Card Online Processing
Redirect Payment Extension

Merchant Specification

Version: 1.0.2

This document has been created by the Wirecard AG. Its contents may be changed without prior notice. External web links are provided for information only. Wirecard does not claim liability for access to and correctness of the referenced content.

COPYRIGHT

Printed in Germany / European Union

Version: 1.0.2

Last Updated: 6/21/2012

TRADEMARKS

CONTACT INFORMATION

For questions relating to this document please contact:

Wirecard AG

Einsteinring 35

D-85609 Aschheim

Germany

phone: +49 89 4424 1640

email: support@wirecard.com

Contents

1 Introduction

This specification is an extension to the Wirecard Card Online Processing Merchant Specification. It describes additional function calls necessary for specific card schemes and/or acquirers.

1.1 Audience

This specification is intended to be read by the technical staff in the merchant's organization responsible for implementing and administering the XML-based card processing interface. It is assumed that the reader has a working knowledge of the programming languages discussed in this specification and is familiar with the Wirecard Card Processing Interface

1.2 Document Conventions

This document uses the following conventions:

- `Monospace/Courier` is used for example code and code listings, file names, commands, path names, directory names, Hypertext Markup Language (HTML) tags, and any text that must be typed on the screen.
- The *italic* font is used in code to represent placeholder parameters (variables) that should be replaced with an actual value, or items that require emphasis.
- Brackets ([]) are used to enclose optional parameters.
- A slash (/) is used to separate directories in a path and to indicate a blank or closing XML parameter

1.3 Software Requirements

To implement the XML interface for standard card processing, the following requirements must be met:

- Internet connection supporting SFTP
- Working knowledge of XML
- SSL server supporting 128-bit (or stronger) encryption
- Support of UTF-8 encoding

1.4 References

- Wirecard Card Online Processing Merchant Specification
- Wirecard 3d Secure Card Online Processing Merchant Specification

1.5 Revision History

This specification is periodically updated to reflect the modifications made to the card-processing interface. With each revision, a new entry is added to the table below, including the date of and the reason for the version change. Additionally, vertical revision bars are placed in the margins to indicate the changes in the text.

Date	Version	Comments
2011-11-23	0.9.0	First public version
2012-03-12	1.0.0	Adjustments for Union Pay Processing
2012-06-21	1.0.2	Small Corrections

2 Overview

The standard Wirecard Card Online Specification describes a backend communication, completely transparent to the merchant. The merchant collects the card holder details from the card holder and provides them to Wirecard via the HTTS/XML server-to-server interface. The card holder does not get into direct contact with either Wirecard, Schemes or Issuer.

Some payment provider / schemes needs additionally direct interaction between the card holder and the card scheme or issuer. In these cases, the card holder's browser has to be redirected to a webpage provided by the scheme or issuer. After the interaction, the browser is redirected back to the merchant's webpage to finalize the checkout.

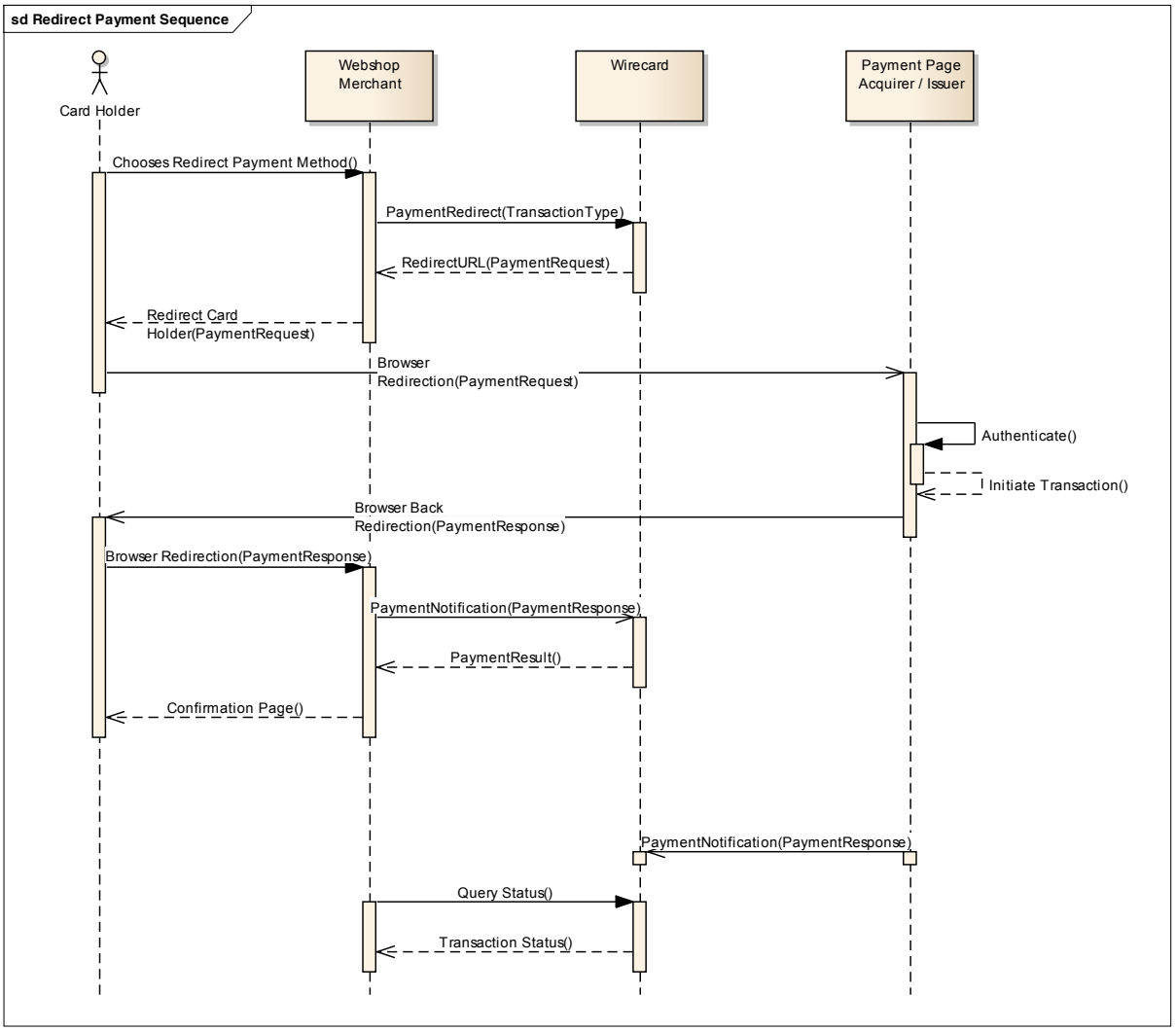
Two different scenarios are possible.

1) Only an additional authentication takes place prior the redirect.
   An example of this scenario is 3d Secure. Please see Wirecard 3d Secure Card Online Processing Merchant Specification for a description.
2) The Preauthorization/Transaction itself is handled during the redirect.
   This scenario is described in this specification.

2.1 Process Sequence

The normal redirect payment flow consists of following steps:

1) The card holder chooses a redirect payment method like Union Pay
2) The merchant initiates the transaction at Wirecard
3) The merchant forwards the card holder to the payment page
4) The payment page authenticates the card holder and triggers the payment
5) The card holder is redirected back to the merchant
6) The merchant sends the returned data to Wirecard to verify the payment
7) Wirecard verifies the payment and sends a confirmation back to the merchant
8) The merchant displays a confirmation page to the card holder



Also this is the normal process it is possible that the card holder is not redirected back to the merchant. This can be caused by different reasons. He can abandon the payment, finalize the payment but close the browser page or is just not aware that there is a “return to shop” button. Therefore it is very important that the merchant covers this case correctly. After a specific time a Query should be sent to Wirecard to retrieve the transaction status.

In case the transaction is still pending, the Query should be retried after a specified time.

In case the transaction failed, a message could be send to the card holder to inform him about the failed transaction and give him the option to retry it or choose another payment method.

In case the transaction was successful and the order can be still fulfilled the confirmation should also be send to the card holder.

In case the transaction was successful but the order cannot be fulfilled anymore, the card holder should be notified and a reversal should be send to Wirecard.

3 Function Types

In the standard online processing flow described in the Wirecard Card Online Processing Specification the merchant collects the card details like the card number directly from the card holder and submits them to Wirecard. For Redirected Payments the card holder is redirected to the payment page of the card scheme and the card holder confirms the Preauthorization or Purchase there.

Wirecard's Redirected Payment Solution consists of three steps:

- ☐ Payment Redirect – Initiates a payment

  The merchant sends a Payment Redirect to the Wirecard platform. The Payment Redirect already specifies the type of transaction (Preauthorization or Purchase) and the processor. Wirecard responds with the RedirectURL and an encrypted message (PayReq) which includes some transaction data.

- ☐ Forward Cardholder to the Payment Page – Card holder enters his details and confirms transaction

  The cardholder is redirected to the PaymentURL. He enters his details and has to pass different authentication checks like SMS TAN. He confirms the payment and is redirected to the Merchant's NotificationURL.

- ☐ Payment Notification – Validation of the result

  The merchant sends a Payment Notification to the Wirecard platform, including the encrypted result, which had been passed to the NotificationURL. Wirecard responds with the decoded result.

Following types are additionally available – please refer to the Wirecard Card Online Processing Specification for a detailed description:

Debit Transaction Types

Capture – Advice to capture the previous authorized amount.

Credit Transaction Types

Bookback – Credit a previously debited amount back to the card holder (limits apply)

Additional Transaction Type

Query – Retrieve the status of a previous transaction

Reversal – Cancel a previous Debit

3.1 Payment Redirect

Description:

With the Payment Redirect the merchant initiates a transaction at the Wirecard platform. The Merchant provides basic transaction data, which will be used later for redirecting the card holder back to the merchant.

Wirecard's response contains the RedirectURL and an encoded PaymentRequest.

Availability and Restrictions:

This function is only available for Union Pay transactions.

Examples:

Please refer to Payment Redirect Example for examples.

Request message:

The following table lists the most common used fields for submitting a PaymentRedirect to Wirecard.

Element	Sett.	Data Type	Description
FNC_CC_PAYMENT_REDIRECT	man	c	This is a collection of transaction data elements and their values.
FunctionID	opt	an..32	This is the merchant system data used for tracking purposes.
CC_TRANSACTION	man	c	This is a collection of transaction data elements and their values
TransactionID	man	an..32	This unique ID is associated with a single transaction, which is created by the merchant and submitted as part of the request.
PaymentType	man	an..32	This element defines the card scheme. Supported Values: - UPOP (Union Pay)
TransactionType	man	an..32	This element defines the subsequent transaction flow. It is only supported for UPOP transactions and ignored otherwise. Supported values: - Preauthorization - Purchase
Amount	man	n..16	This is the integer amount, in smallest currency unit, for which the transaction is requested (e.g., $10.00 would be "1000").
Minorunits	opt	n..1	The attribute <minorunits> specifies the number of decimal places of the amount.

Element	Sett.	Data Type	Description
action		a..8	The attribute <action> defines what action needs to be taken if the value of the attribute <minorunits> does not match the number of decimals defined in ISO standard 4217. The following actions are available: • convert the amount is converted to the number of ISO-defined decimals • validate The transaction is rejected if the value of <minorunits> does not match the number of decimals defined in ISO 4217. This is the default setting.
Currency	man	a 3	This is the ISO 4217 currency code used for the transaction.
Usage	opt	an..256	This is the field, which is shown on the card holder's card statement and can be used by the merchant for reference purposes. This feature is not supported by all the acquirers. The size of this field depends on the acquirer. Please contact Wirecard technical support for further clarification
CONTACT_DATA	opt.	c	This is the collection of the contact information
IPAddress	opt.	an..256	This is the IP address of the end user making the purchase
CORPTRUSTCENTER_DATA	con.	c	This is a collection of additional data fields which may be required for risk management or fraud prevention solutions. Please see Corperate Trustcenter Data in the Wirecard card processing specification for a list of possible elements.
(additional elements)			

Response message:

Element	Sett.	Data Type	Description
FNC_CC_PAYMENT_REDIRECT	man	c	This is a collection of transaction data elements and their values.
FunctionID	opt	an..32	This ID is received as part of the request and echoed back with the response. This is merchant system data used for tracking purposes.
CC_TRANSACTION	man	c	This is a collection of transaction data elements and their values.
TransactionID	man	an..32	This unique ID is associated with a single transaction, which is created by the merchant and submitted as part of the request. This is received as part of the request and echoed back with the response.
PROCESSING_STATUS	man	c	This is a collection of transaction result elements and values.
GuWID	man	an 22	This is the *Global unique Wirecard ID*. It is used for tracking transactions and is required when reporting a problem with a transaction to Wirecard Technical Support (support@wirecard.com).
AuthorizationCode	man.	an..10	This element should be ignored.
StatusType	man	an..32	This element should be ignored.
FunctionResult	man	a 3	This function field reveals if the request succeeded or failed. Valid values are: • ACK (Successful transaction) • NOK (Failed transaction)
ERROR	con	c	This is a collection of error result elements and values. This collection is provided only if the FunctionResult is NOK. This collection can be repeated if there are multiple errors. This collection may also be placed in a higher XML level if the error is more general.
Type	man	an..32	Provides basic information about the type of error. It may have one of the following values: • REJECTED- transaction was rejected by acquirer. • DATA_ERROR – XML request data is not valid and could not be processed • SYSTEM_ERROR - transaction could not be processed because of a system error. • CLIENT_ERROR - error on the client side. This value is returned only if the merchant uses Wirecard's locally installed XML client server software.

Element	Sett.	Data Type	Description
Number	man	n..5	This is the error number associated with the failure.
Message	man	an..1024	This is the error message associated with the failed condition.
Advice	opt.	an..1024	This is the system-generated guidance for correction of the failed condition. This element can be repeated if there is a need for multiple advises.
TimeStamp	man	YYYY-MMDD hh:mm:ss	This is CET (Central European Time) date/time of the completion of the transaction.
PAYMENT_DATA	man.	c	This is a collection of data elements relevant for the redirection of the card holder to the payment RedirectURL
PaymentRequest	con.	an..<16000	A base64-encoded request message
RedirectUrl	con.	an..100	This is the URL to where the cardholder has to be redirected by the merchant. It is only returned if the Payment Redirect Request was processed succesfully.

3.2 Card Holder Redirection

Description:

In addition to the interface setup between Merchant and Wirecard, the successful Payment Redirect implementation requires interaction between merchant and card scheme / issuing bank via the cardholder browser. The cardholder has to be redirected to the URL received by the successful Payment Redirect response. It is imperative to also pass the Payment Request as this contains the Payment relevant data.

The card holder enters card details on the PaymentPage and confirms the payment. With submitting the confirmation the transaction (Preauthorization or Purchase) is immediately carried out and the card holder is redirected back to the Merchant's TermUrl including the transaction result encoded in the PayRes.

Availability and Restrictions:

This function is only available for Union Pay transactions.

Examples:

Please refer to Card Holder Redirection for examples.

Card Holder HTTPS Redirect to PaymentPage

The cardholder has to be redirected to the Url received by the successful Payment Redirect response. It is imperative to also pass the Payment Request as this contains the Payment relevant data.

This HTTPS POST message includes the redirection web address (URL) and three hidden input types: <PayReq>,<TermUrl> and <MD>. The TermUrl defines the web address of the merchant site to which the card scheme or issuer returns forwards the card holder after finalizing the payment process. The parameter type <MD> is reserved for merchant specific data. Although this field is mandatory, it does not need to have a value defined. If this input type is omitted, an authentication error will occur and the payment process is aborted. The MD may be useful for retrieving transaction data from the database or recalling a transaction. The data is returned untouched by the card scheme / issuer to the merchant's TermUrl.

Element	Sett.	Data Type	Description
PayReq	man	an..<16000	A base64-encoded request message containing the result of the payment
TermUrl	man	HTTPS URL	When the card holder leaves the ACS page he is redirected to this page. It has to be a HTTPS URL.
MD	man	Not specified	This field must be provided but can be empty. It is reserved for special purposes.

Card Holder HTTPS Redirect back to Merchant

After finishing, the payment page provided by the scheme / issuer the cardholder is redirected back to the TermUrl previously specified by the merchant.

The response message contains the results of the cardholder's authentication and the untouched merchant data (MD). It is important that a call of PaymentNotification Request is sent.

Element	Sett.	Data Type	Description
PayRes	man	an..<16000	This is the digitally signed, base64-encoded authentication response message received from the issuer.
MD	man	Not specified	This field returns the MD value.

3.3 Payment Notification

Description:

The PayRes received during the redirection of the card holder back to the merchant's URL contains the result of the transaction. At this time the transaction Preauthorization or Purchase has already taken place. The PayRes has to be sent to Wirecard to be encrypted.

The PaymentNotification updates only the transaction created by the previous Payment Redirect. This means also that no new GuWID is created but the GuWID of the previous PaymentRedirect is returned.

Availability and Restrictions:

This transaction type is only available for Union Pay transactions.

Examples:

Please refer to Payment Notification Example for examples.

Request message:

The following table lists the most common used fields for submitting a PaymentNotification Request to Wirecard.

Element	Sett.	Data Type	Description
FNC_CC_PAYMENT_ NOTIFICATION	man	c	This is a collection of transaction data elements and their values
FunctionID	opt	an..32	This is the merchant system data used for tracking purposes
CC_TRANSACTION	man	c	This is a collection of transaction data elements and their values.
TransactionID	man	an..32	This is a unique ID associated with a single transaction, which is created by the merchant and submitted as part of the request.
GuWID	man	an..22	This is the Global unique Wirecard ID, an alphanumeric string returned by the Wirecard system with the Payment Redirect Response. It is used for referencing previous transactions and is required when reporting a problem to Wirecard Technical Support (support@wirecard.com).
PAYMENT_DATA	man.	c	This is a collection of data elements relevant for the redirection of the card holder to the payment RedirectURL
PaymentResponse	man	an..<16000	A base64-encoded request message.

Response message:

The following table lists the fields provided by Wirecard to the merchant in the Payment Notification Response.

Element	Sett.	Data Type	Description
FNC_CC_PAYMENT_ NOTIFICATION	man	c	This is a collection of transaction data elements and their values
FunctionID	opt	an..32	This is the merchant system data used for tracking purposes
CC_TRANSACTION	man	c	This is a collection of transaction data elements and their values.
TransactionID	man	an..32	This is a unique ID associated with a single transaction, which is created by the merchant and submitted as part of the request.
PROCESSING_ STATUS	man	c	This is a collection of transaction result elements and values.
GuWID	man	can..22	This is the Global unique Wirecard ID, an alphanumeric string returned by the Wirecard system with the Payment Redirect Response.
Authorization Code	con	an..10	This is a numerical or alphanumerical code provided by the card issuer. For UPOP no Authorization Code is provided.
StatusType	man	an..32	This element should be ignored.
FunctionResult	opt	a 32	The data returned in this line of the response message shows the result of the executed transaction. Valid values are: • ACK (Successful transaction) • NOK (Failed transaction) • PENDING (successful transaction waiting to be captured) for further details see the definition ’pending’ in the Glossary.
ERROR	con	c	This is a collection of error result elements and values. This collection is provided only if the FunctionResult is NOK. This collection can be repeated if there are multiple errors. This collection may also be placed in a higher XML level if the error is more general.

Element	Sett.	Data Type	Description
Type	man	an..32	Provides basic information about the type of error. It may have one of the following values: • REJECTED- transaction was rejected by acquirer. • DATA_ERROR - XML request data is not valid and could not be processed. • SYSTEM_ERROR - transaction could not be processed because of a system error. • CLIENT_ERROR - error on the client side. This value is returned only if the merchant uses Wirecard's locally installed XML client server software.
Number	man	n..5	This is the error number associated with the failure.
Message	man	an..1024	This is the error message associated with the failed condition.
Advice	opt.	an..1024	This is the system-generated guidance for correction of the failed condition. This element can be repeated if there is a need for multiple advises.
TimeStamp	man	YYYY-MMDD hh:mm:ss	This is CET (Central European Time) date/time of the completion of the transaction.

3.4 Capture

Description:

After an order is shipped, a previous preauthorized amount can be settled (captured). The card-issuing bank credits the funds to the merchant's bank account and updates the cardholder's statement. Card regulations require a merchant to ship goods before settling the funds for an order.

A Capture Request must include a valid GuWID referencing the previous Payment Notification.

A Capture should be made before the preauthorization expires. The preauthorization validity duration depends on the card type and card brand and varies between 7 and 21 days for standard credit cards and standard business. Triggering the capture within 7 days is therefore a good practice.

Most acquirer and issuers also accepts captures send after the preauthorization expired. In this case the issuer has the right to raise a chargeback with reasons like "Late Presentment" or "Non-Authorized Transaction".

Availability and Restrictions:

- ☐ The PaymentRedirect was submitted with the TransactionType Preauthorization
- ☐ The amount of the Capture has to be equal or less than the preauthorized amount.

Specification

Please refer to Wirecard_Card_Online_Processing for a detailed specification of this transaction type.

3.5 Reversal

Description:

The reversal function enables the user to cancel a previous request.

A reversal of a monetary transaction (e.g. ‘Purchase’, ‘Capture’) is possible until the transaction is processed by the Acquirer. The exact deadline depends on the acquirer and the used protocol. Reversed transactions do not appear on the cardholder’s card statement.

A reversal of a preauthorization can be made up to 14 days following the original request, depending on the issuer and processor.

A reversal of a capture reactivates the original preauthorization resulting in reduced credit limit of the credit card. To free reserved amounts on a credit card please cancel also the preauthorization.

For a reversal request, a valid GuWID from a previous request is required. The amount defined in the ‘reversal’ request has to match the amount given in the respective request that needs to be cancelled.

Availability and Restrictions:

For UnionPay transactions only full amount reversals are supported.

Specification

Please refer to Wirecard_Card_Online_Processing for a detailed specification of this transaction type.

3.6 Bookback

Description:

A Bookback allows to credit the customer, e.g. in case of returned goods or cancelation. To post a Bookback request, a valid GuWID from a former Capture or Purchase (debit) transaction is required. It is only possible to credit an amount less than or equal to the initial transaction using the same currency as with the original transaction. A Bookback is listed separately on the cardholder's card statement

Availability and Restrictions:

This transaction type is available for UnionPay.

Bookbacks on UnionPay transactions can be submitted up to 180 days after the original debit transaction.

Specification

Please refer to Wirecard_Card_Online_Processing for a detailed specification of this transaction type.

3.7 Query

Description:

With the Query Request function, the merchant can obtain information about the status of a transaction.

This function is very important for Redirected Payment. Whenever a card holder was redirected to the PaymentPage but does not return to the TermUrl the status of that transaction is unknown. In most cases, the card holder would have probably canceled the payment process. However, it is also possible that the card holder confirms successfully the payment but his browser does not sucessfully performed the redirect back to the merchant (e.g. closing the browser window). In this case, the transaction was succesful. To determine the status a Query is strongly recommended.

Availability and Restrictions:

This transaction type is generally available.

Specification

Please refer to Wirecard_Card_Online_Processing for a detailed specification of this transaction type.

4 Examples

This chapter lists a number of examples of the different functions described in this document.

4.1 Payment Redirect Example

Please refer to Payment Redirect for a description and field definition..

4.1.1 Payment Redirect Request

```
<?xml version="1.0" encoding="UTF-8"?>
<WIRECARD_BXML>
   <W_REQUEST>
      <W_JOB>
         <JobID>JOB-ID-0009</JobID>
         <BusinessCaseSignature>0000000123456</BusinessCaseSignature>
         <FNC_CC_PAYMENT_REDIRECT>
            <FunctionID>PA-12102006-000909</FunctionID>
            <CC_TRANSACTION mode="live">
               <TransactionID>000909-001</TransactionID>
                           <PaymentType>UPOP</PaymentType>
               <TransactionType>Purchase</TransactionType>
               <Amount>45000</Amount>
               <Currency>HKD</Currency>
               <Usage>Item 6727</Usage>
                              <CONTACT_DATA>
                  <IPAddress>192.168.1.1</IPAddress>
               </CONTACT_DATA>
            </CC_TRANSACTION>
         </FNC_CC_PAYMENT_REDIRECT>
      </W_JOB>
   </W_REQUEST>
</WIRECARD_BXML>
```

4.1.2 Payment Redirect Response

```
<?xml version="1.0" encoding="UTF-8"?>
<WIRECARD_BXML xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="pe-card.xsd">
   <W_RESPONSE>
      <W_JOB>
         <JobID>JOB-ID-0009</JobID>
         <FNC_CC_PAYMENT_REDIRECT>
            <FunctionID>PA-12102006-000909</FunctionID>
            <CC_TRANSACTION>
               <TransactionID>061126-02</TransactionID>
               <PROCESSING_STATUS>
                  <GuWID>C862153120247034229240</GuWID>
                  <AuthorizationCode/>
                  <StatusType>C</StatusType>
                  <FunctionResult>PENDING</FunctionResult>
                  <TimeStamp>2011-06-21 11:02:02</TimeStamp>
               </PROCESSING_STATUS>
               <PAYMENT_DATA>
                  <PaymentRequest>
```

```
PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4NCjxXSVJFQ0FSRF9CWE1MIHh
tbG5zOnhzaT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS9YTUxTY2hlbWEtaW5zdGFuY2UiIHhzaT
pub05hbWVzcGFjZVNjaGVtYUxvY2F0aW9uPSJwZS1jYXJkLnhzZCI+DQogIDxXX1JFU1BPTlNFP
g0KICAgIDxXX0pPQj4NCiAgICAgIDxKb2JJRD5KT0ItSUQtMDAwOTwvSm9iSUQ+DQogICAgICA8
Rk5DX0NDX1JFRElSRUNUPg0KICAgICAgICA8RnVuY3Rpb25JRD5QQS0xMjEwMjAwNi0wMDA5MDk
8L0Z1bmN0aW9uSUQ+DQogICAgICAgIDxDQ19UUkFOU0FDVElPTj4NCiAgICAgICAgICA8VHJhbn
NhY3Rpb25JRD4wNjExMjYtMDI8L1RyYW5zYWN0aW9uSUQ+DQogICAgICAgICAgPFBBST0NFU1NJT
kdfU1RBVFVTPg0KICAgICAgICAgICAgPEd1V0lEPkM4NjIxNTMxMjAyNDcwMzQyMjkyNDA8L0d1
V0lEPg0KICAgICAgICAgICAgPEF1dGhvcml6YXRpb25Db2RlLz4NCiAgICAgICAgICAgIDxTdGF
0dXNUeXBlPklORk88L1N0YXR1c1R5cGU+DQogICAgICAgICAgICA8RnVuY3Rpb25SZXN1bHQ+UE
VORElORzwvRnVuY3Rpb25SZXN1bHQ+DQogICAgICAgICAgICA8VGltZVN0YW1wPjIwMTAtMDQtM
jEgMTE6MDI6MDI8L1RpbWVTdGFtcD4NCiAgICAgICAgICA8L1BST0NFU1NJTkdfU1RBVFVTPg0K
ICAgICAgICAgIDxSRURJUkVDVF9EQVRBPg0KCQkJICA8UGF5bWVudFVybD5odHRwczovL2tva29
zLmhlYWQuY29tL2RvPC9QYXltZW50VXJsPg0KICAJCQkgIDxNZXNzYWdlPjwvTWVzc2FnZT4NCi
AgICAgICAgICA8L1JFRElSRUNUX0RBVEE+DQogICAgICAgIDwvQ0NfVFJBTlNBQ1RJT04+DQogI
CAgICA8L0ZOQ19DQ19SRURJUkVDVD4NCiAgICA8L1dfSk9CPg0KICA8L1dfUkVTUE9OU0U+DQo8
L1dJUkVDQVJEX0JYTUw+DQo=</PaymentRequest>

<RedirectUrl>https://c3.wirecard.com/cupclub/front.do</RedirectUrl>
               </PAYMENT_DATA>
            </CC_TRANSACTION>
         </FNC_CC_PAYMENT_REDIRECT>
      </W_JOB>
   </W_RESPONSE>
</WIRECARD_BXML>
```


4.2 Card Holder Redirection

Please refer to Card Holder Redirection for a description and field definition.

4.2.1 Cardholder HTTPS Redirect to Payment Page

The following is an example of a redirect to forward the cardholder's browser to the payment page.

```
<html>
    <head>
        <meta HTTP-EQUIV="Content-Type" content="text/html; charset=UTF-8">
        <meta HTTP-EQUIV="Cache-Control" CONTENT="no cache">
        <meta HTTP-EQUIV="Pragma" CONTENT="no cache">
        <meta HTTP-EQUIV="Expires" CONTENT="0">
        </head>
    <body OnLoad="AutoSubmitForm();">
        <form name="downloadForm" action="<AcsUrl>" method="POST">
            <input type="hidden" name="PayReq" value="<PaReq>">
            <input type="hidden" name="TermUrl" value="<TermUrl>">
            <input type="hidden" name="MD" value="<optionalValue>">
            <SCRIPT LANGUAGE="Javascript"><!--function AutoSubmitForm()
        { document.downloadForm.submit();
            }//-->
            </SCRIPT>
            <input type="submit" name="continue" value="Continue"></center>
        </form>
    </body>
</html>
```

4.2.2 Cardholder HTTPS Redirect back to Merchant

The following is an example how to simulate the redirection generated by the payment page. This has not to be implemented as this redirection is automatically done by the payment page:

```
<html>
    <head>
        <meta http-equiv="Content-type" content="text/html;
              charset=usascii">
        <script language="Javascript" type="text/javascript">
        function OnLoadEvent() {document.downloadForm.submit(); }
        </script>
    </head>
    <body onload="OnLoadEvent();">
        <form name="downloadForm"
        action="http://merchant_URL/checkout_info.html"
        method="post" id="downloadForm">
        <noscript><br>
        <br>
        <center>
        <h1>Processing your 3-D Secure Transaction</h1>
        <h2>JavaScript is currently disabled or is not supported by your
        browser.<br></h2>
        <h3>Please click Submit to continue the processing of your 3D
            Securetransaction.</h3>
        <input type="submit" value="Submit">
        </center>
        </noscript>
        <input type="hidden" name="PayRes"
        value="eJxVj80OgjAQhO8+RdM7FHrgJ1lNeBDPhsCKTaBN2mL07cVQNB5nv53ZWSi
               f40AeaKzUqqBxGFGCqtWdVH1BL/UhyGgpNlDfDWJ1xnYyKOCI1jY9EtkVdO
               u0DU5G9lJdY57EUZzyNMl5TgXsjdFGgI8Xc3rIga0S8IN3ukORZ8B+agH+i
               KhNo6xh9WYcjuTVymGt4x7q2mCp0MxVqGgbP/QSYL8O+Bvb31BtNhVwL">
        <input type="hidden" name="MD" value="111">
        </form>
    </body>
</html>
```

4.3 Payment Notification

Please refer to Payment Notification for a description and field definition.

4.3.1 Payment Notification Request

This is an example of a Payment Notification send by the Merchant to Wirecard.

```
<?xml version="1.0" encoding="UTF-8"?>
<WIRECARD_BXML xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="pe-card.xsd">
   <W_REQUEST>
      <W_JOB>
         <JobID>JOB-ID-0009</JobID>
         <BusinessCaseSignature>0000000123456</BusinessCaseSignature>
         <FNC_CC_PAYMENT_NOTIFICATION>
            <FunctionID>PA-12354</FunctionID>
            <CC_TRANSACTION>
               <TransactionID>000909-001</TransactionID>
               <GuWID>C811971123814650111403</GuWID>
```

```
                <PAYMENT_DATA>
                   <PaymentResponse>
PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4NCjxXSVJFQ0FSRF9CWE1MIHh
tbG5zOnhzaT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS9YTUxTY2hlbWEtaW5zdGFuY2UiIHhzaT
pub05hbWVzcGFjZVNjaGVtYUxvY2F0aW9uPSJwZS1jYXJkLnhzZCI+DQogIDxXX1JFU1BPTlNFP
g0KICAgIDxXX0pPQj4NCiAgICAgICA8Rk5DX0ltSUQtMDAwOTwvSm9iSUQ+DQogICAgICA8
Rk5DX0NDX1JFRElSRUNUPg0KICAgICAgICA8RnVuY3Rpb25JRD5QQS0xMjEwMjAwNi0wMDA5MDk
8L0Z1bmN0aW9uSUQ+DQogICAgICAgIDxDQ19UUkFOU0FDVElPTj4NCiAgICAgICAgICA8VHJhbn
NhY3Rpb25JRD4wNjExMjYtMDI8L1RyYW5zYWN0aW9uSUQ+DQogICAgICAgICAgPFBBST0NFU1NJT
kdfU1RBVFVTPg0KICAgICAgICAgICAgPEd1V0lEPkM4NjIxNTMxMjAyNDcwMzQyMjkyNDA8L0d1
V0lEPg0KICAgICAgICAgICAgPEF1dGhvcml6YXRpb25Db2RlLz4NCiAgICAgICAgICAgIDxTdG
0dXNUeXBlPklORk88L1N0YXR1c1R5cGU+DQogICAgICAgICAgICA8RnVuY3Rpb25SZXN1bHQ+UE
VORElORzwvRnVuY3Rpb25SZXN1bHQ+DQogICAgICAgICAgICA8VGltZVN0YW1wPjIwMTAtMDQtM
jEgMTE6MDI6MDI8L1RpbWVTdGFtcD4NCiAgICAgICAgICA8L1BST0NFU1NJTkdfU1RBVFVTPg0K
ICAgICAgICAgIDxSRURJUkVDVF9EQVRBPg0KCkJICA8UGF5bWVudFVybD5odHRwczovL2tva29
zLmhlYWQuY29tL2RvPC9QYXltZW50VXJsPg0KICAJCQkgIDxNZXNzYWdlPjwvTWVzc2FnZT4NCi
AgICAgICAgICA8L1JFRElSRUNUX0RBVEE+DQogICAgICAgICA8L1RyYW5BQ1RJT04+DQogICAgI
CAgICA8L0ZOQ19DQ19SRURJUkVDVD4NCiAgICA8L1dfSk9CPg0KICA8L1dfUkVTUE9OU0U+DQo8
L1dIUkVDQVJEX0JYTUw+DQo=
                   </PaymentResponse>
                </PAYMENT_DATA>
             </CC_TRANSACTION>
          </FNC_CC_PAYMENT_NOTIFICATION>
       </W_JOB>
    </W_REQUEST>
</WIRECARD_BXML
```

4.3.2 Payment Notification Response

This is an example of a positive answer from

```
<?xml version="1.0" encoding="UTF-8"?>
<WIRECARD_BXML xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="pe-card.xsd">
   <W_RESPONSE>
      <W_JOB>
         <JobID>JOB-ID-0009</JobID>
         <FNC_CC_PAYMENT_NOTIFICATION>
            <FunctionID>PA-12102006-000909</FunctionID>
            <CC_TRANSACTION>
               <TransactionID>061126-02</TransactionID>
               <PROCESSING_STATUS>
                  <GuWID>C062153120247034229240</GuWID>
                  <AuthorizationCode></AuthorizationCode>
                  <StatusType>C</StatusType>
                  <FunctionResult>ACK</FunctionResult>
                  <TimeStamp>2010-04-21 11:02:02</TimeStamp>
               </PROCESSING_STATUS>
            </CC_TRANSACTION>
         </FNC_CC_PAYMENT_NOTIFICATION>
      </W_JOB>
   </W_RESPONSE>
</WIRECARD_BXML>
```