

iDEAL Merchant Integration Guide (EN)



Version: 3.5
01-01-2018

Terms and conditions

Terms and conditions for provision of the iDEAL Merchant Integration Guide:

1. Scheme Owner Currence iDEAL B.V. provides the iDEAL Merchant Integration Guide to Acquiring banks which distribute it to (potential) Merchants and Payment Service Providers to enable them to form a good idea of what the implementation of the iDEAL product would involve and assess how any future use of iDEAL could affect their business operations.
2. Currence iDEAL B.V. reserves the right to deny access to the iDEAL Merchant Integration Guide to (potential) Merchants and Payment Service Providers on reasonable grounds, in consultation with the Acquiring bank with which the Merchant/PSP has a contract.
3. The iDEAL Merchant Integration Guide is explicitly and exclusively provided for the purpose mentioned above, and no other use is permitted. No rights can be derived from the information provided in this document or the accompanying notes. Currence iDEAL B.V. is in no way liable for the consequences of later changes to the iDEAL Standards or the iDEAL Merchant Integration Guide. If banks or other interested parties take decisions and/or make investments on the basis of the information that they obtain via the iDEAL Merchant Integration Guide, Currence iDEAL B.V. accepts no liability for this whatsoever.
4. The iDEAL Merchant Integration Guide is based on the information in the iDEAL Standards documents. In the event of any discrepancy between the iDEAL Merchant Integration Guide and the iDEAL Standards documents, the text in the iDEAL Standards documents shall prevail.

For any questions concerning this document or requests for further information, please contact your iDEAL acquiring bank or Collecting Payment Service Provider

Contents

▼ ...

Terms and conditions

Contents

1. Introduction

1.1 Target audience

1.2 Document structure

1.3 Definitions of internet and online banking

1.4 Revisions

2. Overview

2.1 What is iDEAL?

2.2 What is iDEAL Mobile?

2.3 Four party model

3. Message format

3.1 General

3.2 Header format

3.3 XML Namespace declaration

3.4 Conventions for empty fields

3.5 Merchant information registered with Acquirer

4. Directory protocol

4.1 General

4.2 DirectoryRequest

4.3 DirectoryResponse

4.4 Presentation of the Issuer selection list

5. Payment protocol

5.1 General

5.2 TransactionRequest

5.3 TransactionResponse

5.4 Errors when executing Transaction Protocol

5.5 Redirect to the Internet banking environment (issuerAuthenticationURL)

5.6 Redirect to the Merchant environment (merchantReturnURL)

5.7 Errors during execution of redirects

5.8 Four different scenario's for completion of iDEAL Mobile payment

5.9 Performance and time-out of payment message

5.10 Booking of payments

5.11 Specific requirement iDEAL Mobile: Print or e-mail confirmation message

6. Query protocol

6.1 General

6.2 StatusRequest

6.3 StatusResponse

6.4 Errors during execution of Query Protocol

6.5 Collection duty

6.6 Performance and time-out of status message

7. Error handling

7.1 General

7.2 ErrorResponse

7.3 Non-availability

8. Security and certificates

8.1 General principles of certificates

8.2 Signing iDEAL messages

8.3 Authentication of iDEAL messages

8.4 Creating a key pair

9. Presentation

9.1 General

9.2 Payment method

9.3 Payment button

- 9.4 Payment flow
 - 9.5 Redirect to Issuer
 - 9.6 Frames
 - 9.7 New Window
 - 9.8 Issuer list
 - 9.9 iDEAL banners and logo's
 - 9.10 Explaining iDEAL to Consumers
 - 10. Common Implementation pitfalls
 - 10.1 iDEAL Issuer selection list: keep it up-to-date
 - 10.2 iDEAL transaction status – verify the final status in time
 - 10.3 iDEAL in your mobile app – wrongful use of in-app-browsing via webview
 - Appendix A: Data dictionary
 - Appendix B: Example messages
 - APPENDIX C: Error codes
 - Categories
 - Error codes
 - APPENDIX D: XML messages scheme
 - Interface Merchant – Acquirer
-

1. Introduction

▼ ...

1.1 Target audience

iDEAL Merchants are organisations (Merchants) that want to receive iDEAL payments and have signed an iDEAL contract with a bank. This document is intended for iDEAL Merchants that want to connect to the iDEAL platform of the Acquiring bank they have chosen, and provides a description of all messages that are exchanged between the Merchant and the bank. The messages that are exchanged between the Issuer and the Merchant bank (Acquirer) are not of importance to the Merchant, and therefore will not be discussed in this document. This document is not bank specific, which means that only generic iDEAL specifications are mentioned in this guide. Please contact your Acquiring bank for any information or support on a bank specific connection or implementation.

1.2 Document structure

Chapter 2 provides an overview of all parties involved in iDEAL, and the various messages that are exchanged within the scope of an iDEAL transaction. The overall structure of the exchanged messages is outlined in the third chapter. Chapter 4 describes the Directory protocol, which provides the list of Issuing banks within iDEAL. The Payment protocol is described in chapter 5. The Query protocol, concerning actual status of an iDEAL transaction, is explained in chapter 6. The handling of errors and exceptions is discussed in chapter 7. Chapter 8 describes the security of iDEAL messages. Chapter 9 contains information about the iDEAL presentation (brand and logo). Chapter 10 describes some best practices and common implementation pitfalls.

1.3 Definitions of internet and online banking

In this document there are many references to the 'internet banking' or 'online banking' services of a Consumer bank. For Consumer banks that implement iDEAL Mobile, this should always be read as 'internet banking and/or mobile banking'. For every instance where internet or online related terminology is used, please interpret this as including the mobile channel. Where mobile use of iDEAL leads to specific additional requirements for iDEAL, this is indicated separately in the text.

1.4 Revisions

Version	Description	Release date
0.8	First English version for review by Currence	25-06-2009
2.2.1	Final version reviewed by Currence	30-06-2009
2.2.2	Added extra requirement for Query protocol collection duty	10-09-2009
2.2.3	Added passage greying out in Issuer list Added passage increasing chance of success web shops Added passage payment flow Adjustment passage redirect to Issuer Added passage Consumer error messages Added passage entrance codes Added Merchant-Acquirer XSD scheme	19-05-2010
3.3.0	Final version for approval by CAB	09-01-2012

3.3.1	New version based on iDEAL 3.3.1 Standard	27-03-2012
3.3.1 corrected	Corrections applied to OpenSSL example syntax	22-10-2012
3.3.1 correction November	Minor corrections on use of Byte Order Mark	08-11-2012
3.3.1 correction November II	Minor correction on use of empty fields	20-11-2012
3.3.1 corrections July	<ul style="list-style-type: none"> • Clarification on XMLdsig and RmtInfo • TransactionCreateDateTimestamp, • Removed temporary field length conditions • Clarified supported character set • Clarification on use of canonicalization method • Correction in the openssl command for creating a certificate • Removed information on the Nederlandse credit transfer • Added information on platform versions supporting RSA with SHA-256 digital signatures 	03-07-2013
3.3.1 corrections January 2014	<ul style="list-style-type: none"> • Supplementary information on handling of errors during execution of payment protocol and status protocol 	01-02-2014
3.3.1 corrections February 2014	<ul style="list-style-type: none"> • Mobile addendum and quick guide have been integrated for additional information about Mobile Implementations. 	17-03-2014
3.3.1 corrections October 2014	<ul style="list-style-type: none"> • Added rule regarding the redirect to the Issuer in paragraph 5.5. 	08-10-2014
3.3.1 addition February 2015	<ul style="list-style-type: none"> • Changes made to links to new iDEAL.nl website • Explanation to AP2910 error code • Explanation to use Merchant Signing certificates 	15-02-2015
3.5	<ul style="list-style-type: none"> • General presentation improvements • Added 3-minutes trigger for query protocol • Generalization of Issuer Selection List presentation requirements • Use of Universal Links and Android App links for Merchant Redirect iDEAL mobile • Support for at least TLS 1.1 is required • General clarification on common implementation errors • Consumer message inconsistency adjusted • Removed amount verification requirement in StatusResponse 	Q3 2017

2. Overview

2.1 What is iDEAL?

iDEAL was developed by the Dutch banking community in order to facilitate easier payment for online products and services. iDEAL enables direct and secure real-time online payments between bank accounts of Consumers and iDEAL Merchants. The main characteristics of iDEAL are:

- Real-time payment through accepted and trusted Internet banking that is already familiar to Consumers
- Real-time payment authorisation for the Consumer and real-time confirmation to the Merchant by the Acquiring bank, followed by the irrevocable transfer of funds to the Merchant
- Suitability for online delivery (e.g. downloads, mobile top-ups), offline delivery (e.g. goods) and time-critical payments (e.g. airline tickets)
- Offers the flexibility to make payments for many different purposes (e.g. charitable donations, telephone/e-mail orders)

In practice, nearly every Consumer that uses Internet banking with one of the Issuing banks that support iDEAL can pay with the iDEAL payment method.

2.2 What is iDEAL Mobile?

iDEAL is an online payment method for the Dutch market. Although it was originally developed for use with internet banking services, it is now also possible for Issuers to create iDEAL implementations based on mobile banking services like mobile web sites or mobile apps. This is called iDEAL Mobile.

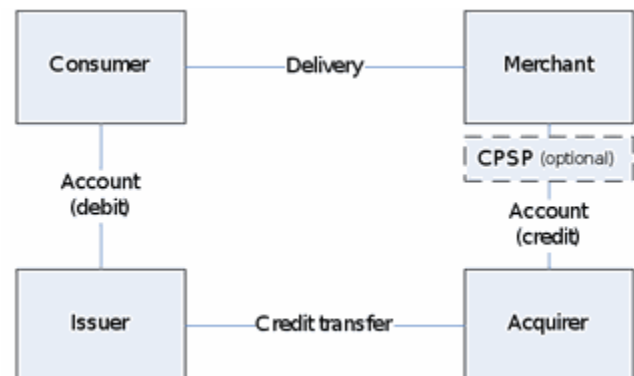
The main characteristics of iDEAL Mobile are:

- There are no changes in the messages sent between Consumer bank and Merchant bank and no changes in messages sent between Merchant and bank;
- Merchant and Consumer do not need to take extra steps for a mobile iDEAL transaction. The redirecting of the Consumer to the mobile banking channel is done automatically by the Consumer's bank; For banks that support iDEAL in their mobile banking app, the Consumer can choose whether to pay using the mobile web browser or the mobile banking app.
- iDEAL Mobile is based on the same mechanisms to ensure trust, security and convenience as used in a desktop environment. In cases mobile technology does not support the same technical security measures as a desktop computer the bank will implement alternative measures to compensate.

Every Consumer that uses internet banking with one of the Issuing banks that supports iDEAL can pay with the iDEAL payment method on a mobile device (although it may be necessary for a Consumer to download and register a mobile application). Those Issuers that do not (yet) have an iDEAL Mobile implementation or that have an implementation that doesn't reach the majority of Consumers will still be able to process transactions through their regular (desktop focussed) iDEAL pages on a mobile device's browser.

2.3 Four party model

There are at least four parties involved in an iDEAL transaction. First there is the Consumer that buys a product or service online. The Consumer buys this from a Merchant that offers the iDEAL payment method; usually this is a web Merchant. The web Merchant is usually referred to as "Merchant" by the banks. The Consumer has a relation with his bank where he can execute iDEAL payments in his Internet banking environment. Within iDEAL, the Consumers' bank is called the Issuer. The Merchant has a relation with his bank in order to accept iDEAL payments. Within iDEAL, the Merchant's bank is referred to as the Acquirer. As stated in the introduction, this document will only cover the iDEAL messages that are exchanged between the Merchant and the Acquirer. However, the iDEAL messages



that are exchanged between the Acquirer and the Issuer will be briefly explained if necessary to provide a good understanding of the entire iDEAL transaction.

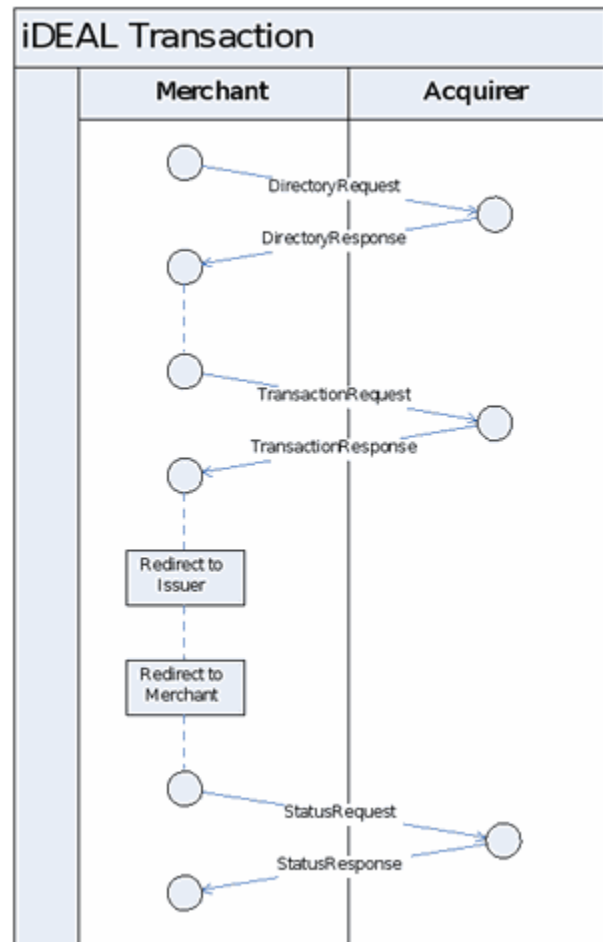
The four party model

Besides the four parties mentioned that are always involved in an iDEAL transaction, additional parties can be involved. The Merchant can, for example, use a Payment Service Provider (PSP) to establish the connection with its Acquiring bank. When this PSP receives/collects the payments before they are paid to the Merchant, this is called a "Collecting PSP" (CPSP). In this case the Collecting PSP acts as the Merchant for the purpose of the iDEAL payments and holds the iDEAL contract with the Acquiring bank on behalf of one or multiple other Merchants. Other roles related to iDEAL payments are out of scope for this document. (Other roles include Distributing PSPs (DPSPs) / Technical PSPs (TPSPs), which act as a software provider for the merchant and can provide a technical connection between merchant and Acquiring bank. They do not play a part in the payment process itself and do not have a contract with a bank and are therefore not included in the figure). The figure shows the parties in this model and their relations.

A demo of an iDEAL payment can be found online at <https://www.ideal.nl/demo/en/>. A typical iDEAL transaction comprises (request-/response-) XML messaging and browser redirects, which handle the initiation, and processing of the transaction in a particular sequence, with all parties involved being informed on the status of the transaction.

The steps in this transaction are shown in the figure.

- By using the Directory protocol the Merchant sends a DirectoryRequest to the Acquirer. This is a request in XML format to obtain the list of participating Consumer banks (Issuers) from the Acquirer. The Acquirer will provide this list back to the Merchant by sending back the DirectoryResponse. The Merchant will show the list of banks, which was sent in the DirectoryResponse to the Consumer. The Consumer will choose his bank from this list. The Directory protocol is explained in more detail in [chapter 4](#).
- By using the Payment protocol the Merchant sends a TransactionRequest to the Acquirer, containing the Issuer chosen by the Consumer, the amount, a purchase id and other transaction details. This message also contains the merchantReturnURL. This URL is used by the Issuer to redirect the Consumer back to the Merchant's website when he has completed the payment process. After the Acquirer has received the message from the Merchant, he sends a separate message to the Issuer that was selected by the Consumer. In return, the Issuer responds with a message that contains the issuerAuthenticationURL (and other data). The Acquirer passes this issuerAuthenticationURL together with a unique TransactionID back to the Merchant via the TransactionResponse message, which is the response to the TransactionRequest. The Merchant now redirects the Consumer to the issuerAuthenticationURL, which refers to the page of the Internet banking portal where the Issuer provides the appropriate prefilled iDEAL transaction details. The Consumer authorizes the payment and receives a confirmation from the Issuer. Then the Consumer is redirected back to the website of



Representation of the steps in an iDEAL transaction

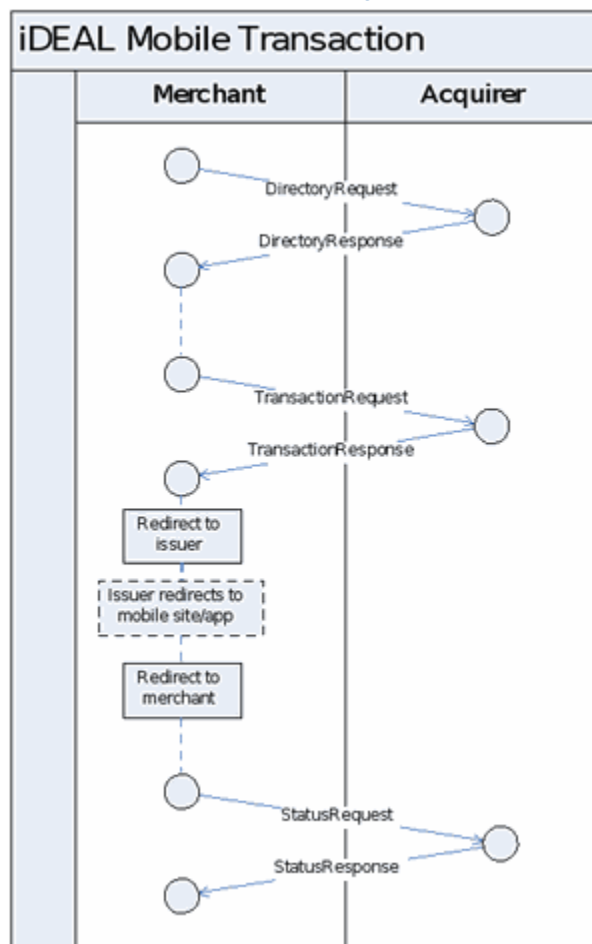
the Merchant via the merchantReturnURL. The entire Payment protocol and the 2 redirects are further described in [chapter 5](#).

- Finally the Merchant initiates the Query protocol by sending a StatusRequest message to the Acquirer. The Acquirer will request the transaction status, if necessary, from the appropriate Issuer and returns the status to the Merchant. If all steps in the transaction were successful this status message contains the proof of payment for the Merchant. [Chapter 6](#) contains more detailed information on the Query protocol. Instead of a regular response to the messages mentioned above, it is also possible that an ErrorResponse is returned. This can be the case if the request contains an error, or if an error occurs during the processing of the request. The ErrorResponse messages are discussed in [chapter 7](#).

The next chapter describes the general format of iDEAL messages. In subsequent chapters the three protocols are discussed in more detail.

✓ [Specific requirement iDEAL Mobile: Redirect to Issuer can be diverted to mobile app or mobile web page of Issuer](#)

The iDEAL Mobile transaction flow is almost identical to the transaction flow in a regular iDEAL transaction. The only difference is the redirect to the 'landing page' (using the issuerAuthenticationURL) where the Consumer, using a mobile device, is either directly sent to the mobile banking app or can choose to be redirected to the Issuer's (mobile) web page or to the Issuer's mobile banking app.



Representation of the steps in an iDEAL Mobile transaction.

3. Message format

3.1 General

This chapter contains a description of the general message structure for the Directory protocol, the Payment protocol and the Query protocol. The subsequent sections will describe the specific fields within the XML messages for each protocol in more detail.

3.2 Header format

The following HTTP header is used for all messages:

Data element	Mandatory	Explanation
Content-Type	Yes	Defines how the remainder of the content is to be interpreted. Contains the value: text/xml; charset="UTF-8".

- All messages must comply with the HTTP 1.1 standard, as defined in RFC 2616 of W3C. For more information: <http://www.w3.org/Protocols/rfc2616/rfc2616.html>
- Each XML request message must be sent as the body of a HTTP POST message.
- Each XML response message must be sent as the body of a HTTP 200 OK message.

The following XML header is used for all messages:

Data element	Mandatory	Explanation
Version	Yes	The XML version according to W3C: 1.0
Encoding	Yes	The character encoding used for (the content of) the XML: UTF-8

Example

```
POST /nl/issuerInformation/getIssuerInformation.xml HTTP/1.1
Content-type: text/xml, charset="UTF-8"
Content-Length: 1201
Host: ideal.abnamro.nl
```

```
<?xml version="1.0" encoding="UTF-8"?>
<DirectoryReq
  xmlns="http://www.idealdesk.com/ideal/messages/mer-acq/3.3.1" version="3.3.1">
  <createDateTimestamp>2008-11-14T09:30:47.0Z</createDateTimestamp>
  <Merchant>
    <merchantID>100000001</merchantID>
    <subID>1</subID>
  </Merchant>
  <Signature
    xmlns="http://www.w3.org/2000/09/xmldsig#">
    <SignedInfo>
      <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
      <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
      <Reference URI="">
        <Transforms>
          <Transform Algorithm="
http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
```

```

        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
        <DigestValue>I7JHyxH/KHcF3KM2xWGVMzSXVQ1MBnD9vInj1XWVNpw=</DigestValue>
    </Reference>
</SignedInfo>
<SignatureValue>
oMve3wVdMSRIMYfMxNfQ8S34BAGebJ0nntobesvTkSaT6pJxhunHjQUQhAt9nIhcwkg3UzdQJZw6UgjX6Pet2tML2nY63
</SignatureValue>
    <KeyInfo>
        <KeyName>7D665C81ABBE1A7D0E525BFC171F04D276F07BF2</KeyName>
    </KeyInfo>
</Signature>
</DirectoryReq>

```

3.3 XML Namespace declaration

XML namespace declaration in iDEAL messages can be done in any way allowed by the XML standards (default namespace declaration or namespace qualification/prefixes).

Most of the example messages given in this document use the default method of namespace declaration. At the end of appendix B one example is given of a message with namespace prefixes. Both types of messages are valid and must be accepted.

3.4 Conventions for empty fields

In iDEAL an XML tag for an optional or conditional field is either:

- present (in which case, the tag must be filled with a valid value)
- or not present at all.

XML tags without content are not allowed and will result in an error message.

3.5 Merchant information registered with Acquirer

Besides the transaction information that the Merchant provides in the iDEAL messages described in the following chapters, the Acquirer also adds information to the iDEAL messages from its own records. Some of this information needs to be registered by a Merchant with the Acquirer before the Merchant can send in its first iDEAL transactions. The relevant iDEAL information is described below:

Data element	Merchant	
Sub-element	Format	Description
legalName	AN..max70	The legal name of the Merchant as registered with the Acquirer. Used together with Merchant.tradeName to represent the Merchant (e.g. Merchant.legalName collecting for Merchant.tradeName). Also used as a name for the Merchant.merchantIBAN for the transfer by the Issuer. It should be the same as the name of the bank account of the Merchant, as registered with the Acquirer.
tradeName	AN..max35	The trade name of the Merchant, as registered with the Acquirer in case it differs from the legalName.
merchantIBAN	ANS..max34	The IBAN of the Merchant, as registered with the Acquirer. (This is linked to Merchant.merchantID.)
merchantBIC	ANS..max11	The BIC of the bank where the Merchant's account is held

Merchant information registered by the acquirer.

▼ [Legenda of used abbreviations](#)

Notation	Meaning
AN	The field can contain any alphanumeric characters. This is followed by a number that indicated the length (or maximum length) of the field.
ANS	The field can contain a restricted set of alphanumeric characters (letters and numbers only). This is followed by a number that indicated the length (or maximum length) of the field.
N	The field contains a numeric value. This is followed by a number that indicated the length (or maximum length) of the field.
PN	The field contains a padded numeric value. The contents are extended to the maximum length by leading zeros.
CL	Indicates a code list of allowed values (enumeration). Which values are allowed is always explained together with the field.
DT	Indicates a date-time field.Format yyyy-MM-ddTHH:mm:ss.SSSZ. (ISO standard 8601). The time used is 'universal time' UTC (formerly GMT), without daylight saving time. Example: 2009-12-28T13:59:59.393Z
RDT	Indicates a relative date-time field (ISO standard 8601). For example: PT10M equals 10 minutes.
DEC(#1,#2)	Indicates a decimal value with #1 being maximum total digits and #2 being maximum fraction digits. DEC(6,2) can be 1234.56 for example.

4. Directory protocol

4.1 General

The Directory protocol allows a Merchant to fetch an up to date list of participating Issuers from his Acquirer, which can be presented to the Consumer. In case of changes in the list of Issuers, the Directory protocol will automatically take care of the update and make it visible in the Issuer lists of all Merchants.

It is not allowed to perform the Directory protocol for each transaction. Since the list of Issuers only changes occasionally, it is sufficient to execute the Directory protocol on a daily basis and check if the list has changed based on the `directoryDateTimestamp`. If the Issuer list has changed, the latest version has to be saved and used for any subsequent transaction. Acquirers will normally also inform all Merchants (e.g. by email) about changes in their Issuer list. The Directory protocol should at least be executed once a month.

The Directory protocol (like the Payment protocol and the Query protocol) consists of a HTTP POST request from the Merchant to the Acquirer, followed by a HTTP response. The `DirectoryRequest` is sent to the URL that is provided to the Merchant by the Acquirer for this specific purpose. This URL can be different from the one that is used for the `TransactionRequest` and the `StatusRequest`, but it can also be the same URL.

The Acquirer validates the authenticity of the message sent by the Merchant by verifying the signature in the message. In order to validate the Acquirer will need the Merchant's Certificate also containing the public key. The way in which the public part of the Merchant certificate is communicated with the Acquirer varies per bank. Please refer to [chapter 8](#) for more information on authentication and security.

4.2 DirectoryRequest

The `DirectoryRequest` consists of an XML message that is sent to the Acquirer with HTTP POST (see [chapter 3](#) for more information). The table below shows all fields and formatting of the `DirectoryRequest`. [For legenda see 3.5.](#)

Fields of the DirectoryRequest

Name	Description	Format
<code>createDateTimestamp</code>	Date and time at which the directory request message was created.	DT
<code>merchantID</code>	MerchantID as supplied to the Merchant by the Acquirer. If the MerchantID has less than 9 digits, leading zeros must be used to fill out the field.	PN..9
<code>subID</code>	Merchant subID, as supplied to the Merchant by the Acquirer, if the Merchant has requested to use this. A Merchant can state a request to the Acquirer to use one or more subIDs. In this way apart from the Legal Name the Trade name will also be shown on the bank statements for each subID used. Unless agreed otherwise with the Acquirer, the Merchant has to use 0 (zero) as subID by default (if no subIDs are used).	N..max6
<code>SignedInfo</code>	This element contains information about the signature as described in W3C XMLdsig specifications	*
<code>SignatureValue</code>	Contains the electronic signature as described in W3C XMLdsig specifications.	*
<code>KeyInfo</code>	Contains information (fingerprint) about the certificate that is used for generating the digital signature, so the receiver can use the right public key for validating the signature as described in W3C XMLdsig specifications.	*

*`SignedInfo`, `SignatureValue` and `KeyInfo` are XML Signature data elements that are defined in the *XML-Signature*

Syntax and Processing (Second Edition) W3C Recommendation 10 June 2008. The signature is described in more detail in chapter 8. The XML Schema for XML Signatures is available from W3C at the following URL: <http://www.w3.org/TR/2002/REC-xmlsig-core-20020212/xmlsig-core-schema.xsd>.

4.3 DirectoryResponse

The Merchant will receive the DirectoryResponse as a reply to the DirectoryRequest. This XML message contains a list of Issuer names with their corresponding issuerID (BIC). Issuers are grouped by country. The banks in the Merchant's country of choice may be presented at the top in the Issuer selection list, the rest are sorted alphabetically by country, then by bank name. The table below shows all fields that appear in the DirectoryResponse message. For legenda see 3.5.

Fields of the DirectoryResponse

Name	Description	Format
createDateTimestamp	Date and time at which the response message was created.	DT
acquirerID	Unique four-digit identifier of the Acquirer within iDEAL.	PN..4
directoryDateTimestamp	The date and time on which the Directory was updated by the Acquirer.	DT
countryNames	Contains the countryNames in the official languages of the country, separated by a '/' symbol (e.g. 'België/Belgique') Country names need only be displayed if there are banks from more than one country on the Issuer list. Currently, all banks in the list are from the Netherlands, so currently the country name can be omitted.	
issuerID	Bank Identifier Code (BIC) of the iDEAL Issuer.	ANS..max11
issuerName	The name of the Issuer (as this should be displayed to the Consumer in the Merchant's Issuer list).	AN..max35
SignedInfo	This element contains information about the signature as described in W3C XMLdsig specifications	*
SignatureValue	Contains the electronic signature as described in W3C XMLdsig specifications.	*
KeyInfo	Contains information (fingerprint) about the certificate that is used for generating the digital signature, so the receiver can use the right public key for validating the signature as described in W3C XMLdsig specifications.	*

*SignedInfo, SignatureValue and KeyInfo are XML Signature data elements that are defined in the *XML-Signature Syntax and Processing (Second Edition) W3C Recommendation 10 June 2008*. The signature is described in more detail in chapter 8. The XML Schema for XML Signatures is available from W3C at the following URL: <http://www.w3.org/TR/2002/REC-xmlsig-core-20020212/xmlsig-core-schema.xsd>.

4.4 Presentation of the Issuer selection list

To ensure that the Consumer experience of an iDEAL transaction is consistent and recognizable through all Merchant websites; all Merchants have to comply with certain presentation standards:

- All Issuers in the DirectoryResponse (to be collected at least monthly) have to be shown in a list (e.g. dropdown list or list of radio buttons) in alphabetic order and exactly as presented in the DirectoryResponse message.
- The list should be accompanied by the instruction phrase "Kies uw bank" (UK: "Choose your bank"). In case of an HTML <SELECT>, the first element in the list states this instruction phrase and is selected by default (to prevent accidental Issuer selection).
- It is not allowed to exclude or grey out any active Issuers from the Issuer list. In case of a new Issuer, the Issuer list should be updated within one month (preferably earlier).
- It is recommended to configure the HTML "value" field of the items in the list box to be the issuerID (BIC) of the corresponding Issuer, because this value is used in subsequent messages (TransactionRequest).
- The Merchant may preselect an Issuer only to allow for an improved user experience (e.g. if the Consumer has previously initiated an iDEAL payment with a specific Issuer). The Consumer must however always be offered the possibility to alter the preselected Issuer.

An example of a correct presentation of the Issuer selection list is shown in the figure.



Example of possible Issuer list presentation method (in Dutch)

If a Merchant learns via the iDEAL Notification System (Central Reporting tool for iDEAL banks to state system non-availability) or via error messages received from the Acquiring bank that a particular Issuing bank is currently not available, the Merchant may display a message on its website informing Consumers that the particular bank is not available. In other words, it is perfectly permissible to display a message conveying such information but it is not permissible to temporarily remove or grey out the Issuing bank concerned from the Issuer selection list.

5. Payment protocol

5.1 General

The Payment protocol initiates the exchange of messages of the actual iDEAL payment initiation. After the Consumer has chosen iDEAL as a payment method and has selected his bank, the Merchant sends a TransactionRequest to the Acquirer. Within the iDEAL standards this message is referred to as the AcquirerTransactionRequest. The Acquirer replies to the TransactionRequest with a TransactionResponse. This TransactionResponse will also (among other fields) contain the issuerAuthenticationURL. This URL will redirect the browser of the Consumer to the Issuer in order to let him authorize the payment.

5.2 TransactionRequest

The XML message sent by the Merchant to the Acquirer to initiate the payment contains the fields shown in below table. For legenda see 3.5.

Fields of the TransactionRequest

Name	Description	Format
createDateTimestamp	Date and time at which the TransactionRequest message was created.	DT
issuerID	The ID (BIC) of the Issuer selected by the Consumer, as stated in the Issuer list.	ANS..max11
merchantID	MerchantID as supplied to the Merchant by the Acquirer. If the MerchantID has less than 9 digits leading zeros are used to fill out the field.	PN..9
subID	Merchant subID, as supplied to the Merchant by the Acquirer, if the Merchant has requested to use this. A Merchant can state a request to the Acquirer to use one or more subIDs. In this way apart from the Legal Name the Trade name will also be shown on the bank statements for each subID used. Unless agreed otherwise with the Acquirer, the Merchant has to use 0 (zero) as subID by default (if no subIDs are used).	N..max6
merchantReturnURL	Valid Merchant URL (not necessarily beginning with http:// or https://) which must redirect the Consumer from the Issuer back to the Merchant website after authorisation of the transaction by the Consumer. Example: https://www.webshop.nl/processpayment	AN..max512
purchaseID	Unique identification of the order within the Merchant's system. This ID ultimately appears on the payment confirmation (Bank statement / account overview of the Consumer and Merchant).	ANS..max35
amount	The amount payable in euro (with a period (.) used as decimal separator).	DEC(12,2)
currency	Currency in which payment should be effected, expressed using the three-letter international currency code as per ISO 4217; Since iDEAL currently only supports Euro payments, value should always be 'EUR'.	ANS..3

expirationPeriod	Optional: the period of validity of the payment request as stated by the Merchant measured from the receipt by the Issuer. The Consumer must authorise the payment within this period. Otherwise the Issuer sets the status of the transaction to 'Expired'. Value period according to ISO 8601: PnYnMnDTnHnMnS. Minimum value:PT1M or PT60S (1 minute); maximum value: PT1H, PT60M or PT3600S (1 hour). If left blank by the Merchant the Issuer sets the default expirationPeriod to PT30M (thirty minutes). Since the majority of the successful payments are executed within fifteen minutes, it is recommended to set the expiration time to a maximum of PT15M. To accommodate the minimum time required for the Consumer to perform a transaction it is recommended to set the expiration period not lower than three minutes (PT180S or PT3M).	RDT
language	This field enables the Issuer's site to select the Consumer's preferred language (e.g. the language selected on the Merchant's site), if the Issuer's site supports this. Code list in accordance with ISO 639-1. (Dutch = 'nl', English = 'en'). If a non-supported or non-existing language is entered the standard language of the Issuer is used. It is recommended to use 'nl' by default since not all Issuers support other languages.	CL AN..2
description	Description of the product(s) or services being paid for. This field must not contain characters that can lead to problems (for example those occurring in HTML editing codes). To prevent any possible errors most iDEAL systems will reject any description that contains HTML-tags and such other code.	AN..max35
entranceCode	The Transaction.entranceCode is an 'authentication identifier' to facilitate continuation of the session between Merchant and Consumer, even if the existing session has been lost. It enables the Merchant to recognise the Consumer associated with a (completed) transaction. The Transaction.entranceCode is sent to the Merchant in the Redirect. The Transaction.entranceCode must have a minimum variation of 1 million and should comprise letters and/or figures (maximum 40 positions). The Transaction.entranceCode is created by the Merchant and passed to the Issuer.	ANS..max40
SignedInfo	This element contains information about the signature as described in W3C XMLdsig specifications	*
SignatureValue	Contains the electronic signature as described in W3C XMLdsig specifications.	*
KeyInfo	Contains information (fingerprint) about the certificate that is used for generating the digital signature, so the receiver can use the right public key for validating the signature as described in W3C XMLdsig specifications.	*

*SignedInfo, SignatureValue and KeyInfo are XML Signature data elements that are defined in the *XML-Signature Syntax and Processing (Second Edition) W3C Recommendation 10 June 2008*. The signature is described in more detail in [chapter 8](#). The XML Schema for XML Signatures is available from W3C at the following URL: <http://www.w3.org/TR/2002/REC-xmlsig-core-20020212/xmlsig-core-schema.xsd>.

5.3 TransactionResponse

If everything goes well the Acquirer will reply to the TransactionRequest with the TransactionResponse. The table below shows all fields of the TransactionResponse message. [For legenda see 3.5](#).

Fields of the TransactionResponse

Name	Description	Format
------	-------------	--------

createDateTimestamp	Date and time at which TransactionResponse message was created.	DT
acquirerID	Unique four-digit identifier of the Acquirer within iDEAL.	PN..4
issuerAuthenticationURL	The complete Issuer URL to which the Consumer shall be redirected by the Merchant for authentication and authorisation of the transaction.	AN..max512
transactionID	Unique 16-digit number within iDEAL. The number consists of the acquirerID (first four positions) and a unique number generated by the Acquirer (12 positions). This ID ultimately appears on payment confirmation (bank statement or account overview of the Consumer and Merchant).	PN..16
transactionCreateDateTimestamp	Date and time at which the transaction was first registered by the Acquirer. This time can be used by Merchant, Acquiring bank and Issuing bank for reporting on the transaction.	DT
purchaseID	Unique identification of the order within the Merchant's system. This ID ultimately appears on the payment confirmation (Bank statement / account overview of the Consumer and Merchant). This field has the same value as in the TransactionRequest.	ANS..max35
SignedInfo	This element contains information about the signature as described in W3C XMLdsig specifications	*
SignatureValue	Contains the electronic signature as described in W3C XMLdsig specifications.	*
KeyInfo	Contains information (fingerprint) about the certificate that is used for generating the digital signature, so the receiver can use the right public key for validating the signature as described in W3C XMLdsig specifications.	*

*SignedInfo, SignatureValue and KeyInfo are XML Signature data elements that are defined in the *XML-Signature Syntax and Processing (Second Edition) W3C Recommendation 10 June 2008*. The signature is described in more detail in chapter 8. The XML Schema for XML Signatures is available from W3C at the following URL: <http://www.w3.org/TR/2002/REC-xmlsig-core-20020212/xmlsig-core-schema.xsd>.

5.4 Errors when executing Transaction Protocol

A number of errors may occur when executing the iDEAL Transaction Protocol. These may be related to unavailability within your own web store environment (Merchant), the Acquirer environment or the Issuer environment.

The following situations may occur:

- The iDEAL payment cannot successfully be initiated.
- You receive an error response (report 'X') from your Acquirer within the set time-out period.
- You do not receive any response within the set time-out period.

In all of the above cases, the transaction protocol cannot be successfully executed. This means it is not possible for the iDEAL payment to take place at that time. In these cases, except when you received a consumermessage in the ErrorResponse, it is recommended that you display a message along the following lines to the Consumer: '*Unfortunately, it is not possible to pay using iDEAL at this time. Please try again later or use an alternative method of payment*'. Or in Dutch: '*Op dit moment is betalen met iDEAL helaas niet mogelijk. Probeer het op een later moment nog eens of gebruik een andere betaalmethode*'.

5.5 Redirect to the Internet banking environment (issuerAuthenticationURL)

After receiving the TransactionResponse the Merchant has to redirect the Consumer to the issuerAuthenticationURL of the selected Issuer bank, as stated in the TransactionResponse message. If the Merchant's page contains HTML frames, these will be removed by the Issuer ('frame busting'). If and when the Consumer returns to the Merchant's website (with the merchantReturnURL), the Merchant will have to completely rebuild its own page for showing the Merchant's order confirmation. For privacy reasons it is not allowed to include any personal and order information of the Consumer in the HTTP referer header (contains information of the URL that the Consumer was redirected from) , when redirecting the Consumer to the chosen bank.

▼ **Specific requirement iDEAL Mobile: Redirect to Issuer (no in-app browser)**

The Merchant needs to provide the redirect to the Issuer from the browser window or Merchant app where the Consumer selected the Issuing bank. If it is not possible to keep the Consumer in the same browser window then this should be communicated to the Consumer (e.g. 'You will now be redirected to the app or mobile website of your bank'). In case of a payment initiated in the Merchant app, it is *not* allowed to present the Issuer approval screens in a webview component within the Merchant's own app (in-app browser via webview). The complete payment flow, up to the redirect back to the Merchant's app, must take place in an app that is trusted by the Consumer, either the Consumer's chosen browser or the Issuer's mobile app. Thus, the issuerAuthenticationURL must, for execution, be offered to the operating system or opened with [SafariViewController](#) of [Chrome Custom Tabs](#). For more information see paragraph 10.3. During the payment flow it must not be possible for the Consumer to initiate another payment through the Merchant's original app.

Relevant details about the redirect from the Merchant to the Issuer's mobile channel:

- The Issuer decides which Consumers to redirect to which channel. For example some Issuers may treat users of tablet devices the same as mobile users while others will treat them like PC users;
- The Merchant should not intervene with the redirect, there is only one issuerAuthenticationURL for the Merchant to use in all transactions, not a separate URL for mobile iDEAL transactions. The issuerAuthenticationURL should be executed by the operating system at all times;
- If the Issuing bank has integrated iDEAL mobile in its mobile banking app, the Consumer is automatically directed to the mobile banking app or offered the option, on a 'landing page', to open the app or pay via the (mobile) web page. On this 'landing page' the Consumer might be offered the option to download the latest version of the mobile banking app, if it is not yet installed on the Consumer's device.

▼ **Specific requirement iDEAL Mobile: Downloading app during the payment and expiration period**

As mentioned above, an Issuer may allow the Consumer to download an (updated) mobile payment app as part of the transaction. In some cases, the extra time needed to download the app and personalise it may cause the payment to exceed the expiration period of the transaction if this is set to a very short interval. In such cases the payment cannot be completed successfully. The Merchant is able to collect the status of the transaction.

In general, Merchants can avoid such failed transactions in the mobile channel by setting longer expiration periods for payments that are done from a Consumer's mobile device. This is also recommended because the bandwidth and signal strength available to mobile internet will often fluctuate, which can cause transactions on this channel to take longer to complete.

5.6 Redirect to the Merchant environment (merchantReturnURL)

After the Consumer has performed the necessary steps at the Issuer he will be presented with a 'Continue' button that must redirect him back to the website of the Merchant with the merchantReturnURL as supplied in the TransactionRequest.

Two GET parameters are appended to this URL: the entranceCode (see paragraph 5.2) with 'ec' as GET parameter name, and the transactionID (see paragraph 5.3) with 'trxid' as GET parameter name. It is also possible for a Merchant to add additional parameters. For example, if the Merchant defines the merchantReturnURL as follows:

<http://www.webshop.nl/processpayment?producttype=electronics>

the final URL will look something like:

<http://www.webshop.nl/processpayment?producttype=electronics&trxid=0010123456789012&ec=4hd7TD9wRn76w>

The entranceCode field, as previously described in paragraph 5.2, should contain a unique value, with the object of preventing message 'sniffing'. Use of the same entranceCode each time would allow malevolent individuals to intercept the data from the merchantReturnURL and make fraudulent use of this information. This is why using unique values for the entranceCode is extremely important.

Note that a Consumer may not always use the redirect back to the Merchant environment that is offered by the Issuer. Also note that in exceptional cases the Issuer may be unable to match the `transactionID` in its system or another error occurs, which makes it impossible to redirect the Consumer back to the Merchant. In all other cases the Consumer is redirected with the parameters defined above, regardless of the final status of the payment (successful, cancelled, failed). The Merchant must then use the query protocol (see [next chapter](#)) to determine the status of the transaction.

✓ Requirements for iDEAL Mobile: redirect to the Merchant environment.

After the Consumer has been authenticated in either the mobile or regular channel and has approved the payment, he is redirected back to the Merchant as normal (using the `merchantReturnURL`). The `merchantReturnURL` usually starts with 'https', redirecting the Consumer back to a page on the mobile device's browser. If the Consumer has initiated the payment from the Merchant's mobile app, the `merchantReturnURL` can be an app handler, which will redirect the Consumer directly to the Merchant app. An app handler is a call that can be used to start an app and request it to initiate a specific action. For example a Merchant's app handler can start with 'nl-companyname-ideal://' and this will open the Merchant's app. Also it is possible to make use of [Universal Links \(iOS\)](#) or [Android App Links](#) techniques to redirect the Consumer back to a Merchant app.

Note: the `merchantReturnURL` should always direct to a web page or app of the Merchant (or party acting on behalf of the Merchant).

5.7 Errors during execution of redirects

The following errors may occur during execution of the redirect to the online banking environment (Issuer), the execution of the payment at the Issuer and/or the redirect back to your (Merchant) environment:

- The bank page is unavailable as a result of which the Consumer cannot pay, but neither can they be properly redirected to your confirmation page.
- The bank page is available but the Consumer cannot (after paying or otherwise) be properly redirected to your confirmation page.

In both situations the Consumer cannot (as the result of a disturbance) return to your confirmation page in the normal way. In that case the Consumer can return to your website by using the 'back' button or entering the URL, for example. If the Consumer can be identified (for example because he or she has logged into the Merchant environment or via the browser session), the advice in these situations is to check the status of the payment and notify the Consumer of this.

If the Consumer can be identified and the status can be checked but is found to still be 'open', we recommend that the Consumer is shown the following message: *We didn't receive a confirmation from your bank. When you see that your payment has been completed, we will deliver your order after we have received the payment.* Or in Dutch: *We hebben van uw bank nog geen bevestiging van uw betaling ontvangen. Als u in uw Internetbankieren ziet dat uw betaling heeft plaatsgevonden, zullen wij na ontvangst van de betaling tot levering overgaan.*

If the Consumer cannot be identified, your system should check the status of the payment at the end of the expiration period. In such situations, we also recommend that the status of the payment is reported to the Consumer – as soon as it has become final – in one of the ways indicated below:

- By e-mail.
- On your website, for example in the account of the Consumer or via the Consumer's browser session.

5.8 Four different scenario's for completion of iDEAL Mobile payment

Four different scenario's that are possible in case of an iDEAL Mobile payment have been specified. There are four different scenarios possible because either the Issuer or the Merchant can both use a (mobile) web page or a mobile app.

Scenario	Merchant	Issuing Bank
1	(Mobile) web page	(Mobile) web page
2	(Mobile) web page	Mobile Banking App
3	Mobile app	Mobile Banking App
4	Mobile app	(Mobile) web page

Four different scenario's for the completion of an iDEAL mobile payment

▼ [This overview show specific attention points in the iDEAL mobile process flow for different scenario's](#)

Step	Description	<i>Scenario 1 - Merchant (mobile) web page to Issuer (mobile) web page</i>	<i>Scenario 2 - Merchant (mobile) web page to Issuer Mobile Banking App</i>	<i>Scenario 3 - Merchant mobile app to Issuer Mobile Banking App</i>	<i>Scenario 4 - Merchant mobile app to Issuer (mobile) web page</i>
1	The Consumer selects the items to purchase, selects iDEAL as the payment method, selects his Issuing Bank.				
2	The Consumer is redirected to the Issuer of his choice			It is mandatory for the Merchant to let the Operating System, which is installed on the Consumer's mobile device, handle the issuerAuthenticationURL or make use of the safe in-app browsing techniques SafariViewController and Chrome Custom Tabs. See paragraph 5.5 for more information.	
3	The Issuer automatically leads the consumer to the mobile banking app (step 4 is then skipped) or presents the Issuer 'landing page' to the Consumer, which offers the option to complete the iDEAL payment in the Issuer's mobile banking app or in the Issuer's (mobile) web page.				
4	The Consumer selects the (mobile) web page or the Issuer Mobile Banking app.				

5	The Consumer is redirected to the Issuer's (mobile) web page or Mobile Banking app where he can log in and authorize the iDEAL payment. After completion of the payment the consumer is shown the result of the payment by the Issuer.			
6	The Consumer is redirected back, by the Issuer, to the Merchant's app or (mobile) web page using the merchantReturnURL, which was received from the Merchant.		Because the payment takes place in the bank's app, outside of the web-browser setting, the browser session may be lost. This means the Merchant may not be able to recognize the Consumer using the browser session. Next to this, when redirecting the Consumer back to the Merchant from the bank-app, the merchantReturnURL is handled by the Operating System of the mobile device. The OS uses the native (default) browser to handle this URL. This discontinues the original browser session if the transaction was initiated in a non-native browser.	
7	The Merchant shows the Consumer the result of the iDEAL payment.	The merchantReturnURL usually starts with https:// and contains two parameters (entranceCode and transactionID) that can be used to correctly identify the Consumer upon his return.	The merchantReturnURL contains an app handler and two parameters (entranceCode and transactionID) that can be used to correctly identify the Consumer upon his return. Also it is possible to make use of Universal Links (iOS) or Android App Link techniques to redirect the consumer directly back to the merchant app.	

Steps iDEAL Mobile + attention points per scenario

5.9 Performance and time-out of payment message

The performance of the Issuer and Acquirer systems has a direct influence on the Consumer's user experience. Therefore iDEAL sets a target time and time-out for the transaction response message. For a Merchant the relevant target time and time-out apply to the communication with its iDEAL Acquirer:

Communication	Target time (in seconds)	Time-out (in seconds)
TransactionRequest --> TransactionResponse	2.0	7.6

Performance requirements (for the 95th percentile)

The target time is the time (in seconds) within which a *TransactionResponse* message should be received by the Merchant after sending a *TransactionRequest*. The time-out is the length of time after which the Merchant should no longer expect a response (most likely an error has occurred) and should act accordingly (for example by displaying an appropriate error message to the Consumer). 95th percentile is a statistical term indicating that 95% of transactions in a tested sample should be within the set target time.

5.10 Booking of payments

An iDEAL Transaction Request that has been successfully authorized by the Consumer will eventually result in a payment from the Consumer's account at the Issuing bank to the Merchant's account. This has the form of a SEPA Credit Transfer (SCT). Below is a description of the information in this payment message and how it relates to the information in the iDEAL messages.

SEPA Credit Transfer

EPC pacs.008.001.02 data elements	iDEAL data elements
2.3 End to End Identification	<code>Transaction.statusDateTimeStamp</code> ; <code>Transaction.transactionID</code> (optional)
2.12 Local Instrument	Literal: IDEAL
2.15 Category Purpose	Literal: EPAY
2.18 Interbank Settlement Amount - XML attribute: Currency	<code>Transaction.amount</code> <code>Transaction.currency</code>
2.49 Debtor - Name	<code>Transaction.consumerName</code> Note: in the exceptional case where the Issuer is not able to determine the consumer name the Issuer fills the field <code>consumerName</code> with "N/A". The SCT payment must be SCT compliant however and therefore must always include the consumer name.
2.50 Debtor Account - Identification - IBAN	<code>Transaction.consumerIBAN</code>
2.51 Debtor Agent - Financial Institution Identification - BIC	<code>Transaction.consumerBIC</code>
2.53 Creditor Agent - Financial Institution Identification - BIC	<code>Merchant.merchantBIC</code>
2.55 Creditor - Name	<code>Merchant.legalName</code>
2.56 Creditor Account - Identification - IBAN	<code>Merchant.merchantIBAN</code>
2.75 Remittance Information	<ul style="list-style-type: none"> • <code>Transaction.purchaseID</code> (max 35 chars); • <code>Transaction.transactionID</code> (16 chars); • <code>Transaction.description</code> (max 35 chars); • <code>Merchant.tradeName</code> (max 35 chars, optional); <p>These elements always appear in this order and are separated by exactly one space.</p> <div style="border: 1px solid orange; padding: 5px; margin-top: 10px;"> <p>Note: the element <code>Transaction.description</code> and the element <code>Merchant.tradeName</code> (if present) may also contain one or more spaces.</p> </div>

Mapping of transaction details for a SEPA Credit Transfer

In 2.3 End to End Identification, `Transaction.statusDateTimeStamp` is mandatory and is presented in the local time zone

of the Issuer (including daylight saving changes) and format of the Consumer with a maximum of 16 positions (e.g. DD-MM-YYYY HH:MM). This can optionally be followed by `Transaction.transactionID` (which is also included in Remittance Information), separated by a space.

5.11 Specific requirement iDEAL Mobile: Print or e-mail confirmation message

After a successful regular iDEAL payment the Issuer will always give the Consumer the option to print a confirmation of the payment. However in a mobile payment environment printing will often not be possible so this requirement is expanded to include alternatives such as e-mailing the confirmation to the Consumer or receiving confirmation in the Consumer's internet banking portal.

6. Query protocol

6.1 General

To verify whether a transaction was successful the Merchant will start the Query protocol by sending a StatusRequest to the Acquirer. This can be done after the return of the Consumer to the Merchant's website (after the redirect from the Issuer), or after a specified amount of time (for example 5 or 10 minutes after the expirationPeriod expired). To avoid unnecessary system load status requests should not be made unnecessarily often, see 6.5 for more details on what is allowed. Within the iDEAL standards this message is also referred to as the AcquirerStatusRequest.

6.2 StatusRequest

The table below contains all fields that are part of the StatusRequest XML message. For legenda see 3.5.

Fields of the StatusRequest

Name	Description	Format
createDateTimestamp	Date and time at which the StatusRequest message was created.	DT
merchantID	MerchantID as supplied to the Merchant by the Acquirer. If the merchantID has less than 9 digits leading zeros are used to fill out the field.	PN...9
subID	Merchant subID, as supplied to the Merchant by the Acquirer, if the Merchant has requested to use this. A Merchant can request permission from the Acquirer to use one or more subIDs. In this way apart from the Legal Name the Trade name will also be shown on the bank statements for each subID used. Unless agreed otherwise with the Acquirer, the Merchant has to use 0 (zero) as subID by default (if no subIDs are used).	N..max6
transactionID	Unique 16-digit number within iDEAL. The number consists of the acquirerID (first four positions) and a unique number generated by the Acquirer (12 positions). This ID ultimately appears on payment confirmation (statement of the Consumer and Merchant or confirmation screen).	PN..16
SignedInfo	This element contains information about the signature as described in W3C XMLdsig specifications	*
SignatureValue	Contains the electronic signature as described in W3C XMLdsig specifications.	*
KeyInfo	Contains information (fingerprint) about the certificate that is used for generating the digital signature, so the receiver can use the right public key for validating the signature as described in W3C XMLdsig specifications	*

*SignedInfo, SignatureValue and KeyInfo are XML Signature data elements that are defined in the *XML-Signature Syntax and Processing (Second Edition) W3C Recommendation 10 June 2008*. The signature is described in more detail in [chapter 8](#). The XML Schema for XML Signatures is available from W3C at the following URL: <http://www.w3.org/TR/2002/REC-xmlsig-core-20020212/xmlsig-core-schema.xsd>.

6.3 StatusResponse

The reply to the StatusRequest contains the fields listed in Table 14. This message communicates the status of the transaction (related to the transactionID which was sent in the StatusRequest) to the Merchant. If the status equals "Success" some additional fields with information about the Consumer are added to the message. If necessary, this information can be used to

refund (part of) the transaction amount to the Consumer. For legenda see 3.5.

Fields of the StatusResponse.

Name	Description	Format
createDateTimestamp	Date and time at which the StatusResponse message was created.	DT
acquirerID	Unique four-digit identifier of the Acquirer within iDEAL.	PN..4
transactionID	Unique 16-digit number within iDEAL. The number consists of the acquirerID (first four positions) and a unique number generated by the Acquirer (12 positions). This ID ultimately appears on payment confirmation (bank statement or account overview of the Consumer and Merchant).	PN..16
status	Indicates one of the following statuses of the transaction: Success: Positive result; the payment is guaranteed. Cancelled: Negative result due to cancellation by Consumer; no payment has been made. Expired: Negative result due to expiration of the transaction; no payment has been made. Failure: Negative result due to other reasons; no payment has been made. Open: Final result not yet known. A new status request is necessary to obtain the status.	CL AN..max9
statusDateTimestamp	If Status = Success, Cancelled, Expired or Failure This is the date and time at which the Issuer established the Transaction.status for this transaction and recorded it as part of the transaction details.	DT
consumerName	Only included if Status = Success Name of the Consumer according to the name of the account used for payment. In the exceptional case that the consumerName cannot be retrieved by the Issuer, this is filled with 'N/A'. If governing law prevents Issuers outside the Netherlands from disclosing this information, field may be omitted.	AN..max70
consumerIBAN	Only included if Status = Success The IBAN of the Consumer Bank account used for payment. If governing law prevents Issuers outside the Netherlands from disclosing this information, field may be omitted.	ANS..max34
consumerBIC	Only included if Status = Success BIC of the bank where the Consumer account is held. If governing law prevents Issuers outside the Netherlands from disclosing this information, field may be omitted.	ANS..max11
amount	Only included if Status = Success The amount in euro guaranteed by the Acquirer to the Merchant (including decimal separator).	DEC(12,2)
currency	Only included if Status = Success Currency of the amount guaranteed, expressed using the three-letter international currency code as per ISO 4217; Since iDEAL only supports Euro payments at this moment, the value should always be 'EUR'.	ANS..3
SignedInfo	This element contains information about the signature as described in W3C XMLdsig specifications	*
SignatureValue	Contains the electronic signature as described in W3C XMLdsig specifications.	*
KeyInfo	Contains information (fingerprint) about the certificate that is used for generating the digital signature, so the receiver can use the right public key for validating the signature as described in W3C XMLdsig specifications	*

*SignedInfo, SignatureValue and KeyInfo are XML Signature data elements that are defined in the *XML-Signature Syntax and Processing (Second Edition) W3C Recommendation 10 June 2008*. The signature is described in more detail in chapter 8. The XML Schema for XML Signatures is available from W3C at the following URL: <http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/xmldsig-core-schema.xsd>.

6.4 Errors during execution of Query Protocol

When using the Query Protocol to fetch the iDEAL status, errors can occur as a result of which it is impossible for you to obtain the status of the payment at that moment. It is therefore not possible to show the Consumer the final status of the transaction at that moment.

It is recommended that you use the following message if it is not possible to obtain status details: *We didn't receive a confirmation from your bank. When you see that your payment has been completed, we will deliver your order after we have received the payment.* Or in Dutch: *We hebben van uw bank nog geen bevestiging ontvangen. Als u in uw Internetbankieren ziet dat uw betaling heeft plaatsgevonden, zullen wij na ontvangst van de betaling tot levering overgaan.*

In addition to this, we recommend informing your customer how he or she will be notified of the payment status when it is known or where he or she can obtain that information (online).

You can then try to obtain the status through your Acquirer, in accordance with the guidelines (see MIG section 6.5). As soon as a definitive status has been received, you can notify the Consumer of the status of payment in the following ways, for example:

- By e-mail.
- On your website, for example in the account of the Consumer or via the Consumer's browser session.

6.5 Collection duty

The Merchant has to initiate a StatusRequest when the Consumer returns to the page to which he was redirected by the Issuer (the merchantReturnURL from the TransactionRequest). However, it is possible for the Consumer to close his browser before returning to this merchantReturnURL (even when he has authenticated the payment). It is mandatory for Merchants to perform a StatusRequest for every transaction, also in case the Consumer does not return to the Merchant's website. The iDEAL protocol prescribes a so called "collection duty" for the result of every transaction. The Merchant can comply with this "collection duty" by performing a StatusRequest for every transaction when

- the Consumer is redirected to the MerchantReturnURL
- more than 3 minutes have passed after the TransactionResponse and no final status has been retrieved yet
- the transaction's expiration period has passed and no final status has been retrieved yet.

Also when a Consumer does not return to your website because he does not appropriately complete or cancel the iDEAL payment (e.g. when he/she clicks away its browser screen), the Merchant must at least perform the Query Protocol after the expiration time has ended, to collect the final status at the iDEAL Acquiring bank.

If the retrieved status of a transaction is "Open", the StatusRequest for this transaction has to be repeated after a short period of time. All other statuses (Cancelled, Expired, Success and Failure) are final statuses. Since these statuses cannot change anymore, it is not necessary (and not allowed) to perform another StatusRequest. To avoid unnecessary load on the iDEAL systems, Merchants shall not perform unnecessary StatusRequests.

The following situations must **never** occur:

- Request the status of a transaction more than 5 times before the expiration period has passed;
- Perform repeated StatusRequests with a time interval shorter than 60 seconds

The following situations should **not** occur after the expiration period has passed:

- Perform repeated StatusRequests with a time interval shorter than 60 minutes
- Request the status of a transaction more than 5 times per day
- Request the status of a transaction after the final status of the transaction has been received;
- Perform StatusRequests for transactions with a timestamp older than 7 days;
- Stop requesting the status of a transaction before the final status of the transaction has been received (unless timestamp

is older than 7 days).

Usually one of the final statuses should be returned shortly after the expiration period. If the "Open" status is still returned after the expiration period, this can indicate a system failure. If this failure is not solved within 24 hours, please contact the Acquirer and stop sending StatusRequests. When the returned status is "Success" the Merchant will be able to proceed to deliver of the ordered product or service, using the saved order data. When the Consumer returns to your website inappropriately (e.g. using browser buttons to get back from the Issuing bank Internet Banking screens to the Merchant website) and decides to start a new iDEAL payment request, the Merchant must send a new transaction request to the Acquiring Bank, after first trying to collect the final status of the iDEAL transaction that was initiated earlier.

6.6 Performance and time-out of status message

The performance of the Issuer and Acquirer systems has a direct influence on the Consumer's user experience. Therefore iDEAL sets a target time and time-out for the status response message. For a Merchant the relevant target time and time-out apply to the communication with its iDEAL Acquirer:

Communication	Target time (in seconds)	Time-out (in seconds)
StatusRequest - StatusResponse	2.0	7.6

Performance requirements (for the 95th percentile)

The target time is the time (in seconds) within which a StatusResponse message should be received by the Merchant after sending a StatusRequest. The time-out is the length of time after which the Merchant should no longer expect a response (most likely an error has occurred) and should act accordingly (for example by displaying an appropriate error message to the Consumer). 95th percentile is a statistical term indicating that 95% of transactions in a tested sample should be within the set target time.

7. Error handling

7.1 General

If an error occurs while processing a *DirectoryRequest*, *TransactionRequest* or *StatusRequest*, for example because a request contains an invalid value, an *ErrorResponse* will be returned instead of the regular response. The *ErrorResponse* has the same structure for all three types of requests.

7.2 ErrorResponse

Instead of the regular response (*DirectoryResponse*, *TransactionResponse* or *StatusResponse*) the Acquirer will return an *ErrorResponse* if an error occurs during the reception or processing of the request, or if the request contains values that are not allowed or do not comply with the required format. The table below lists the fields that appear in the *ErrorResponse*. For legenda see 3.5.

Fields of the ErrorResponse.

Name	Description	Format
createDateTimestamp	Date and time at which the <i>ErrorResponse</i> message was created.	DT
errorCode	Unique characteristic of the occurred error within the iDEAL system. Please refer to Appendix C for the error code list.	CL AN..6
errorMessage	Descriptive text accompanying the <i>ErrorCode</i> .	AN..max128
errorDetail	Details of the error. As determined and described by the Acquirer.	AN..max256
suggestedAction	Suggestions aimed at resolving the problem.	AN..max512
consumerMessage	An Acquirer can include a (standardised) message here which the Merchant should show to the Consumer. Below you will find more information about the <i>consumerMessage</i>	AN..max512
SignedInfo	This element contains information about the signature as described in W3C XMLdsig specifications	*
SignatureValue	Contains the electronic signature as described in W3C XMLdsig specifications.	*
KeyInfo	Contains information (fingerprint) about the certificate that is used for generating the digital signature, so the receiver can use the right public key for validating the signature as described in W3C XMLdsig specifications	*

**SignedInfo*, *SignatureValue* and *KeyInfo* are XML Signature data elements that are defined in the *XML-Signature Syntax and Processing (Second Edition) W3C Recommendation 10 June 2008*. The signature is described in more detail in chapter 8. The XML Schema for XML Signatures is available from W3C at the following URL: <http://www.w3.org/TR/2002/REC-xmlsig-core-20020212/xmlsig-core-schema.xsd>.

The *consumerMessage* is a message intended to show to the Consumer in case of an error. It contains one of the following messages (in Dutch):

- "Betalen met iDEAL is nu niet mogelijk. Probeer het later nogmaals of betaal op een andere manier." This message is returned in case something goes wrong in the Payment protocol.
- "Het resultaat van uw betaling is nog niet bij ons bekend. U kunt desgewenst uw betaling controleren in uw

internetbankieren." This message is returned in case something goes wrong in the Query protocol.

- "De geselecteerde iDEAL bank is momenteel niet beschikbaar i.v.m. onderhoud tot naar verwachting date time from Notification System. Probeer het later nogmaals of betaal op een andere manier." This message is returned in case an Issuer is unavailable due to scheduled maintenance.
- "De geselecteerde iDEAL bank is momenteel niet beschikbaar. Probeer het later nogmaals of betaal op een andere manier." This message is returned in case an Issuer is unexpectedly unavailable.

Merchants should always show the standardised consumerMessage generated by the Acquirer to their customers. The other fields in the ErrorResponse should be used for local error detection. [Appendix C](#) contains a list of the possible error codes for the various messages and the associated text of the errorDetail field.

7.3 Non-availability

It might be possible that one of the Issuers is temporarily unavailable. In this case transactions that have to be processed by this Issuer will generate an ErrorResponse (see previous). When the Acquirer has determined unavailability at an Issuer he will communicate this to the Issuer immediately. This means that Merchant will never have to contact the Issuer directly.

It might also be possible that the Acquirer itself is temporarily unavailable. In this case, unless the Merchant has more than one Acquirer, no iDEAL transactions can be processed and the Acquirer will generate an ErrorResponse.

Also your confirmation page may fail to work properly. In that case, we recommend that you display a clear error message to the Consumer.

Subsequently, we also recommend that the status of the payment is reported to the Consumer in one of the ways indicated below:

- By e-mail.
 - On your website, in the Consumer's account
 - On your website, as part of the session information of the order.
-

8. Security and certificates

8.1 General principles of certificates

For asymmetric encryption 2 keys are used: one public key and one private key. The public key is linked to the certificate and can be shared with anyone. The private key must be kept confidential by the owner of the certificate. The specific characteristics of the private and public part of the certificates will allow the encryption of a message with the public part while the result can be decrypted with the private part, and vice versa. It is not possible to decrypt a text with the same key that was used to encrypt it. These specific characteristics enable two applications of a certificate:

1. Encryption of a message. By encrypting a message with the public key of the receiving party the information can only be read by the recipient (who has sole knowledge of the private key).
2. Creating an electronic signature of a message. By encrypting the (hash of a) message with the private key, the recipient can determine the authenticity of the (sender of the) message by successfully decrypting the signature with the public part of the certificate. The recipient will also verify the integrity of the message to make sure the contents of the message was not changed by a third party.

The one-sided TLS connection that is used within iDEAL between the Merchant and the Acquirer is based on the first application. The TLS connection uses at least 128-bit encryption based on a server side certificate of the Acquirer. It is important that your TLS implementation always complies with the latest security standards and patches (at least TLS 1.1 should be supported). Not implementing these patches might lead to the situation that it is no longer possible to submit transaction requests. Since iDEAL does not put any constraints on the communication between the Consumer and the Merchant, this can be either with or without a TLS connection. Merchants are advised, however, to always use TLS on the payment pages of their website.

The iDEAL standard also uses electronic signatures to ensure the authenticity, integrity and non- repudiation of all messages with the exception of redirects. The electronic signature of the Acquirer in the StatusResponse message, for example, enables the Merchant to verify the authenticity of the payment confirmation.

8.2 Signing iDEAL messages

All messages that are sent by the Merchant to the Acquirer (DirectoryRequest, TransactionRequest and StatusRequest) have to be signed by the Merchant. Messages are signed in accordance with the "XML Signature Syntax and Processing (2nd Edition) W3C Recommendation" of 10 June 2008 <http://www.w3.org/TR/xmlsig-core/>, with the following settings and restrictions applied:

1. The entire XML message must be signed. Signature reference to the signed info URI is left blank, see example messages in [Appendix B](#).
2. For the purpose of generating the digest of the main message, the [inclusive canonicalization algorithm](#) must be used. This method of canonicalization of the main message is not (always) explicitly indicated in the iDEAL XML messages. For this reason this transform has not been included in the example messages in this document. Merchants are not required to explicitly indicate this transform in their messages.
3. For the purpose of generating the signature value, the [exclusive canonicalization algorithm](#) must be used.
4. The syntax for an [enveloped signature](#) must be used. The signature itself must be removed from the XML message using the default transformation prescribed for this purpose.
5. For hashing purposes the [SHA-256 algorithm](#) must be used.
6. For signature purposes the [RSAWithSHA256 algorithm](#) must be used. RSA keys must be 2,048 bits long.
7. The public key must be referenced using a fingerprint of an X.509 certificate. The fingerprint must be calculated according to the following formula: HEX(SHA-1(DER certificate)). See example messages in [Appendix B](#).

Note: the key reference is backwards compatible with all previous versions of iDEAL.

Note: According to Base64 specifications line breaks are allowed to be inserted after each 76 characters using a CR/LF <http://tools.ietf.org/html/rfc2045#section-6.8>.

In general Merchants don't need to have extensive knowledge of RSA since most programming languages have libraries available that implement XML Digital Signature processing. It is strongly recommended to use these standard libraries. Standard

functionality for creation and verification of RSAWithSHA256 digital signatures is available in commonly used software platforms, from the following versions and higher: PHP version 5.3.0, Microsoft .NET version 3.5 sp1 en Java version 1.6 u18. This functionality may also be available in earlier versions of these platforms and in other platforms (e.g. Python, Ruby). For information about creating the public and private key pair please refer to paragraph 8.4.

8.3 Authentication of iDEAL messages

To ensure the status of a transaction the Merchant has to verify the signature of the Acquirer in the Response messages. To verify the signature in the SignatureValue field it is recommended that merchants use standard XML Digital Signature libraries which are available in most (web) programming languages.

8.4 Creating a key pair

If you want to use a so-called "self signed certificate" this paragraph will explain how to do so. It is also possible to purchase a certificate at a company specialized in this field (Certificate Authority), see below.

In order to create a public and a private key execute the following steps:

1. Download the "OpenSSL Library" from <http://www.openssl.org>. You can find more information on the "certificate generating utility" at: <http://www.openssl.org/docs/apps/req.html>. You may also generate the key pair using other software. If so please use the manual that comes with your software. Always make sure you have the latest version installed to be sure that your implementation applies to the latest security standards.
2. Generate an "RSA private key" using the following command(choose your own password for the field [privateKeyPass]):

```
openssl genrsa -aes-128 -out priv.pem -passout pass:[privateKeyPass] 2048
```

3. Create a certificate based on the "RSA private key" (use the same password as in the previous step for the field [privateKeyPass]):

```
openssl req -x509 -sha256 -new -key priv.pem -passin pass:[privateKeyPass] -days 1825 -out cert.cer
```

This OpenSSL command will generate a certificate in X.509 format, with a validity period of 5 years (1825 days), the maximum for iDEAL signing certificates.

4. The file priv.pem contains the private key. The file cert.cer contains the certificate with the public key. The Merchant has to keep the priv.pem file private, which is used in the RSA encryption. The cert.cer file has to be communicated to the Acquirer. The method of communication will depend on the Acquirer.

Buying a certificate from a Certificate Authority

When buying a certificate from a Certificate Authority (CA), rather than generating the certificate yourself it is important to note the following: the CA signing certificate (and the rest of the certificate chain) must use hashing algorithms and key lengths that are at least as secure or better than those of the Merchant certificate. Therefore CA-certificates used to sign certificates for electronic signatures must use at least SHA-256 for hashing and 2,048 bits for RSA keys. Signing certificates should also have a maximum validity period of 5 years.

9. Presentation

9.1 General

There are some requirements regarding the presentation of the iDEAL payment method on the Merchant's website. The main purpose of these requirements is to create a uniform user experience for Consumers whenever they pay with iDEAL, regardless which Merchant's website they use. The requirements are further explained in the following paragraphs.

9.2 Payment method

A Merchant that accepts iDEAL as a payment method has to place the iDEAL payment method in its list of offered payment methods, in such a way that it is a logical step in the Merchant's ordering process.

The iDEAL payment method must be presented in the list of payment methods in such a way that it receives at least the same amount of attention as other payment methods. Since iDEAL has the highest frequency of use in the Netherlands, it is recommended to place iDEAL at the top.

9.3 Payment button

It must be clear for the Consumer how and when the iDEAL payment method has been chosen. This is achieved by displaying a 'Payment' button, generally on that part of the page where one of the payment methods is normally selected.

The permitted depictions of the iDEAL 'Payment' button are supplied via the Merchant's portal on the website <http://www.ideal.nl/en/payee/logos-banners/>.

Specific Requirements iDEAL Mobile: Payment button iDEAL Mobile

In cases where the iDEAL logo cannot be displayed at the required size on a mobile device (for example if this would violate the requirements for app icons on the mobile device) the mandatory free space of 15px around the logo may be reduced to accommodate the requirements of the mobile device.

9.4 Payment flow

When the iDEAL button is clicked to select a payment method, the Consumer should immediately be presented with the iDEAL Issuer selection list without any intermediate screens being displayed by the Merchant (e.g. Consumer login and/or registration screens). And when the required Issuing bank has been selected by the Consumer, the Consumer should be immediately redirected to the iDEAL environment of the selected Issuing bank.

In other words, Consumer login and registration with the Merchant should have taken place before selection of the iDEAL payment method by clicking the iDEAL button.

9.5 Redirect to Issuer

The Merchant needs to perform the redirect to the Issuer, from the browser window where the Consumer selected the Issuing bank. The complete page of the Merchant shall be replaced by the complete page of the selected Issuing bank. Therefore it is not allowed to open the redirect to the Issuer in a new browser window. It is however allowed to open a new window, with visible address bar, before the Consumer selects his bank from the Issuer list.

9.6 Frames

Frames used on the Merchant's site are allowed. The page of the Issuing bank will remove these frames using a frame busting technique. This will allow the Consumer to verify whether the iDEAL payment is really taking place at the bank chosen from the

Issuer list. After the redirect to the Merchant the Merchant shall completely rebuild the Merchant page to show the Merchant's order confirmation.

9.7 New Window

The iDEAL payment may take place in a new browser window, as long as the Merchant will have this window appear at (or before) the moment the Consumer chooses the payment method. A new window is only allowed if initiated by the Consumer (no pop-ups are allowed). The complete payment flow must take place in this window, including the Merchant's order confirmation. The new window must also contain an address bar that allows the Consumer to check the Internet address URL and SSL-certificate of the Issuing bank. During the payment flow it should not be possible for the Consumer to initiate another payment through the Merchant's original browser window.

Specific Requirements iDEAL Mobile: New window or app iDEAL Mobile

The iDEAL Mobile payment may redirect the Consumer to a different mobile page or app as part of the transaction. The Merchant should strive to keep the Consumer on one browser page as much as possible but should not make use of an in-app browser via webview in the Merchant app (see 5.5). In those cases where changing to another app or window is necessary (such as the redirect to the Issuer) the Consumer should be informed beforehand in order to avoid confusion (e.g. " You are being redirected to the mobile web site or the app of your bank").

9.8 Issuer list

The Issuer list has to be presented as described in [paragraph 4.4](#).

9.9 iDEAL banners and logo's

Merchants that want to use a banner on their website to promote iDEAL can download the most recent banners at <http://www.ideal.nl/en/payee/logos-banners/>. These banners need to be installed only once, since the URL will always refer to the latest version of the banner containing the right logos of the participating Issuing banks.

When using a iDEAL banner on mobile devices it is recommended to either user the logo without the pay-off or use a pay-off spread over 2 lines to reduce the width taken up on the mobile device screen.

9.10 Explaining iDEAL to Consumers

Merchants that want to provide specific help and instructions to Consumers regarding the iDEAL payment method are advised to use the following text:

How does iDEAL work?

A few simple steps is all it takes to pay using iDEAL:

- Place your order
- Select iDEAL as payment method
- Select your bank
- This opens the familiar online banking application of your own bank
- The relevant details of your purchase will already be shown
- You merely authorise the payment in the way you are accustomed to at your Issuing bank
- Your bank confirms your payment
- You return to the online store – order accepted and payment successful!

▼ [Or, in Dutch](#)

Hoe werkt iDEAL?

Betalen met iDEAL doet u in een paar stappen:

- U bestelt een product
 - Kies iDEAL als betaalmethode
 - Selecteer de bank waar u uw bankzaken online regelt
 - U komt direct in de bekende internetbankieromgeving van uw bank
 - De relevante gegevens van uw aankoop zijn al ingevuld
 - Op de voor u bekende manier keurt u de betaling goed
 - U ontvangt een bevestiging van uw bank
 - U keert terug naar de webwinkel. Bestelling en betaling geslaagd!
-

10. Common Implementation pitfalls

10.1 iDEAL Issuer selection list: keep it up-to-date

- When you don't keep the iDEAL issuer selection list (bank list) up-to-date, some of your customers are unable to pay with iDEAL and furthermore you are in breach with iDEAL regulations.
- New iDEAL Issuers need to be added to your iDEAL issuer list as soon as possible, but no later than 1 month after the new Issuer joined iDEAL.
- The easiest way is to keep your issuer list updated fully automatic. You can do this in the following ways:
 - When possible, make use of an iDEAL landing page provided by your bank or CPSP. The Issuer selection list that is used on this page is automatically kept up-to-date by these parties;
 - In case you have an iDEAL contract with an Acquiring bank (you will receive the iDEAL payments directly at your account at this bank), then make sure your Issuer-list is connected to the Directory protocol. When you have a contract with a CPSP for iDEAL, often an iDEAL issuer list API is provided. Keep your issuer selection list up-to-date by automatically executing the iDEAL Directory protocol or API call on a daily or weekly basis.
- Do you use a plug-in for iDEAL (in e.g. Woocommerce or Magento)? Then make sure that you regularly check that you have implemented the latest version of this plug-in.
- Do you opt to manually adjust the issuer selection list? Then make sure that the needed technical resources are available in your organization to do this within the required time period (one month). This adjustment is usually small, but does require some programming skills. Contact your software provider or CPSP in case you have any questions about the manual adjustments of your iDEAL issuer selection list.
 - Please also make sure that with a manual adjustment, you also remove issuers that do not offer iDEAL anymore (e.g. Friesland bank)

10.2 iDEAL transaction status – verify the final status in time

- Make sure that you at least have two triggers in place for collecting the iDEAL transaction final status (status Success, Canceled, Expired or Failed):
 1. The return of the user after payment to your merchant environment (via the MerchantReturnURL)
 2. Expiring of the iDEAL transaction expirationPeriod (which is configurable in your transaction request, and by default set to 30 min)
- Not having the second trigger in place can lead to mismatches between your order system and the iDEAL end status. This can result in situations where customers have paid for an order but no confirmation of payment is administered in your order system and, as a result, no product is delivered to the consumer.
- Also make sure that you retrieve the final status of the iDEAL transaction *before* your order system has automatically timed-out the order.
- Make sure you do not make delivery solely based on the return of the consumer to your merchant environment (via the MerchantreturnURL) but solely after receipt of the final status "Success".
- The final end status can only be determined in the following ways:
 - Execute the Query protocol (sending the iDEAL StatusRequest), after which you will receive the StatusResponse which includes the status of the iDEAL transaction, or;
 - In the dashboard of your iDEAL contracting party (if present) look up the final status of the iDEAL transaction.

10.3 iDEAL in your mobile app – wrongful use of in-app-browsing via webview

- When you as a merchant have an app in which you offer iDEAL as a payment option, specifically mind the following aspects regarding the Issuer redirect:
 1. The consumer must be able to check the URL and https "lock" icon of the Issuer webpages at all times
 2. The browser in which the redirect to the Issuer takes place, must be securely safeguarded for the Merchant (the Merchant should not be able to eavesdrop on user typing)
 3. The browser, in which the redirect to the Issuer takes place, must be able to open bank-apps (app schemes like "bank://ideal/12392").
- To be able to comply to the above requirements, we strongly advise you to always offer the IssuerAuthenticationURL to

the operating system of the mobile device. As a result, the IssuerAuthenticationURL will be opened within the browser of choice of the user or directly in the bank-app

- It is strictly forbidden to make use of custom made in-app browsers for the redirect to the Issuer, because by doing this the above requirements are not complied to!
- In case you chose to make use of an in-app browser for opening the IssuerauthenticationURL, you must make use of [SafariViewController](#) for Apple iOS, so that you comply to the above requirements.
- With the use of [Chrome Custom Tabs](#) as in-app web browser, requirements 1 and 2 are complied to, however in some cases it does not allow for a redirect of the user to a bank-app (ING and ABN Amro). When Chrome Custom Tabs is used and one of these two banks is chosen by the user, the IssuerAuthenticationURL must still be offered to the Android Operating system.

Please note that if you as a merchant do not comply to the above requirements, this can lead to the event that your iDEAL contract must be ended.

For tips on how to increase the chances of success of your web shop, see the iDEAL website <http://www.ideal.nl/en/payee/getting-started/recommendations/>.

Appendix A: Data dictionary

The table below contains all data elements that appear in iDEAL messages relevant to the Merchant, together with information about the format and permitted values:

Data element	Sub element	Format	Values, explanation	Occurs in messages
Root	Version	AN..8	3.3.1	All
Root	Xmlns	AN..1024	http://www.idealdesk.com/ideal/messages/mer-acq/3.3.1 http://www.w3.org/2000/09/xmlns#	All
Acquirer	acquirerID	PN..4		All Responses
createDateTimestamp		DT	ISO-8601	All
Directory	directory Date Timestamp	DT	ISO-8601	<i>DirectoryResponse</i>
Directory	countryNames	AN..128		<i>DirectoryResponse</i>
Directory	Issuer - issuerID	ANS..max11	ISO 9362	<i>DirectoryResponse</i>
Directory	Issuer - issuerName	AN..max35		<i>DirectoryResponse</i>
Error	consumerMessage	AN..max512		<i>ErrorResponse</i>
Error	errorCode	CL AN..6		<i>ErrorResponse</i>
Error	errorMessage	AN..max128		<i>ErrorResponse</i>
Error	errorDetail	AN..max256		<i>ErrorResponse</i>
Error	suggestedAction	AN..max512		<i>ErrorResponse</i>
Issuer	issuer Authentication URL	AN..max512	Also in Issuer Redirect	<i>TransactionResponse</i>
Issuer	issuerID	ANS..max11	ISO 9362	<i>TransactionRequest</i>
Merchant	legalName	AN.. max70	Registered by Acquirer	N/A
Merchant	tradeName	AN..max35	Registered by Acquirer	N/A
Merchant	IBAN	ANS..max34	ISO 13616, registered by Acquirer	N/A
Merchant	BIC	ANS..max11	ISO 9362, registered by Acquirer	N/A
Merchant	merchantID	PN..9	Unique within Acquirer	All Requests
Merchant	merchantReturnURL	AN..max512	Also in MerchantRedirect	<i>TransactionRequest</i>
Merchant	subID	N..max6		All Requests
SignedInfo	Exclusive Canonicalization Method		http://www.w3.org/2001/10/xml-exc-c14n#	All

SignedInfo	Inclusive Canonicalization Method		www.w3.org/TR/2001/REC-xml-c14n-20010315	All
SignedInfo	Signaturemethod		http://www.w3.org/2001/04/xmldsig-more#rsa-sha256	All
SignedInfo	Reference - URI		Must be empty	All
SignedInfo	Reference - Transforms		http://www.w3.org/2000/09/xmldsig#enveloped-signature	All
SignedInfo	Reference - DigestMethod		http://www.w3.org/2001/04/xmlenc#sha256	All
SignedInfo	Reference - DigestValue		Base64	All
SignatureValue			Base64	All
KeyInfo	KeyName		Fingerprint of the certificate	All
Transaction	amount	DEC(12,2)	In euro, with period (.) for decimals	<ul style="list-style-type: none"> TransactionRequest StatusResponse
Transaction	consumerIBAN	ANS..max34	ISO13616	StatusResponse
Transaction	consumerBIC	ANS..max11	ISO 9362	StatusResponse
Transaction	consumerName	AN..max70		StatusResponse
Transaction	currency	ANS..3	ISO 4217: EUR for Euro	<ul style="list-style-type: none"> TransactionRequest StatusResponse
Transaction	description	AN..max35	No HTML layout codes	TransactionRequest
Transaction	entranceCode	ANS..max40	Also in MerchantRedirect	TransactionRequest
Transaction	expirationPeriod	RDT	ISO 8601	TransactionRequest
Transaction	language	CL AN..2	ISO 639-1	TransactionRequest
Transaction	purchaseID	ANS..max35		<ul style="list-style-type: none"> TransactionRequest TransactionResponse
Transaction	status	CL AN..max9	Open, Success, Failure, Cancelled, Expired	StatusResponse
Transaction	statusDateTimestamp	DT	ISO 8601	StatusResponse
Transaction	transactionID	PN..16	Also in redirects	<ul style="list-style-type: none"> TransactionResponse StatusRequest StatusResponse
Transaction	transaction Create DateTimestamp	DT	ISO 8601	TransactionResponse

Data elements in iDEAL messages

▼ [Notations Leganda](#)

Notation	Meaning
AN	Alphanumeric, free text
ANS	Alphanumeric, strict (letters and numbers only)
N	Numerical
PN	Numerical (padded), the contents are extended to the maximum length by leading zeros
..max #	Maximum number of positions for alphanumeric and numerical values
..#	Fixed number of positions for alphanumeric and numerical values
CL	Code list, enumeration
DT	Date time field in UTC (no daylight saving time): yyyy-MM-ddTHH:mm:ss.SSSZ
RDT	Relative date time field: PnYnMnDTnHnMnS
DEC(#1,#2)	Decimal, #1 total digits, #2 fraction digits: DEC(6,2) can be 1234.56 for example

Character set

- In all iDEAL messages the Unicode character set must be used. Only the MES-2 subset must be supported.
- Encoding must be used as indicated in the HTTP and XML headers UTF-8 (Unicode Transformation Format).
- The use of characters that are not part of the Unicode character set will not lead to a refusal of a batch or post, but the character may be changed to a space, question mark or asterisk in transit.
- The Byte Order Mark (BOM) must not be used. The UTF-8 representation of the BOM is the byte sequence 0xEF,0xBB,0xBF.

Appendix B: Example messages

The shown examples are not related to each other. For instance, the example DirectoryResponse message is not necessarily the result of the DirectoryRequest example message. Values used in the examples, like the IDs for the Acquirers and Issuers, may differ from those used in the actual iDEAL system.

▼ DirectoryRequest

```
<?xml version="1.0" encoding="UTF-8"?>
<DirectoryReq
  xmlns="http://www.idealdesk.com/ideal/messages/mer-acq/3.3.1" version="3.3.1">
  <createDateTimestamp>2008-11-14T09:30:47.0Z</createDateTimestamp>
  <Merchant>
    <merchantID>100000001</merchantID>
    <subID>1</subID>
  </Merchant>
  <Signature
    xmlns="http://www.w3.org/2000/09/xmldsig#">
    <SignedInfo>
      <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
      <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
      <Reference URI="">
        <Transforms>
          <Transform Algorithm="
http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
        <DigestValue>I7JHyxH/KHcF3KM2xWGVMzSXVQ1MBnD9vInj1XWVNpw=</DigestValue>
      </Reference>
    </SignedInfo>
    <SignatureValue>
oMve3wVdMSRIMYfMxNfQ8S34BAGebJ0nntobesvTkSaT6pJxhunHjQUQhAt9nIhcwkg3UzdQJZw6UgjX6Pet2tML2nY63
</SignatureValue>
    <KeyInfo>
      <KeyName>7D665C81ABBE1A7D0E525BFC171F04D276F07BF2</KeyName>
    </KeyInfo>
  </Signature>
</DirectoryReq>
```

▼ DirectoryResponse

```
<?xml version="1.0" encoding="UTF-8"?>
<DirectoryRes
  xmlns="http://www.idealdesk.com/ideal/messages/mer-acq/3.3.1" version="3.3.1">
  <createDateTimestamp>2008-11-14T09:30:47.0Z</createDateTimestamp>
  <Acquirer>
    <acquirerID>0001</acquirerID>
  </Acquirer>
  <Directory>
    <directoryDateTimestamp>2004-11-10T10:15:12.145Z</directoryDateTimestamp>
    <Country>
      <countryNames>Nederland</countryNames>
    <Issuer>
      <issuerID>ABNANL2AXXX</issuerID>
      <issuerName>ABN AMRO Bank</issuerName>
    </Issuer>
    <Issuer>
```

```

        <issuerID>INGBNL2AXXX</issuerID>
        <issuerName>ING</issuerName>
    </Issuer>
    <Issuer>
        <issuerID>RABONL2UXXX</issuerID>
        <issuerName>Rabobank</issuerName>
    </Issuer>
</Country>
<Country>
    <countryNames>België/Belgique</countryNames>
    <Issuer>
        <issuerID>KREDBE22XXX</issuerID>
        <issuerName>KBC</issuerName>
    </Issuer>
</Country>
</Directory>
<Signature
    xmlns="http://www.w3.org/2000/09/xmldsig#">
    <SignedInfo>
        <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
        <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
        <Reference URI="">
            <Transforms>
                <Transform Algorithm="
http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
            </Transforms>
            <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
            <DigestValue>0R00+jQyZlSLrTA+lgSmsH9vtbzbNbI IoWzh9SDJXGc=</DigestValue>
        </Reference>
    </SignedInfo>
    <SignatureValue>
H5FBG+1ceWVginZdK1bT77g1HL0p8/OXJ1FTw2QWNBOzkKgg1q0L7OvbQTsW3i+RP1TVfvXr35X+C3KUQbZNrn59w6J7
</SignatureValue>
    <KeyInfo>
        <KeyName>7D665C81ABBE1A7D0E525BFC171F04D276F07BF2</KeyName>
    </KeyInfo>
</Signature>
</DirectoryRes>

```

TransactionRequest

```

<?xml version="1.0" encoding="UTF-8"?>
<AcquirerTrxReq
    xmlns="http://www.idealdesk.com/ideal/messages/mer-acq/3.3.1" version="3.3.1">
    <createDateTimeStamp>2008-11-14T09:30:47.0Z</createDateTimeStamp>
    <Issuer>
        <issuerID>RABONL2UXXX</issuerID>
    </Issuer>
    <Merchant>
        <merchantID>100000001</merchantID>
        <subID>1</subID>
        <merchantReturnURL>https://www.ashop.eu/paymentHandling</merchantReturnURL>
    </Merchant>
    <Transaction>
        <purchaseID>iDEALaankoop21</purchaseID>
        <amount>59.99</amount>
        <currency>EUR</currency>
        <expirationPeriod>PT3M30S</expirationPeriod>
        <language>nl</language>
        <description>Documenten Suite</description>
    </Transaction>
</AcquirerTrxReq>

```

```

    <entranceCode>4hd7TD9wRn76w6gGwGFDgdL7jEtb</entranceCode>
  </Transaction>
  <Signature
    xmlns="http://www.w3.org/2000/09/xmldsig#">
    <SignedInfo>
      <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
      <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
      <Reference URI="">
        <Transforms>
          <Transform Algorithm="
http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
        <DigestValue>DsP5NLca+plhp9tZvGwykfb2whQYt3CQ5sbsVd9Q9aE=</DigestValue>
      </Reference>
    </SignedInfo>
    <SignatureValue>
LrfE0po3YPvVxB/m77iBWWiR07Ghiuhuj7tO2C2LKqZK2cLrAiidt+3tjbJ3m16quCFxfh7bmjRtJsGi7a3HKtKqY4aug
    </SignatureValue>
    <KeyInfo>
      <KeyName>7D665C81ABBE1A7D0E525BFC171F04D276F07BF2</KeyName>
    </KeyInfo>
  </Signature>
</AcquirerTrxReq>

```

TransactionResponse

```

<?xml version="1.0" encoding="UTF-8"?>
<AcquirerTrxRes
  xmlns="http://www.idealdesk.com/ideal/messages/mer-acq/3.3.1" version="3.3.1">
  <createDateTimestamp>2008-11-14T09:30:47.0Z</createDateTimestamp>
  <Acquirer>
    <acquirerID>0001</acquirerID>
  </Acquirer>
  <Issuer>
    <issuerAuthenticationURL>
https://www.issuingbank.eu/ideal?random=1Y98dHjPwe2qq3s&trxid=0001000000000001</
issuerAuthenticationURL>
    </Issuer>
  <Transaction>
    <transactionID>0001000000000001</transactionID>
    <transactionCreateDateTimestamp>2008-11-14T09:30:50.125Z</
transactionCreateDateTimestamp>
    <purchaseID>iDEAL21</purchaseID>
  </Transaction>
  <Signature
    xmlns="http://www.w3.org/2000/09/xmldsig#">
    <SignedInfo>
      <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
      <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
      <Reference URI="">
        <Transforms>
          <Transform Algorithm="
http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
        <DigestValue>fU+UQd8JBswjQruM4MOLau8fmFrcfXaZ/kwufu45JUw=</DigestValue>
      </Reference>
    </SignedInfo>
    <SignatureValue>

```

```
U/gCgvwtFOrEaRKctJUw+t/vmx7+VfjQnbKxpkmWpZfzDTC0Pp9/e8A7SQQsiWG8AWk31j1fV1zgUG0fQTXH4bEk3WRQT
</SignatureValue>
  <KeyInfo>
    <KeyName>7D665C81ABBE1A7D0E525BFC171F04D276F07BF2</KeyName>
  </KeyInfo>
</Signature>
</AcquirerTrxRes>
```

▼ StatusRequest

```
<?xml version="1.0" encoding="UTF-8"?>
<AcquirerStatusReq
  xmlns="http://www.idealdesk.com/ideal/messages/mer-acq/3.3.1" version="3.3.1">
  <createDateTimestamp>2008-11-14T09:30:47.0Z</createDateTimestamp>
  <Merchant>
    <merchantID>100000001</merchantID>
    <subID>1</subID>
  </Merchant>
  <Transaction>
    <transactionID>0001000000000001</transactionID>
  </Transaction>
  <Signature
    xmlns="http://www.w3.org/2000/09/xmldsig#">
    <SignedInfo>
      <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
      <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
      <Reference URI=" ">
        <Transforms>
          <Transform Algorithm="
http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
        <DigestValue>8E3Jw9y/wLoSYC56l7TjDuml03B1Cv1117azX/PfIJo=</DigestValue>
      </Reference>
    </SignedInfo>
    <SignatureValue>
JyK+rFRZ6ohYjLsrc98iElbp+xv9rhJgd9tfn2tj1lkqJ7Hm3xLnQtjAe49/b6n+/upkVjrdzRtbQVvHuPH6wuzlxTrvE
</SignatureValue>
    <KeyInfo>
      <KeyName>7D665C81ABBE1A7D0E525BFC171F04D276F07BF2</KeyName>
    </KeyInfo>
  </Signature>
</AcquirerStatusReq>
```

▼ StatusResponse

```
<?xml version="1.0" encoding="UTF-8"?>
<AcquirerStatusRes
  xmlns="http://www.idealdesk.com/ideal/messages/mer-acq/3.3.1" version="3.3.1">
  <createDateTimestamp>2008-11-14T09:30:47.0Z</createDateTimestamp>
  <Acquirer>
    <acquirerID>0001</acquirerID>
  </Acquirer>
  <Transaction>
    <transactionID>0001000000000001</transactionID>
    <status>Success</status>
    <statusDateTimestamp>2008-11-14T09:32:47.0Z</statusDateTimestamp>
    <consumerName>Onderheuvell</consumerName>
    <consumerIBAN>NL44RABO0123456789</consumerIBAN>
    <consumerBIC>RABONL2U</consumerBIC>
    <amount>59.99</amount>
```

```

    <currency>EUR</currency>
  </Transaction>
  <Signature
    xmlns="http://www.w3.org/2000/09/xmldsig#" >
    <SignedInfo>
      <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
      <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
      <Reference URI="" >
        <Transforms>
          <Transform Algorithm="
http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
        <DigestValue>P9rQXN4+XSNXe2XTKT1A5mOxAC5CzaFxpPD0VMxlO2s=</DigestValue>
      </Reference>
    </SignedInfo>
    <SignatureValue>
Tdtj0eUdP9t/WahIBty9yL/k42uuOZybFoKT1Uc/jWs3geYF+fhJtPIJbNxS5W18T6FdHZmlsNnTBzvYg81eFQ1qni1f3
    </SignatureValue>
    <KeyInfo>
      <KeyName>7D665C81ABBE1A7D0E525BFC171F04D276F07BF2</KeyName>
    </KeyInfo>
  </Signature>
</AcquirerStatusRes>

```

▼ ErrorResponse

```

<?xml version="1.0" encoding="UTF-8"?>
<AcquirerErrorRes
  xmlns="http://www.idealdesk.com/ideal/messages/mer-acq/3.3.1" version="3.3.1">
  <createDateTimestamp>2008-11-14T09:30:47.0Z</createDateTimestamp>
  <Error>
    <errorCode>S01100</errorCode>
    <errorMessage>Issuer not available</errorMessage>
    <errorDetail>System generating error: Rabobank</errorDetail>
    <consumerMessage>
De geselecteerde iDEAL bank is momenteel niet beschikbaar i.v.m. onderhoud tot naar
verwachting 31-12-2010 03:30. Probeer het later nogmaals of betaal op een andere manier.
    </consumerMessage>
  </Error>
  <Signature
    xmlns="http://www.w3.org/2000/09/xmldsig#" >
    <SignedInfo>
      <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
      <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
      <Reference URI="" >
        <Transforms>
          <Transform Algorithm="
http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
        <DigestValue>LGCQRlTT14Y5BPe0p0kKLoF7+PxHJ49kJxVfOMIBsSw=</DigestValue>
      </Reference>
    </SignedInfo>
    <SignatureValue>
FfZbFiOhLRiVZ6Kf7eGEEjFH9Wyr8t9WVcSYmKLUzwZZzvsLi51SVwiBTUvQNHxnszYbjXjrQ6Y8Xu2Mcj6bXA9kF1VÇ
    </SignatureValue>
    <KeyInfo>
      <KeyName>7D665C81ABBE1A7D0E525BFC171F04D276F07BF2</KeyName>
    </KeyInfo>
  </Signature>

```

```
</Signature>
</AcquirerErrorRes>
```

▼ **ErrorResponse (incl namespace prefixes)**

```
<?xml version="1.0" encoding="UTF-8"?>
<ns:AcquirerErrorRes
  xmlns:ns="http://www.idealdesk.com/ideal/messages/mer-acq/3.3.1" version="3.3.1">
  <ns:createDateTimestamp>2008-11-14T09:30:47.0Z</ns:createDateTimestamp>
  <ns:Error>
    <ns:errorCode>S01100</ns:errorCode>
    <ns:errorMessage>Issuer not available</ns:errorMessage>
    <ns:errorDetail>System generating error: Rabobank</ns:errorDetail>
    <ns:consumerMessage>
      De geselecteerde iDEAL bank is momenteel niet beschikbaar i.v.m. onderhoud tot naar
      verwachting 31-12-2010 03:30. Probeer het later nogmaals of betaal op een andere manier.
    </ns:consumerMessage>
  </ns:Error>
  <ds:Signature
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
      <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"
    />
      <ds:Reference URI="">
        <ds:Transforms>
          <ds:Transform Algorithm="
http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
        </ds:Transforms>
        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
        <ds:DigestValue>LGCQRlTT14Y5BPe0p0kKLoF7+PxHJ49kJxVfOMIBsSw=</ds:DigestValue
      >
        </ds:Reference>
      </ds:SignedInfo>
      <ds:SignatureValue>
FfZbFiOhLRiVZ6Kf7eGEEjFH9Wyr8t9WVcSYmKLUzwZZzvsLi5lSVwiBTUvQNHxnszYbjJXjRQ6Y8Xu2Mcj6bXA9kF1VQ
</ds:SignatureValue>
      <ds:KeyInfo>
        <ds:KeyName>7D665C81ABBE1A7D0E525BFC171F04D276F07BF2</ds:KeyName>
      </ds:KeyInfo>
    </ds:Signature>
  </ns:AcquirerErrorRes>
```

APPENDIX C: Error codes

Categories

The `Error.errorCode` is composed of:

- a category (two letters)
- a number (four digits)

The following categories are distinguished:

Category	Meaning
IX	Invalid XML and all related problems. Such as incorrect encoding, invalid version, otherwise unreadable.
SO	System maintenance. The errors that are communicated in the event of system maintenance or system failure. Also covers the situation where new requests are no longer being accepted but requests already submitted will be dealt with (until a certain time).
SE	Security and authentication errors. Incorrect authentication methods and expired certificates.
BR	Field errors. Additional information on incorrect fields.
AP	Application errors. Errors relating to IDs, account numbers, time zones, transactions, currencies.

Error code categories

Error codes

errorCode	errorMessage	errorDetail
IX1100	Received XML not valid	See 1)
IX1200	Encoding type not UTF-8	See 1)
IX1300	XML version number invalid	See 1)
IX1600	Mandatory value missing	See 1)
SO1000	Failure in system	See 2)
SO 1100	Issuer unavailable	See 5)
SO1200	System busy. Try again later	See 2)
SO1400	Unavailable due to maintenance	See 2)
SE2000	Authentication error	See 1)
SE2100	Authentication method not supported	See 1)
BR1200	iDEAL version number invalid	See 1)
BR1210	Value contains non-permitted character	See 1)
BR1220	Value too long	See 1)

BR1230	Value too short	See 1)
BR1270	Invalid date/time	See 1)
BR1280	Invalid URL	See 1)
AP1100	MerchantID unknown	See 1)
AP1200	IssuerID unknown	See 1)
AP1300	SubID unknown	See 1)
AP1500	MerchantID not active	See 1)
AP2600	Transaction does not exist	See 1)
AP2900	Selected currency not supported	See 1)
AP2910	Maximum amount exceeded. (Detailed record states the maximum amount).	See 3)
AP2915	Amount too low. (Detailed record states the minimum amount).	See 4)
AP2920	Expiration period is not valid.	See 1)

Error codes

The field `errorDetail` in the table above contains one of the values shown in the table below. The italic printed words shall be replaced by actual values, as indicated.

Indication	errorDetail
1)	Field generating error: <i>location-reference in XML message</i>
2)	System generating error: <i>Issuer/Acquirer</i>
3)	Maximum amount is <i>amount</i>
4)	Minimum amount is <i>amount</i>
5)	System generating error: <i>Name of Issuer</i>

errorDetail

Words printed in italics are to be replaced by actual values, as indicated.

Explanation regarding indication 3: in the maximum amount in the error detail the Issuer will only state the transaction limit or the day limit which are applicable at the Issuer. No personal limit set by the Consumer will be stated here.

The value of `consumerMessage` is specified in *AcquirerErrorRes (X')* by the Acquirer based on the criteria described in the following table. Words printed in italics are to be replaced by actual values, as indicated.

Situation	consumerMessage
error occurred in sending or receiving message <i>A, A', B, B''</i>	Paying with iDEAL is currently not possible. Please try again later or pay using another payment method.
error occurred in sending or receiving message <i>F, F''</i>	The result of your payment can not yet be determined. Please check your Internet banking statement for further details.

<p>error occurred because of unavailability of Issuer (SO1000, SO1100, SO1200 , SO1400 or no response received from Issuer received by Acquirer after sending message C)</p>	<p>The selected iDEAL bank is currently unavailable. Please try again later or pay using another payment method.</p>
<p>error occurred because of unavailability of Issuer (see above) AND additional information is available from the iDEAL Notification System</p>	<p>The selected iDEAL bank is currently unavailable due to maintenance until the expected time of <i>date time from Notification System</i>. Please try again later or pay using another payment method.</p>

consumerMessage

Words printed in italics are to be replaced by actual values, as indicated.

APPENDIX D: XML messages scheme

Interface Merchant – Acquirer

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- iDEAL Messages version 3.3.1: interface Merchant/Acquirer -->
<!-- Copyright © Currence iDEAL B.V. -->
<xs:schema
  xmlns="http://www.idealdesk.com/ideal/messages/mer-acq/3.3.1"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#" targetNamespace="
http://www.idealdesk.com/ideal/messages/mer-acq/3.3.1" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:import namespace="http://www.w3.org/2000/09/xmldsig#" schemaLocation="
xmldsig-core-schema.xsd"/>
  <xs:annotation>
    <xs:documentation>elements defined</xs:documentation>
  </xs:annotation>
  <xs:element name="DirectoryReq">
    <xs:annotation>
      <xs:documentation>Directory Request (A)</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="createDateTimestamp" type="iDEAL.dateTime"/>
        <xs:element name="Merchant">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="merchantID" type="Merchant.merchantID"/>
              <xs:element name="subID" type="Merchant.subID"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element ref="ds:Signature"/>
      </xs:sequence>
      <xs:attributeGroup ref="MessageAttributes"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="DirectoryRes">
    <xs:annotation>
      <xs:documentation>Directory Response (A')</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="createDateTimestamp" type="iDEAL.dateTime"/>
        <xs:element name="Acquirer">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="acquirerID" type="Acquirer.acquirerID"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="Directory">
          <xs:complexType>
            <xs:sequence>
```

```

        <xs:element name="directoryDateTimestamp" type="xs:dateTime" />
        <xs:element name="Country" maxOccurs="unbounded" >
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="countryNames" type=
"Country.countryNames" />
                    <xs:element name="Issuer" maxOccurs="unbounded" >
                        <xs:complexType>
                            <xs:sequence>
                                <xs:element name="issuerID" type=
"Issuer.issuerID" />
                                <xs:element name="issuerName" type=
"Issuer.issuerName" />
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
        <xs:element ref="ds:Signature" />
    </xs:sequence>
    <xs:attributeGroup ref="MessageAttributes" />
</xs:complexType>
</xs:element>
<xs:element name="AcquirerTrxReq">
    <xs:annotation>
        <xs:documentation>Acquirer Transaction Request (B)</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:sequence>
            <xs:element name="createDateTimestamp" type="iDEAL.dateTime" />
            <xs:element name="Issuer">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="issuerID" type="Issuer.issuerID" />
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="Merchant">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="merchantID" type="Merchant.merchantID" />
                        <xs:element name="subID" type="Merchant.subID" />
                        <xs:element name="merchantReturnURL" type="iDEAL.url" />
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="Transaction">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="purchaseID" type="Transaction.purchaseID" />
                        <xs:element name="amount" type="Transaction.amount" />
                        <xs:element name="currency" type="Transaction.currency" />
                        <xs:element name="expirationPeriod" type=
"Transaction.expirationPeriod" minOccurs="0" />
                        <xs:element name="language" type="Transaction.language" />
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>

```

```

        <xs:element name="description" type="Transaction.description"/>
        <xs:element name="entranceCode" type="Transaction.entranceCode"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element ref="ds:Signature"/>
</xs:sequence>
<xs:attributeGroup ref="MessageAttributes"/>
</xs:complexType>
</xs:element>
<xs:element name="AcquirerTrxRes">
    <xs:annotation>
        <xs:documentation>Acquirer Transaction Response (B')</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:sequence>
            <xs:element name="createDateTimestamp" type="iDEAL.dateTime"/>
            <xs:element name="Acquirer">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="acquirerID" type="Acquirer.acquirerID"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="Issuer">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="issuerAuthenticationURL" type="
"Issuer.issuerAuthenticationURL"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="Transaction">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="transactionID" type="Transaction.transactionID"/>
                        <xs:element name="transactionCreateDateTimestamp" type="
iDEAL.dateTime"/>
                        <xs:element name="purchaseID" type="Transaction.purchaseID"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element ref="ds:Signature"/>
        </xs:sequence>
        <xs:attributeGroup ref="MessageAttributes"/>
    </xs:complexType>
</xs:element>
<xs:element name="AcquirerStatusReq">
    <xs:annotation>
        <xs:documentation>Acquirer Status Request (F)</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:sequence>
            <xs:element name="createDateTimestamp" type="iDEAL.dateTime"/>
            <xs:element name="Merchant">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="merchantID" type="Merchant.merchantID"/>
                        <xs:element name="subID" type="Merchant.subID"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>

```

```

        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="Transaction">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="transactionID" type="Transaction.transactionID"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element ref="ds:Signature"/>
</xs:sequence>
<xs:attributeGroup ref="MessageAttributes"/>
</xs:complexType>
</xs:element>
<xs:element name="AcquirerStatusRes">
    <xs:annotation>
        <xs:documentation>Acquirer Status Response (F')</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:sequence>
            <xs:element name="createDateTimestamp" type="iDEAL.dateTime"/>
            <xs:element name="Acquirer">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="acquirerID" type="Acquirer.acquirerID"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="Transaction">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="transactionID" type="Transaction.transactionID"/>
                        <xs:element name="status" type="Transaction.status"/>
                        <xs:element name="statusDateTimestamp" type="iDEAL.dateTime"
minOccurs="0"/>
                        <xs:sequence minOccurs="0">
                            <xs:element name="consumerName" type="Transaction.consumerName"
minOccurs="0"/>
                            <xs:element name="consumerIBAN" type="Transaction.consumerIBAN"
minOccurs="0"/>
                            <xs:element name="consumerBIC" type="Transaction.consumerBIC"
minOccurs="0"/>
                            <xs:element name="amount" type="Transaction.amount"/>
                            <xs:element name="currency" type="Transaction.currency"/>
                        </xs:sequence>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element ref="ds:Signature"/>
        </xs:sequence>
        <xs:attributeGroup ref="MessageAttributes"/>
    </xs:complexType>
</xs:element>
<xs:element name="AcquirerErrorRes">
    <xs:annotation>
        <xs:documentation>Acquirer Error Response (X')</xs:documentation>
    </xs:annotation>
    <xs:complexType>

```

```

    <xs:sequence>
      <xs:element name="createDateTimestamp" type="xs:dateTime" />
      <xs:element name="Error">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="errorCode" type="Error.errorCode" />
            <xs:element name="errorMessage" type="Error.errorMessage" />
            <xs:element name="errorDetail" type="Error.errorDetail" minOccurs="0"
/>
            <xs:element name="suggestedAction" type="Error.suggestedAction"
minOccurs="0" />
            <xs:element name="consumerMessage" type="Error.consumerMessage"
minOccurs="0" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element ref="ds:Signature" />
    </xs:sequence>
    <xs:attributeGroup ref="MessageAttributes" />
  </xs:complexType>
</xs:element>
<xs:annotation>
  <xs:documentation>simpleTypes defined</xs:documentation>
</xs:annotation>
<xs:simpleType name="Acquirer.acquirerID">
  <xs:restriction base="xs:token">
    <xs:length value="4" fixed="true" />
    <xs:pattern value="[0-9]+" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="Country.countryNames">
  <xs:restriction base="xs:token">
    <xs:minLength value="1" />
    <xs:maxLength value="128" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="Error.consumerMessage">
  <xs:restriction base="xs:string">
    <xs:maxLength value="512" fixed="true" />
    <xs:minLength value="1" fixed="true" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="Error.errorCode">
  <xs:restriction base="xs:token">
    <xs:length value="6" fixed="true" />
    <xs:pattern value="[A-Z]{2}[0-9]{4}" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="Error.errorDetail">
  <xs:restriction base="xs:string">
    <xs:maxLength value="256" fixed="true" />
    <xs:minLength value="1" fixed="true" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="Error.errorMessage">
  <xs:restriction base="xs:string">
    <xs:minLength value="1" />
    <xs:maxLength value="128" />
  </xs:restriction>
</xs:simpleType>

```

```

    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="Error.suggestedAction">
  <xs:restriction base="xs:string">
    <xs:maxLength value="512" fixed="true"/>
    <xs:minLength value="1" fixed="true"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="Issuer.issuerAuthenticationURL">
  <xs:restriction base="iDEAL.url"/>
</xs:simpleType>
<xs:simpleType name="Issuer.issuerID">
  <xs:restriction base="iDEAL.BIC"/>
</xs:simpleType>
<xs:simpleType name="Issuer.issuerName">
  <xs:restriction base="xs:token">
    <xs:maxLength value="35" fixed="true"/>
    <xs:minLength value="1" fixed="true"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="Merchant.merchantID">
  <xs:restriction base="xs:token">
    <xs:length value="9" fixed="true"/>
    <xs:pattern value="[0-9]+"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="Merchant.merchantReturnURL">
  <xs:restriction base="iDEAL.url"/>
</xs:simpleType>
<xs:simpleType name="Merchant.subID">
  <xs:restriction base="xs:nonNegativeInteger">
    <xs:maxInclusive value="999999" fixed="true"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="Transaction.amount">
  <xs:restriction base="xs:decimal">
    <xs:totalDigits value="12"/>
    <xs:fractionDigits value="2"/>
    <xs:minExclusive value="0"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="Transaction.consumerBIC">
  <xs:restriction base="iDEAL.BIC"/>
</xs:simpleType>
<xs:simpleType name="Transaction.consumerIBAN">
  <xs:restriction base="iDEAL.IBAN"/>
</xs:simpleType>
<xs:simpleType name="Transaction.consumerName">
  <xs:restriction base="xs:token">
    <xs:minLength value="1"/>
    <xs:maxLength value="70"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="Transaction.currency">
  <xs:restriction base="xs:token">
    <xs:length value="3" fixed="true"/>
    <xs:pattern value="EUR"/>
  </xs:restriction>

```

```

</xs:simpleType>
<xs:simpleType name="Transaction.description">
  <xs:restriction base="xs:token">
    <xs:minLength value="1" fixed="true" />
    <xs:maxLength value="35" fixed="true" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="Transaction.entranceCode">
  <xs:restriction base="xs:token">
    <xs:minLength value="1" fixed="true" />
    <xs:maxLength value="40" fixed="true" />
    <xs:pattern value="[a-zA-Z0-9]+" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="Transaction.expirationPeriod">
  <xs:restriction base="xs:duration">
    <xs:minInclusive value="PT1M" fixed="true" />
    <xs:maxInclusive value="PT1H" fixed="true" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="Transaction.language">
  <xs:restriction base="iDEAL.language" />
</xs:simpleType>
<xs:simpleType name="Transaction.purchaseID">
  <xs:restriction base="xs:token">
    <xs:minLength value="1" />
    <xs:maxLength value="35" />
    <xs:pattern value="[a-zA-Z0-9]+" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="Transaction.status">
  <xs:restriction base="xs:token">
    <xs:pattern value="Open|Success|Failure|Expired|Cancelled" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="Transaction.transactionID">
  <xs:restriction base="xs:token">
    <xs:length value="16" fixed="true" />
    <xs:pattern value="[0-9]+" />
  </xs:restriction>
</xs:simpleType>
<xs:annotation>
  <xs:documentation>basic simpleTypes defined</xs:documentation>
</xs:annotation>
<xs:simpleType name="iDEAL.BIC">
  <xs:restriction base="xs:token">
    <xs:pattern value="[A-Z]{6,6}[A-Z2-9][A-NP-Z0-9]([A-Z0-9]{3,3}){0,1}" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="iDEAL.IBAN">
  <xs:restriction base="xs:token">
    <xs:pattern value="[a-zA-Z]{2,2}[0-9]{2,2}[a-zA-Z0-9]{1,30}" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="iDEAL.dateTime">
  <xs:restriction base="xs:dateTime">
    <xs:pattern value=".+Z" />
  </xs:restriction>

```



```

</xs:simpleType>
<xs:simpleType name="iDEAL.language">
  <xs:restriction base="xs:token">
    <xs:length value="2" fixed="true"/>
    <xs:pattern value="[a-z]"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="iDEAL.url">
  <xs:restriction base="xs:anyURI">
    <xs:maxLength value="512"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="iDEAL.version">
  <xs:restriction base="xs:string">
    <xs:pattern value="3\.3\.1"/>
  </xs:restriction>
</xs:simpleType>
<xs:annotation>
  <xs:documentation>attributeGroups defined</xs:documentation>
</xs:annotation>
<xs:attributeGroup name="MessageAttributes">
  <xs:annotation>
    <xs:documentation>attributes of each iDEAL message</xs:documentation>
  </xs:annotation>
  <xs:attribute name="version" type="iDEAL.version" use="required"/>
</xs:attributeGroup>
</xs:schema>

```