



Bibit Implementation Manual

Version 2.5.3 - March 2004

Table of content

0	Quick Start Guide	6
0.1	Create and send an order.....	6
0.2	Parse the Bibit reply and redirect the shopper	7
0.3	Receiving payment statuses and sending an [OK].....	8
1	About this Manual	10
1.1	Intended audience.....	10
1.2	Content overview.....	10
1.3	Feedback.....	10
2	Overview of the System.....	11
2.1	What is the Bibit Payment Service?.....	11
2.2	What does the Bibit Payment Service do?.....	11
2.2.1	Web shop (Redirect model).....	11
2.2.2	Call Center	11
2.2.2.1	Option 1 (Redirect):.....	11
2.2.2.2	Option 2 (Hosted Call Center Application):.....	12
2.2.3	Swipe Application (Point Of Sale/POS)	12
2.2.4	XML Direct.....	13
2.2.5	XML Batch.....	13
2.2.6	Schematic overview of the payment process.....	13
2.3	What it doesn't do.....	13
3	System requirements for the Merchant	15
3.1	Bibit contract.....	15
3.1.1	Required information for a contract.....	15
3.2	Standard communication protocols	15
3.3	Web server	15
3.4	XML	15
3.5	Shopping application.....	16
3.6	Back-office.....	16
4	Anatomy of an Order (Redirect model).....	17
4.1	Orders versus Payments.....	17
4.2	Creating the order in XML	17
4.2.1	Example: XML order.....	17
4.2.2	Interpreting the XML order	18
4.2.3	Character Data (CDATA).....	19
4.2.4	Payment Method Mask (payment method (pre-)selection).....	20
4.2.5	Shopper information (the Shopper element).....	20
4.3	Setting up a connection and posting the order.....	22
4.4	Bibit's XML response.....	22
4.5	Parsing (interpreting) the XML response.....	23
5	Anatomy of a Redirect Payment	25
5.1	Why redirect?	25
5.2	How to obtain the redirection location?.....	25
5.3	How to redirect?.....	26
5.4	The payment method selection screen.....	26
5.4.1	Changing the appearance of the payment selection screen.....	26
5.5	Actions of the shopper.....	27
5.5.1	Choose payment method	27
5.5.2	Payment specific actions	28
5.6	Sending feedback to the shopper.....	28
5.6.1	Result URLs	28
5.6.1.1	Success URL	28
5.6.1.2	Pending URL	28
5.6.1.3	Failure URL	29
5.6.1.4	Request variables in the result URLs.....	29
5.6.1.5	URL encoding.....	29
5.6.1.6	HTTPS proxy.....	29
5.6.2	Email notification.....	30

6	Communicating the results to the Merchant.....	32
6.1	Overview of payment statuses	32
6.1.1	SIGNED_FORM_RECEIVED.....	32
6.1.2	SENT_FOR_AUTHORISATION.....	32
6.1.3	AUTHORISED.....	32
6.1.4	CAPTURED	33
6.1.5	REFUSED.....	33
6.1.6	ERROR.....	33
6.1.7	CANCELLED.....	34
6.1.8	CHARGED_BACK.....	34
6.1.9	SETTLED.....	34
6.1.10	SENT_FOR_REFUND	34
6.1.11	REFUNDED.....	34
6.1.12	EXPIRED.....	34
6.2	When does Bibit send a status message?.....	34
6.3	Overview of communication channels	35
6.3.1	Merchant interface vs. messaging.....	35
6.3.2	Email confirmations.....	35
6.3.3	HTTP confirmations.....	36
6.3.4	Replying to the HTTP(S) confirmation with [OK].....	36
6.3.5	Example of an [OK].....	37
6.3.6	Examples of HTTP(S) confirmation messages.....	37
7	Integration with the merchant's Back-Office system	40
7.1	Back-office code.....	40
7.2	Authenticity.....	41
7.2.1	Secure server.....	41
7.2.2	Authenticating the sender (Bibit).....	41
8	Order Modification and Order Inquiry.....	42
8.1	Order Modifications	42
8.1.1	Cancel.....	42
8.1.2	Refund.....	43
8.1.3	Capture	44
8.1.4	Authorisation code.....	45
8.2	Order Inquiries	47
8.2.1	Authorised.....	47
8.2.2	Refused (referred).....	47
8.2.3	Refused (card blocked).....	48
8.2.4	Signed form received.....	48
8.2.5	Sent for refund.....	48
8.2.6	Settled.....	49
8.2.7	Charged-back.....	49
8.2.8	Cancelled.....	50
8.2.9	Could not find order.....	50
8.2.10	Could not find payment.....	50
8.2.11	Captured.....	51
9	Extended functionalities.....	52
9.1	Recurring payments (Pay as Order/Split Payments).....	52
9.2	Card deposit.....	53
10	Other Service Models.....	55
10.1	XML Direct.....	55
10.1.1	Direct vs. Redirect.....	55
10.1.2	Security.....	55
10.1.3	Anatomy of an order (XML direct).....	55
10.1.4	Payment details	57
10.1.5	Bibit's response.....	58
10.1.6	Reduced number of payment methods	58
10.1.7	Order Modifications and Order Inquiries.....	58
10.1.8	Risk Management in combination with XML Direct.....	58
10.1.9	Informing the Shopper	58
10.2	XML Batch	59
10.2.1	Batch vs. other models	59
10.2.2	Anatomy of an order batch.....	59
10.2.3	Bibit's response.....	60
10.2.4	Payment results.....	60

10.2.5	Batch inquiries	60
10.2.6	Batch Payment methods	61
10.2.7	Possible batch statuses	62
10.2.7.1	ORDERS_SAVED	62
10.2.7.2	PROCESSED	62
10.2.7.3	PROCESSED_WITH_ERROR	62
10.2.8	Order Modifications and Order Inquiries	62
10.2.9	Risk Management in combination with Batch payments	62
10.3	Hosted Call Center Application	62
10.3.1	Sending an order	62
10.3.2	Feedback to the shopper	63
10.3.3	Feedback to the merchant	63
10.3.4	Testing	63
10.3.5	Call Center Payment methods	63
10.3.6	Accepted currencies	63
10.3.7	Ordercode	63
10.3.8	Order Modifications and Order Inquiries	63
10.3.9	Risk Management in combination with the Hosted Call Center Application	64
10.4	Bibit Swipe Application (P.O.S.)	65
10.4.1	Requirements	65
10.4.2	Installation	65
10.4.3	Sending an order	65
10.4.4	Accepted payment methods	65
10.4.5	Order code	65
10.4.6	Order Modifications and Order Inquiries	65
10.4.7	Risk Management in combination with the Swipe Application	65
10.4.8	Printing the payment slip	66
11	(Corporate) Purchase Cards	67
11.1	XML orders for Purchase cards	67
11.1.1	Partial Refunds Purchase card orders	69
11.1.2	Partial Capture Purchase card orders	69
12	Testing	70
12.1	Test locations	70
12.2	Test credit card numbers	70
12.3	Testing ELV transactions	71
12.4	What to test?	71
12.5	Test scenario's	71
12.5.1	REFUSED	71
12.5.2	AUTHORISED	71
12.5.3	ERROR	71
12.5.4	REFERRED	71
12.5.5	CAPTURED	72
12.5.6	CVC2	72
12.5.7	AVS	72
12.6	Reporting	73
12.7	Testing in the production environment	74
13	Troubleshooting	75
13.1	Merchant not active	75
13.2	IP check failed	75
13.3	Access denied	76
13.4	Missing doc type declaration	76
13.5	Empty body in message	76
13.6	Missing quote characters	77
13.7	No description	77
13.8	Duplicate order	77
13.9	Maximum transaction amount exceeded	78
13.10	Unsupported currency	78
13.11	Internal Error	78
13.12	NMTOKEN	79
13.13	Service temporarily not available	79
13.14	Duplicate batch	79
13.15	Reduced number of payment methods	79

14	How to go live?	81
14.1	Bibit acceptance test procedure.....	81
14.1.1	Applying for the test.....	81
14.1.2	What does Bibit test?.....	81
14.1.3	How long will the test take?.....	81
14.1.4	When can you go live?.....	82
14.1.5	Directing payments to the production environment.....	82
15	Risk Management	83
15.1	What type of risks does a merchant run?.....	83
15.2	Risk Management Module	83
15.3	CVC2.....	83
15.3.1	Incorrect CVC2 code resulting in authorised payment.....	85
15.4	Verified by VISA (formerly 3-D Secure).....	85
15.5	AVS.....	85
15.5.1	Failed AVS check resulting in authorised payment.....	85
15.6	Risk reporting methods.....	85
16	Order Fulfilment	85
16.1	When to deliver goods or services to the customer?	86
16.2	How to cancel an order?	86
16.3	Capture and partial capture	86
16.4	How to do a refund?.....	86
17	Payments to the Merchants	87
17.1	What does Bibit do with the money?.....	87
17.2	Accepting multiple settlement currencies	87
17.3	Why doesn't this order receive a payment status?.....	87
17.4	Why did the shopper have to pay a higher amount than the merchant originally charged?.....	87
17.5	Dispute management.....	87
17.6	When to expect payments from Bibit?	88
18	Account Types and Financial Reporting	89
18.1	Account types.....	89
18.2	Financial reporting.....	91
18.2.1	Order report.....	91
18.2.2	Batch reports.....	91
18.2.3	Transfer report.....	91
18.2.4	Capture report.....	91
19	Merchant account information	92
20	Payment method codes	98
	Credit Cards.....	98
	Online debit methods	99
	Offline payment methods.....	100
21	ISO currency codes (ISO 4217)	101
22	Country and language table	103
23	Acquirer Response Codes	104
Index		106

0 Quick Start Guide

This chapter is intended as a quick reference to connect to the Bibit Payment Service. It lacks technical details and should only be used to quickly set up a working implementation of the Bibit Payment Service with the intent to realise a more integrated solution at a later date.

If you want to get a full grasp and understanding of the Bibit Payment Service and its possibilities you should skip this chapter and proceed to chapter 1.

The following sections describe how to connect to the Bibit Payment Service using the redirect service model and provide examples on how to:

- Create and send an order
- Parse the Bibit reply and redirect the shopper
- Receive payment statuses and send an [OK]

0.1 Create and send an order

Before a shopper can make a payment in the Bibit Payment Service your system will need to send an order so the Bibit Payment Service knows to expect a payment. The order has to be in XML format and looks like this:

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit/DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService version="1.4" merchantCode="TECHMAN">
  <submit>
    <order orderCode="AY845">
      <description>ACME Webshop int. Inc.</description>
      <amount value="1982" currencyCode="EUR" exponent="2" />
      <orderContent>
        <![CDATA[

<center><table>
<tr><td bgcolor="#ffff00">Your Internet Order:</td><td colspan="2" bgcolor="#ffff00"
align="right">AY 845</td></tr>
<tr><td bgcolor="#ffff00">Description:</td><td>14 Tulip bulbs</td><td
align="right">1,00</td></tr>
<tr><td colspan="2">Subtotal:</td><td align="right">14,00</td></tr>
<tr><td colspan="2">VAT: 13%</td><td align="right">1,82</td></tr>
<tr><td colspan="2">Shipping and Handling:</td><td align="right">4,00</td></tr>
<tr><td colspan="2" bgcolor="#c0c0c0">Total cost:</td><td bgcolor="#c0c0c0"
align="right">Euro 19,82</td></tr>
<tr><td colspan="3">&nbsp;</td></tr>
<tr><td bgcolor="#ffff00" colspan="3">Your billing address:</td></tr>
<tr><td colspan="3">Mr. Doe,<br>11 Hereortherestreet,<br>1234 KL
Somewhereorother,<br>Thisplace.</td></tr>
<tr><td colspan="3">&nbsp;</td></tr>
<tr><td bgcolor="#ffff00" colspan="3">Your shipping address:</td></tr>
<tr><td colspan="3">Mr. Doe,<br>11 Hereortherestreet,<br>1234 KL
Somewhereorother,<br>Thisplace.</td></tr>
<tr><td colspan="3">&nbsp;</td></tr>
<tr><td bgcolor="#ffff00" colspan="3">Our contact information:</td></tr>
<tr><td colspan="3">ACME Webshops Int. Inc.,<br>11 Strangewood Blv.,<br>1255 KZ
Thisisit,<br>Nowhereataill.<br><br>acmeweb@acme.inc<br>(555) 1235 456</td></tr>
<tr><td colspan="3">&nbsp;</td></tr>
<tr><td bgcolor="#c0c0c0" colspan="3">Billing notice:</td></tr>

]]>
      </orderContent>
    </order>
  </submit>
</paymentService>
```

```

<tr><td colspan="3">Your payment will be handled by Bibit Global Payments
Services<br>This name may appear on your bank
statement<br>http://www.bibit.com</td></tr>
</table></center>

```

```

    ]]>
  </orderContent>
  <paymentMethodMask>
    <include code="ALL"/>
  </paymentMethodMask>
  <shippingAddress>
    <address>
      <firstName>John</firstName>
      <lastName>Doe</lastName>
      <street>11 Hereortherestreet</street>
      <postalCode>1234KL</postalCode>
      <city>Somewhereorother</city>
      <countryCode>TP</countryCode>
      <telephoneNumber>0123456789</telephoneNumber>
    </address>
  </shippingAddress>
</order>
</submit>
</paymentService>

```

The XML orders should be sent to: <https://secure.bibit.com/jsp/merchant/xml/paymentService.jsp>

Once you have set up the connection to the Bibit Payment Service your system has to post the XML message. Before being able to send the message the XML password needs to be set up in the Merchant Interface. Please note that there is a time delay of about 15 minutes before password changes take effect.

Please make sure the HTTP content type is 'text/xml'. **It is important to make sure that the 'content length' attribute in the HTTP headers is specified correctly. Not specifying 'content length' will not create errors, while specifying an incorrect length will.**

0.2 Parse the Bibit reply and redirect the shopper.

After an order has been successfully stored, the Bibit Payment Service will send a reply that looks like this:

```

<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
    "http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="TECHMAN" version="1.4">
  <reply>
    <orderStatus orderCode="AY845">
      <reference id="1234567">
        https://secure.bibit.com/jsp/shopper/SelectPaymentMethod.jsp?OrderKey=
        TECHMAN^AY845
      </reference>
    </orderStatus>
  </reply>
</paymentService>

```

The URL in the `reference` element is the URL where the shopper should be redirected. You can optionally add parameters to the URL to change the appearance and functionality of the payment selection page, as explained in section 5.6.1.

After you have redirected the shopper he will select his preferred payment method and fill in the necessary details.

0.3 Receiving payment statuses and sending an [OK].

When the shopper has paid the order, Bibit Payment Service can send out a confirmation message (order notification). These can be sent through email or HTTP.

Email confirmation messages can be sent to one or more valid email addresses. Below you find an example of an email confirmation (in both text format and in XML).

Text message:

```

Subject:
Payment Result for order DEMO1021272851471
Body:
*****

OrderCode: DEMO1021272851471
PaymentId: 2619062
PaymentAmount : 75,00
PaymentCurrency: EUR
PaymentStatus : AUTHORISED
PaymentMethod : ECMC-SSL
Date : Mon May 13 08:54:39 CEST 2002
*****

```

XML message:

```

Subject:
Payment Result for order DEMO1021273208485
Body:
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService version="1.4" merchantCode="TECHMAN">
  <notify>
    <orderStatusEvent orderCode="DEMO1021273208485">
      <payment>
        <paymentMethod>ECMC-SSL</paymentMethod>
        <amount value="2000" currencyCode="EUR" exponent="2"
debitCreditIndicator="credit"/>
        <lastEvent>AUTHORISED</lastEvent>
        <balance accountType="IN_PROCESS_AUTHORISED">
          <amount value="2000" currencyCode="EUR" exponent="2"
debitCreditIndicator="credit"/>
        </balance>
        <cardNumber>4111*****1111</cardNumber>
      </payment>
      <journal journalType="AUTHORISED">
        <bookingDate>
          <date dayOfMonth="23" month="08" year="2002"/>
        </bookingDate>
        <accountTx accountType="IN_PROCESS_AUTHORISED">
          <amount value="2000" currencyCode="EUR" exponent="2"
debitCreditIndicator="credit"/>
        </accountTx>
      </journal>
    </orderStatusEvent>
  </notify>
</paymentService>

```


Contrary to email confirmations, the HTTP confirmation messages **NEED** to be confirmed with an unconditional [OK] reply. This is mandatory. If the Bibit Payment Service does not receive [OK] replies for HTTP confirmation messages you will not get access to the live environment.

Below is an example of an HTTP confirmation message for a refused Switch payment for 152,90 Pound Sterling.

```
https://www.techman.com/bibit/confirmation.asp?OrderCode=DEMO1021272851471&
PaymentId=2620362&PaymentStatus=REFUSED&PaymentAmount=15290&PaymentCurrency=GBP&PaymentMet
hod=SWITCH-SSL
```

1 About this Manual

This manual describes how to connect to the Bibit Payment System. It describes the process of realising a payment for an order from beginning to end: from the creation of that order to reporting the status to the back office of the merchant. The main focus is on the information exchange between the system of the merchant and the Bibit Payment Service and what the messages that are exchanged should look like. It will not explain how to create, send and interpret the messages as this can be accomplished in many different ways depending on the application(s) used by the merchant.

Throughout this manual the following convention is used: machine code, whether sent by the Bibit system or to be received by the Bibit system, is indicated by *Courier New* typeface. All indentation in machine code is done to increase legibility.

All external links (outside of the www.bibit.com domain) are provided for information purposes only. Bibit does not sponsor these sites in any way, nor does Bibit control any part of their content. Bibit is cannot be held responsible for any information found on these sites.

1.1 Intended audience

The intended audience for this manual is the technical staff at the merchant or the merchant's systems integrator. Even though this manual is quite technical at some points, also the non-technical staff at the merchant can read this manual to get a better understanding of the Bibit Payment Service.

1.2 Content overview

Chapter 2 gives an overview of the Bibit Payment Service and its different service models. In chapter 3, the requirements for the merchant (system) are described. Chapter 4 explains the syntax of an order and describes how to post the order to the Bibit system for the Redirect service model. Payment process details for the Redirect model are discussed in chapter 5. Chapter 6 describes the different steps involved in processing a payment, including the different ways Bibit reports the payment status to the merchant. Additional details on integration of the merchant's back-office system with the Bibit Payment service are given in chapter 7. The standard functionalities for order modification and –inquiry, and extended functionalities are covered in chapters 8 and 9. Chapter 10 focusses on the other service models and chapter 11 on the details for corporate purchase cards. Chapter 12 describes how to test the connection to the Bibit Payment Service. Error codes and troubleshooting is discussed in chapter 13. Details and prerequisites on how to go live are presented in chapter 14. Chapter 15 explains the details of Risk Management. General information on order- and payment fulfilment is presented in chapters 16 and 17. Detailed technical information on the payment information flow through the Bibit Payment Service, including the Bibit financial reports, is given in chapter 18. In Chapter 19 an overview is given of all merchant information required to set up an account. Payment method codes, ISO currency codes, country and language codes, and acquirer response codes are listed in chapters 20, 21, 22 and 23.

1.3 Feedback

In order to improve the quality of this manual we would like to invite you to provide us with constructive feedback. You can do this by sending an email to: QA@bibit.com.

2 Overview of the System

This section provides a basic overview of the Bibit Payment Service system. In later chapters, the distinct functionalities of the system will be discussed in detail.

2.1 What is the Bibit Payment Service?

Bibit Global Payment Services has developed a unified system for handling a large number of international, multi-currency payment methods suitable for merchants using the Internet, Call Centers and/or Point of Sale channels.

Communication between the system of the merchant and the Bibit Payment Service is done through predefined XML messages over the Internet using standard protocols. This unified XML messaging ensures the merchant only needs to implement *one* system for all supported international payment methods and currencies. Furthermore the usage of XML enables the merchant to use the system or platform of his choice to conduct his business, be it 'out of the box' shopping software or proprietary software, and integrate the Bibit Payment Service with their own back-office and logistical systems.

2.2 What does the Bibit Payment Service do?

The Bibit Payment Service provides an interface between the shopping basket of the merchant and the international financial institutions. The interface enables consumers/shoppers to pay for their purchases using a payment method and currency of their choice, thus providing the optimal shopping experience for the consumer/shopper.

Below you find the different models available to connect to the Bibit Payment Service. A merchant can choose from these the model that is best suited to his business model.

2.2.1 Web shop (Redirect model)

The standard process flow is as follows:

- A shopper chooses the desired goods or services in the web shop
- The merchant's system generates a so called order, containing all the data necessary for Bibit to be able to process the payment, and sends this to the Bibit Payment Service
- The merchant's system redirects the shopper to the secured Bibit environment (from this moment the Bibit system takes over)
- The shopper can choose his preferred payment method
- The shopper fills out the payment details in the online forms
- Bibit will send the payment details to the appropriate financial institution for the chosen payment method
- The financial institution reports back to the Bibit system
- The merchant's system is notified by Bibit
- The shopper is directed back to the merchant's system

See also the schematic overview in 2.2.6 (figure 1).

2.2.2 Call Center

There are two possible call center models. Their standard process flow is as follows:

2.2.2.1 Option 1 (Redirect):

- The call center operator is logged on to the merchant's website

- The shopper calls the call center
- The call center operator types in the required information into the website
- The merchant's server prepares and sends the order to the Bibit Payment Service.
- The merchant's system directs the operator to the secured Bibit environment (from this moment the Bibit system takes over)
- The operator can choose the shopper's preferred payment method
- The operator fills out the payment details in the online forms
- Bibit sends the payment details to the appropriate financial institution for the chosen payment method
- The financial institution reports back to the Bibit system
- The merchant's system is notified by Bibit
- The operator is directed back to the merchant's system for the next order

2.2.2.2 Option 2 (Hosted Call Center Application):

- The call center operator is logged on to the Hosted Call Center Application on the Bibit system
 - The shopper calls the call center
 - The operator enters the required data (order number, description, amount, currency)
 - The order is created in the Bibit Payment Service.
- (A) The operator selects the payment method and enters the relevant payment information for the selected payment method (credit card number, card holder name, bank account information etc)
- Bibit sends the information to the financial institutions for authorisation
 - Bibit presents the results to the call center operator in the Hosted Call Center Application or (in development) sends an email to the shopper with an URL where the shopper can choose payment methods not suitable for call centers such as SET payment methods.
 - The merchant's system is notified by Bibit
 - The operator is directed back to the order entry page of the Hosted Call Center Application
- (B) The operator sends an email to the shopper with a redirection URL where the shopper can select an (off-line) payment method:
- The shopper receives the email with a URL
 - The URL redirects the shopper to a page (pre-populated with order information) within the secured Bibit environment
 - The shopper selects the desired payment method
 - The shopper fills out the payment details in the online forms
 - Bibit will send the payment details to the appropriate financial institution for the chosen payment method
 - The financial institution will report back to the Bibit system
 - The merchant's system is notified by Bibit
 - The shopper is directed to a page informing him about the status of the payment.

2.2.3 Swipe Application (Point Of Sale/POS)

In a POS scenario the standard process flow is as follows:

- The merchant is logged in to the Bibit Swipe Application
- The merchant enters the order details
- The order is stored in the Bibit system
- The merchant swipes the card through the special keyboard or card reader

- Bibit will send the payment details to the appropriate financial institution for the chosen payment method
- The financial institution will report back to the Bibit system
- The merchant's system is notified by Bibit
- The merchant is informed about the payment status
- If the merchant so chooses a slip can be printed for the cardholder to sign
- The merchant is redirected to the order entry page to process the next payment

2.2.4 XML Direct

In the case of XML Direct, the shopper is not redirected to the secured Bibit environment. Instead, he will remain on the secured environment of the merchant. The XML Direct model requires vigorous safety measures on the system of the merchant. All data is gathered on the secured environment of the merchant and sent to Bibit in the shape of XML messages. These messages contain more information than the messages used in the Redirect model. More information about this model can be found in section 10.1 (Other models: XML Direct).

2.2.5 XML Batch

The XML Batch model operates in much the same way as XML Direct. The difference is that all orders and payment details are saved on the merchant's system and are only periodically sent to the Bibit Payment Service to be processed. These dispatches may be triggered at pre-specified times or after a specified number of orders have been collected on the merchant's system.

2.2.6 Schematic overview of the payment process

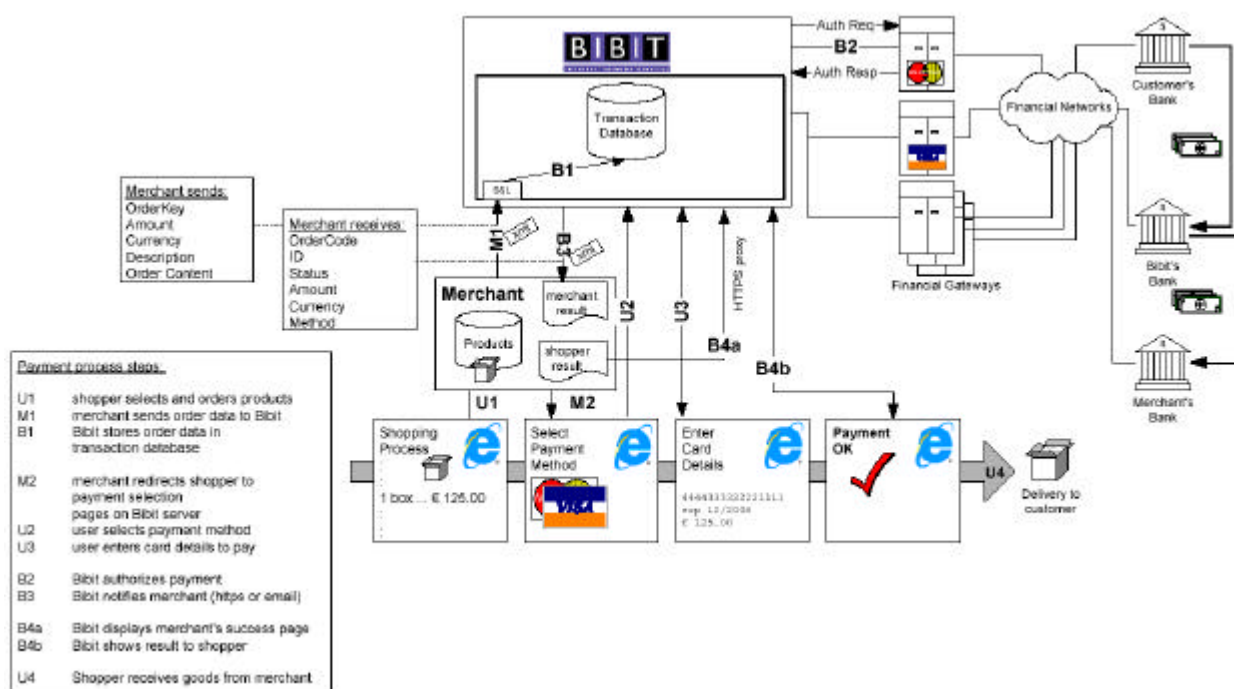


Figure 1 Standard process flow of an order resulting in a payment using the Bibit Payment Service.

2.3 What it doesn't do

It is important to realise that the Bibit Payment Service only processes *payment* information. This means that the whole process of gathering the information of the order, the information about the shopper (billing address,

shipping address) and all other information relevant to the merchant takes place on the merchant's system. Therefore, some sort of shopping application has to run on the merchant's system. This can be any type of application, running on any kind of platform, as long as it can send and interpret the XML messages specified in this manual, and set-up the required HTTP(S) connections to the Bibit Payment Service.

The Bibit Payment Service only communicates through XML messages and therefore does not accept information (orders) in a format other than specified in this manual and in the Document Type Definition (DTD) found at <http://dtd.bibit.com/>.

3 System requirements for the Merchant

This section describes the requirements for the merchant to be able to connect to the Bibit Payment Service.

3.1 Bibit contract

In order to be able to use the Bibit Payment Service you need to have a signed contract with Bibit Global Payment Services. After the contract has been signed, all relevant technical and administrative information will be entered in the Bibit system and an account will be created for you. There will be two separate environments available to merchants: a test environment to test the integration between your system(s) and the Bibit Payment Service, and a production (live) environment to use for the actual payments. Both environments will be available during the entire contract period. Please note that you can only use the production environment after an acceptance test by Bibit.

3.1.1 Required information for a contract

The Bibit Payment Service needs a certain amount of information about your system before you can actually use the service. This information includes the IP addresses of your system, your preferred channel to use for Bibit Payment Service's feedback, the preferred format of this feedback, bank details for payout, financial and technical contact details, etc. All this information needs to be filled out in the 'online sign-up form' that can be found on Bibit's website: <http://www.bibit.com>. Section 20 contains a full list of all the information required to set up a merchant account.

3.2 Standard communication protocols

In order to provide an accessible modular, platform independent service, the Bibit Payment Service only makes use of standardised Internet communication protocols such as HTTP, HTTPS and XML. These protocols are available for all platforms.

The website of the World Wide Web Consortium is a good reference point for more information about standard Internet protocols and of protocols in the process of being standardised (<http://www.w3.org>).

3.3 Web server

You need to have access to a web server that hosts the actual web shop. In the case of Point Of Sale or Hosted Call Center Application, this is not a requirement. The web server running the shopping application (or rather the server that will process the messages between your system and Bibit's system) has to have a fixed IP address (for authentication purposes), has to be able to set up and accept connections with other servers and has to be able to compose, send, receive and parse XML messages. Please check this with your System Administrator or your hosting provider.

3.4 XML

XML is an abbreviation for eXtensible Mark-up Language and is a platform independent, clearly defined format to structure, describe and interchange data. XML is a subset of the Standard Generalised Mark-up Language (SGML). XML is used to describe data within a message, explaining to the machine(s) what the data contained

within the message actually represents. XML is 'extensible' because the mark-up symbols are unlimited and self-defining.

Bibit uses a Document Type Definition (DTD) to make sure that the XML exchanged between Bibit and the merchant is interpreted correctly and built using the correct elements. In effect, the DTD checks each message's validity. All elements and attributes that are not in the DTD will be ignored or produce errors. The Bibit DTD can be found at <http://dtd.bibit.com/> and should be referred to as an external DTD in each XML message as shown in the XML examples provided later on in this document.

More information can be found at the W3 Schools website <http://www.w3schools.com>.

3.5 Shopping application

As described in an earlier paragraph your system is responsible for gathering all the necessary information about the order and the shopper. You will need some sort of software running on your system that will act as the actual shop. In this shop, the shopper will be able to select the goods or services required and put them in a shopping basket. After indicating to the system that he wants to pay for the order, your system will have to prepare the appropriate XML message and send it to the Bibit Payment Service.

Bibit doesn't have any preference to a specific shopping application. It can be any application, 'out of the box' or custom built. The only requirements are that the output of the system complies with the XML specifications as laid out in this document and the DTD and is able to receive and interpret Bibit's XML output.

You decide what kind of integration is best for your situation; Call Centers or Point of Sale situations have different requirements for the merchant's system.

3.6 Back-office

Most merchants want to keep track of their orders using a back-office and/or logistic system. Bibit reports the results of the payments through two channels, email and/or HTTP(S), in a few different formats (HTML, XML, HTML, with or without back-office code, comma separated). Note that not all reporting is available in all formats. The reports are sent in a standardised way so that your system can be programmed to interpret the incoming reports (messages) and up-date the data in the back office and/or logistical system(s) accordingly.

This automated standardised reporting means that the payment system can be fully integrated into your back-office system(s).

Additionally, you will have access to the so-called Merchant Interface, which can be found at <http://www.bibit.com/merchant>. At all times, the Merchant Interface provides near real-time information about all orders supplied to Bibit, all payments processed by Bibit and all payments to the merchant from Bibit. The Merchant Interface also provides access to special functionalities such as: (partial) refunds, (partial) manual capture of payments and the information and settings of the Risk Management Module.

4 Anatomy of an Order (Redirect model)

This chapter describes Bibit's Redirect payment service model. Most of the information is also valid for all other service models. Therefore, all merchants wishing to connect to the Bibit Payment Service are strongly recommended to read this chapter. Details for other service models will be described later on in the manual.

4.1 Orders versus Payments

It is important to understand the difference between an *order* and a *payment*. Before a shopper can do an actual payment, Bibit Payment Service will have to know the details of the order so that they can be presented to the shopper. These details are called an order. An order consists of the following elements: *description*, *amount*, *orderContent*, *paymentMethodMask* and *shopper*.

Once an actual attempt to pay an order delivered to the Bibit Payment Service has been made by the shopper and a response from the financial institutions has been processed, an order will receive a payment status and a payment id. This is what Bibit refers to as payment; an order with a payment status, be it authorised, refused or any having any other payment status.

If a shopper doesn't complete the payment process, the order will not receive a payment status. These orders will, however, be visible in the Merchant Interface. Your back-office system(s) should interpret these orders, which have not received a payment status, correctly.

Orders without payment status do not expire in the Bibit system. Nor can they be removed from the database on your request.

4.2 Creating the order in XML

After your system has gathered all the order information from the shopper, it has to compose an order in XML. There are many different software environments available and even more ways of creating XML from the data in your system. For this reason this manual can not go into details on to how to create the actual XML order; this manual will only show an example of what a correctly formatted XML order should look like.

There is a lot of free information on XML available on the Internet and of course, excellent books are available¹.

4.2.1 Example: XML order

This example shows an order for 14 tulip bulbs, at Euro 1 a piece, all payment methods are allowed. The merchant is TECHMAN and the orderCode is AY845.

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit/DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService version="1.4" merchantCode="TECHMAN">
  <submit>
    <order orderCode="AY845">
      <description>ACME Webshop int. Inc.</description>
      <amount value="1982" currencyCode="EUR" exponent="2"/>
      <orderContent>
        <![CDATA[
```

¹ Some online resources/books:

<http://www.4guysfromrolla.com/webtech/xml.shtml>

<http://www.xml101.com/>

or the "XML Bible" by John Wiley & Sons; ISBN: 0764547607; 2nd edition (June 2001)

```

<center><table>
<tr><td bgcolor="#ffff00">Your Internet Order:</td><td colspan="2"
bgcolor="#ffff00" align="right">AY 845</td></tr>
<tr><td bgcolor="#ffff00">Description:</td><td>14 Tulip bulbs</td><td
align="right">1,00</td></tr>
<tr><td colspan="2">Subtotal:</td><td align="right">14,00</td></tr>
<tr><td colspan="2">VAT: 13%</td><td align="right">1,82</td></tr>
<tr><td colspan="2">Shipping and Handling:</td><td align="right">4,00</td></tr>
<tr><td colspan="2" bgcolor="#c0c0c0">Total cost:</td><td bgcolor="#c0c0c0"
align="right">Euro 19,82</td></tr>
<tr><td colspan="3">&nbsp;</td></tr>
<tr><td bgcolor="#ffff00" colspan="3">Your billing address:</td></tr>
<tr><td colspan="3">Mr. Doe,<br>11 Hereortherestreet,<br>1234 KL
Somewhereorother,<br>Thisplace.</td></tr>
<tr><td colspan="3">&nbsp;</td></tr>
<tr><td bgcolor="#ffff00" colspan="3">Your shipping address:</td></tr>
<tr><td colspan="3">Mr. Doe,<br>11 Hereortherestreet,<br>1234 KL
Somewhereorother,<br>Thisplace.</td></tr>
<tr><td colspan="3">&nbsp;</td></tr>
<tr><td bgcolor="#ffff00" colspan="3">Our contact information:</td></tr>
<tr><td colspan="3">ACME Webshops Int. Inc.,<br>11 Strangewood Blv.,<br>1255 KZ
Thisisit,<br>Nowhereatall.<br><br>acmeweb@acme.inc<br>(555) 1235 456</td></tr>
<tr><td colspan="3">&nbsp;</td></tr>
<tr><td bgcolor="#c0c0c0" colspan="3">Billing notice:</td></tr>
<tr><td colspan="3">Your payment will be handled by Bibit Global Payments
Services<br>This name may appear on your bank
statement<br>http://www.bibit.com</td></tr>
</table></center>
    
```

```

    ]]>
</orderContent>
<paymentMethodMask>
  <include code="ALL"/>
</paymentMethodMask>
<shippingAddress>
  <address>
    <firstName>John</firstName>
    <lastName>Doe</lastName>
    <street>11 Hereortherestreet</street>
    <postalCode>1234KL</postalCode>
    <city>Somewhereorother</city>
    <countryCode>TP</countryCode>
    <telephoneNumber>0123456789</telephoneNumber>
  </address>
</shippingAddress>
</order>
</submit>
</paymentService>
    
```

4.2.2 Interpreting the XML order

In the example in the previous section the first two lines inform the receiving machine that it is looking at XML (and which version), and which DTD to use to validate the content of the message.

The third line identifies the merchant to the Bibit system through the `merchantCode` attribute; in this case, the merchant code is TECHMAN. This merchant code is issued by Bibit and is always in capitals.

Next, the message tells Bibit it wants to submit something, and specifies this as an order with order code "AY845". Order codes can be up to 64 characters long, no spaces or special characters are allowed and they have to be unique per merchant. Orders submitted to the Bibit Payment Service with an order code that has already been used by the merchant cannot be processed.

The description is a simple one-line description of the order. The description can be up to 50 characters long.

The element `amount` specifies the value (no decimal point or comma), the currency (in capital letters, ISO4217 code) and the exponent (specifies where the decimal point or comma should be placed, counting from the right). The amount should be the total amount the shopper is expected to pay. The currency codes and their respective exponents can be found in the section "ISO currency codes (ISO 4217)" at the end of this manual.

The order content as specified in the `orderContent` element is what the shopper will see when he is redirected to the payment selection screen, and what can be seen in the Merchant Interface. This information can be in HTML format. Note that only HTML tags that are allowed between the `<body>` and `</body>` tags of HTML documents can be used. No form of scripting is allowed in the order content. The size of `orderContent` must be less than 10 kilobytes and should always be included in a CDATA block.

Please note that you should NOT use `<html><body></body></html>` tags in the `orderContent`!

The order content should reflect a traditional invoice and MUST contain the following elements:

- order code
- product(s) and/or service(s) ordered
- item price
- total amount
- shopper billing address
- shopper shipping address
- contact details merchant
- the fact that Bibit processes the payment
- the fact that the name Bibit may appear on the shopper's bank or credit card statements

Failing to provide this information in the order content will result in Bibit denying you access to the live environment. Any other additional information is optional.

In the `paymentMethodMask`, all payment methods available to the merchant are 'activated' for this order.

Bibit does not require merchants to deliver the card address and shipping address in this format when using the redirect method, however it can be useful to provide this information in machine-readable format. Card address is only mandatory in the case of credit card payments when a merchant uses the XML direct model to connect to the Bibit Payment Service.

The order as shown above is an example of a basic order. What the `orderContent` in this order looks like in a browser is shown in figure 2.

4.2.3 Character Data (CDATA)

Elements that contain user input, such as cardholder name and address, can be placed in a CDATA block to allow for special characters in the input. In a CDATA block (`<![CDATA[content here]]>`), cardholder names such as "O'Hara" or "O'Reilly & Sons" containing 'special characters' will not be a problem. The CDATA block tells the XML parser that the data within that block does not have to be parsed. `orderContent` should always use a CDATA block to prevent any parsing problems.

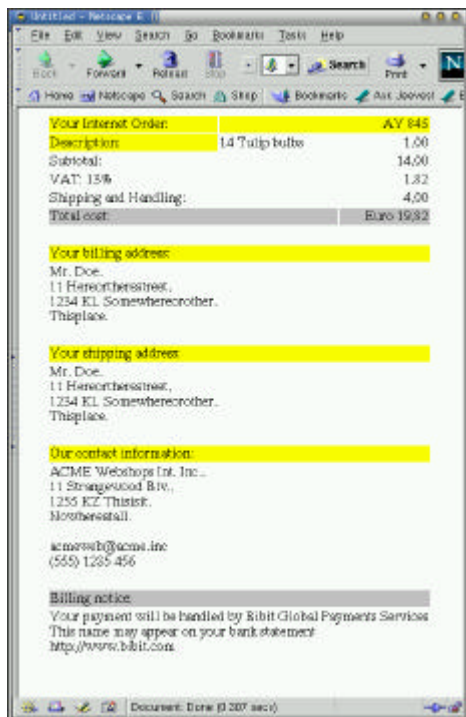


Figure 2 CDATA in the orderContent from the above example looks like this in a web browser.

4.2.4 Payment Method Mask (payment method (pre-)selection)

If you want to influence the number and type of payment methods shown to the shopper, you can use the `paymentMethodMask` element. The `paymentMethodMask` can be different per order.

The `paymentMethodMask` is used to inform the system which payment methods should be shown to the shopper for this order. Using `<include code="ALL"/>` element will simply show all available payment methods for the default or selected country. Individual payment methods can be included or excluded using the `<include code="XXX"/>` or the `<exclude code="XXX"/>` elements respectively for each payment method you want to include or exclude. The payment method codes can be found in section 21 of this manual.

The `paymentMethodMask` element has to contain at least one `<include code="XXX"/>` element. When a specific payment method has been included, it automatically excludes all other payment methods, except when `<include code="ALL"/>` is used - it includes all available payment methods for your account.

Example of the `paymentMethodMask`:

```
<paymentMethodMask>
  <include code="VISA-SSL"/>
  <include code="AMEX-SSL"/>
  <include code="ECMC-SSL"/>
</paymentMethodMask>
```

In this example only VISA, AMEX and MasterCard SSL payments will be available for the shopper to choose from. All payment methods that don't fall into the inclusion will automatically be excluded.

4.2.5 Shopper information (the Shopper element)

You can provide extra information on the shopper. This is done using the `shopper` element:

```

<shopper>
  <shopperEmailAddress>me@techman.com</shopperEmailAddress>
  <authenticatedShopperID>1234567890</authenticatedShopperID>
</shopper>

```

This shopper element and its 'children' can be placed within the submit element of the order. Providing the shopper email address in this format can be used by the system to identify known fraudsters or possibly fraudulent transactions, or to send email to the shopper when the payment is authorised or refused. The authenticated shopper ID can be used to help Bibit keep track of this particular shopper for fraud assessment purposes or to identify the shopper when using recurring payments (as described later on in this manual).

Combining these three examples into one complete order would result in the following XML order to be sent to Bibit:

```

<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit/DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService version="1.4" merchantCode="TECHMAN">
  <submit>
    <order orderCode="AY845">
      <description>ACME Webshop int. Inc.</description>
      <amount value="1982" currencyCode="EUR" exponent="2"/>
      <orderContent>
        <![CDATA [
          <center><table>
            <tr><td bgcolor="#ffff00">Your Internet Order:</td><td colspan="2"
            bgcolor="#ffff00" align="right">AY 845</td></tr>
            <tr><td bgcolor="#ffff00">Description:</td><td>14 Tulip bulbs</td><td
            align="right">1,00</td></tr>
            <tr><td colspan="2">Subtotal:</td><td align="right">14,00</td></tr>
            <tr><td colspan="2">VAT: 13%</td><td align="right">1,82</td></tr>
            <tr><td colspan="2">Shipping and Handling:</td><td align="right">4,00</td></tr>
            <tr><td colspan="2" bgcolor="#c0c0c0">Total cost:</td><td bgcolor="#c0c0c0"
            align="right">Euro 19,82</td></tr>
            <tr><td colspan="3">&nbsp;</td></tr>
            <tr><td bgcolor="#ffff00" colspan="3">Your billing address:</td></tr>
            <tr><td colspan="3">Mr. Doe,<br>11 Hereotherestreet,<br>1234 KL
            Somewhereorother,<br>Thisplace.</td></tr>
            <tr><td colspan="3">&nbsp;</td></tr>
            <tr><td bgcolor="#ffff00" colspan="3">Your shipping address:</td></tr>
            <tr><td colspan="3">Mr. Doe,<br>11 Hereotherestreet,<br>1234 KL
            Somewhereorother,<br>Thisplace.</td></tr>
            <tr><td colspan="3">&nbsp;</td></tr>
            <tr><td bgcolor="#ffff00" colspan="3">Our contact information:</td></tr>
            <tr><td colspan="3">ACME Webshops Int. Inc.,<br>11 Strangewood Blv.,<br>1255 KZ
            Thisisit,<br>Nowhereatall.<br><br>acmeweb@acme.inc<br>(555) 1235 456</td></tr>
            <tr><td colspan="3">&nbsp;</td></tr>
            <tr><td bgcolor="#c0c0c0" colspan="3">Billing notice:</td></tr>
            <tr><td colspan="3">Your payment will be handled by Bibit Global Payments
            Services<br>This name may appear on your bank
            statement<br>http://www.bibit.com</td></tr>
          </table></center>
        ]]>
      </orderContent>
      <paymentMethodMask>
        <include code="VISA-SSL"/>
        <include code="AMEX-SSL"/>
        <include code="ECMC-SSL"/>
      </paymentMethodMask>
    </submit>
  </order>
</paymentService>

```

```

<shopper>
  <shopperEmailAddress>me@techman.com</shopperEmailAddress>
  <authenticatedShopperID>1234567890</authenticatedShopperID>
</shopper>
<shippingAddress>
  <address>
    <firstName>John</firstName>
    <lastName>Doe</lastName>
    <street>11 Hereortherestreet</street>
    <postalCode>1234KL</postalCode>
    <city>Somewhereorother</city>
    <countryCode>TP</countryCode>
    <telephoneNumber>0123456789</telephoneNumber>
  </address>
</shippingAddress>
</order>
</submit>
</paymentService>

```

4.3 Setting up a connection and posting the order

When your system has created the order in XML format, it should set up a connection to the Bibit Payment Service to submit this order. The order has to be sent to the Bibit Payment Service. Each platform has its specific way of setting up a connection with another server. You have to be sure that you are using your merchant code (always in capitals) as the login and your XML password as the password when setting up the connection. The XML password has to be set up by you in the Merchant Interface (<http://www.bibit.com/merchant> or for the test environment <http://www.bibit.com/merchant-test>) in the profile page by clicking on the yellow pencil next to 'xml password'. The password should contain at least eight characters and have three numbers in the first eight characters. The password is case sensitive. Please note that there is a time delay of about 15 minutes before the password changes take effect.

Once you have set up the connection to the Bibit Payment Service, your system has to post the XML message.

Please make sure the HTTP content type is "text/xml"! **It is important to check whether 'content length' is specified correctly. Not specifying 'content length' will not create errors, while specifying it incorrectly will.**

The URLs to post orders to are:

- Test environment: <https://secure-test.bibit.com/jsp/merchant/xml/paymentService.jsp>
- Production or live environment: <https://secure.bibit.com/jsp/merchant/xml/paymentService.jsp>

Before Bibit Payment Service accepts the message, the originating IP address has to be registered in your account information in Bibit's systems. You can specify two IP addresses, one for the test environment and one for the live or production environment. The Bibit Payment Service checks on IP addresses within the C-range (last octet).

Sometimes a router or a firewall can mask the sending machine's IP address and replace it by another IP address used for all outgoing IP traffic from your network. It is important that the IP address sent out by your network for the machines used to send the orders to the Bibit Payment Service is registered with Bibit.

4.4 Bibit's XML response

After the Bibit Payment Service receives an XML order, it will check the originating IP address with the IP address we have registered for your account in the Bibit system. Please note that the servers connecting to the test and

the live environment can each have a different IP address. Each environment, however, can only have an IP C-range specified (totalling 255 IP addresses).

If the originating IP address does not match the IP address stored in our system the order will not be accepted and an error message in XML format will be returned. (IP check failed). You should keep Bibit informed about changes in the IP addresses. This can be done by simply creating an incident in the incident console at <http://support.bibit.com>.

Once the Bibit Payment Service has received the order in the example, it will send an XML response that will look like this:

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="TECHMAN" version="1.4">
  <reply>
    <orderStatus orderCode="AY845">
      <reference id="1234567">
        https://secure.bibit.com/jsp/shopper/SelectPaymentMethod.jsp?OrderKey=
                                                    TECHMAN^AY845
      </reference>
    </orderStatus>
  </reply>
</paymentService>
```

This reply originates from the live environment. The same reply from the test environment would look like this:

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="TECHMAN" version="1.4">
  <reply>
    <orderStatus orderCode="AY845">
      <reference id="1234567">
        https://secure-test.bibit.com/jsp/shopper/SelectPaymentMethod.jsp?OrderKey=
                                                    TECHMAN^AY845
      </reference>
    </orderStatus>
  </reply>
</paymentService>
```

4.5 Parsing (interpreting) the XML response

The response from Bibit has to be parsed by your system. Be sure to use a proper XML parser for this and DO NOT depend on a homemade one. As the Bibit DTD changes over time the home made parser might not be in synch with the changes and thus will not be able to correctly interpret the messages received from Bibit. This will not happen when you use a proper XML parser. Each different platform has it's own XML parsers.²

The XML message states it is a reply (`reply` element). It includes the order code so that your system will know which order the reply refers to. The reference id can be used as a payment reference if the shopper is expected to do a bank transfer or you use Accept Giro payments. In the latter case, this number (reference id) should be printed on the Accept Giros as the payment reference. If you are sending the order solely to acquire this reference id, there is no need to use the redirection URL and redirect the shopper. You can choose to store this reference id with the corresponding orderCode. Shoppers who paid an order using a bank transfer or Accept Giro sometimes refer to this number instead of the orderCode.

² There are a lot of resources on the Internet about platform specific parsers i.e. <http://www.xml.org>.

The URL included in the `reference` element is the so-called redirection URL, pointing to where the shopper needs to be redirected to. This URL can be supplemented with variables to change the appearance of the payment selection page in the secure Bibit environment (more on this later).

The order is now stored in the Bibit Payment Service and the system is ready to receive a payment for this order.

5 Anatomy of a Redirect Payment

When the order is saved at Bibit, the system is ready to process a payment for this order. When the order has been stored, it is time to redirect the shopper to the payment selection screen.

Redirecting the shopper means that the shopper's Internet browser is redirected from your shop to the payment selection screen in the secure Bibit environment where a payment can be initiated.

5.1 Why redirect?

When redirecting the shopper to the Bibit environment:

- all payment methods are available (subject to conditions of the contract). Some payment methods are not available with other connection types;
- active risk assessment by the Bibit Payment Service is possible;
- Bibit secures sensitive data, such as payment details;
- Bibit secures the communication with the shopper.

As the payment information (credit card number, CVC2 code, cardholder name, etc) is very sensitive, you should make sure that the data transfer and the data storage are extremely secure. As the costs of maintaining such a high level of security for all processes involved is very high, it's often easier for a merchant to simply redirect the shopper to the secured Bibit environment.

When shoppers are redirected to the secure Bibit environment, Bibit can gather information that can be used for risk assessment purposes.

Other ways of connecting to the Bibit Payment Service are 'XML Direct', 'Batch', 'Hosted Call Center Application' and 'Swipe'. These are discussed later on in this manual.

5.2 How to obtain the redirection location?

As explained in a previous section, the redirection location can be extracted from the XML response from the Bibit system after an order is successfully saved.

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="TECHMAN" version="1.4">
  <reply>
    <orderStatus orderCode="AY845">
      <reference id="1234567">
        https://secure-test.bibit.com/jsp/shopper/SelectPaymentMethod.jsp?OrderKey=
TECHMAN^AY845
      </reference>
    </orderStatus>
  </reply>
</paymentService>
```

The URL in the example above:

```
https://secure-test.bibit.com/jsp/shopper/SelectPaymentMethod.jsp?OrderKey=TECHMAN^AY845
```

is the URL the shopper needs to be redirected to in the test environment. In the live environment the URL would be the same except that the host name will be `secure.bibit.com`.

Note on 'reference id': this reference id would be saved in your back-office system as this number is used as a payment reference for off-line payment methods. Shoppers might use this number in their communication about orders.

5.3 How to redirect?

How the actual redirection is brought about depends on the implementation of your system. It can be done by providing the shopper with a simple link or by using a redirection method like the META refresh method. It is up to you to choose how to redirect the shoppers.

5.4 The payment method selection screen

After you have redirected the shopper his browser will display the so-called payment method selection screen. This screen will have the order description and the orderContent (included in HTML) as specified in the corresponding order.

Below the order content the shopper will find a list of the available payment methods for this particular order. The number and type of payment methods depends on the `paymentMethodMask` used and on the redirection variables appended to the redirection URL (see 'changing the appearance of the payment selection screen' below) and the country selection.

5.4.1 Changing the appearance of the payment selection screen

You can influence the appearance of the payment selection screen by adding variables to the redirection URL provided in the Bibit XML response. This is done on a per order basis.

An example of a redirection URL making use of all optional variables looks like this:

```
https://secure.bibit.com/jsp/shopper/SelectPaymentMethod.jsp?orderKey=TECHMAN^AY845&country=GB&language=EN&bodyAttr=bgcolor%3D%22black%22&fontAttr=face%3D%22arial%22+color%3D%22white%22&successURL=http%3A%2F%2Fwww.techman.com%2Fsuccess.asp&failureURL=http%3A%2F%2Fwww.techman.com%2Ffailure.php&pendingURL=http%3A%2F%2Fwww.techman.com%2Fpending.html&preferredPaymentMethod=SWITCH-SSL
```

Please note that the values of the parameters are URL-encoded. This means that all special characters are replaced by codes so they will be processed correctly. More information on URL-encoding will follow later in this manual.

Dissecting the redirection URL above:

- `https://secure.bibit.com/jsp/shopper/SelectPaymentMethod.jsp?orderKey=TECHMAN^AY845`

This part is what Bibit returns in the XML response once the order AY845 has been stored successfully in the system for merchant TECHMAN. In principle this would be enough to redirect the shopper, in that case the shopper would be presented the standard payment selection screen.

- `&country=GB&language=EN`

These parameters are optional. These variables set the country and the language. The language only influences the language in which the information originating from the Bibit system in the payment selection screen is presented. The country settings influence the payment methods presented to the shopper in the payment selection screen as payment methods are sorted and presented per country. Setting the country to GB will present the international credit cards and the country specific payment methods for Great Britain. Country and language can be specified independently from each other. In other words, you can present country specific payment methods for the Netherlands in Swedish.

- `&bodyAttr=bgcolor%3D%22black%22`

This parameter is optional. This sets the body attributes of the page. These body attributes can be anything that can be used in the <BODY> tag in HTML. In this example only the background colour has been set, black. This attribute can also be used to specify an image that should be used as a background to the payment page. Please refer to external HTML documentation for more attributes, and the syntax that can be used in the HTML <BODY> tag. Please note that if you are going to use a background image, the image has to originate from the same IP address as the order, and the HTTPS proxy has to be used, otherwise a pop-up alert from the browser will appear. If such a pop-up appears, Bibit will deny you access to the live environment. In order to use the Bibit HTTPS proxy, the URL for the image should be preceded by `https://secure.bibit.com/servlet/HTTPSProxy?`

- `&fontAttr=face%3D%22arial%22+color%3D%22white%22`

This parameter is optional. This sets the font attributes of the payment selection screen. In this example, the font face is set to Arial and the font colour set to white. Please note that when the font indicated is not available on the shopper's system the browser's default font will be used. It is possible to define alternative fonts by separating them with a comma (for example Arial, Verdana, Helvetica).

- `&successURL=http%3A%2F%2Fwww.techman.com/success.asp`
`&failureURL=http%3A%2F%2Fwww.techman.com/failure.php`
`&pendingURL=http%3A%2F%2Fwww.techman.com/pending.html`

These set the success URL, failure URL and pending URL. These URLs must reside on your server and are used to provide feedback about the payment to the shopper.

- `&preferredPaymentMethod=SWITCH-SSL`

This parameter is optional. This variable sets the preferred payment method. You have the possibility to pre-select the payment method for the shopper. In this example, the payment method is Switch. Reasons for setting the preferred payment method could be that you only want to accept one specific payment method (for this transaction) or because you want to bypass the payment selection screen presented by Bibit because the shopper chooses his preferred payment method in the shopping application on your server.

Except when you have specified a preferred payment method, the shopper has the possibility to select a language and country at the bottom (or at the top) of the page. Again, the language parameter only influences the language in which the information is shown, but the country parameter will determine the subset of payment methods shown on the page. The default position of the language and country selection boxes is at the bottom of the page and can be changed by Bibit on request. The language and country selection feature can also be switched-off by Bibit. You can contact Bibit through the incident console (<http://support.bibit.com>) to have it switched off. Please note that this feature is switched on or off for all transactions, it cannot be done on a per transaction basis.

5.5 Actions of the shopper

Once you have redirected the shopper to the payment selection screen, the shopper needs to choose the payment method and perform some payment method specific actions.

5.5.1 Choose payment method

Once the shopper has checked the details presented in the order content, he can choose a different language and or a different country using the corresponding selection boxes. Please note that this feature can be switched off if the merchant so chooses. After selecting the preferred language and/or country, the shopper has to click the 'select' button. The page will reload with the appropriate changes. After that, or if the settings matched his needs in default mode, the shopper selects his preferred payment method from the list presented on the screen by clicking on the name of the payment method.

5.5.2 Payment specific actions

After the shopper has selected his preferred payment method, Bibit will present the shopper with a screen specific to that payment method. If a shopper has chosen a credit card, he will be asked to fill out all necessary details such as credit card number, expiration date, cardholder name and CVC2 code, if required. The shopper clicks on 'submit' to send the payment information to the Bibit Payment Service.

The online authorisation of a credit card transaction will take 10 to 15 seconds. During this time, the Bibit system will digest the payment information and reformat it so it can be sent through the financial networks. Once Bibit has received the answer (authorised, refused, referred) from the financial institution, the order receives a payment status. From then on, we refer to it as a 'payment', even if the payment may have been refused. For the Bibit system, a 'payment' is an order that has received a payment status. The merchant's system will be informed about the payment status through two possible channels (email and/or HTTP).

Off-line payment methods require other actions from the shopper. For these payments, the merchant will be informed of the payment status after the actual transfer of money has taken place.

Please note that if a shopper (inadvertently) terminates the process from the moment the order has been stored at Bibit, the order will stay there indefinitely, never reaching a payment status. In other words, not all orders will reach a payment status.

5.6 Sending feedback to the shopper

The shopper has to be informed about the result of the payment. This can be done in various ways. For a purchase at an on-line web shop the shopper has to be presented with a screen informing him about the result of the payment. This is done through result URLs. In addition to on-line confirmation, the merchant can choose to send an email to the shopper informing him about the order and optionally providing extra information about delivery, etc.

5.6.1 Result URLs

There are three types of payment results possible:

- Success (the payment is authorised)
- Failure (the payment is refused)
- Pending (the payment result isn't directly available, when using off-line payment methods).

Each situation triggers a separate URL that is specified in the redirection URL as shown in an earlier example. If for some reason none of the specified URLs are appropriate or there is an error, the Bibit Payment Service will show a separate error message.

5.6.1.1 Success URL

In the earlier example (section 5.6.1) the success URL was set to `http://www.techman.com/success.asp (&successURL=http%3A%2F%2Fwww.techman.com/success.asp)`. This means that if the payment attempt is successful, i.e. results in an authorisation, the shopper will be redirected from the Bibit secure environment to `http://www.techman.com/success.asp`. This URL should be on your system and informs the shopper of the successful authorisation of the payment. Please note that your back office system should NOT interpret redirections to this page as authorisations, as this location can be reached by typing in the correct URL manually. Your back office system should only rely on the confirmation messages (through email or HTTP) from the Bibit system.

5.6.1.2 Pending URL

In the earlier example (section 5.6.1) the pending URL was set to `http://www.techman.com/pending.html (&pendingURL=http%3A%2F%2Fwww.techman.com/pending.html)`. The shopper

will be redirected to this page after using a payment method that does not provide online status reporting. One of those methods would be a bank transfer. As the system doesn't know whether or not the shopper will do the actual bank transfer and if so, when, the system cannot provide immediate online feedback. The shopper will therefore be redirected to a 'pending' page informing the shopper that the order has been placed and you will wait for the payment before sending out the goods. Please note that no payment status was reached so no confirmation message is sent!

5.6.1.3 Failure URL

In the earlier example (section 5.6.1) the failure URL was set to `http://www.techman.com/failure.php (&failureURL=http%3A%2F%2Fwww.techman.com/failure.php)`. If a payment is sent to the financial institution and is not authorised (refused) the shopper has to be informed about this. As many merchants only provide an extremely basic message that would leave many a shopper in doubt about the payment result, an example message is provided below.

"Unfortunately your payment was refused by the financial institution. No money will be deducted from your credit card or bank account. The order will not be processed. Click here to try again."

It is important that you inform the shopper about the fact that the financial institutions have refused the transaction and that no money will be transferred. This information will ease the minds of shoppers who otherwise would be confused about whether or not their credit card or bank account would be debited. This is one of the things Bibit checks during an acceptance test.

5.6.1.4 Request variables in the result URLs

It is possible to use request variables in the result URLs. Please note that the orderKey (merchantCode^orderCode) will be appended to the result URLs. So in the earlier example the success URL: `http://www.techman.com/success.asp?shopper_id=1234` will be changed into `http://www.techman.com/success.asp?shopper_id=1234&orderKey=TECHMAN^AY845`.

If you want to send the request variables `twip=no&c11=20` through the result URL the above URL should look like:

```
http://www.techman.com/success.asp?shopper_id=1234&orderKey=TECHMAN^AY845&twip=no&c11=20.
```

5.6.1.5 URL encoding

Please note that the result URLs (and their request variables) should be URL-encoded for correct processing. URL encoding transforms characters that might potentially create a problem in the correct processing of the URL into the hexadecimal representation of the ISO-Latin code for those characters. A space has the ISO-Latin code 32, in hexadecimal 32 is 20; the URL-encoded space will look like %20. Many platforms have tools (built-in functions) that can automatically URL-encode information.³

5.6.1.6 HTTPS proxy

A security warning may appear when the shopper is redirected from a secure location on the Bibit system to an insecure location on a merchant's system. The shopper's browser displays this warning. Even though the shopper can turn off this warning, most browsers will have this warning switched on after a standard installation.

A shopper may get confused after receiving the security warning. He may believe that the whole payment process was insecure and start to worry about having supplied his bank or credit card information. Alternately the shopper may choose not to follow the link and not see the result URL.

³ More information on URL-encoding can be found on the Internet, for instance at <http://www.bloobery.com/indexdot/html/topics/urlencoding.htm>

To avoid this warning message the Bibit system uses an HTTPS proxy. This proxy is a piece of software residing in the secure Bibit environment, requesting the result URL and showing it through the existing secure connection, instead of redirecting the shopper directly. This way the shopper will not receive a security warning. This feature is activated by default for each merchant account but can be switched off after a request for this by the merchant. Bibit recommends leaving this feature switched on.

Merchants may choose to use a secured environment themselves, but setting this up for the sole purpose of the redirection URLs is costly and can be technically challenging. Merchants who already have a secure environment in place need not use the proxy.

The proxy does have some restrictions:

- For security reasons the feature will only work directly after a payment has been done. This means that to test the proxy functionality you will have to go through the whole payment cycle.
- The URLs should reside on the same machine (IP address) that sends the orders to the Bibit Payment Service.
- Pages that redirect through the 302 HTTP return code do not function in combination with the proxy.

If the result page has a redirection itself, the way to achieve this in a manner compatible with the proxy would be to use a HTTP-refresh in the Meta tag of the document: `<meta http-equiv="refresh" content="0;url=somewhere.asp">`. A non-W3 supported redirection method that can be used in the result URL in combination with the Bibit proxy is:

```
<html>
  <head>
    <script language="JavaScript">
      <!--
      self.location='/redirectedfolder/thanyou.asp';
      //-->
    </script>
  </head>
</html>
```

This method is supported but only if implemented in the way shown above. You should replace the redirection URL with the desired URL.

5.6.2 Email notification

In addition to the online notification to the shopper an extra notification through email can be sent to provide the shopper with information he can archive for later reference. This can be done in several ways depending on the wishes of the merchant. The merchant's system can generate an email with all the necessary information for the shopper (1), or the Bibit system can be triggered to send an email to the shopper (2). In both cases, the shopper email address has to be available to the respective system.

1. Sent by merchant's system

Your system will send an email after it receives an automated notification from Bibit through the chosen feedback channel (email or HTTP). As the email is initiated by your system, you can choose when to send the email and exactly which information is provided to the shopper.

2. Sent by Bibit's system

If you don't want to implement this feature in your system you can have your Bibit account set-up to have Bibit send an email to the shopper. This email can be triggered by one of two events: Refused and Authorised.

If you would like to use this method, this feature has to be activated (at no extra cost) in your account by Bibit. You can change the settings and the text of the actual emails through the Merchant Interface (<http://www.bibit.com/merchant>). This feature can only be configured in the live environment, after which it will function in the test environment as well. When setting up the text for

the emails, two variables can be used to automatically import data from the Bibit system into the email. The variables that can be used are: &ordercode to import the ordercode, and &ordercontent, to import the orderContent as sent to Bibit in the original XML order. If the merchant wants to use HTML he should not use the <BODY> and <HTML> tags, as they will be added by the Bibit system automatically. Also, note that if the merchant wants to use HTML he should indicate this when requesting this feature to be activated in his account.

6 Communicating the results to the Merchant

Once a payment status has been reached, the Bibit system will notify the merchant's system. It will also keep the merchant informed about changes in the payment status of orders that already have a payment status.

6.1 Overview of payment statuses

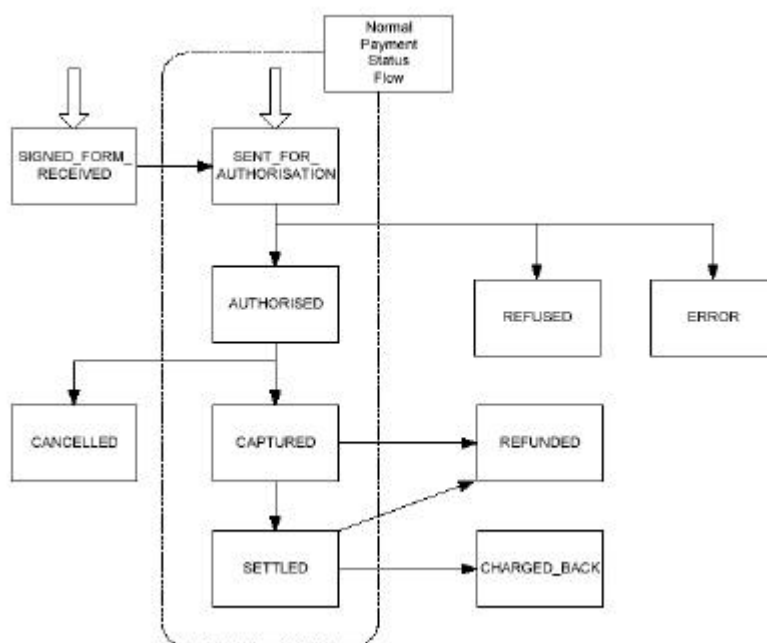


Figure 3. This image represents the normal flow of payment statuses through the Bibit Payment Service.

6.1.1 SIGNED_FORM_RECEIVED

This status applies only to Dutch and German Incasso payments and unsigned direct debits (available to selected merchants only). Bibit received the physical standing order form (to be printed by the shopper during the payment process) with the shoppers' signature either by fax or by conventional mail. The payment request is sent to the banks for further processing. This status does not assure a successful payment, since the automatic debit action can fail, because of insufficient balance, for instance.

6.1.2 SENT_FOR_AUTHORISATION

The payment has been sent to the financial institution and awaits a response. If the response is processed the following status can be AUTHORISED, CAPTURED, REFUSED or ERROR. If the payment has the status SENT_FOR_AUTHORISATION, no new payment attempt can be done for this order.

6.1.3 AUTHORISED

The usual procedure is that an order amount is first authorised. This means that the financial institution validates the payment, and in the case of a credit card payment, "reserves" the amount for the merchant and deducts it from the shopper's spending limit. The amount is not actually transferred yet. In the case of bank transfers and similar payment methods, the payment can reach this status immediately followed by CAPTURED.

You will be informed of the authorization and can commence shipping the ordered goods. In case of a credit card payment, the authorization remains valid for some time, depending on the credit card institution.

This payment status will be reached after the shopper finished selecting his payment method and entering relevant payment data, provided, of course, the payment is authorised. Payments through bank transfer and other off-line payment methods will reach this status in an off-line stage.

6.1.4 CAPTURED

The financial institution successfully processed the payment and the money will be transferred to Bibit. The standard time between AUTHORISED and CAPTURED statuses, the so-called capture delay, is five days, with a minimum of 0 days and a maximum of 14 days. The Bibit system will automatically capture any authorised payment after the appropriate capture delay for that account. This feature can be switched off under special circumstances to create an authorisation-only account. You can manually capture an authorised payment before the end of the capture delay through the Merchant Interface, or by sending a capture instruction in an order modification message.

As all credit card companies *dictate*, it is not allowed for a merchant to capture a payment before shipping has completed. You should only start shipping the order once the status AUTHORISED is reached!

If only part of the order can be shipped, it is allowed to capture an amount equal to the shipped part of the order; this is called a partial payment. Please note that only ONE capture per authorisation is possible. Therefore, if you do a partial capture, the remaining amount can't be captured anymore. The Payment System offers the partial capture functionality in the Merchant Interface or through an XML message when it is available for that specific order. Payments that were authorised can be captured as long as the authorisation hasn't expired. For selected merchants, and under specific conditions, the capability to do more than one partial capture can be enabled (the total captured amount never exceeding the originally authorised amount). Multiple partial captures is not a standard feature and extra fees will be charged.

Some payment methods, like bank transfers or direct debits, reach the CAPTURED status straight away, though always via SIGNED_FORM_RECEIVED, SENT_FOR_AUTHORISATION and AUTHORISED.

6.1.5 REFUSED

The financial institution refused to authorise the payment. This can be due to, for instance, the credit limit being exceeded, the expiry date being incorrect, the balance being insufficient, or many other reasons, depending on the selected payment method. Bibit is not always informed of the refusal reason by the financial institutions. Whatever information Bibit receives from the financial institutions will be communicated to you through the Merchant Interface, under 'payment details'.

If you are using the Risk Management Module, Bibit can refuse payments because of increased risk. More information on the Risk Management Module can be found in chapter 17.

The refusal reason REFERRED means that if you use your own connections to the financial institution instead of Bibit's, you will be able to call the respective financial institution to check this particular order and possibly obtain an authorisation code. This code can be entered through the Merchant Interface, or sent to Bibit through a so-called order modification message.

If a payment (attempt) ends in the REFUSED state, the shopper can choose a different payment method to try again. You will have to provide a link on his result URL(s) back to the payment selection screen.

6.1.6 ERROR

The payment was not completely processed by the Payment System and ended in error. There can be different reasons for this payment status. Payments with this status can be processed again. This means that the shopper can go through the payment process again.

Status ERROR is a final status for a payment attempt. Bibit does not report this status.

6.1.7 CANCELLED

You cancelled a payment that had the status AUTHORISED but which had not yet reached the status CAPTURED. Status CANCELLED means that the authorisation, or your right to collect the money, is cancelled. You can choose to cancel for various reasons: because the goods or services are no longer available, or he suspects fraud, or any other reason. A payment can be canceled either manually in the Merchant Interface or by sending a cancel instruction in an order modification message.

CANCELLED is a final status for a payment. Bibit reports this status.

6.1.8 CHARGED_BACK

Cardholders can reverse a capture by their bank or credit card company. The cardholder contacts his credit card company and denies having made the transaction. The credit card company then revokes a payment that was already captured. Please note the legal difference between 'shopper' (he who ordered the goods) and 'cardholder' (he who owns the credit card and ends up paying for the order).

The CHARGED_BACK status only occurs incidentally, and when it does, contacting the shopper can often solve the situation. Sometimes it is an indication of credit card fraud.

Bibit is not always informed in advance about pending charge backs and therefore cannot always inform its merchants in an earlier stage.

6.1.9 SETTLED

Bibit has added the payment to a merchant batch and will pay out the amount during the next four days OR the payment has already been transferred to your bank account.

6.1.10 SENT_FOR_REFUND

You have requested the financial institution to do a (partial) refund. You should inform the cardholder that it generally takes two to three weeks before he receives the refund, as he might expect the money sooner. We advise to use cancellation or a partial capture as an alternative whenever possible.

Please note that both partial refunds and full refunds will reach this status. The system doesn't differentiate between them.

6.1.11 REFUNDED

The refund has been processed by the financial institution and has been transferred to the cardholder's account. The amount will be deducted from the next merchant settlement batch.

6.1.12 EXPIRED

This means that the time to capture an authorised payment has expired. This is only applicable to merchants who have automatic capture by the Bibit Payment Service switched off (capture delay = 365 days). See also section 6.1.4 CAPTURED.

6.2 When does Bibit send a status message?

The Bibit system sends on-line status messages (also known as 'order notifications' or 'confirmation messages') through email and/or HTTP(S) for the following statuses, or changes from one of these statuses to another:

- SIGNED_FORM_RECEIVED
- AUTHORISED
- CANCELED
- CAPTURED
- CAPTURE_FAILED
- CHARGED_BACK
- SENT_FOR_REFUND

- REFUNDED
- REFUSED

No other payment statuses are sent as confirmation messages. Those other statuses can be checked through the Merchant Interface or by sending an information request in the form of an XML order inquiry.

The SETTLED status is reported in the transfer (batch) reports.

6.3 Overview of communication channels

Bibit can communicate these confirmation messages through two different channels: email (SMTP) and HTTP(S). In addition to these two channels the status information can always be viewed online through the Merchant Interface at <http://www.bibit.com/merchant>.

The two channels are: email and HTTP(S). Both channels have three possible formats: cgi, text or XML.

6.3.1 Merchant interface vs. messaging

The difference between the Merchant Interface and the confirmation messages is the fact that the Merchant Interface requires human interaction and is not very suitable for integration in the merchant's back-office system. The Merchant Interface is presented through a web browser whereas the confirmation messages are sent through email and/or HTTP(S). The Merchant Interface always shows all payment statuses whereas the confirmation messages are only sent once for the most important ones.

6.3.2 Email confirmations

Email confirmation messages can be sent to one or more valid email addresses. The recipient does not reply to these messages.

Text message:

```

Subject:
Payment Result for order DEMO1021272851471
Body:
*****

OrderCode: DEMO1021272851471
PaymentId: 2619062
PaymentAmount : 75,00
PaymentCurrency: EUR
PaymentStatus : AUTHORISED
PaymentMethod : ECMC-SSL
Date : Mon May 13 08:54:39 CEST 2002
*****

```

XML message:

```

Subject:
Payment Result for order DEMO1021273208485
Body:
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService version="1.4" merchantCode="TECHMAN">
  <notify>
    <orderStatusEvent orderCode="DEMO1021273208485">
      <payment>
        <paymentMethod>ECMC-SSL</paymentMethod>

```

```

    <amount value="2000" currencyCode="EUR" exponent="2"
      debitCreditIndicator="credit"/>
    <lastEvent>AUTHORISED</lastEvent>
    <balance accountType="IN_PROCESS_AUTHORISED">
      <amount value="2000" currencyCode="EUR" exponent="2"
        debitCreditIndicator="credit"/>
    </balance>
    <cardNumber>4111*****1111</cardNumber>
  </payment>
  <journal journalType="AUTHORISED">
    <bookingDate>
      <date dayOfMonth="23" month="08" year="2002"/>
    </bookingDate>
    <accountTx accountType="IN_PROCESS_AUTHORISED">
      <amount value="2000" currencyCode="EUR" exponent="2"
        debitCreditIndicator="credit"/>
    </accountTx>
  </journal>
</orderStatusEvent>
</notify>
</paymentService>

```

6.3.3 HTTP confirmations

Confirmation messages can also be sent through HTTP(S). They are sent to a URL on your system that is able to process the information contained in the HTTP(S) message. These URL's usually point to a page with some sort of script (asp, php, Perl, cgi) that will interpret the message and send the results of the interpretation through to the back-office system. This way a seamless integration is possible.

The most important difference between email confirmation messages and confirmation messages sent through HTTP(S) is the fact that HTTP(S) messages NEED to be acknowledged. This makes HTTP(S) a channel for guaranteed delivery: you can be sure that all confirmation messages sent by Bibit are received by your system.

Please note that if your system makes use of a firewall or other construction to restrict incoming traffic, it should never filter incoming messages from Bibit (e.g. HTTP(S) confirmations) based on Bibit's IP address. The IP address is subject to change without notice; instead, the system should filter on the Bibit domain name (bibit.com).

6.3.4 Replying to the HTTP(S) confirmation with [OK]

After the Bibit system has sent a confirmation message to a URL on your server it will wait for a acknowledgement from your system that the message is actually received. The acknowledgement that needs to be sent by your system is "[OK]". This is mandatory for each received message. The [OK] sent back to the Bibit system is not to certify that the information sent by Bibit is compliant to what your system is expecting, but only to acknowledge the receipt of the message! Therefore, an [OK] has to be sent upon receipt of each single message. Before HTTP(S) messaging can go live, Bibit will check if your system is sending the mandatory [OK] correctly.

If the Bibit system does not receive an [OK] as reply to a confirmation message sent through HTTP(S) it will wait for 5 minutes and it then tries to deliver the message again. This will be repeated for about 4 hours, until an [OK] is received. All new confirmation messages to be sent to your system are placed in a message queue. The Bibit system will try to deliver the first message first until it receives an [OK].

If no [OK] is received after 4 hours and no new message has triggered the Bibit system to try again, the system will stop trying to send the message (or actually trying to receive an [OK]) until a new confirmation message is created for you. The system will start the cycle again: trying to receive an [OK] for the first message in the queue every 5 minutes for four hours.

6.3.5 Example of an [OK]

When sending the [OK] please bear in mind the following:

- it should be an **unconditional** [OK],
- the [OK] is the literal string as shown here in quotation marks "[OK]",
- it should be in capital letters and between square brackets,
- it is meant to confirm the receipt of the Bibit confirmation messages, and does not convey any validation of their content,
- the HTTP response should be code 200, other codes will be interpreted as errors,
- Bibit guarantees that these messages are sent, not that each message is only sent once!

An example:

```
<HTML>
  <BODY>
    [ OK ]
  </BODY>
</HTML>
```

The [OK] does not need to be displayed in the browser, but it needs to appear somewhere in the response. Please note that sending "not [OK]" or "233wss[OK]sskjeKLD" would both constitute as a proper [OK] messages as the Bibit system will only scan for an "[OK]".

6.3.6 Examples of HTTP(S) confirmation messages

Below you find examples of Bibit HTTP(S) confirmation messages in cgi format, sent to <https://www.techman.com/bibit/confirmation.asp>. Please note that though the examples show HTTPS messages, the messages will look the same when sent through HTTP.

SIGNED_FORM_RECEIVED

```
https://www.techman.com/bibit/confirmation.asp?OrderCode=DEM01021272851471&PaymentId=2620415&PaymentStatus=SIGNED_FORM_RECEIVED&PaymentAmount=1250&PaymentCurrency=EUR&PaymentMethod=INCASSO_NL-FAX
```

AUTHORISED

```
https://www.techman.com/bibit/confirmation.asp?OrderCode=DEM01021272851471&PaymentId=15390&PaymentStatus=AUTHORISED&PaymentAmount=1000&PaymentCurrency=EUR&PaymentMethod=VISA-SSL
```

REFUSED

```
https://www.techman.com/bibit/confirmation.asp?OrderCode=DEM01021272851471&PaymentId=2620362&PaymentStatus=REFUSED&PaymentAmount=15290&PaymentCurrency=GBP&PaymentMethod=SWITCH-SSL
```

CAPTURED

```
https://www.techman.com/bibit/confirmation.asp?OrderCode=DEM01021272851471&PaymentId=2572542&PaymentStatus=CAPTURED&PaymentAmount=4950&PaymentCurrency=EUR&PaymentMethod=ECMC-SSL
```

CANCELLED

```
https://www.techman.com/bibit/confirmation.asp?OrderCode=DEM01021272851471&PaymentId=2615922&PaymentStatus=CANCELLED&PaymentAmount=40000&PaymentCurrency=EUR&PaymentMethod=ECMC-SSL
```

CHARGED_BACK

https://www.techman.com/bibit/confirmation.asp?OrderCode=DEM01021272851471&PaymentId=796345&PaymentStatus=CHARGED_BACK&PaymentAmount=5500&PaymentCurrency=EUR&PaymentMethod=AMEX-SSL

The next example is an XML confirmation message for status AUTHORISED:

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService version="1.4" merchantCode="DEMO">
  <notify>
    <orderStatusEvent orderCode="TEST-ORDER-1">
      <payment>
        <paymentMethod>VISA-SSL</paymentMethod>
        <amount currencyCode="EUR" debitCreditIndicator="credit" exponent="2"
          value="2600"/>
        <lastEvent>AUTHORISED</lastEvent>
        <balance accountType="IN_PROCESS_AUTHORISED">
          <amount currencyCode="EUR" debitCreditIndicator="credit" exponent="2"
            value="2600"/>
        </balance>
      </payment>
      <journal journalType="AUTHORISED">
        <bookingDate>
          <date dayOfMonth="30" month="10" year="2002"/>
        </bookingDate>
      </journal>
    </orderStatusEvent>
  </notify>
</paymentService>
```

An example of a CAPTURED confirmation message is:

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService version="1.4" merchantCode="VIBES">
  <notify>
    <orderStatusEvent orderCode="5141">
      <payment>
        <paymentMethod>VISA-SSL</paymentMethod>
        <amount value="2000" currencyCode="EUR" exponent="2"
          debitCreditIndicator="credit"/>
        <lastEvent>CAPTURED</lastEvent>
        <balance accountType="IN_PROCESS_CAPTURED">
          <amount value="2000" currencyCode="EUR" exponent="2"
            debitCreditIndicator="credit"/>
        </balance>
        <riskScore value="0"/>
      </payment>
      <journal journalType="CAPTURED">
        <bookingDate>
          <date dayOfMonth="27" month="09" year="2002"/>
        </bookingDate>
      </journal>
    </orderStatusEvent>
  </notify>
</paymentService>
```

The reply Bibit receives after sending any of these messages should ALWAYS be: [OK].

7 Integration with the merchant's Back-Office system

Integration with the merchant's back-office system is not mandatory, but most merchants will want to keep track of the orders processed by Bibit in their own back-office system(s). This is where the confirmation messages come in. Both the email and HTTP(S) channel can be used for this, depending on the wishes of the merchant and the level of integration. When confirmation messages are received, the merchant's system should interpret them and send the results to the appropriate process(es) on the merchant's system.

Please note that not all possible payment statuses are reported through confirmation messages .

Integration details depend on the platform(s) used by the merchant and the level of integration that is required. It is therefore not possible to provide any (generic) examples.

If no integration is done, merchants can keep track of the orders and payments through the Merchant Interface.

7.1 Back-office code

Some merchants use different codes in their back-office system to identify their orders that have been processed by the Bibit system. It is possible to "attach" these different codes to their corresponding codes used in the Bibit system. Please note that the merchant also provides the order codes used in the Bibit system.

The other order code, or back-office code, can be used by the merchant to differentiate between orders that have reached a capture status at one point, and thus need to be fulfilled, and orders that haven't. This back-office code can be attached to an order by sending an XML order modification message to the Bibit system at the moment of your choosing. This back-office code will appear in the transfer reports.

In the example below merchant TECHMAN adds the back-office code "CAP1234" to an existing order with the order code "1234".

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit/DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService version="1.4" merchantCode="TECHMAN">
  <modify>
    <orderModification orderCode="1234">
      <addBackOfficeCode backOfficeCode="CAP1234" />
    </orderModification>
  </modify>
</paymentService>
```

Upon receipt of this order modification, the Bibit Payment system replies with the following acknowledgement:

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService version="1.4" merchantCode="TECHMAN">
  <reply>
    <ok>
      <backofficeCodeReceived orderCode="1234" backOfficeCode="CAP1234" />
    </ok>
  </reply>
</paymentService>
```


7.2 Authenticity

As the confirmation messages sent to your system contain sensitive information it is advised that you put some measures in place to ensure the messages are genuine. Possible measures are: receiving the messages on a secure server, checking Bibit's client certificate for authenticity and checking the sender's domain name (this should be bibit.com).

7.2.1 Secure server

Using a secure server on your side will enable you to use the HTTPS protocol to receive the Bibit confirmation messages. HTTPS prevents eavesdropping and tampering with the actual messages. HTTPS is a standard Internet protocol and can be implemented on most web servers. Even though the connection is secured, this method will not ensure that actually Bibit sent the message.

Please note that using HTTPS for receiving messages from Bibit Payment Service is **not** a requirement!

7.2.2 Authenticating the sender (Bibit)

In order to be sure that it was Bibit who sent the confirmation messages, you can check Bibit's certificate. This feature has to be switched on per merchant; certificates aren't sent by default.

The Bibit Payment Service can send its certificate with all its communication to the merchant. A merchant can then check the authenticity of the certificate. Using the certificate will ensure that the information has not been tampered with during communication and that the sender was indeed Bibit.

To obtain Bibit's client certificates, please contact your Bibit contact person or request it through the online support console (<http://support.bibit.com>).

The HTTP(S) confirmation URLs should only be accessible by Bibit. When configuring your system to only accept incoming traffic from Bibit for that URL, your system should check the domain name, not the IP address. This is important as the IP addresses of the different systems that make up the Bibit Payment Service are subject to change without any notice.

8 Order Modification and Order Inquiry

Using the XML interface, merchants can send order modifications to perform actions on an existing order and order inquiries to actively request the status of an existing order from the Bibit Payment Service.

8.1 Order Modifications

Possible order modifications are:

- cancel
- capture
- refund
- add back office code
- authorisation code

8.1.1 Cancel

The following example shows an XML request to cancel order 1234 for merchant TECHMAN

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit/DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="TECHMAN" version="1.4">
  <modify>
    <orderModification orderCode="1234">
      <cancel/>
    </orderModification>
  </modify>
</paymentService>
```

An order can only be cancelled after it has reached the status AUTHORISED and can no longer be cancelled after it has reached the status CAPTURED. If the payment is not authorised there is nothing to cancel, after the payment has been captured it cannot be cancelled as it is in fact the authorisation that is cancelled. There can be many reasons to cancel an authorisation: the products or services are no longer available, you don't want to deliver to this particular shopper or the shopper has cancelled his order.

Upon receipt of this order modification, the Bibit Payment system replies with the following acknowledgement:

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService version="1.4" merchantCode="TECHMAN">
  <reply>
    <ok>
      <cancelReceived orderCode="1234"/>
    </ok>
  </reply>
</paymentService>
```

Later, when the cancellation has been successfully processed, Bibit sends a confirmation message (order notification). Please refer to section 6.3 for details on confirmation messages. For the above example Bibit sends the following confirmation message:

```

<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="TECHMAN">
  <notify>
    <orderStatus orderCode="1234">
      <payment>
        <paymentMethod>ECMC-SSL</paymentMethod>
        <amount value="2269" currencyCode="EUR" exponent="2"
          debitCreditIndicator="credit"/>
        <lastEvent>CANCELLED</lastEvent>
      </payment>
    </orderStatus>
  </notify>
</paymentService>
  
```

8.1.2 Refund

The following example shows an XML request to refund an amount of GBP 127,20 for order 1234 by merchant TECHMAN:

```

<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="TECHMAN" version="1.4">
  <modify>
    <orderModification orderCode="1234">
      <refund>
        <amount value="12720" currencyCode="GBP" exponent="2"
          debitCreditIndicator="credit"/>
      </refund>
    </orderModification>
  </modify>
</paymentService>
  
```

Refunds can be processed after a payment has reached the CAPTURED status. If a payment still has the AUTHORISED status, it can be cancelled but not refunded, because no money has yet been debited from the card/account. Refunds can never exceed the original captured amount. It is possible to do (several) partial refund(s). Once a refund request has been received by the Bibit system, the system checks whether the financial institution accepts refund instructions (not all do) and whether the amount to be refunded is still available for that order code. If so, the refund request will be scheduled. Bibit processes refunds in batches every 90 minutes. Please note that Bibit will send the refund instruction to the applicable financial institution within those 90 minutes, the financial institutions may take up to 3 weeks to actually process them.

Upon receipt of this order modification, the Bibit Payment system replies with the following acknowledgement:

```

<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService version="1.4" merchantCode="TECHMAN">
  <reply>
    <ok>
      <refundReceived orderCode="1234">
        <amount value="12720" currencyCode="GBP" exponent="2"/>
      </refundReceived>
    </ok>
  </reply>
</paymentService>
  
```

After the refund has been successfully processed, Bibit sends the confirmation message:

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit/DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="TECHMAN" version="1.4">
  <notify>
    <orderStatus orderCode="1234">
      <payment>
        <paymentMethod>VISA-SSL</paymentMethod>
        <amount value="20000" currencyCode="GBP" exponent="2"
          debitCreditIndicator="credit"/>
        <lastEvent>SENT_FOR_REFUND</lastEvent>
      </payment>
      <orderModification orderCode="1234">
        <refund>
          <amount currencyCode="GBP" debitCreditIndicator="credit" exponent="2"
            value="12720"/>
        </refund>
      </orderModification>
    </orderStatus>
  </notify>
</paymentService>
```

Please note that the system does not differentiate its responses for full or partial refunds, they are all 'Sent for refund'.

Only Credit card, Dankort, Carte Bancaire, Switch and Solo transactions can be refunded using orderModification messages.

8.1.3 Capture

The following example shows an XML request to capture an amount of EUR 109,65 for order 1234 for merchant TECHMAN:

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit/DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="TECHMAN" version="1.4">
  <modify>
    <orderModification orderCode="1234">
      <capture>
        <amount value="10965" currencyCode="EUR" exponent="2"
          debitCreditIndicator="credit"/>
      </capture>
    </orderModification>
  </modify>
</paymentService>
```

It is possible to do a partial capture. This can be useful in cases where not all ordered products or services can be delivered. Doing a partial capture at an early stage can prevent a refund at a later stage. Please note that only ONE (partial) capture can be done per authorised payment! Therefore, if a partial capture has been done the remaining amount for that order cannot be captured, as the authorisation expires immediately after the capture instruction.

For selected merchants the functionality to do more than one partial capture can be enabled, under specific conditions, and with the total captured amount never exceeding the originally authorised amount. Multiple partial captures functionality is not a standard feature and extra fees will be charged.

Upon receipt of this order modification, the Bibit Payment system replies with the following acknowledgement:

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService version="1.4" merchantCode="TECHMAN">
  <reply>
    <ok>
      <captureReceived orderCode="5678">
        <amount value="4000" currencyCode="EUR" exponent="2"/>
      </captureReceived>
    </ok>
  </reply>
</paymentService>
```

After the capture has been successfully processed, Bibit sends a confirmation message:

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="TECHMAN" version="1.4">
  <notify>
    <orderStatus orderCode="1234">
      <payment>
        <paymentMethod>VISA-SSL</paymentMethod>
        <amount currencyCode="EUR" debitCreditIndicator="credit" exponent="2"
          value="10965" />
        <lastEvent>CAPTURED</lastEvent>
        <balance accountType="IN_PROCESS_CAPTURED">
          <amount currencyCode="EUR" debitCreditIndicator="credit" exponent="2"
            value="10965" />
        </balance>
        <riskScore value="0" />
      </payment>
    </orderStatus>
  </notify>
</paymentService>
```

8.1.4 Authorisation code

One of the refusal reasons is 'referred'. This means that the appropriate financial institution should be contacted for more information. This status is only interesting if you connect to the Bibit system using your own accounts with the acquiring banks.

In such a case you should refer to your contact person with the appropriate acquiring bank about the transaction involved. The acquiring bank can give an authorisation code when they decided to authorise the payment. You can send this code either through the Merchant Interface or through an XML message so the payment status in the Bibit system can be changed to AUTHORISED and the money can be captured. Below is an example of the latter case.

```
<?xml version="1.0">
<!DOCTYPE paymentService PUBLIC "-//Bibit/DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService version="1.4" merchantCode="TECHMAN">
  <modify>
    <orderModification orderCode="1239">
      <authorise authorisationCode="acbsdf"/>
    </orderModification>
  </modify>
</paymentService>
```

This message will send the authorisation code (acbsdf) obtained from the acquiring bank for order 1239 for merchant TECHMAN to the Bibit Payment Service.

Upon receipt of this order modification, the Bibit Payment system replies with the following acknowledgement:

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService version="1.4" merchantCode="TECHMAN">
  <reply>
    <ok>
      <authorisationCodeReceived orderCode="1239" authorisationCode="acbsdf"/>
    </ok>
  </reply>
</paymentService>
```

After the authorisation code has been successfully processed, Bibit sends a confirmation message:

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="TECHMAN" version="1.4">
  <notify>
    <orderStatus orderCode="1239">
      <payment>
        <paymentMethod>VISA-SSL</paymentMethod>
        <amount currencyCode="EUR" debitCreditIndicator="credit" exponent="2"
          value="1180"/>
        <lastEvent>AUTHORISED</lastEvent>
        <ISO8583ReturnCode code="2" description="REFERRED"/>
        <balance accountType="IN_PROCESS_AUTHORISED">
          <amount currencyCode="EUR" debitCreditIndicator="credit" exponent="2"
            value="1180"/>
        </balance>
        <cardNumber>4111*****1111</cardNumber>
      </payment>
      <orderModification orderCode="1239">
        <authorise authorisationCode="acbsdf"/>
      </orderModification>
    </orderStatus>
  </notify>
</paymentService>
```

8.2 Order Inquiries

The XML message below will request the Bibit system to send the status of order 5678 for merchant TECHMAN.

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="TECHMAN" version="1.4">
  <inquiry>
    <orderInquiry orderCode="5678" />
  </inquiry>
</paymentService>
```

All Bibit reply messages to order inquiries contain a timestamp (date element), indicating the exact moment in time the replied status corresponds to.

The possible reply messages are:

8.2.1 Authorised

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="TECHMAN" version="1.4">
  <reply>
    <orderStatus orderCode="5678">
      <payment>
        <paymentMethod>VISA-SSL</paymentMethod>
        <amount value="4000" currencyCode="EUR" exponent="2"
debitCreditIndicator="credit"/>
        <lastEvent>AUTHORISED</lastEvent>
        <balance accountType="IN_PROCESS_AUTHORISED">
          <amount value="4000" currencyCode="EUR" exponent="2"
debitCreditIndicator="credit"/>
        </balance>
        <cardNumber>4568*****0015</cardNumber>
      </payment>
      <date dayOfMonth="21" month="05" year="2003"
hour="09" minute="47" second="24"/>
    </orderStatus>
  </reply>
</paymentService>
```

8.2.2 Refused (referred)

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="TECHMAN" version="1.4">
  <reply>
    <orderStatus orderCode="5678">
      <payment>
        <paymentMethod>VISA-SSL</paymentMethod>
        <amount value="4000" currencyCode="EUR" exponent="2"
debitCreditIndicator="credit"/>
        <lastEvent>REFUSED</lastEvent>
        <ISO8583ReturnCode code="2" description="REFERRED"/>
      </payment>
    </orderStatus>
  </reply>
</paymentService>
```

```

        <date dayOfMonth="21" month="05" year="2003"
            hour="09" minute="47" second="24"/>
    </orderStatus>
</reply>
</paymentService>

```

8.2.3 Refused (card blocked)

```

<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
    "http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="TECHMAN" version="1.4">
    <reply>
        <orderStatus orderCode="5678">
            <payment>
                <paymentMethod>VISA-SSL</paymentMethod>
                <amount value="4000" currencyCode="EUR" exponent="2"
                    debitCreditIndicator="credit"/>
                <lastEvent>REFUSED</lastEvent>
                <ISO8583ReturnCode code="76" description="CARD BLOCKED"/>
            </payment>
            <date dayOfMonth="21" month="05" year="2003"
                hour="09" minute="47" second="24"/>
        </orderStatus>
    </reply>
</paymentService>

```

8.2.4 Signed form received

```

<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
    "http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="TECHMAN" version="1.4">
    <reply>
        <orderStatus orderCode="5678">
            <payment>
                <paymentMethod>INCASSO_NL-FAX</paymentMethod>
                <amount value="4000" currencyCode="EUR" exponent="2"
                    debitCreditIndicator="credit"/>
                <lastEvent>SIGNED_FORM_RECEIVED</lastEvent>
            </payment>
            <date dayOfMonth="21" month="05" year="2003"
                hour="09" minute="47" second="24"/>
        </orderStatus>
    </reply>
</paymentService>

```

8.2.5 Sent for refund

```

<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
    "http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="TECHMAN" version="1.4">
    <reply>
        <orderStatus orderCode="5678">
            <payment>

```



```

        <paymentMethod>VISA-SSL</paymentMethod>
        <amount value="4000" currencyCode="EUR" exponent="2"
            debitCreditIndicator="credit"/>
        <lastEvent>SENT_FOR_REFUND</lastEvent>
    </payment>
    <date dayOfMonth="21" month="05" year="2003"
        hour="09" minute="47" second="24"/>
</orderStatus>
</reply>
</paymentService>

```

8.2.6 Settled

```

<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
    "http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="TECHMAN" version="1.4">
    <reply>
        <orderStatus orderCode="5678">
            <payment>
                <paymentMethod>TRANSFER_NL-BANK</paymentMethod>
                <amount value="4000" currencyCode="EUR" exponent="2"
                    debitCreditIndicator="credit"/>
                <lastEvent>SETTLED</lastEvent>
                <balance accountType="SETTLED_ACQUIRER_TOTAL">
                    <amount value="4000" currencyCode="EUR" exponent="2"
                        debitCreditIndicator="debit"/>
                </balance>
                <balance accountType="SETTLED_BIBIT_COMMISSION">
                    <amount value="23" currencyCode="EUR" exponent="2"
                        debitCreditIndicator="credit"/>
                </balance>
                <balance accountType="SETTLED_BIBIT_NET">
                    <amount value="3977" currencyCode="EUR" exponent="2"
                        debitCreditIndicator="credit"/>
                </balance>
            </payment>
            <date dayOfMonth="21" month="05" year="2003"
                hour="09" minute="47" second="24"/>
        </orderStatus>
    </reply>
</paymentService>

```

8.2.7 Charged-back

```

<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
    "http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="TECHMAN" version="1.4">
    <reply>
        <orderStatus orderCode="5678">
            <payment>
                <paymentMethod>VISA-SSL</paymentMethod>
                <amount value="4000" currencyCode="EUR" exponent="2"
                    debitCreditIndicator="credit"/>
                <lastEvent>CHARGED_BACK</lastEvent>
            </payment>
            <date dayOfMonth="21" month="05" year="2003"
                hour="09" minute="47" second="24"/>
        </orderStatus>
    </reply>
</paymentService>

```

```

    hour="09" minute="47" second="24"/>
  </orderStatus>
</reply>
</paymentService>

```

8.2.8 Cancelled

```

<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
    "http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="TECHMAN" version="1.4">
  <reply>
    <orderStatus orderCode="5678">
      <payment>
        <paymentMethod>VISA-SSL</paymentMethod>
        <amount value="4000" currencyCode="EUR" exponent="2"
            debitCreditIndicator="credit"/>
        <lastEvent>CANCELLED</lastEvent>
      </payment>
      <date dayOfMonth="21" month="05" year="2003"
          hour="09" minute="47" second="24"/>
    </orderStatus>
  </reply>
</paymentService>

```

8.2.9 Could not find order

```

<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
    "http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="TECHMAN" version="1.4">
  <reply>
    <orderStatus orderCode="5678" >
      <error code="5"><![CDATA[Could not find payment for order]]></error>
      <date dayOfMonth="21" month="05" year="2003"
          hour="09" minute="47" second="24"/>
    </orderStatus>
  </reply>
</paymentService>

```

This message indicates that there is no order with order code 5678 for merchant TECHMAN in the system.

8.2.10 Could not find payment

```

<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
    "http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="TECHMAN" version="1.4">
  <reply>
    <orderStatus orderCode="5678">
      <error code="5"><![CDATA[Could not find payment for order]]></error>
      <date dayOfMonth="21" month="05" year="2003"
          hour="09" minute="47" second="24"/>
    </orderStatus>
  </reply>

```

```
</paymentService>
```

This message indicates that no payment was found for this order. This order hasn't reached a payment status yet.

8.2.11 Captured

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
    "http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="TECHMAN" version="1.4">
  <reply>
    <orderStatus orderCode="5678">
      <payment>
        <paymentMethod>RABO-DIRECTBETALEN</paymentMethod>
        <amount value="4000" currencyCode="EUR" exponent="2"
            debitCreditIndicator="credit"/>
        <lastEvent>CAPTURED</lastEvent>
        <balance accountType="IN_PROCESS_CAPTURED">
          <amount value="4000" currencyCode="EUR" exponent="2"
              debitCreditIndicator="credit"/>
        </balance>
      </payment>
      <date dayOfMonth="21" month="05" year="2003"
          hour="09" minute="47" second="24"/>
    </orderStatus>
  </reply>
</paymentService>
```

The `orderStatus` element within the `lastEvent` element provides the status of the order/payment.

9 Extended functionalities

Depending on the contract details, a merchant has access to extended functionalities of the Bibit Payment Service. The availability of these functionalities is subject to the Bibit service model used, the type of goods or services on sale, order and payment history of the merchant, among other things. Bibit decides which merchants can have access to these functionalities.

9.1 Recurring payments (Pay as Order/Split Payments)

This functionality is restricted to selected merchants for the Direct model only and has to be activated by Bibit Operations. Recurring payments can be used for monthly subscriptions of any kind or regularly recurring payments such as upgrading online credits for services.

Below is an example of a recurring payment submitted to Bibit:

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit/DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService version="1.4" merchantCode="TECHMAN">
  <submit>
    <order orderCode="1234">
      <description>Monthly subscription.</description>
      <amount value="1399" currencyCode="EUR" exponent="2" />
      <orderContent>Provide your oderContent here as described earlier.</orderContent>
      <payAsOrder orderCode="1111">
        <amount value="1399" currencyCode="EUR" exponent="2" />
      </payAsOrder>
    </order>
  </submit>
</paymentService>
```

This message sends order 1234 to Bibit and instructs the system to use the same payment details as for the previous order 1111. As a result, order 1234 for merchant TECHMAN will be created for an amount of EUR 13.99 and the payment details for order 1111 will be used to pay this order.

Bibit's reply displays the payment details for order 1111:

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit/DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="TECHMAN" version="1.4">
  <reply>
    <orderStatus orderCode="1234">
      <payment>
        <paymentMethod>VISA-SSL</paymentMethod>
        <amount value="1399" currencyCode="EUR" exponent="2"
          debitCreditIndicator="credit"/>
        <lastEvent>AUTHORISED</lastEvent>
        <balance accountType="IN_PROCESS_AUTHORISED">
          <amount value="1399" currencyCode="EUR" exponent="2"
            debitCreditIndicator="credit"/>
        </balance>
        <cardNumber>4568*****0015</cardNumber>
      </payment>
    </orderStatus>
```

```
</reply>
</paymentService>
```

If your account hasn't been set up to allow split payments, the following error message will be received instead:

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit/DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="TECHMAN" version="1.4">
  <reply>
    <orderStatus orderCode="1234">
      <error code="2"><![CDATA[Your contract does not allow split payment.]]></error>
    </orderStatus>
  </reply>
</paymentService>
```

9.2 Card deposit

In some industries it is common to be able to pay out money to customers. In the normal situation, a merchant would only be able to send money to a credit card if he had first been authorised to capture that amount. In effect, this means that a merchant can only do refunds and not pay out extra money.

The card deposit functionality makes it possible to pay out money to a card without first having to get an authorisation for that card.

WARNING: this feature involves considerable risk. Fraudsters who capture this message can deposit large amounts of money on credit cards. Because of the risks involved this feature is switched off by default and only after rigorous investigation of business policies and financial situation of the merchant by Bibit this feature may (or may not) be switched on.

The XML is identical to a regular refund (section 8.1.2), extended with payment details. These payment details are important because the system has to do a refund on an order that didn't previously exist.

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit/DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService version="1.4" merchantCode="TECHMAN">
  <submit>
    <order orderCode="depos1234">
      <description>short description </description>
      <amount value="1000" currencyCode="EUR" exponent="2"/>
      <orderContent>Order Content</orderContent>
      <paymentDetails action="REFUND">
        <VISA-SSL>
          <cardNumber>4444333322221111</cardNumber>
          <expiryDate>
            <date month="02" year="2003"/>
          </expiryDate>
          <cardHolderName>John Doe</cardHolderName>
          <cvc>123</cvc>
        </VISA-SSL>
      </paymentDetails>
    </order>
  </submit>
</paymentService>
```

Bibit's reply:

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
    "http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="TECHMAN" version="1.4">
  <reply>
    <orderStatus orderCode="depos1234">
      <payment>
        <paymentMethod>VISA-SSL</paymentMethod>
        <amount value="1000" currencyCode="EUR" exponent="2"
            debitCreditIndicator="credit"/>
        <lastEvent>SENT_FOR_REFUND</lastEvent>
      </payment>
      <orderModification orderCode="1234">
        <refund>
          <amount currencyCode="EUR" debitCreditIndicator="credit" exponent="2"
              value="1000"/>
        </refund>
      </orderModification>
    </orderStatus>
  </reply>
</paymentService>
```

If you do not have access to this feature, the following error message will be sent instead:

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
    "http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="TECHMAN" version="1.4">
  <reply>
    <orderStatus orderCode="depos1234">
      <error code="2">
        <![CDATA[Your contract does not allow refunds on non-existing payments.]]>
      </error>
    </orderStatus>
  </reply>
</paymentService>
```

10 Other Service Models

In the preceding chapters, the Redirect service model has been described. There are however other models for processing payments through the Bibit Payment Service. These other models are available for selected merchants only and each model has its specific advantages and field of application. This chapter describes the other models:

- XML Direct
- XML Batch
- Hosted Call Center Application
- Bibit Swipe

10.1 XML Direct

10.1.1 Direct vs. Redirect

In the Redirect model the shopper is redirected to the secure Bibit environment to choose the payment method and enter the payment details such as credit card number, cardholder name, etc. In the Direct model, ALL information has to be collected on your system, as the shopper is not redirected to the secured Bibit environment. In your XML message to Bibit you must not only provide the order information, but also the payment information.

10.1.2 Security

The core issue in the Direct model is security. The collection and storage of information such as card numbers, cardholder names, expiry dates, CVC2 codes and other relevant payment information has to be done in a secure environment. All this information is very sensitive and in most countries falls under some form of privacy legislation.

If you consider using the Direct model you should be aware of the security measures that have to be put in place when connecting to Bibit. It is not simply a matter of setting up a secure server to provide an SSL (Secure Socket Layer) connection to the shopper's browser. Once the credit card data has been collected, it needs to be stored in a secure database using encryption algorithms. It is important to realise that the biggest threat to on-line transactions are vulnerable databases with all information of thousands of valid credit cards. Be aware that databases like this fall under special privacy protection laws and special legal provisions may have to be put in place.

10.1.3 Anatomy of an order (XML direct)

The order in the XML Direct model is very similar to an order in the Redirect model. In addition to the usual order information, the order in the Direct model must provide payment information. You should also provide additional information about the session. This information is used for fraud detection processes and cannot be gathered by Bibit, as the shopper is not redirected to the secured Bibit environment.

Below is an example of a complete order in the Direct model:

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit/DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService version="1.4" merchantCode="TECHMAN">
  <submit>
    <order orderCode="AY845">
      <description>ACME Webshop int. Inc.</description>
      <amount value="1982" currencyCode="EUR" exponent="2"/>
    <orderContent>
      <![CDATA[
```

```

<center><table>
<tr><td bgcolor="#ffff00">Your Internet Order:</td><td colspan="2"
bgcolor="#ffff00" align="right">AY 845</td></tr>
<tr><td bgcolor="#ffff00">Description:</td><td>14 Tulip bulbs</td><td
align="right">1,00</td></tr>
<tr><td colspan="2">Subtotal:</td><td align="right">14,00</td></tr>
<tr><td colspan="2">VAT: 13%</td><td align="right">1,82</td></tr>
<tr><td colspan="2">Shipping and Handling:</td><td align="right">4,00</td></tr>
<tr><td colspan="2" bgcolor="#c0c0c0">Total cost:</td><td bgcolor="#c0c0c0"
align="right">Euro 19,82</td></tr>
<tr><td colspan="3">&nbsp;</td></tr>
<tr><td bgcolor="#ffff00" colspan="3">Your billing address:</td></tr>
<tr><td colspan="3">Mr. Doe,<br>11 Hereortherestreet,<br>1234 KL
Somewhereorother,<br>Thisplace.</td></tr>
<tr><td colspan="3">&nbsp;</td></tr>
<tr><td bgcolor="#ffff00" colspan="3">Your shipping address:</td></tr>
<tr><td colspan="3">Mr. Doe,<br>11 Hereortherestreet,<br>1234 KL
Somewhereorother,<br>Thisplace.</td></tr>
<tr><td colspan="3">&nbsp;</td></tr>
<tr><td bgcolor="#ffff00" colspan="3">Our contact information:</td></tr>
<tr><td colspan="3">ACME Webshops Int. Inc.,<br>11 Strangewood Blv.,<br>1255 KZ
Thisisit,<br>Nowhereatall.<br><br>acmeweb@acme.inc<br>(555) 1235 456</td></tr>
<tr><td colspan="3">&nbsp;</td></tr>
<tr><td bgcolor="#c0c0c0" colspan="3">Billing notice:</td></tr>
<tr><td colspan="3">Your payment will be handled by Bibit Global Payments
Services<br>This name may appear on your bank
statement<br>http://www.bibit.com</td></tr>
</table></center>
  ]]>
</orderContent>
<paymentDetails>
  <VISA-SSL>
    <cardNumber>444433332221111</cardNumber>
    <expiryDate>
      <date month="09" year="2003" />
    </expiryDate>
    <cardHolderName>J. Doe</cardHolderName>
    <cvc>123</cvc>
    <cardAddress>
      <address>
        <firstName>John</firstName>
        <lastName>Doe</lastName>
        <street>11 Hereortherestreet</street>
        <postalCode>1234KL</postalCode>
        <city>Somewhereorother</city>
        <countryCode>TP</countryCode>
        <telephoneNumber>0123456789</telephoneNumber>
      </address>
    </cardAddress>
  </VISA-SSL>
  <session shopperIPAddress="123.123.123.123" id="0215ui8ib1" />
</paymentDetails>
<shopper>
  <shopperEmailAddress>me@techman.com</shopperEmailAddress>
  <authenticatedShopperID>1234567890</authenticatedShopperID>
</shopper>
<shippingAddress>
  <address>
    <firstName>John</firstName>
    <lastName>Doe</lastName>
    <street>11 Hereortherestreet</street>
  </address>
</shippingAddress>

```



```

        <postalCode>1234KL</postalCode>
        <city>Somewhereorother</city>
        <countryCode>TP</countryCode>
        <telephoneNumber>0123456789</telephoneNumber>
    </address>
</shippingAddress>
</order>
</submit>
</paymentService>

```

10.1.4 Payment details

Compared to the orders in the Redirect model two new elements, `paymentDetails` and `session` have been added.

The element `paymentDetails` contains the details for the selected payment method. In this case, the payment method is VISA-SSL. The `cvc` element is optional but recommended. More information about the CVC(2) code can be found in section 15.3

Each payment method has its own set of sub-elements and attributes. For VISA-SSL the `paymentDetails` look as follows:

```

<paymentDetails>
  <VISA-SSL>
    <cardNumber>4444333322221111</cardNumber>
    <expiryDate>
      <date month="09" year="2003" />
    </expiryDate>
    <cardHolderName>J. Doe</cardHolderName>
    <cvc>123</cvc>
    <cardAddress>
      <address>
        <firstName>John</firstName>
        <lastName>Doe</lastName>
        <street>11 Hereortherestreet</street>
        <postalCode>1234KL</postalCode>
        <city>Somewhereorother</city>
        <countryCode>TP</countryCode>
        <telephoneNumber>0123456789</telephoneNumber>
      </address>
    </cardAddress>
  </VISA-SSL>
  <session shopperIPAddress="123.123.123.123" id="0215ui8ib1" />
</paymentDetails>

```

The `paymentDetails` for the German payment method ELV-SSL has the following format:

```

<paymentDetails>
  <ELV-SSL>
    <accountHolderName>John Doe</accountHolderName>
    <bankAccountNr>1234567</bankAccountNr>
    <bankName>My bank</bankName>
    <bankLocation>Koeln</bankLocation>
    <bankLocationId>12345678</bankLocationId>
  </ELV-SSL>
</paymentDetails>

```

Please check the Bibit DTD (<http://dtd.bibit.com>) for details on the formats for other payment methods.

10.1.5 Bibit's response

The Bibit reply to the submitted order in the above example is:

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="TECHMAN" version="1.4">
  <reply>
    <orderStatus orderCode=" AY845">
      <payment>
        <paymentMethod>VISA-SSL</paymentMethod>
        <amount value="1982" currencyCode="EUR" exponent="2"
          debitCreditIndicator="credit"/>
        <lastEvent>AUTHORISED</lastEvent>
        <CVCRresultCode description="APPROVED" />
        <balance accountType="IN_PROCESS_AUTHORIZED">
          <amount value="1982" currencyCode="EUR" exponent="2"
            debitCreditIndicator="credit"/>
        </balance>
        <cardNumber>4444*****1111</cardNumber>
      </payment>
    </orderStatus>
  </reply>
</paymentService>
```

10.1.6 Reduced number of payment methods

It is important to understand that certain payment methods are not available in the Direct model due to the nature of the interaction with the shopper. Any payment method that needs direct interaction between the shopper or the shopper's system and the payment system cannot be used.

10.1.7 Order Modifications and Order Inquiries

Order modification and order inquiries for the Direct model are identical to those on the Redirect service model.

10.1.8 Risk Management in combination with XML Direct

Because the shopper is not redirected to the secured Bibit environment, Bibit is not able to collect certain information about the shopper that is used for fraud assessment. Therefore, you are advised to provide this information to Bibit while submitting the order.

Information that is not required for processing an order but that can be used for fraud assessment includes: shopper IP address, shopper email address, session, authenticated shopper ID and CVC2. We do recommend that `shopperIPAddress` attribute in the `session` element is *always* used, see example in section 10.1.3. Other information that is used for fraud assessment is present in the required attributes. Please note that 'authenticated shopper ID' is required in certain types of recurring payments.

10.1.9 Informing the Shopper

As the shopper is not redirected to the secured Bibit environment no result URLs can be specified in the redirection URL (there is no redirection URL to begin with). Bibit reports to your system in the same way as in the Redirect model. Your system itself will have to determine whether or not a payment was successful by interpreting the confirmation messages sent by Bibit, and will have to provide the appropriate feedback to the shopper.

10.2 XML Batch

10.2.1 Batch vs. other models

The Batch model can best be compared to the Direct model. The only difference is that in the Batch model the orders are gathered in a batch and sent to the Bibit Payment Service in one batch, be it once or several times a day. All transactions done using this model are inherently off-line, as no direct feedback to the shopper is possible.

The Batch model is specifically useful for merchants who have a very high transaction volume.

10.2.2 Anatomy of an order batch

One of the most common problems, when submitting batches to the Bibit Payment Service, is the fact that the merchantBatchCode HAS to be unique. Even if a batch, for whatever reason, is resubmitted to Bibit with the exact same information, it will have to have another merchantBatchCode!

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit/DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService version="1.4" merchantCode="TECHMAN">
  <submit>
    <orderBatch transactions="3" merchantBatchCode="B1234">
      <order orderCode="1234">
        <description>A meaningful description here.</description>
        <amount value="9800" currencyCode="EUR" exponent="2"
          debitCreditIndicator="credit"/>
        <orderContent>Your order content here.</orderContent>
        <paymentDetails>
          <SINGLE_UNSIGNED_DD_NL-SSL>
            <BankAccount_NL>
              <bankAccountNr>1234567</bankAccountNr>
              <accountHolderName>J. Doe</accountHolderName>
              <accountHolderResidence>Maastricht</accountHolderResidence>
            </BankAccount_NL>
          </SINGLE_UNSIGNED_DD_NL-SSL>
        </paymentDetails>
      </order>
      <order orderCode="1235">
        <description> A meaningful description here.</description>
        <amount value="14300" currencyCode="EUR" exponent="2"
          debitCreditIndicator="credit"/>
        <orderContent> Your order content here .</orderContent>
        <paymentDetails>
          <SINGLE_UNSIGNED_DD_NL-SSL>
            <BankAccount_NL>
              <bankAccountNr>1234568</bankAccountNr>
              <accountHolderName>M.N. De Clerck</accountHolderName>
              <accountHolderResidence>Utrecht</accountHolderResidence>
            </BankAccount_NL>
          </SINGLE_UNSIGNED_DD_NL-SSL>
        </paymentDetails>
      </order>
      <order orderCode="1236">
        <description> A meaningful description here.</description>
        <amount value="9800" currencyCode="EUR" exponent="2"
          debitCreditIndicator="credit"/>
        <orderContent> Your order content here .</orderContent>
        <paymentDetails>
          <SINGLE_UNSIGNED_DD_NL-SSL>
```

```

        <BankAccount_NL>
          <bankAccountNr>1234569</bankAccountNr>
          <accountHolderName>J.J. Smith</accountHolderName>
          <accountHolderResidence>Amsterdam</accountHolderResidence>
        </BankAccount_NL>
      </SINGLE_UNSIGNED_DD_NL-SSL>
    </paymentDetails>
  </order>
</orderBatch>
</submit>
</paymentService>

```

A batch can contain different sorts of payments, as long as each payment has the correct format and the correct data is supplied.

10.2.3 Bibit's response

When all orders within a batch are formatted correctly, the Payment Service will send an XML response to confirm the batch was received.

```

<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit/DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="TECHMAN" version="1.4">
  <reply>
    <batchStatus transactions="3" merchantBatchCode="B1234"
      status="ORDERS_SAVED"/>
  </reply>
</paymentService>

```

This reply only means the XML formatting of the orders and the batch was correct and the batch has been saved in the Bibit Payment Service for processing. The content of the batch is not yet validated at this point; there could still be invalid orders in the batch. The response simply means that the formatting was correct from a DTD point of view.

10.2.4 Payment results

At scheduled times the Bibit Payment Service will send all valid orders to the acquiring banks for further processing. Invalid orders within a batch will not be sent and will generate errors. These errors will describe what's wrong with those particular orders and will be available through a batch inquiry. Orders are invalid, for instance, if the used currency is not accepted for the merchant's account, if there are too many characters in the description field, in cases of duplicate orders, etc.

The individual payments will trigger email or http confirmation messages if this is specified in the merchant's account.

The batch will reach a new status after it has been processed. Once all orders within the batch have been processed without error, the new batch status will be PROCESSED". If one or more orders within the batch ended in ERROR (from the acquirers) the status for the batch will be PROCESSED_WITH_ERROR. The Bibit Payment Service will distinguish between errors at the acquiring bank or due to incorrect XML content.

10.2.5 Batch inquiries

To find out the status of the orders within a certain batch the merchant can send a 'batch inquiry' to the Bibit Payment Service:

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit/DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService version="1.4" merchantCode="TECHMAN">
  <inquiry>
    <batchInquiry merchantBatchCode="B1235" />
  </inquiry>
</paymentService>
```

Bibit's response *after* the batch has been processed:

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit/DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService version="1.4" merchantCode="TECHMAN">
  <batchStatus transactions="1" merchantBatchCode="123456" status="PROCESSED">
    <orderStatus orderCode="B1237">
      <payment>
        <paymentMethod>VISA-SSL</paymentMethod>
        <amount value="36442" currencyCode="EUR" exponent="2"
          debitCreditIndicator="credit"/>
        <lastEvent>CAPTURED</lastEvent>
        <balance accountType="IN_PROCESS_CAPTURED">
          <amount value="36442" currencyCode="EUR" exponent="2"
            debitCreditIndicator="credit"/>
        </balance>
      </payment>
    </orderStatus>
  </batchStatus>
</paymentService>
```

This is a reply for a batch inquiry for batch B123 for merchant TECHMAN containing 1 transaction. This batch has been processed. Last event indicates the status of the order. This status will also have been sent as an http or email confirmation message, if this has been activated on for this merchant's account.

10.2.6 Batch Payment methods

It is important to realise that certain payment methods that need direct interaction with the shopper or the shopper's system cannot be used. Due to their nature, they will not be available for the Batch model.

Payment methods that can be used in the Batch model are:

- VISA-SSL
- AMEX-SSL
- JCB-SSL
- DISCOVER-SSL
- MC-SSL
- DINERS-SSL
- LASER-SSL
- SWITCH-SSL
- SOLO-SSL
- ELV-SSL
- UNSIGNED_DIRECT_DEBIT

10.2.7 Possible batch statuses

10.2.7.1 ORDERS_SAVED

This status indicates that the Bibit Payment Service has saved the batch for processing at a scheduled time. It also means that the Bibit Payment Service was able to parse the XML in the batch order. It doesn't mean that the orders themselves were valid (duplicate order numbers for instance are not allowed). Invalid orders will generate individual error messages. These messages can be retrieved using a batch inquiry.

10.2.7.2 PROCESSED

This status indicates that all orders within the batch have been processed and no errors were encountered.

10.2.7.3 PROCESSED_WITH_ERROR

This status indicates that the batch has been processed but some errors were encountered. Using a batch inquiry the merchant can get a list of all results for each payment within the batch.

10.2.8 Order Modifications and Order Inquiries

Similar to the Redirect model, in the Batch model it is possible to send an order modification or order inquiry to the Bibit Payment. Batch order modifications do not get an acknowledgement reply from Bibit upon receipt.

10.2.9 Risk Management in combination with Batch payments

Similar to the Direct model, the fraud detection is more accurate if the merchant supplies as much information about the shopper and the session as possible.

10.3 Hosted Call Center Application

The Hosted Call Center Application is a way of connecting to the Bibit Payment Service created especially for call centers, or more specifically, for call centers that, for technical or logistical reasons, cannot integrate their systems with the Bibit Payment Service. In that respect the Hosted Call Center Application is different from call centers that use any of the other models (Redirect, Direct or Batch) to process payments through Bibit.

As the name already implies, every aspect of Hosted Call Center Application is hosted on the Bibit system. There is no need for the call center to generate XML messages. The only requirement is that the call center operators have access to a workstation with Internet access and a webbrowser.

Under normal conditions the Hosted Call Center Application can be up and running within a few hours, as no integration from the merchant's side is required.

10.3.1 Sending an order

The call center operator logs into the application through the webbrowser at <https://secure.bibit.com/jsp/callcenter/CallCenter.jsp> using the appropriate login and password.

The operator is presented with an online form where he can enter the order details, the amount and a description. The application can be configured to generate order codes so the operator does not have to fill out this field. After clicking the 'send' button the order is sent to the Bibit Payment Service and the operator is presented a payment selection screen. The operator clicks on the desired payment method and fills out the payment details and clicks "send" again. After the payment has been processed (a

payment status is reached) the operator will be presented a screen with the payment result and he is ready to process the next payment.

10.3.2 Feedback to the shopper

The only feedback the shopper can get is the feedback from the call center operator. In order to avoid confusion, the shopper should be informed that the name Bibit Internet Payments might appear on the bank or credit card statement. The account can be configured to send the shopper an email when status 'refused' or 'authorised' has been reached.

10.3.3 Feedback to the merchant

The feedback to the merchant occurs in the same way as with the other models. This means that Bibit can send email confirmations and or HTTP(S) confirmations. Normal reporting (transfer reports, order reports) through the usual channels (email and or HTTP(S)) is possible in this application too. Several levels of integration are possible, from no integration at all to fully automatic processing of the HTTP confirmation messages updating the back-office and logistical systems. This is up to the merchant.

10.3.4 Testing

Connecting to the Bibit Payment Service through the Hosted Call Center Application requires no mandatory acceptance test, as this application runs entirely on the Bibit system.

10.3.5 Call Center Payment methods

Normally the payment methods available to the shopper are determined by the settings in the paymentMethodMask. As the Hosted Call Center Application runs entirely on the Bibit systems the merchant cannot define this himself. The paymentMethodMask settings can be added by Bibit Operations only. It is best to communicate these payment methods to Bibit at the time of setting up the account.

It is important to realise that certain payment methods will not be available due to the nature of the interaction with the shopper. Any payment methods that require direct interaction with the shopper or the shopper's system cannot be used. These payment methods include all SET payment methods, Rabobank Direct betalen, ABN e-Wallet, Paybox, etc.

10.3.6 Accepted currencies

Similar to the paymentMethodMask, the accepted currencies of the Hosted Call Center Application can only be set in the account settings of the merchant. This can be done by Bibit Operations.

10.3.7 Ordercode

There are two methods for generating the ordercodes for the orders processed through the Hosted Call Center Application. The ordercodes can be generated by the merchant (copying them from another, external, system) and typed in by the operator. Or they can be generated by the Bibit system, automatically ensuring they are unique, and relieving the operator from having to type them in. When using the latter method the call center has to make sure there is a connection between the ordercodes generated by Bibit and the internal back-office and logistical systems.

10.3.8 Order Modifications and Order Inquiries

Order modification and order inquiry messages are available for orders processed through the Hosted Call Center Application by sending the XML instructions as discussed in previous chapter. As most Hosted Call Center Application merchants will have a fairly low level of integration, the extended functionalities will probably be accessed through the Merchant Interface (<http://www.bibit.com/merchant>).

10.3.9 Risk Management in combination with the Hosted Call Center Application

Due to the nature of interaction between the Bibit system and the call center operator-mediated shopper, risk assessment is done on card level.

10.4 Bibit Swipe Application (P.O.S.)

The Bibit Swipe Application differs from the other models, as this is the only model that can process so-called 'card present' transactions. All other Bibit models of processing transactions are IOMOTO (Internet Order/Mail Order/Telephone Order). The Swipe Application is suitable for all Point-Of-Sale situations where card present transactions need to be processed, such as conference registrations, check-in procedures at hotels or in-shop situations.

10.4.1 Requirements

The merchant needs a Windows NT 4.0, Windows 2000 or Windows XP workstation that can run JAVA applications, has a connection to the Internet and a keyboard that can be used to read the magnetic strip on the credit cards. The application has been pre-configured for a Cherry MY8000 keyboard but can be reconfigured for other keyboards or card readers at Bibit's standard fees.

10.4.2 Installation

This Swipe application comes with its own installation CD-ROM and manual. Merchants should note that installing the Swipe application also installs a new Java Runtime environment on the merchant's workstation that will be used for the Swipe Application. This may influence existing Java applications on the workstation.

10.4.3 Sending an order

To send orders to the Bibit Payment Server the merchant starts up the Bibit Swipe application. The Swipe Application will fire up and connect to the Bibit system through the Internet. An order form will be presented to the operator. The form needs to be filled out and the order can be sent to Bibit by clicking on the 'Send' button. The order is delivered to the Bibit Payment Service and is ready for payment. The operator selects the card the shopper wants to use and swipes the card through the card-reader.

10.4.4 Accepted payment methods

The Bibit Swipe Application is meant for card present transactions, i.e. only card payment methods are possible.

10.4.5 Order code

The order code for each order sent to Bibit has to be unique. Even if a transaction fails at first and is tried a second time, the second try should get a new order code. Each order SENT to Bibit has to be unique.

10.4.6 Order Modifications and Order Inquiries

Order modification and order inquiry messages are available for orders processed through the Bibit Swipe Application by sending the XML instructions as discussed in previous chapter of this manual. As most Bibit Swipe Application merchants will have a fairly low level of integration, the extended functionalities will probably be accessed through the Merchant Interface (<http://www.bibit.com/merchant>).

10.4.7 Risk Management in combination with the Swipe Application

Due to the nature of the communication between the shopper and Bibit (through the Swipe Application) fraud assessment is done on card basis. Due to the fact that the card is physically present for the transaction, the cardholder does not have the right to claim his money back and the merchant is not accountable for charge-backs.

10.4.8 Printing the payment slip

After the transaction has been authorised by the financial institution the operator has the possibility to print a payment slip that can be signed by the shopper/cardholder. This can serve as legal proof of a card present sale.

11 (Corporate) Purchase Cards

Purchase card payments differ from 'normal' credit card payments in a few ways:

- The statements of purchase cards provide a lot more information than regular statements
- The statements of purchase cards are accepted by Inland Revenue Services in many countries for accounting purposes
- Most purchase cards are NOT cleared for on-line transactions. In those cases the payment refusal reason will be 'referred'

11.1 XML orders for Purchase cards.

Orders for purchase cards have to contain extra information required by the acquiring and issuing banks. This can be achieved by including in the XML order a new element, which is called: `branchSpecificExtension`. This Branch Specific Extension (BSE) contains the required extra purchase information.

However, it is not mandatory to have BSE already in the XML order that you submit. It can also be sent later in the XML capture for that order. In all cases, BSE should be complete, for incomplete BSE will result in an invalid order.

Please note that BSE is not a replacement for `orderContent` and/or `description`. If BSE is used in the order XML, these elements must still be used. Bibit sends the BSE to the acquirer at the moment of CAPTURE and/or REFUND.

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService version="1.4" merchantCode="TECHMAN">
  <submit>
    <order orderCode="AY845">
      <description>Computer Products</description>
      <amount value="1982" currencyCode="EUR" exponent="2" />
      <orderContent>
        <![CDATA[
<center><table>
<tr><td bgcolor="#ffff00">Your Internet Order:</td><td colspan="2" bgcolor="#ffff00"
align="right">AY 845</td></tr>
<tr><td bgcolor="#ffff00">Description:</td><td>14 Tulip bulbs</td><td
align="right">1,00</td></tr>
<tr><td colspan="2">Subtotal:</td><td align="right">14,00</td></tr>
<tr><td colspan="2">VAT: 13%</td><td align="right">1,82</td></tr>
<tr><td colspan="2">Shipping and Handling:</td><td align="right">4,00</td></tr>
<tr><td colspan="2" bgcolor="#c0c0c0">Total cost:</td><td bgcolor="#c0c0c0"
align="right">Euro 19,82</td></tr>
<tr><td colspan="3">&nbsp;</td></tr>
<tr><td bgcolor="#ffff00" colspan="3">Your billing address:</td></tr>
<tr><td colspan="3">Mr. Doe,<br>11 Hereortherestreet,<br>1234 KL
Somewhereorother,<br>Thisplace.</td></tr>
<tr><td colspan="3">&nbsp;</td></tr>
<tr><td bgcolor="#ffff00" colspan="3">Your shipping address:</td></tr>
<tr><td colspan="3">Mr. Doe,<br>11 Hereortherestreet,<br>1234 KL
Somewhereorother,<br>Thisplace.</td></tr>
<tr><td colspan="3">&nbsp;</td></tr>
<tr><td bgcolor="#ffff00" colspan="3">Our contact information:</td></tr>
<tr><td colspan="3">ACME Webshops Int. Inc.,<br>11 Strangewood Blv.,<br>1255 KZ
Thisisit,<br>Nowhereatall.<br><br>acmeweb@acme.inc<br>(555) 1235 456</td></tr>
<tr><td colspan="3">&nbsp;</td></tr>
]]>

```

```

<tr><td bgcolor="#c0c0c0" colspan="3">Billing notice:</td></tr>
<tr><td colspan="3">Your payment will be handled by Bibit Global Payments
Services<br>This name may appear on your bank
statement<br>http://www.bibit.com</td></tr>
</table></center>
  ]>
</orderContent>
<paymentDetails>
  <VISA-SSL>
    <cardNumber>4444333322221111</cardNumber>
    <expiryDate><date month="11" year="2004" /></expiryDate>
    <cardHolderName><![CDATA[John Doe]]></cardHolderName>
    <cvc>771</cvc>
  </VISA-SSL>
  <cardAddress>
    <address>
      <firstName>John</firstName>
      <lastName>Doe</lastName>
      <street>11 Hereotherestreet</street>
      <postalCode>1234KL</postalCode>
      <city>Somewhereorother</city>
      <countryCode>TP</countryCode>
      <telephoneNumber>0123456789</telephoneNumber>
    </address>
  </cardAddress>
</paymentDetails>
<branchSpecificExtension>
  <purchase>
    <costCenter><![CDATA[ACME inc]]></costCenter>
    <shopperVATNumber>NL123456789</shopperVATNumber>
    <merchantVATNumber>NL987654321</merchantVATNumber>
    <purchaseCardAccount>1</purchaseCardAccount>
    <item quantity="14" unit="PIECE">
      <description><![CDATA[14 Tulip bulbs ]]></description>
      <commodity code="1209"/>
      <netUnitPrice>
        <amount currencyCode="EUR" value="100" exponent="2"/>
      </netUnitPrice>
      <netDiscount>
        <amount currencyCode="EUR" value="0" exponent="2"/>
      </netDiscount>
      <grossAmount>
        <amount currencyCode="EUR" value="1645" exponent="2"/>
      </grossAmount>
      <VAT percentage="17.5"/>
    </item>
    <freight>
      <departure countryCode="NL" zipCode="1255KZ"/>
      <destination countryCode="NL" zipCode="1234KL"/>
      <netAmount>
        <amount currencyCode="EUR" value="400" exponent="2"/>
      </netAmount>
      <grossAmount>
        <amount currencyCode="EUR" value="400" exponent="2"/>
      </grossAmount>
      <VAT percentage="13.00"/>
    </freight>
  </purchase>
  <total>
    <netAmount>
      <amount currencyCode="EUR" value="1982" exponent="2"/>
    </netAmount>
  </total>

```

```
<netDiscount>
  <amount currencyCode="EUR" value="0" exponent="2"/>
</netDiscount>
<VAT percentage="13.00"/>
</total>
</purchase>
</branchSpecificExtension>
</order>
</submit>
</paymentService>
```

Most new elements are self-explanatory. The element `commodity` has an attribute that corresponds with the TARIC code (Tarif Intégré de la Communauté). More information about this code can be found at:

http://europa.eu.int/comm/taxation_customs/dds/en/tarhome.htm.

It is compulsory to specify the VAT rate and net discount in each `item` element. For information about which fields are mandatory, please check the current DTD at <http://dtd.bibit.com>.

11.1.1 Partial Refunds Purchase card orders

When a partial refund needs to be done on a purchase card order, the merchant will have to send the (adjusted) BSE in the `orderModification`. When a full refund is done the BSE of the original order will be used, this does not need any changes.

11.1.2 Partial Capture Purchase card orders

When a partial capture is performed on a purchase card order the merchant will have to send the (adjusted) BSE in the `orderModification`. Full captures will automatically use the current (existing) BSE for that order.

12 Testing

Testing is an iterative process that usually starts the moment developers start building the application that will interact between the Bibit Payment Service and the merchant's system. All merchants who have a contract with Bibit have access to two environments: the test environment and the production environment. The test environment simulates the most important functionalities of the Bibit Payment Service. The transactions sent to the test environment will not be debited from the cards used.

Though many merchants want to test transactions in the production environment, this can only be done once Bibit has granted the merchant access to this part of the system. Access to the production environment will only be granted after Bibit has performed a so-called 'acceptance test' (see chapter 14).

Please note that tests done in the production environment are treated as valid, transactions, and money will be debited from cards and or bank accounts unless the transactions are cancelled. Testing in the production environment means that you will be charged for all authorisations, refunds, possible charge-backs and extra services you use.

12.1 Test locations

Test orders should be sent to <https://secure-test.bibit.com/jsp/merchant/xml/paymentService.jsp>. The Merchant Interface for the test environment can be found at <http://www.bibit.com/merchant-test>.

The database used in the test environment is regularly cleaned up. Please be aware that as a result all test transaction data is removed.

The Hosted Call Center Application and the Swipe Application do not have a test environment. These two applications are set to the production environment from the start.

12.2 Test credit card numbers

For test purposes only, Bibit has defined a set of test credit card numbers. These can only be used in the test environment.

- VISA 4111 1111 1111 1111 and 4444 3333 2222 1111
- AMEX 3434 343434 34343
- ECMC 5555 5555 5555 4444
- SWITCH 6333333333333333336
- SOLO_GB 67676767676767671

The Switch and SOLO_GB test cards have no issue number but use a start date.

These card numbers should always be used in combination with an expiry date in the future and with a cardholder name that is either "REFUSED", "AUTHORISED", "ERROR", "REFERRED" or "CAPTURED".

Cardholder name "REFUSED" will simulate a refused payment situation whereas "AUTHORISED" will simulate an authorised payment situation, etc.

12.3 Testing ELV transactions

Merchants can also send (German) ELV test transactions to Bibit's test server. The transaction will end in status 'AUTHORISED' by default. Only if the AccountHolderName in the authorisation request contains the value 'REFUSED' the transaction will end in status 'REFUSED'. A preliminary condition is that the payment method ELV must be activated in the production environment for the testing merchant.

12.4 What to test?

This chapter describes what MUST be tested before you are granted access to the production environment, as well as a few additional scenario's that can be simulated on the test environment. Please note that the test environment is not an exact copy of the production environment and therefore not all situations can be simulated.

The capture and refund functionality are available in the test environment, but payments cannot reach the "SETTLED" status in this environment.

12.5 Test scenario's

To test your webshop in the Bibit test environment make sure that the orders are sent to the correct URL of the Bibit test environment (<https://secure-test.bibit.com/jsp/merchant/xml/paymentService.jsp>). Go to your (test-)shop to send transactions to the Bibit test environment. Use the provided test credit card numbers for the test transactions. Please note that not all payment methods are available in the test environment. Only a few credit cards payment methods will be shown despite the settings of the paymentMethodMask. This is due to the fact that not all payment method 'issuers' allow test transactions on their systems.

12.5.1 REFUSED

Place an order in your (test-)shop and send it to the Bibit test environment. In the payment method selection screen select one of the credit cards shown, use the appropriate test credit card number and use "REFUSED" (in capital letters) as cardholder name. Bibit will simulate a situation where the transaction is refused by the financial institutions.

12.5.2 AUTHORISED

Place an order in the your (test-)shop and send it to the Bibit test environment. In the payment method selection screen select one of the credit cards shown, use the appropriate test credit card number and use "AUTHORISED" (in capital letters) as cardholder name. Bibit will simulate a situation where the transaction is authorised by the financial institutions.

12.5.3 ERROR

Place an order in your (test-)shop and send it to the Bibit test environment. In the payment method selection screen select one of the credit cards shown, use the appropriate test credit card number and use "ERROR" (in capital letters) as cardholder name. Bibit will simulate a situation where the transaction ends in error.

12.5.4 REFERRED

Place an order in your (test-)shop and send it to the Bibit test environment. In the payment method selection screen select one of the credit cards shown, use the appropriate test credit card number use "REFERRED" (in capital letters) as cardholder name. Bibit will simulate a situation where the financial institution replies with "REFERRED".

This is only applicable to merchants who have their own accounts with acquiring banks.

12.5.5 CAPTURED

Place an order in your (test-)shop and send it to the Bibit test environment. In the payment method selection screen select one of the credit cards shown, use the appropriate test credit card number use "CAPTURED" (in capital letters) as cardholder name. Bibit will simulate a situation where payment is captured.

12.5.6 CVC2

To be able to test a transaction with the CVC2 check, a check on a security code printed on credit cards (refer to section 15.3 for details), this functionality has to be activated for your account first. Bibit Operations can do this. Once the functionality has been activated, the CVC2 field will be visible in the payment screen in the Redirect service model and can be used.

The following CVC2 scenario's can be tested:

CVC2 code	Simulated situation	Numeric response
Left blank	NOT SUPPLIED BY SHOPPER	1
111	NOT SENT TO ACQUIRER	2
222	NO RESPONSE FROM ACQUIRER	3
333	NO CHECKED BY ACQUIRER	4
444	FAILED	5
555	APPROVED	6

12.5.7 AVS

AVS is short for (Cardholder) Address Verification Service. It provides a means of comparing the address provided by the shopper with the cardholder address registered at the credit card company. As cardholder address is not embossed on the card or encoded on the magnetic stripe of the card, anyone wanting to use a card number to commit fraudulent transactions would have to know the registered address. This provides an extra layer of security.

The AVS functionality has to be switched on for the merchant. Bibit Operations can do this. Unfortunately not all acquirers provide this service so it is currently only available in England (June 2002).

The cardholder address can only be compared if it is provided in the XML order sent by the merchant or if it is entered into the payment details screen when using the Redirect model. The order should include the `cardAddress` element:

```

...
<cardAddress>
  <address>
    <firstName>John</firstName>
    <lastName>Doe</lastName>
    <street>11 Hereortherestreet</street>
    <postalCode>1234KL</postalCode>
    <city>Somewhereorother</city>
    <countryCode>TP</countryCode>
    <telephoneNumber>0123456789</telephoneNumber>
  </address>
</cardAddress>
...

```


The `houseNumber` element can only contain numerical data. House number extensions as in "59A" or "4bis" should be placed in the `houseNumberExtension` element.

In the UK it is common not to have a house number, but just a name, like "Rose cottage". In this case the `houseName` element should be used instead of the `houseNumber` element.

The `countryCode` element should contain a valid ISO-3166 two-character country code; e.g. not "USA" or "GBR" but "US" and "GB".

Below is an example of Bibit's response to an order with AVS.

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="TECHMAN" version="1.4">
  <reply>
    <orderStatus orderCode="1234">
      <payment>
        <paymentMethod>ECMC-SSL</paymentMethod>
        <amount currencyCode="GBP" debitCreditIndicator="debit" exponent="2"
          value="6200" />
        <lastEvent>AUTHORISED</lastEvent>
        <AVSResultCode description="APPROVED"/>
        <balance accountType="IN_PROCESS_AUTHORISED">
          <amount currencyCode="GBP" debitCreditIndicator="credit" exponent="2"
            value="6200" />
        </balance>
        <cardNumber>5555*****4444</cardNumber>
      </payment>
    </orderStatus>
  </reply>
</paymentService>
```

For test purposes the following postal codes can be used in the test environment.

Postal code	Response
Left empty	NOT SUPPLIED BY SHOPPER
1111	NOT SENT TO ACQUIRER
2222	NO RESPONSE FROM ACQUIRER
3333	NO CHECKED BY ACQUIRER
4444	FAILED
5555	APPROVED

12.6 Reporting

The reporting structure in the test environment is identical to the reporting structure in the production environment. This means that you can receive order reports, email confirmation messages and HTTP(S) confirmation messages. Please note that, as in the production environment, the HTTP(S) confirmations have to be answered with an [OK] reply to Bibit.

Because no real transactions are processed in the test environment, no transfer reports can be sent.

12.7 Testing in the production environment

As not every situation can be simulated in the test environment, you may want to test the integration with your system on the Bibit production environment. Though this is no problem there are a few things you should know:

- Before being granted access to the Bibit production environment, Bibit Operations will perform an acceptance test to check whether the merchant's system complies to the specifications as described in this manual.
- Payments processed in the production environment will be routed to the appropriate financial institutions and money **will** be deducted from the credit card or bank account. This can be avoided by cancelling the order before it reaches captured status. Some payment methods will reach captured status immediately and cannot be cancelled. Those payments can be refunded.
- Bibit will charge the merchant for services rendered. Normal authorisation, refund, charge-back and other fees will be invoiced for all transactions done in the production environment.

13 Troubleshooting

This chapter lists examples of possible errors that can be encountered when sending orders, order modifications or order inquiries, as well as suggested solutions. Please note that the exact format and wording are subject to change without notice.

There are eight types of errors:

1. internal error in the Bibit Payment Service
2. parse error, invalid xml
3. invalid number of transactions in batch
4. security error
5. invalid request
6. invalid content, occurs when xml is valid but content of xml not
7. payment details in the order element are incorrect
8. service temporarily not available

13.1 Merchant not active

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="MERCHANTCODE" version="1.4">
  <reply>
    <error code="4"><![CDATA[Merchant not active.]]></error>
  </reply>
</paymentService>
```

This error means that you don't have access to the environment (test or production) you are trying to send the order to. If you are still in the testing phase it is most likely that the order was sent to the production environment, which is only accessible after an acceptance test by Bibit. If you sent the order to the test environment and received this reply, you should contact Bibit to open access to the test environment. If you are ready to go live and want Bibit to do an acceptance test, you should contact Bibit support. Please note that during the acceptance test the orders still need to be sent to the test environment.

13.2 IP check failed

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="MERCHANTCODE" version="1.4">
  <reply>
    <error code="4"><![CDATA[IP check failed. Access denied.]]></error>
  </reply>
</paymentService>
```

This error indicates that the IP address you sent the XML order from is not identical to the IP address Bibit has registered for you. Bibit checks on IP addresses within a C-range. This means that the last octet of the registered IP number can be anything, enabling you to have several (test-) machines per Bibit account to work from. If the IP address you have supplied to Bibit is the IP address of the machine that actually sends the XML orders, and you

still get this error, you could check with your System or Network Administrator whether a router or a firewall is masking the machine's real IP address and replacing it by another IP address. If so, Bibit will need that IP address for its checks.

13.3 Access denied

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="MERCHANTCODE" version="1.4">
  <reply>
    <error code="4"><![CDATA[Security violation. Access denied.]]></error>
  </reply>
</paymentService>
```

This error indicates that the username/XML password combination is not correct. The username is always your merchant code and always in capitals. Please note that anyone within your organisation who has access to the Merchant Interface can change the XML password. Bibit does not have access to your passwords and can only assign new passwords, rendering the current password inoperable.

13.4 Missing doc type declaration

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="MERCHANTCODE" version="1.4">
  <reply>
    <error code="2"><![CDATA[Missing DOCTYPE declaration.]]></error>
  </reply>
</paymentService>
```

This error indicates that the XML data sent to Bibit does not contain the required DOCTYPE declaration. The doctype declaration is used by Bibit to determine what kind of information to expect. The DOCTYPE declaration must be:

```
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
```

13.5 Empty body in message

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="MERCHANTCODE" version="1.4">
  <reply>
    <error code="2"><![CDATA[Empty body in message.]]></error>
  </reply>
</paymentService>
```

This error indicates that you posted an XML message to Bibit with an empty message body. Another possibility is that 'Content-length' in the HTTP headers was set incorrectly (too few characters specified). A third possibility is that you tried to use the HTTP GET method instead of POST to send the request to Bibit.

13.6 Missing quote characters

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
    "http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="MERCHANTCODE" version="1.4">
  <reply>
    <error code="2">
      <![CDATA[The value of attribute "code" must begin with either a single or double
        quote character.]]>
    </error>
  </reply>
</paymentService>
```

The values for `orderCode`, `value`, `currencyCode` and `exponent` elements need to be between either single or double quotes. Each XML message can contain only one type of quotes.

13.7 No description

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
    "http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="MERCHANTCODE" version="1.4">
  <reply>
    <error code="2"><![CDATA[No description for XMLOrder]]></error>
  </reply>
</paymentService>
```

This error indicates that the `description` element was empty. A `description` is mandatory and can contain a maximum of 50 characters.

13.8 Duplicate order

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
    "http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="MERCHANTCODE" version="1.4">
  <reply>
    <orderStatus orderCode="123456">
      <error code="5"><![CDATA[Duplicate Order]]></error>
    </orderStatus>
  </reply>
</paymentService>
```

This error indicates that you tried to send an order with `orderCode` 123456 to Bibit, while this `orderCode` already exists for you in the Bibit database. The `orderCode` has to be unique per merchant. If an order has been

successfully submitted to the Bibit Payment Service, it will not accept a new order with the same `orderCode` for the same merchant. The Bibit Payment Service does not look at the payment status of the previous, existing order. An easy way to make the `orderCode` unique is to use a date/time-stamp, an incrementing number or a combination of both.

13.9 Maximum transaction amount exceeded

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
    "http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="MERCHANTCODE" version="1.4">
  <reply>
    <error code="2"><![CDATA[Your contract does not allow payments of EUR
    1.620,95]]></error>
  </reply>
</paymentService>
```

This error indicates that the transaction amount for the submitted order is too high to be processed on your account. Each merchant contract type has a specific maximum transaction amount. The standard amounts are EUR 450,- for Standard contracts and EUR 1000,- for Plus contracts.

13.10 Unsupported currency

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
    "http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="MERCHANTCODE" version="1.4">
  <reply>
    <error code="2"><![CDATA[Currency USD is not supported for your contract
    type]]></error>
  </reply>
</paymentService>
```

This reply indicates that the currency stated in the order (in this case US Dollars) is not supported by the contract type you are using.

13.11 Internal Error

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
    "http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="MERCHANTCODE" version="1.4">
  <reply>
    <orderStatus orderCode="9008050210213990">
      <error code="1"><![CDATA[Internal error]]></error>
    </orderStatus>
  </reply>
</paymentService>
```

Internal errors originate from the Bibit system itself and can be caused by a number of things. When you receive this error reply it is best to retry after a short period. When a serious internal error occurs, Bibit's technical staff is informed automatically and the problem will be corrected.

13.12 NMTOKEN

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="MERCHANTCODE" version="1.4">
  <reply>
    <error code="2"><![CDATA[Attribute value currencyCode = "NLG GBP" of type NMTOKEN
      must be a name token.]]></error>
  </reply>
</paymentService>
```

This error reply indicates that the value of an XML attribute contains a space.

13.13 Service temporarily not available

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService version="1.4" merchantCode="TECHMAN">
  <reply>
    <error code="8">
      <![CDATA[Service temporarily not available. Please try again later]]>
    </error>
  </reply>
</paymentService>
```

This error reply indicates that (a part of) the Bibit Payment Service is temporarily not available. When you receive this error reply it is best to retry after a short period.

13.14 Duplicate batch

This error means that a batch with the same code (`merchantBatchCode`) already exists in the Bibit system. Every batch you send to Bibit must have a unique `merchantBatchCode`. Even if a batch of transactions hasn't yet been processed and it is sent to Bibit a second time, it should have a different `merchantBatchCode`, even if the transactions in the batch are exactly the same as during a previous attempt.

13.15 Reduced number of payment methods

If the payment selection screen doesn't display all payment methods you expect there are three possible reasons.

- You are using the Bibit test environment. Only a few credit card payment methods are available because not all payment methods have test functionality.
- You are using the production environment, but the transaction amount exceeds the maximum transaction amount stated in your contract. In this case only non-reversible payment methods will be shown.

- The payment method selection mask or preferred payment method hasn't been configured correctly on your system.

14 How to go live?

After you have tested your application and the interaction with the Bibit Payment Service successfully, you can start using the Bibit production environment.

The Hosted Call Center Application and the Swipe Application are set to 'live' during the set-up of the account. If you use one of these models you are connected to the production environment from the start and can therefore skip this chapter.

14.1 Bibit acceptance test procedure

Before you are allowed to go live on the production environment, Bibit Operations will test your application from Bibit's perspective. This means that Bibit staff will visit your webshop/call center (Hosted Call Center Application and the Swipe Application are exempt from this test) and perform a number of test transactions.

Please note that during the acceptance test, the transactions must be sent to Bibit's test environment.

14.1.1 Applying for the test

When you are ready for it, you request for an acceptance test you by creating an incident in the incident console (<http://support.bibit.com>). In this request you should provide your merchant code, the URL where Bibit staff can test your application and, if necessary, a password and login for your application. If shoppers have to register at your site in order to be able to order goods or services, you should provide Bibit with a username and a password of a pre-existing shopper in your system. This way Bibit can perform the test without delay. Bibit staff will not fill out customer information/registration forms during acceptance tests. If necessary, provide detailed information on how to order goods or products and how to get to the Bibit payment method selection screen.

14.1.2 What does Bibit test?

During the acceptance test Bibit staff will check the following items:

- Does a secure/insecure message pop up during redirection?
- Does the `orderContent` have all the required elements? (refer to section 4.2.2 for details)
- Does the merchant send the orders in XML format?
- In case the merchant uses http(s) confirmations, does Bibit receive the proper [OK] reply?
- Is the shopper correctly informed in case of a refused transaction?
- Is the shopper correctly informed in case of an authorised transaction?

Bibit performs these tests by placing two test transactions in the shopping application of the merchant, one simulating the REFUSED situation and one simulating the AUTHORISED situation. Bibit will check the feedback to the shopper via the web browser and check the log files to see if the interaction between the merchants system, on system level, takes place according to the Bibit specifications.

14.1.3 How long will the test take?

The execution of an acceptance tests usually takes about 15 minutes. Please inform us at least a day in advance so we can plan the tests. During the test we may find things that need adjusting before you can go live. You should take into account the time it might take to implement these adjustments.

14.1.4 When can you go live?

The testing process is iterative. If Bibit staff notices during the tests anything that is not compliant to the specifications described in this manual, you will be informed about this in a reply to the original incident where you requested the acceptance test. Bibit will inform you about the findings and request you to correct your application. After you have made the required changes you should reply to the message sent by Bibit to inform Bibit that another acceptance test can be done.

Please note that the test environment is available to you during the contract period even after you have gone live.

14.1.5 Directing payments to the production environment

When all tests are positive, Bibit Operations will give you access to the production environment. Once you have been granted access you can start sending live transactions to the production environment.

15 Risk Management

This section briefly touches the issues of risks involved in accepting credit card transactions over the Internet or call centers. Please make sure to read the separate white paper about risk management and the product sheet for the Risk Management Module.

15.1 What type of risks does a merchant run?

Though only an extremely small percentage of all transactions processed through the Bibit system turns out to be fraudulent, it is important for you to realise that accepting credit cards implicitly means accepting a risk. The risk lies in the fact that in all so-called "card not present", or MOTO (Mail Order/Telephone Order) transactions, the cardholder has the right to deny a transaction. Since the card was not present at the time of the transaction, there is no cardholder signature, and the credit card companies will hold the merchant liable for any fraud. All remote sales from webshops, call centres etc. are considered MOTO transactions by the credit card companies.

As the card was not present at the time of the transaction and no signature of the cardholder is available, it is very difficult for you to prove that the actual cardholder made the transaction. The card number and expiry date may have fallen into the hands of fraudsters who use this information to purchase goods and services on the net.

Certain types of goods and services are more attractive to fraudsters than others. Very attractive are expensive and easily tradable goods (game consoles, software, cameras, computer hardware, etc). Certain services that represent money or have some kind of spending power are also attractive (pre paid telephone cards, credits for online services etc.). If you sell these types of goods you should be aware of the risks and of the measures you can take to minimize the risk.

When a cardholder discovers unauthorised transactions on his statement he can contact his credit card company, deny the transaction and request his money back. The card company will almost always assume the cardholder is telling the truth, as there was no card present, nor was there a signature from the cardholder. The card company will refund the money to the cardholder and charge it back to the merchant. This is called a charge-back. The credit card companies, not Bibit, initiate the charge-backs; Bibit only executes them. Whenever Bibit receives a warning about a pending charge-back the merchant will be notified.

Not all charge-backs are due to fraudulent transactions. Sometimes the fact that the name Bibit Internet Payments appears on a credit card statement causes confusion and the shopper contacts his card company. By communicating this fact in advance to the shopper the charge-back risk can very easily be reduced.

Extensive information on Risk Management and how to implement it in your organisation can be found in Bibit's white paper 'Risk Management' (<http://www.bibit.com/documentation.shtml>)

15.2 Risk Management Module

To minimize your risk as a merchant, Bibit offers a separate service called 'Risk Management Module'. This module provides you with an easily configurable tool to reduce the risk of fraudulent transactions. As with all 'security systems' this module is no guarantee against fraud but it has proven to reduce the risk dramatically.

More information about the Risk Management Module can be found in a separate product sheet available at the Bibit documentation website (<http://documentation.bibit.com>).

15.3 CVC2

Credit card companies have realised the fact that cards are being used for fraudulent transactions. In order to protect their clients from fraud (and thus safeguarding their profits) they are now introducing measures to reduce

the risk of fraud. One of them is the CVC2 code (Card Verification Code). This code is NOT embossed on the card nor is it stored on the magnetic stripe. Most cards have it printed on the back of the card (the last three digits of the large number on the signature strip), some have it printed on the front of the card (AMEX) and use four digits.

The shopper has the possibility to enter the CVC2 code on the payment pages in the secured Bibit environment. This functionality can be switched on on a per merchant basis. If you want to benefit from the CVC2 code in the redirect model you should ask Bibit to activate it for you. If you are using the XML Direct model the CVC2 code has to be provided by you in the XML message.

Below is an example of an XML order in combination with a CVC2 code:

```
<?xml version="1.0"?>
<!DOCTYPE paymentService SYSTEM "http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="TECHMAN" version="1.4">
  <submit>
    <order orderCode="1234">
      <description>Meaningful description goes here.</description>
      <amount value="4000" exponent="2" currencyCode="EUR"
        debitCreditIndicator="credit" />
      <orderContent>HTML order content goes here</orderContent>
      <paymentDetails action="AUTHORISE">
        <VISA-SSL>
          <cardNumber>4512123412341234</cardNumber>
          <expiryDate>
            <date month="09" year="2003" />
          </expiryDate>
          <cardHolderName>TEST, Mr.</cardHolderName>
          <cvc>123</cvc>
        </VISA-SSL>
      </paymentDetails>
    </order>
  </submit>
</paymentService>
```

This XML Direct model message will submit an order with the orderCode "1234" for merchant "TECHMAN" for the amount of EUR 40,00 using a VISA card with the card number 4512123412341234, expiry date 09/2003 for cardholder "TEST, Mr," and CVC2 code "123".

Bibit's response message will be:

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//Bibit//DTD Bibit PaymentService v1//EN"
  "http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService merchantCode="TECHMAN" version="1.4">
  <reply>
    <orderStatus orderCode="1234">
      <payment>
        <paymentMethod>VISA-SSL</paymentMethod>
        <amount currencyCode="EUR" debitCreditIndicator="credit" exponent="2"
          value="4000" />
        <lastEvent>AUTHORISED</lastEvent>
        <CVCTestResultCode description="APPROVED" />
        <balance accountType="IN_PROCESS_AUTHORISED">
          <amount currencyCode="EUR" debitCreditIndicator="credit" exponent="2"
            value="4000" />
        </balance>
        <cardNumber>4512*****1234</cardNumber>
      </payment>
    </orderStatus>
  </reply>
</paymentService>
```

Please note that Bibit has this functionality fully integrated in its system, but not all financial institutions have. Even if you provide Bibit with the CVC2 code, in some cases it is not actually checked by the financial institution involved. This situation will occur less frequently in the future as the credit card companies have planned to make this test mandatory for all financial institutions involved in processing credit card transactions.

15.3.1 Incorrect CVC2 code resulting in authorised payment

Even if the acquiring bank checks the CVC2 code, an incorrect CVC2 code will not result in a refused payment. This check, as well as the Address Verification Service (AVS) check is separated from the authorisation process. An incorrect CVC2 code can mean one of two things:

- 1) The shopper does not have the physical card available
- 2) The shopper does not know what a CVC2 code is.

It is up to the merchant to decide whether or not to go through with the order, based on the result of the CVC2 check.

15.4 Verified by VISA (formerly 3-D Secure)

This is an initiative by VISA to provide more security during card not present transactions. The cardholder registers a password with VISA and uses this (secret) password as a signature at the time of the transaction. VISA then can verify whether it was the actual cardholder who initiated the payment.

The new technology will become available to all of Bibit's merchants in Germany, Austria, Switzerland and the Benelux that accept VISA payments online. Citibank and Bank of Ireland were the first acquiring banks in Europe to support Verified by VISA - Visa's new authentication method. Visa is offering protection for online retailers who adopt Verified by VISA technology.

15.5 AVS

AVS is short for (Cardholder) Address Verification Service. It provides a means of comparing the address provided by the shopper with the cardholder address registered at the credit card company. As cardholder addresses are not embossed in the cards or coded in the magnetic strip of the card, anyone who wants to use a card number to commit fraudulent transactions would have to know this address. This provides an extra layer of security.

The AVS functionality can be switched on for the merchant by Bibit staff. Unfortunately not all the acquirers provide this service, so it is currently only available in England (June 2002).

15.5.1 Failed AVS check resulting in authorised payment

As it is the case with the CVC2 check, it is up to the merchant to decide whether or not to continue with the order if the AVS check fails.

15.6 Risk reporting methods

The Risk Score of each payment is not yet communicated through XML reply messages sent by Bibit. The risk score can be viewed through the Merchant Interface (<http://www.bibit.com/merchant>).

16 Order Fulfilment

After the payment has been processed, the ordered products and or services have to be delivered to the consumer.

16.1 When to deliver goods or services to the customer?

It is more or less up to you to decide when to send the goods or start delivering the services ordered. However, you should note that in certain countries there are specific laws governing this issue. You should also be aware that credit card companies *dictate* that goods and/or services must be delivered when the payment reaches the status AUTHORISED.

16.2 How to cancel an order?

You can choose to cancel an order, or rather an authorisation for a capture. This can be done in the Merchant Interface (<http://www.bibit.com/merchant>) or by sending an XML cancel message as described in section 8.1.

16.3 Capture and partial capture

The capture delay of the payment can be set to a moment between 0 and 14 days after a payment reaches the status authorised; default is 5 days. If you want to capture the payment **before** the capture delay set for his account has expired, you can use the capture button for this transaction in the Merchant Interface or send an XML capture message as described in a section 8.1.

If only a part of the goods or services can be delivered, you may choose to capture only the corresponding amount from the credit card. This way you don't have to perform a refund at a later stage, which is cheaper for you and more pleasant for the shopper. You can do a partial capture through the Merchant Interface by filling out the amount to be captured and clicking on the "capture now" button or by sending an XML message as described in a previous section of this manual. Only one (partial) capture is possible per authorisation. Please note that after a partial capture has been done it is no longer possible to capture the remaining amount.

16.4 How to do a refund?

There are several ways to do a refund. Depending on the type of transaction, you can use the following methods:

- Use the Merchant Interface or send an XML message for regular credit card transactions.
- Send a refund request to Bibit for payments per bank transfer.
- Payments per ELV can be refunded by sending a refund request to Bibit
- Payments per SET methods can be refunded by sending a refund request to Bibit

Refund requests can be sent through the incident console at <http://support.bibit.com> and should contain the merchant code, the order code, reference id or payment id, shopper name, and the amount to be refunded. For SET and ELV refunds, the banking details of the shopper should also be submitted. Due to the particularities of these payment methods Bibit does not have access to the banking information of the shopper.

An example of an XML refund message (order modification) can be found in section 8.1.

17 Payments to the Merchants

Though this manual is a technical manual, a few points about merchant payments will be addressed. More information about the financial process can be found in Bibit's Administrative manual.

17.1 What does Bibit do with the money?

It is not Bibit who receives the money from the merchant's customers. The money that is processed for the merchants is held by an independent foundation (Stichting Derdegelden Bibit Internet Payments or Stichting Trusted Third Party Bibit Internet Payments). Bibit uses this foundation to make sure that no money from the merchants can ever be appropriated for general business purposes; received money can always be paid out to the merchants. The only things Bibit does with the money is to receive it, to reconcile it with orders and to transfer it to the appropriate merchants.

17.2 Accepting multiple settlement currencies

A merchant account on the Bibit Payment Service can only settle in one currency. This means that the merchant has to indicate in which currency he would like to receive the revenue generated through his account through the Bibit Payment Service. Multi-national organisations wanting to settle in multiple currencies have to set-up an account for each settlement currency.

17.3 Why doesn't this order receive a payment status?

As explained earlier in this manual not all orders receive a payment status, due to the process flow of the Bibit Payment Service. When a shopper has filled his shopping basket on the site and clicks to pay for the goods or services, the merchant's system will send all information (the order) to the Bibit Payment Service. The shopper can then decide not to continue, thus a payment status will not be reached for that order.

17.4 Why did the shopper have to pay a higher amount than the merchant originally charged?

The Bibit Payment Service only debits the shopper's card or account for the amount the merchant sets in the order information send to the system. However, some credit-card companies and banks have hidden costs for their customers. These are so called Cross Border Surcharges.⁴ These surcharges may vary per Card brand or banking institution and can change over time. The Bibit Payment Service does NOT impose these fees! Shoppers should contact their bank for more information about these charges. Currently the EU is working to out-law this practise within its member states.

17.5 Dispute management

Whenever a cardholder requests information from his card company about a transaction processed by the Bibit Payment Service, the merchant will be notified *whenever this is possible*. These transactions will appear in the 'Dispute management' function of the Merchant Interface. This functionality provides the merchant with a way to

⁴ Some extra information can be found at http://europeforvisitors.com/europe/articles/credit_card_surcharges.htm

communicate with the acquiring bank about information request. The merchant can enter free text in the web form presented on the screen and can attach one (1) file to the communication. If the merchant wants to add more files they should be bundled into one zipped file. The merchant should provide the acquiring bank with as much proof as possible that the person ordering the goods was actually the cardholder himself and that the goods or services have been delivered. When the merchant sends off this message the Bibit Payment Service will automatically attach extra information needed by the acquiring bank. The information will be signed and encrypted before it is sent to the acquiring bank. Please note that a merchant can use this function only once per disputed transaction.

17.6 When to expect payments from Bibit?

It is important to realise that the money Bibit receives for its merchants is not accessible to the corporate business entity Bibit. Nor is it within Bibit's business model to make money from the interest gained on money received on behalf of the merchants. Bibit will always do everything possible to pay out the money to its merchants as soon as possible. This is done twice a week.

As not all payment methods can be automated to the same extent small fluctuations in the pay out frequency may occur. This also depends on other links within the chain of financial institutions used to process payments.

It is important to realise that some payment methods can be settled within a smaller timeframe than other payment methods, this is due to the nature of the payment method. Where one payment done on the same day as another may be settled within 3 days, the other payment may be settled in three weeks.

18 Account Types and Financial Reporting

Payment information, that represents money, flows through different accounts that are part of the financial structure of the Bibit Payment Service. This flow can be followed through the <journal> information in the XML messages the Bibit Payment Service sends to the merchant's system.

The different account types that are used in the journals are described below.

18.1 Account types

Account Type	Description
IN_PROCESS_AUTHORISED	This account contains the money that is reserved for the merchant by the financial institutions. This amount is the amount that can (still) be captured. For more information about capture, see section 6.1.4 of this manual. When an amount is not captured within 14 days the authorisation will expire, see section 6.1.12 of this manual.
IN_PROCESS_CAPTURED	This amount is captured with the financial institutions on behalf of the merchant. It will be settled in the merchant's SETTLED_BIBIT_NET and SETTLED_BIBIT_COMMISSION accounts, which are described below.
SETTLED_BIBIT_NET	The account contains the net amount that will be transferred to the merchant on a regular basis (twice a week). The actual amount transferred is the balance of this account, MINUS the amount needed to keep the DEPOSIT account on the required level.
SETTLED_BIBIT_COMMISSION	The account contains the commission, or MSC, that is withheld from the merchant at the time of settlement.
DEPOSIT	This account contains the merchant's deposit that is used for possible charge-backs. The money in this account is kept on a preset level. The 'profile' page in the Merchant Interface shows the exact amount that is kept in deposit.
PAYOUT	Special account type, outside the scope of this manual.

Below you find an example of a journal for a payment that has reached the status 'settled'.

```
<journal journalType="SETTLED" note="M" description=" ">
  <bookingDate>
    <date dayOfMonth="01" month="08" year="2002"/>
  </bookingDate>
  <accountTx accountType="SETTLED_BIBIT_COMMISSION">
    <amount value="560" currencyCode="EUR" exponent="2" debitCreditIndicator="credit"/>
  </accountTx>
  <accountTx accountType="SETTLED_BIBIT_NET" batchId="182">
    <amount value="9412" currencyCode="EUR" exponent="2" debitCreditIndicator="credit"/>
  </accountTx>
  <accountTx accountType="IN_PROCESS_CAPTURED">
    <amount value="9972" currencyCode="EUR" exponent="2" debitCreditIndicator="debit"/>
  </accountTx>
</journal>
```

The third accountTx element:

```
<accountTx accountType="IN_PROCESS_CAPTURED">
  <amount value="9972" currencyCode="EUR" exponent="2" debitCreditIndicator="debit"/>
</accountTx>
```

indicates this account contained EUR 99.72 debit.

The second accountTx element:

```
<accountTx accountType="SETTLED_BIBIT_NET" batchId="182">
  <amount value="9412" currencyCode="EUR" exponent="2" debitCreditIndicator="credit"/>
</accountTx>
```

indicates this account contains EUR 94.12 credit.

The first accountTx element:

```
<accountTx accountType="SETTLED_BIBIT_COMMISSION">
  <amount value="560" currencyCode="EUR" exponent="2" debitCreditIndicator="credit"/>
</accountTx>
```

indicates this account contains EUR 5.60 credit.

The amount EUR 99.72 has been transferred to two accounts. An amount of EUR 94.12 has been transferred to the SETTLED_BIBIT_NET account and an amount of EUR 5.60 has been transferred to the SETTLED_BIBIT_COMMISSION account.

Using these messages, the merchant can keep track of how the money flows through the Bibit Payment Service. A journal follows general accounting rules and is therefore always balanced.

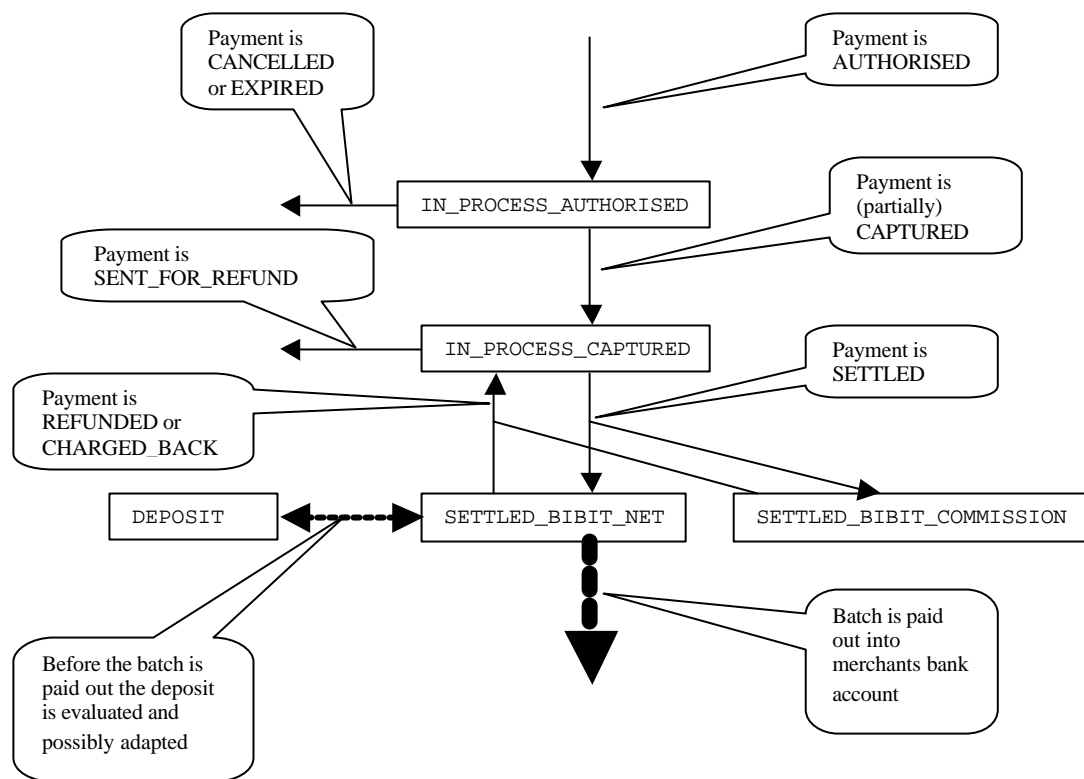


Figure 4 Overview of the flow of money through the different accounts in the Bibit Payment Service.

Pre version 1.4 (DTD)

In the Bibit Payment Service specifications prior to version 1.4 the accounts were organised differently. In the past the system used one account IN_PROCESS for both the authorised amounts (now transferred to IN_PROCESS_AUTHORISED) and the captured amounts (now transferred to IN_PROCESS_CAPTURED).

In version 1.4 of the Bibit Payment Service it is easier for you to keep track of how the money travels through the system. To make your system compatible to version 1.4, make sure it is able to interpret the account types as described in this section correctly. Once your system is ready to interpret these messages you should make the following change to the merchant environment properties in the Merchant Interface (<http://www.bibit.com/merchant>). Go to the 'Profile' page, click on the 'Environment properties' button at the bottom of the page. In 'Production specific properties' and 'Test specific properties' set 'Production XML reply/orderStatusEvent version' and 'Test production XML reply/orderStatusEvent version' to 'Standard latest XML version' for the production and test environment respectively.

18.2 Financial reporting

The Bibit Payment Service can provide you with four types of reports to be used for financial purposes. These reports are the Order report, the Transfer report, the Capture report and the Event report.

18.2.1 Order report

This daily report lists all payments that have been authorised that day or payments that have changed status that day.

18.2.2 Batch reports

There are two Batch report types: IN_PROCESS_AUTHORISED and IN_PROCESS_CAPTURED. These are daily financial reports that list all transactions that are booked on these accounts. In other words: a report on all authorised transactions ready to be captured, and a report on all captured transactions ready to be settled.

18.2.3 Transfer report

This report is sent to you every time a payment batch has been closed for you and money is being transferred to your bank account. This happens twice a week. This report shows all the settled, refunded and charged-back transactions of that period. It also specifies the financial details of these orders. If you use so-called back office codes these will be shown, next to the order codes, in this report.

18.2.4 Capture report

This daily report lists all the captured transactions. This report is available for selected merchants only.

Reports can be sent through email and/or http(s) in different formats. You should indicate which channel you want to use for each report and in which format the report should be sent. This can be done through the Merchant Interface.

19 Merchant account information

This section gives an overview of all the information Bibit needs to set-up a working account for a merchant.

Item	Issuer/Set by/format	Description
Merchant Code	Issued by Bibit, all capital letters.	The code the Bibit Payment Service uses to uniquely identify a merchant.
Administration Code	Issued by Bibit to merchants who have several accounts. All capital letters.	Code used to group multiple accounts from one particular merchant.
Password	Issued by Bibit, at least eight characters, at least 3 numbers in the first 8 characters.	<p>Password that in combination with the merchant code gives access to the Merchant Interface.</p> <p>You can change this password through the Merchant Interface. Please note that there is a time delay of about 15 minutes before the password changes take effect.</p>
XML Password	Set by the merchant in the Merchant Interface, at least 3 numbers in the first 8 characters.	<p>Password that in combination with the merchant code gives access to the Bibit Payment Service to deliver orders in XML.</p> <p>You can change this password through the Merchant Interface. Please note that there is a time delay of about 15 minutes before the password changes take effect.</p>
AVS	Set by Bibit, yes or no. This option is not available for all acquirers and therefore not for all merchants.	AVS is only available in a few countries.
CVC	Set by Bibit, yes or no.	Activating CVC will generate a CVC2 data entry field in the payment details screen for the shopper. However, you should realize that not all acquirers check the code.
Card deposit	Set by Bibit, yes or no. This option is available to selected merchants only.	Card deposit allows merchants to pay out money to shoppers.
Supports referral	Set by Bibit, yes or no. This option is only available for merchants who use their own accounts with acquirers who allow sending of authorisation codes.	In case a payment gets the refusal reason 'referred' the merchant can contact his acquirer for an authorisation code. This code can then be sent to the acquirer for authorisation.

Risk Management Module	Set by Bibit, yes or no. Merchants are charged extra for this option.	Setting this to 'yes' will give the merchant access to the Risk Management Module, allowing him to change the settings of the checks.
Pay Out module	Set by Bibit, yes or no. Available for selected merchants only.	Selected merchant will be able to pay out money to their shoppers' credit cards.
Allow Split Payments	Set by Bibit, default is no. Available for selected merchants only.	Split payments enable a merchant to use the 'pay as order' or 'recurring payments' functionality.
Shop name	Set by Bibit.	This name will be sent through the financial network for printing on the statements of the shopper.
Shop Country	Set by Bibit.	Indicates the location of the merchant.
Shop URL	Set by Bibit. Indicates the location of the online shop of the merchant.	To be sent through the financial network for printing on the statements of the shopper.
Default customer interaction	Set by Bibit. E-commerce is standard for web shops; MOTO is standard for call centers.	Determines what types of transactions are possible in the account.
Bibit (pay-out) Bank account	Set by Bibit (reconciliation).	Determines which bank account Bibit uses to payout this merchant. This setting can be changed by reconciliation only.
Settlement Currency Code	Set by Bibit, determined by merchant.	Determines the currency in which the merchant is paid out.
Deposit amount	Set by Bibit.	The amount kept in deposit to safeguard Bibit Foundation from costs occurring after charge-backs and other financial risks.
Capture delay	Set by Bibit, determined by merchant. Between 0 and 14 days, or no auto-capture.	The number of days between the authorisation and the automatic capture of payments. The value 365 indicates no automatic capture.
HTTPS proxy	Set by Bibit, yes or no.	Default is 'yes'. If a merchant has a secure environment where the result URLs reside, this can be changed to 'no'.
Country language selection	Set by Bibit, yes or no.	Default is 'yes'. Determines whether the country and language selection menu is shown to the shoppers at the bottom of the payment method selection screen.
Continue button	Set by Bibit, yes or no.	Default is 'yes'. Determines whether a 'continue' button will be shown after selecting an off-line payment method.

Shopper e-mail confirmation	Set by Bibit, determined by merchant, yes or no.	Default is 'no'. Determines whether the Bibit Payment Service will send the shopper an email after the payment has reached the REFUSED or AUTHORISED status.
Shopper e-mail content type	Set by Bibit, determined by merchant.	Determines whether the content type of the shopper email will be 'text' or 'text/plain'.
Maximum transaction amount	Set by Bibit, determined by contract type of merchant.	Default amount for 'standard' is Euro 450 and for 'plus' Euro 1000. If the transaction amount exceeds this amount only SET payment methods will be shown.
Call center interface used	Set by Bibit, determined by contract type of the merchant.	Default is 'no'. When set to 'yes' the merchant can use the Hosted Call Center Application.
Call Center phone number	Set by Bibit, determined by merchant.	Contains the phone number of the call center. This number is sent through the financial institutions for printing on the statements.
Payment Methods	Set by Bibit, determined by merchant.	When using the Hosted Call Center Application the payment method mask has to be filled out in this field. Included payment methods should be separated by the '^'-sign.
Accepted currencies	Set by Bibit, determined by merchant.	When using the Hosted Call Center Application, the accepted currencies have to be filled out in this field. Accepted currencies should be separated by the '^'-sign.
Call center country	Set by Bibit, determined by location of the merchant.	Determines the standard country specific payment methods for the call center operators.
Call center language code	Set by Bibit, determined by location of the merchant.	Determines the standard language of the Bibit interface for the Hosted Call Center Application.
Call center generates order code	Set by Bibit, determined by merchant, yes or no.	Determines whether the Bibit Payment Service will generate unique order codes for the merchant using the Hosted Call Center Application.
Technical Contact person name	Set by Bibit, provided by merchant.	The person Bibit can contact about technical issues regarding the account.
Technical Contact person phone number	Set by Bibit, provided by merchant.	The telephone number of the technical contact person.
Technical Contact person e-mail	Set by Bibit, provided by merchant.	The email address of the technical contact person.

Contract type code	Set by Bibit, determined by contract type.	Determines the billable amount for the account.
Contract sign date	Set by Bibit.	Indicates the sign date of the contract.
Contract start date	Set by Bibit, determined by contract.	Indicates the start date of the contract.
Contract end date	Set by Bibit, determined by contract / merchant.	Indicates the end date of the contract. If 1st of June 2002 is the end date, the entry should be 2002-05. Determines whether or not invoices are sent to the merchant.
Invoice Contract disabled	Set by Bibit, yes or no.	Determines whether or not invoices are sent to the merchant. No longer in use.
Specified invoices	Set by Bibit, yes or no.	Default is 'no'. Indicates whether the merchant will receive specified invoices. A separate fee will be charged for this option.
Invoice name	Set by Bibit, determined by the contract.	The name of the merchant's legal entity that will pay the invoice.
Invoice attention name	Set by Bibit, determined by the merchant.	The name of the person the invoice is addressed to.
Invoice address 1	Set by Bibit, determined by the merchant.	First part of the invoice address.
Invoice address 2	Set by Bibit, determined by the merchant.	Second part of the invoice address, country can be left out.
Telephone number Financial contact	Set by Bibit, determined by the merchant.	Telephone number of the person Bibit can contact about any financial issues regarding the account.
E-mail address financial contract	Set by Bibit, determined by the merchant.	Email address of the financial contact person.
Invoice Country	Set by Bibit, determined by the merchant.	Country where the invoices are sent to.
Invoice language	Set by Bibit, determined by the merchant.	Language of the invoices.
Invoice payment method	Set and determined by Bibit.	For merchants operating from the Netherlands standard type is 'Incasso' and next 3 fields need to be filled out. For other merchants it is 'Transfer', no extra information is required in the next 3 fields.
Incasso bank account	Set by Bibit, determined by the merchant.	The bank account number where Bibit can withdraw the invoice amounts for this account.
Incasso bank account holder name	Set by Bibit, determined by the merchant.	The name of the legal owner of the bank incasso bank account.

Incasso bank account city	Set by Bibit, determined by the merchant.	City where incasso bank account is located.
---------------------------	---	---

Merchant channels		
E-mail (test)	Set by Bibit, determined by the merchant.	The email address where payment confirmations are sent to, can be set to 'active yes' or 'active no'. Formats can be text, cgi or html.
http (test)	Set by Bibit, determined by the merchant.	The URL where HTTP confirmations of payments are sent to, can be set to 'active yes' or 'active no'. Formats can be text, cgi or html. Set to 'POST' or 'GET'. Need to be confirmed with an [OK].
E-mail (production)	Set by Bibit, determined by the merchant.	The email address used to send payment confirmations to, can be set to 'active yes' or 'active no'. Formats can be text, cgi or html.
http (production)	Set by Bibit, determined by the merchant.	The URL where http confirmations of payments are sent to, can be set to 'active yes' or 'active no'. Formats can be text, cgi or html. Set to 'POST' or 'GET'. Need to be confirmed with an [OK].
Active on test	Set by Bibit, yes or no.	Determines whether the merchant has access to the test environment.
Active on production	Set by Bibit, yes or no.	Determines whether the merchant has access to the live/production environment.
Test shop IP-address	Set by Bibit, determined by merchant.	Contains the IP address from where Bibit accepts orders for the merchant for testing. /24 indicates checking on C-range, /12 indicates checking on B-range, /0 indicates no IP check.
Production shop IP address	Set by Bibit, determined by merchant.	Contains the IP address from where Bibit accepts orders for the merchant for production. /24 indicates checking on C-range, /12 indicates checking on B-range.
Order / transfer / capture report test	Set by Bibit, determined by merchant.	Determines where the merchant reports in test are sent, can be set to active or inactive. Can be sent in several formats. Can be sent through email or http.
Order / transfer / capture report production	Set by Bibit, determined by merchant.	Determines where the merchant reports in production are sent, can be set to active or inactive. Can be sent in several formats. Can be sent through email or http.

Bank information		The details of the bank account Bibit uses to pay out the merchant.
Bank name	Set by Bibit, determined by merchant.	The name of the bank the merchant is using
Bank country	Set by Bibit, determined by merchant.	Country where the bank is located.
Bank address	Set by Bibit, determined by merchant.	Address of the bank.
Branch code, Bankleitzahl, Sort code	Set by Bibit, determined by merchant.	Code used in Germany, UK, US.
Account currency	Set by Bibit, determined by contract.	Currency the amounts are sent in.
Account number	Set by Bibit, determined by merchant.	Actual bank account number
Account holder city	Set by Bibit, determined by merchant.	City of the account holder.
Account holder address	Set by Bibit, determined by merchant.	Address of account holder.

20 Payment method codes

The merchant can use the `paymentMethodMask` or the `preferredPaymentMethod` variable to determine which payment method(s) the shopper will be able to choose. The codes for the payment methods can be found in the tables below.

Credit Cards

Name	PaymentMethod code	Area	Remarks
American Express SSL	AMEX-SSL	International	
VISA	VISA-SSL	International	
MasterCard	ECMC-SSL	International	The name Eurocard is no longer in use
Carte Bancaire	CB-SSL	France	
Carte Bleue	CARTEBLEUE-SSL	France	
Diners	DINERS-SSL	International	
Laser Card	LASER-SSL	Ireland	
Discover Card	DISCOVER-SSL	United States	
Japanese Credit Bank	JCB-SSL	International, Japan	
Dankort	DANKORT-SSL	Denmark	

Online debit methods

Name	PaymentMethod code	Area	Remarks
Rabobank DirectBetalen	RABO-DIRECTBETALEN	Netherlands	For Rabobank shoppers only. Shopper is redirected to Rabobank server.
ING Homepay	HOMEPAY-SSL	Belgium	For ING shoppers only. Shopper needs to have a ING Homepay account at his bank.
Solo	SOLO_GB-SSL	United Kingdom	
Switch	SWITCH-SSL	United Kingdom	
WWW-Bon	ICCHEQUE-SSL	Netherlands	Internet voucher, will be discontinued.
Nordea Bank	SOLO-SSL (Fi) EBETALNING-SSL (Se)	Finland, Sweden	
Paybox	PAYBOX-SSL	Germany, Austria, Spain, UK	Payment method using mobile phone.
Elektronisches Lastschriftverfahren	ELV-SSL	Germany	

Offline payment methods

Name	PaymentMethod code	Area	Remarks
Domestic Bank transfer	TRANSFER_NL-BANK TRANSFER_BE-BANK TRANSFER_DE-BANK TRANSFER_FI-BANK TRANSFER_FR-BANK TRANSFER_IT-BANK TRANSFER_ES-BANK TRANSFER_GB-BANK TRANSFER_SE-BANK TRANSFER_AT-BANK TRANSFER_LU-BANK TRANSFER_CH-BANK TRANSFER_DK-BANK TRANSFER_GR-BANK TRANSFER_NO-BANK	Netherlands, Belgium, Germany, Finland, France, Italy, Spain, United Kingdom, Sweden, Austria, Luxemburg, Switzerland, Denmark, Greece, Norway	Shopper transfers the money using a bank transfer, either manually or through an electronic banking system. If bank transfers are used for international payments, the shopper may be presented with extra charges (cross border fees) from the banks
Cheque	CHEQUE-BANK		Cheques
Cheque	CHEQUE_GB-BANK	United Kingdom,	Regular cheque payments
Direct Debit	INCASSO_NL-FAX INCASSO_DE-FAX	The Netherlands, Germany	Forms have to be printed, signed and sent to Bibit
Rembours / Cash on Delivery	CASH-DELIVERY	The Netherlands	
Deutsche Bank 24	DB24-BANK	Germany	
Dresdner Bank InternetBanking	DRESDNER-BANK	Germany	
Commerz Bank Online Banking Web	COMLINE-BANK	Germany	
AcceptGiro	ACCEPTGIRO_NL-BANK	The Netherlands	Merchant has to use the 'reference id' as payment reference to be printed on the accept giro forms
	PERMANENT_SIGNED_DD_NL	The Netherlands	
	SINGLE_UNSIGNED_DD_FR SINGLE_UNSIGNED_DD_NL	France, The Netherlands	

21 ISO currency codes (ISO 4217)

Currencies accepted by the Bibit Payment Service⁵. Exponent is the number of decimals available in the currency. Please note that amounts in the orders sent to Bibit NEVER have any decimal delimiters. Merchants should use 'exponent' instead! Currency code is always in capitals. In the following example the amount payable by the shopper is Euro 19,82: `<amount value="1982" currencyCode="EUR" exponent="2" />`

Code	Name	Exponent
ARS	Nuevo Argentine Peso	2
AUD	Australian Dollar	2
BRL	Brazilian Real	2
CAD	Canadian Dollar	2
CHF	Swiss Franc	2
CLP	Chilean Peso	2
CNY	Yuan Renminbi	2
COP	Colombian Peso	2
CZK	Czech Koruna	2
DKK	Danish Krone	2
EUR	Euro	2
GBP	Pound Sterling	2
HKD	Hong Kong Dollar	2
HUF	Hungarian Forint	2
IDR	Indonesian Rupiah	0
ISK	Iceland Krona	2
JPY	Japanese Yen	0
KES	Kenyan Shilling	2
KRW	South-Korean Won	0
MXP	Mexican Peso	2
MYR	Malaysian Ringgit	2
NOK	Norwegian Krone	2
NZD	New Zealand Dollar	2
PHP	Philippine Peso	2
PLN	New Polish Zloty	2
PTE	Portugese Escudo	2
SEK	Swedish Krone	2
SGD	Singapore Dollar	2
SKK	Slovak Koruna	2
THB	Thai Baht	2
TWD	New Taiwan Dollar	2
USD	US Dollars	2

⁵ The full ISO 4217 list can be found at: <http://www.id3.org/iso4217.html>.

VND	Vietnamese New Dong	2
ZAR	South African Rand	2

22 Country and language table

The Bibit Payment Service uses the two letter ISO 3166 country codes. These codes are always in capital letters. The language codes used by the Bibit Payment Service are the two letter ISO 639 codes; these are always in lower case.

Consult the Bibit Demo shop (<http://www.bibit.com/demo>) to find the current list of accepted countries and languages.

Complete list of all country codes:

http://www.din.de/gremien/nas/nabd/iso3166ma/codlstp1/en_listp1.html

http://ftp.std.com/obi/Standards/ISO/ISO_3166.

Be sure to only use the two letter codes.

Complete list of all language codes: http://ftp.std.com/obi/Standards/ISO/ISO_639.

23 Acquirer Response Codes

Bibit uses the ISO 8583 Response Codes in the `orderStatusEvent` messages to indicate the status of the payment: AUTHORISED, REFUSED, etc.

The Bibit Payment Service maps these responses to a simplified list. Below you will find all possible response codes, their numeric value and the Bibit mapping.

Code	Message	Status
0	AUTHORISED	AUTHORISED
2	REFERRED	REFUSED
4	HOLD CARD	REFUSED
5	REFUSED	REFUSED
8	APPROVE AFTER IDENTIFICATION	REFUSED
13	INVALID AMOUNT	REFUSED
15	INVALID CARD ISSUER	REFUSED
17	ANNULATION BY CLIENT	REFUSED
28	ACCESS DENIED	REFUSED
29	IMPOSSIBLE REFERENCE NUMBER	REFUSED
33	CARD EXPIRED	REFUSED
34	FRAUD SUSPICION	REFUSED
38	SECURITY CODE EXPIRED	REFUSED
41	LOST CARD	REFUSED
43	STOLEN CARD, PICK UP	REFUSED
51	LIMIT EXCEEDED	REFUSED
55	INVALID SECURITY CODE	REFUSED
56	UNKNOWN CARD	REFUSED
57	ILLEGAL TRANSACTION	REFUSED
62	RESTRICTED CARD	REFUSED
63	SECURITY RULES VIOLATED	REFUSED
75	SECURITY CODE INVALID	REFUSED
76	CARD BLOCKED	REFUSED
85	REJECTED BY CARD ISSUER	REFUSED
91	CREDITCARD ISSUER TEMPORARILY NOT REACHABLE	REFUSED
97	SECURITY BREACH	REFUSED
3	INVALID ACCEPTOR	ERROR
12	INVALID TRANSACTION	ERROR
14	INVALID ACCOUNT	ERROR
19	REPEAT OF LAST TRANSACTION	ERROR
20	ACQUIRER ERROR	ERROR
21	REVERSAL NOT PROCESSED, MISSING AUTHORISATION	ERROR
24	UPDATE OF FILE IMPOSSIBLE	ERROR
25	REFERENCE NUMBER CANNOT BE FOUND	ERROR
26	DUPLICATE REFERENCE NUMBER	ERROR

27	ERROR IN REFERENCE NUMBER FIELD	ERROR
30	FORMAT ERROR	ERROR
31	UNKNOWN ACQUIRER ACCOUNT CODE	ERROR
40	REQUESTED FUNCTION NOT SUPPORTED	ERROR
58	TRANSACTION NOT PERMITTED	ERROR
64	AMOUNT HIGHER THAN PREVIOUS TRANSACTION AMOUNT	ERROR
68	TRANSACTION TIMED OUT	ERROR
80	AMOUNT NO LONGER AVAILABLE, AUTHORISATION EXPIRED	ERROR
92	CREDITCARD TYPE NOT PROCESSED BY ACQUIRER	ERROR
94	DUPLICATE REQUEST	ERROR

Index

- [OK], 36, 37, 39, 73, 81, 96
- 3D secure**, 85
- acceptance test, 15, 29, 63, 70, 74, 75, 81, 82
- AcceptGiro, 23, 100
- Access denied, 75, 76
- Address Verification Service, 72, 85
- amount, 6, 12, 15, 17, 19, 21, 32, 33, 34, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 58, 59, 61, 62, 73, 78, 79, 84, 86, 93, 94, 95, 101
- authenticated shopper ID, 21, 58
- Authenticating the sender, 41
- Authenticity**, 41
- authorisation code, 33, 45, 46, 92
- AUTHORISED, 32, 34, 37, 46, 47, 52, 58, 71, 73, 84
- AVS, 72, 85, 92
- background image, 27
- back-office, 11, 16, 17, 26, 35, 36, 40, 63
- Back-office**, 16, 40
- bank transfer, 23, 29, 33, 86, 100
- batch, 25, 34, 35, 59, 60, 61, 62, 75, 79, 91
- Batch, 13, 55, 59, 60, 62
- BSE, 67
- Call Center, 11, 12
- Call Centre, 12, 15, 55, 62, 94
- cancel, 34, 42, 86
- CANCELLED, 34, 37, 43, 50
- capture, 16, 33, 34, 40, 44, 45, 53, 86, 91, 93, 96
- capture delay, 33, 86
- Capture report, 91
- CAPTURED, 32, 33, 34, 37, 51, 61
- card address, 19
- Card deposit**, 53, 92
- card-reader, 65
- CDATA, 6, 17, 19, 21, 50, 53, 54, 55, 67, 75, 76, 77, 78, 79
- certificate, 41
- channels, 16, 28, 35, 40, 63, 96
- Character Data, 19
- charge-back, 74, 83
- charge-backs, 65, 70, 83, 93
- CHARGED_BACK, 34, 35, 38, 49
- Cherry MY8000 keyboard, 65
- confirmation messages, 8, 28, 34, 35, 36, 37, 40, 41, 58, 60, 63, 73
- connection, 7, 22, 25, 30, 41, 55, 63, 65
- content length**, 7, 22, 77
- C-range, 22, 23, 75, 96
- Cross Border Surcharges, 87
- CVC, 84, 92
- CVC2, 28, 55, 58, 72, 83, 84, 85, 92
- date/time-stamp, 78
- DEPOSIT, 89
- description, 6, 12, 17, 19, 21, 26, 46, 47, 48, 52, 53, 55, 59, 60, 62, 73, 77, 84
- Dispute management**, 87
- Document Type Definition, 14, 16
- DTD, 6, 7, 8, 14, 16, 17, 20, 21, 23, 25, 35, 40, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 57, 58, 59, 60, 61, 69, 73, 75, 76, 77, 78, 79, 84
- Duplicate**, 77
- Elektronisches Lastschriftverfahren, 99
- ELV, 57, 61, 86, 99
- E-mail confirmations, 35
- E-mail notification, 30
- Empty body in message**, 76
- ERROR, 32, 33, 60, 62, 71
- EXPIRED, 104, 105
- failure url, 27, 29
- Failure url**, 29
- feedback, 15, 27, 28, 29, 30, 58, 59, 63, 81
- firewall, 22, 36
- Fulfilment, 85
- hosted call centre application, 25, 62, 63, 81, 94
- HTTP confirmations, 36
- HTTP-refresh, 30
- https proxy, 27
- HTTPS proxy**, 29, 30, 93
- IN_PROCESS_AUTHORISED, 8, 36, 89
- IN_PROCESS_CAPTURED, 89, 90
- Incasso, 32, 95, 96
- internal error, 75, 79
- Internal Error**, 78
- invalid content, 75
- invalid request, 75
- IP address, 15, 22, 23, 27, 30, 36, 41, 75, 96
- IP check failed, 23, 75
- live, 15, 19, 22, 23, 25, 27, 30, 36, 70, 75, 81, 82, 96
- look and feel, 26
- Mask, 20
- merchant interface, 16, 17, 19, 30, 33, 34, 35, 40, 45, 63, 65, 76, 86, 87, 92
- Merchant not active, 75
- merchantBatchCode, 59, 60, 61, 79
- Missing doc type declaration**, 76
- Missing quote characters**, 77
- MOTO, 65, 83, 93
- oderInquiry, 42
- online sign-up form, 15
- order, 6, 7, 8, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 31, 32, 33, 34, 35, 40, 41, 42, 43, 44, 46, 47, 50, 51, 52, 53, 55, 57, 58, 59, 60, 61, 62, 63, 65, 71, 72, 73, 74, 75, 77, 78, 81, 83, 84, 86, 87, 91, 93, 94
- order modification, 33, 40
- Order report, 91
- Ordercode, 63
- ordercontent, 26, 31
- orderContent, 6, 7, 17, 18, 19, 21, 26, 31, 52, 53, 55, 56, 59, 81, 84
- OrderInquiry**, 47
- orderModification, 42
- ORDERS_SAVED, 60, 62
- parse error, 75
- parser, 19, 23

- Parsing**, 23
- partial capture, 33, 44, 86
- payment, 7, 11, 12, 13, 16, 17, 18, 19, 20, 21, 23, 24, 25, 26, 27, 28, 29, 30, 32, 33, 34, 35, 40, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 60, 61, 62, 63, 65, 66, 68, 71, 72, 73, 74, 75, 78, 79, 80, 81, 84, 85, 86, 87, 88, 91, 92, 94, 95, 96, 98, 100
- Payment results, 60
- paymentMethodMask, 7, 17, 18, 20, 21, 26, 63, 71, 98
- PAYOUT, 89
- pending url, 27, 28
- Pending url**, 28
- Point Of Sale, 12, 15, 65
- POS**, 12
- preferred payment, 27, 80
- PROCESSED, 60, 61, 62
- PROCESSED_WITH_ERROR, 60
- production (live) environment, 15
- production environment, 15, 22, 70, 71, 73, 74, 75, 81, 82, 96
- protocols, 11, 15
- purchase cards, 67
- Purchase cards, 67
- Recurring payments**, 52
- redirect model, 11, 17
- Redirect model, 13
- redirecting, 25, 30
- redirection location**, 25
- redirection url, 23, 24, 26, 28, 30, 58
- refund, 34, 43, 44, 48, 53, 54, 74, 83, 86
- REFUNDED, 34
- refunds, 16, 34, 43, 44, 53, 54, 70, 86
- REFUSED, 8, 9, 32, 33, 34, 35, 37, 47, 48, 71
- reporting, 16, 29, 63, 73, 85, 89, 91
- request variables, 29
- Request variables**, 29
- Result urls, 28
- Risk Management, 16, 33, 58, 62, 64, 65, 83, 93
- Risk Management Module, 16, 33, 83, 93
- router, 22
- Secure server, 41
- Security, 55, 76
- security error, 75
- SEND_FOR_AUTHORISATION, 32
- SENT_FOR_REFUND, 34, 44, 49, 54
- Setting up a connection. See connection**
- SETTLED, 34, 49
- SETTLED_BIBIT_COMMISSION, 49, 89, 90
- SETTLED_BIBIT_NET, 89, 90
- settlement currencies**, 87
- SGML. See Standard GEneralised Mark-up Language
- shopper element, 21
- Shopper element, 20
- Shopper information, 20
- Shopping application**, 16
- shopping basket, 11, 16, 87
- SIGNED_FORM_RECEIVED, 32, 34, 37, 48
- SSL, 8, 20, 21, 35, 37, 38, 43, 44, 46, 47, 48, 49, 50, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 73, 84, 98, 99
- Standard Generalised Mark-up Language, 15
- success url, 27, 28
- Success url**, 28
- support**, 23, 27, 41, 81, 86
- surcharges, 87
- SWIPE application, 12, 65
- System requirements, 15
- Test credit card numbers**, 70
- test environment, 15, 22, 23, 31, 70, 71, 73, 74, 75, 79, 82, 96
- Test locations**, 70
- Testing, 63, 70, 74, 75
- Transfer report, 91
- unauthorised, 83
- Unsupported currency**, 78
- URL encoding**, 29
- urlencoded, 26
- Verified by VISA**, 85
- Web server**, 15
- webshop, 11, 71, 81
- Webshop, 6, 11, 17, 21, 55
- XML, 7, 8, 11, 13, 14, 15, 16, 17, 19, 21, 22, 23, 25, 31, 33, 35, 38, 40, 45, 47, 53, 55, 58, 60, 62, 63, 65, 72, 75, 77, 81, 84, 86, 92
- xml password, 76