRN>
  <Batch>3</Batch>
  <A4>
    <Version>1000</Version>
    <MessageNumber>2</MessageNumber>
    <InvoiceNumber>071897</InvoiceNumber>
    <InvoiceDate>20060118</InvoiceDate>
    <MerchantSSN>5415986459</MerchantSSN>
    <MerchantVATNumber>079318</MerchantVATNumber>
    <Amount1>100000</Amount1>
    <TotalVAT>24500</TotalVAT>
    <AmountWithVAT>124500</AmountWithVAT>
    <InvoiceAmount>125000</InvoiceAmount>
    <RecordNumber>1</RecordNumber>
    <NumberOfRecords>1</NumberOfRecords>
  </A4>
  <A6>
    <Version>1000</Version>
    <MessageNumber>2</MessageNumber>
    <ProductCode>054</ProductCode>
    <ItemDescription>Chocolate</ItemDescription>
    <ItemQuantity>5</ItemQuantity>
    <GrossValue>20000</GrossValue>
    <Amount>1000</Amount>
    <VATPercent>0,245</VATPercent>
    <VATAmount>24500</VATAmount>
    <AmountWithVAT>124500</AmountWithVAT>
    <RecordNumber>1</RecordNumber>
    <NumberOfRecords>1</NumberOfRecords>
  </A6>
</A4>
</DetailData>
```

Appendix

Action Codes

Action Code	Explanation	ISO Reply
100-199	<i>Show reason for transaction dismissal by issuer, card pick-up not required.</i>	
000	<i>Accepted</i>	00
100	<i>Do not honor (Not accepted)</i>	05
101	<i>Expired card</i>	54
102	<i>Suspected card forgery (fraud)</i>	
103	<i>Merchant call acquirer</i>	01, 66, 08
104	<i>Restricted card</i>	62
106	<i>Allowed PIN retries exceeded</i>	75, 76
109	<i>Merchant not identified</i>	03
110	<i>Invalid amount</i>	13, 64
111	<i>Invalid card number</i>	14
112	<i>PIN required</i>	
116	<i>Not sufficient funds</i>	51
117	<i>Invalid PIN</i>	55
118	<i>Unknown card</i>	
119	<i>Transaction not allowed to cardholder</i>	57
120	<i>Transaction not allowed to terminal</i>	12, 58
121	<i>Exceeds limits to withdrawal</i>	61, 65
125	<i>Card not valid</i>	
126	<i>False PIN block</i>	
129	<i>Suspected fraud</i>	
130	<i>Invalid Track2</i>	
131	<i>Invalid Expiration Date</i>	
161	<i>DCC transaction not allowed to cardholder</i>	
162	<i>DCC cardholder currency not supported</i>	
163	<i>DCC exceeds time limit for withdrawal</i>	
164	<i>DCC transaction not allowed to terminal</i>	
165	<i>DCC not allowed to merchant</i>	
166	<i>DCC unknown error</i>	
200-299	<i>Show reason for transaction dismissal by issuer, card pick-up required.</i>	
200	<i>No not honor</i>	04
201	<i>Card not valid</i>	33
202	<i>Suspected card forgery (fraud)</i>	
203	<i>Merchant contact acquirer</i>	37
204	<i>Limited card</i>	36
205	<i>Merchant contact police</i>	
206	<i>Allowed PIN-retries exceeded</i>	38
207	<i>Special occasion</i>	
208	<i>Lost card</i>	41
209	<i>Stolen card</i>	43
210	<i>Suspected fraud</i>	

902	<i>False transaction</i>	89, 06
904	<i>Form error</i>	30
907	<i>Issuer not responding</i>	
908	<i>Message can not be routed</i>	15, 80, 81, 92
909	<i>System error</i>	83, 96
910	<i>Issuer did not respond</i>	68
911	<i>Issuer timed out</i>	82
912	<i>Issuer not reachable</i>	91
913	<i>Double transaction transmission</i>	94
914	<i>Original transaction can not be traced</i>	
916	<i>Merchant not found</i>	
950	<i>No financial record found for detail data</i>	
951	<i>Batch already closed</i>	
952	<i>Invalid batch number</i>	
953	<i>Host timeout</i>	
954	<i>Batch not closed</i>	
955	<i>Merchant not active</i>	
956	<i>Transaction number not unique</i>	

FAQ

Frequently Asked Questions

1. Is it possible to perform partial refunds (TransType = 3)? Can this happen multiple times until reaching the total amount authorized?

This is supported, for example

1. Perform a normal purchase transaction (TransType = 1) with amount 10.
2. Then, perform a refund transaction (TransType = 5) with amount 5.
3. Finally, perform another refund transaction for the remaining 5.

The sum of the refunds must be equal or lower than the original sale amount.

2. Is it possible to send normal purchase transactions (TransType = 1) with less amount than the one specified in the original pre-authorization (TransType = 5) so that the final authorized amount is the one sent in the normal purchase?

This is supported, for example

1. Perform a pre-authorize transaction (TransType = 5) with amount 10.
2. Capture the previous pre-authorization performing a normal purchase transaction (TransType = 1) with amount 5.

Note that the cardholder will have the withdrawal limit corrected 1-2 days later.

3. Has RRN any kind of uniqueness restriction?

RRN has must be size 12 alphanumeric, the last 6 letters cannot be 000000. For reference sake it is recommended to be a sequence with fixed first part, i.e. ACME12345678

4. Are batches automatically closed at the same time every day? In that case, could you tell us the exact time and time zone when this happens? Can this time be configured or is it fixed?

Default is to have the batch close time fixed, then it will close at midnight GMT-0 (no daylight changes).

It can be controlled by calling party by using the newBatch() command, at execution the old batch is closed and a new one is opened. This is the preferred way to use for custom closure.