



ICEPAY
global payments • local service

IMPLEMENTATION GUIDE V1.3.0



TABLE OF CONTENTS

1	PREFACE	4
1.1	Who should read this document?	4
2	GLOSSARY	4
3	REQUIREMENTS & IMPLEMENTATION	5
3.1	Requirements	5
3.2	Implementation	5
4	DATAFLOW	6
4.1	Pathway 1: Redirection	6
4.1.1	The dataflow of pathway 1	6
4.2	Pathway 2: Server-to-server communication or Server-to-flash communication.	9
5	PARAMETER DEFINITIONS	10
5.1	Domain and Server IP range	10
5.1.1	Domain	10
5.1.2	Server ip range	10
5.2	Checkout.aspx	10
5.2.1	Parameters	11
5.2.2	Checksum	12
5.2.2.1	Checksum generation Sample code	13
5.2.3	Payment request form sample	15
5.3	Parameters returned in step 4 only for FLASH and XML response type	16
5.3.1	Parameters	16
5.3.2	Checksum	17
5.3.3	Sample	17
5.4	Parameters sent to IC_URLCompleted and IC_URLRError	19
5.4.1	Parameters	19
5.4.2	Checksum	20
5.4.3	Sample	20
5.5	Handling the postback notifications, parameters sent to IC_Postback	21



5.5.1	Parameters	21
5.5.2	Possible statuses	22
5.5.3	Detailed status description	22
5.5.4	Possible status transitions	24
5.5.5	Checksum	24
5.5.6	Sample	25
APPENDIX		26
A.	Code samples, document updates and other documents	26
B.	Changelog	27
C.	Contact	27



1 PREFACE

This document explains how to implement a coupling between your website and the ICEPAY payment methods in *Advanced Mode*.

NOTE: This document is not applicable for other payment initialization methods like Basic Mode or “API Mode”

1.1 WHO SHOULD READ THIS DOCUMENT?

This document is intended for the technical staff (webmaster, software engineer, etc.) at your company.

2 GLOSSARY

Merchant	The direct users of ICEPAY. This can be an individual or a company that operates the website that will be coupled to the ICEPAY system.
User	A customer of the Merchant. It is the person that wants to make a payment through the website of the Merchant.
Payment method provider	Any provider of payment services, such as ABN AMRO (iDEAL), Wallie, Interconnection (phone payments) etc.



3 REQUIREMENTS & IMPLEMENTATION

3.1 REQUIREMENTS

The following things are required to get started:

- A server-side scripting environment that is capable of generating SHA1 hashes. PHP4, PHP5, Perl 5.x and ASP.NET 2.0 are capable of doing this. Although other languages and environments might also be capable of generating SHA1 hashes, it has not been tested by us.
- An ICEPAY account. When your account is activated, you will receive a *MerchantID* and an *Encryption Code* from us. If you do not have them, please contact us.
Warning: You must never reveal these two codes to third parties! The *MerchantID* and *Encryption Code* are used to verify your identity.
NOTE: Each *MerchantID* (+ *Encryption Code*) is meant to be used only for 1 (one) web site. You may request as much you need merchant accounts for free!

3.2 IMPLEMENTATION

You must implement the following items:

- One scripted page for server-to-server communication. You can set this value in web interface when you login into client area. This URL is named *IC_Postback* in this document (see 5.5).
- A “Shopping Cart Checkout” page. This is where your system must initiate the ICEPAY payment.
- An “OK” page. This is where the user will go to when the ICEPAY payment has been completed successfully. In the more technical part of this document, it is referred to as *IC_URLCompleted* (see 5.4).
- An “Error” page. This is where the user will go to when the ICEPAY payment cannot be completed, aborted, failed, or canceled. In the more technical part of this document, it is referred to as *IC_URLError* (see 5.4).
Hint: it is possible to put an option on this page where the user is able to try again. Don't forget to create new OrderID.



4 DATAFLOW

There are two implementation methods. You should decide which implementation method suits your situation better.

4.1 PATHWAY 1: REDIRECTION

This method is the “classic” approach and is most suited for most websites. The user will first be routed to ICEPAY. Immediately after that, the user will be routed to the site of the payment method provider where they will see a payment screen. Depending on the payment method this can be a page for iDEAL, Interconnection, Wallie or any other supported payment method. This is where the user is able to complete the payment. After the payment is completed successfully, the user will be redirected to the “OK” page (*IC_URLCompleted*). If the payment was unsuccessful, then the user will be redirected to the “Error” page (*IC_URL_Error*). All payment status change events are pushed to IC_Postback.

4.1.1 THE DATAFLOW OF PATHWAY 1

To understand complete payment dataflow as in Figure 1 we will explain every step.

1. The user requests an order checkout at the merchant’s website.
2. The merchant prepares the order information, generates the encrypted checksum and using plain FORM sends all required information to ICEPAY’s Check-out.aspx page. See chapter 5.2 for more information.
Note: use *IC_ResponseType* = REDIRECT
Tip: you can do this 2nd step in step 1 already by using a hidden FORM (see the sample code that is provided separately)
3. ICEPAY requests a transaction initialization with the chosen payment method provider.
4. A response is generated based on the reply received from the payment method provider in step 3. There are 2 possible courses of action:
 - a. The transaction initialization with the payment method provider was OK. The user is redirected to the site of the payment method provider.
 - b. Something went wrong. The user is redirected to the “Error” page of the merchant along with a description of what went wrong. It is very important that the user understands what happened, and as such should be presented with an informative message.
The payment request for the current order ends here.
Note: if you want to retry the payment (perhaps with another payment method?) a new *OrderID* must be generated!
5. The user makes a payment at the website of the payment method provider.



6. In the background, ICEPAY will periodically query the payment method provider for the payment status information and update its internal data. This is repeated until the payment is completed, aborted, failed or canceled.
7. In every update cycle in step 6, the merchant will be informed through the “IC_Postback” about any change in the payment status. It is very important that this URL is set to a correct (and existing!) page.
8. The user finished or canceled the payment. The user is sent back to ICEPAY for further processing.
Note: A payment can also timeout or generate an error. In such case the payment status will also be collected in step 6.
9. After the payment process has finished, the user is routed to ICEPAY. Here, based on information from step 6, 7 and 8, the user will be redirected:
 - a. The payment was completed successfully: the user is redirected to *IC_URLCompleted*
 - b. The payment was not completed for any reason, canceled or aborted by the user: the user is redirected to *IC_URL_Error*
 - c. The payment is still being processed by the payment method provider. For online payment methods the user is redirected to *IC_URL_Error*, for off-line payment methods the user is redirected to *IC_URLCompleted*, with Status=OPEN in both cases. See table 5.4.1 for more information.

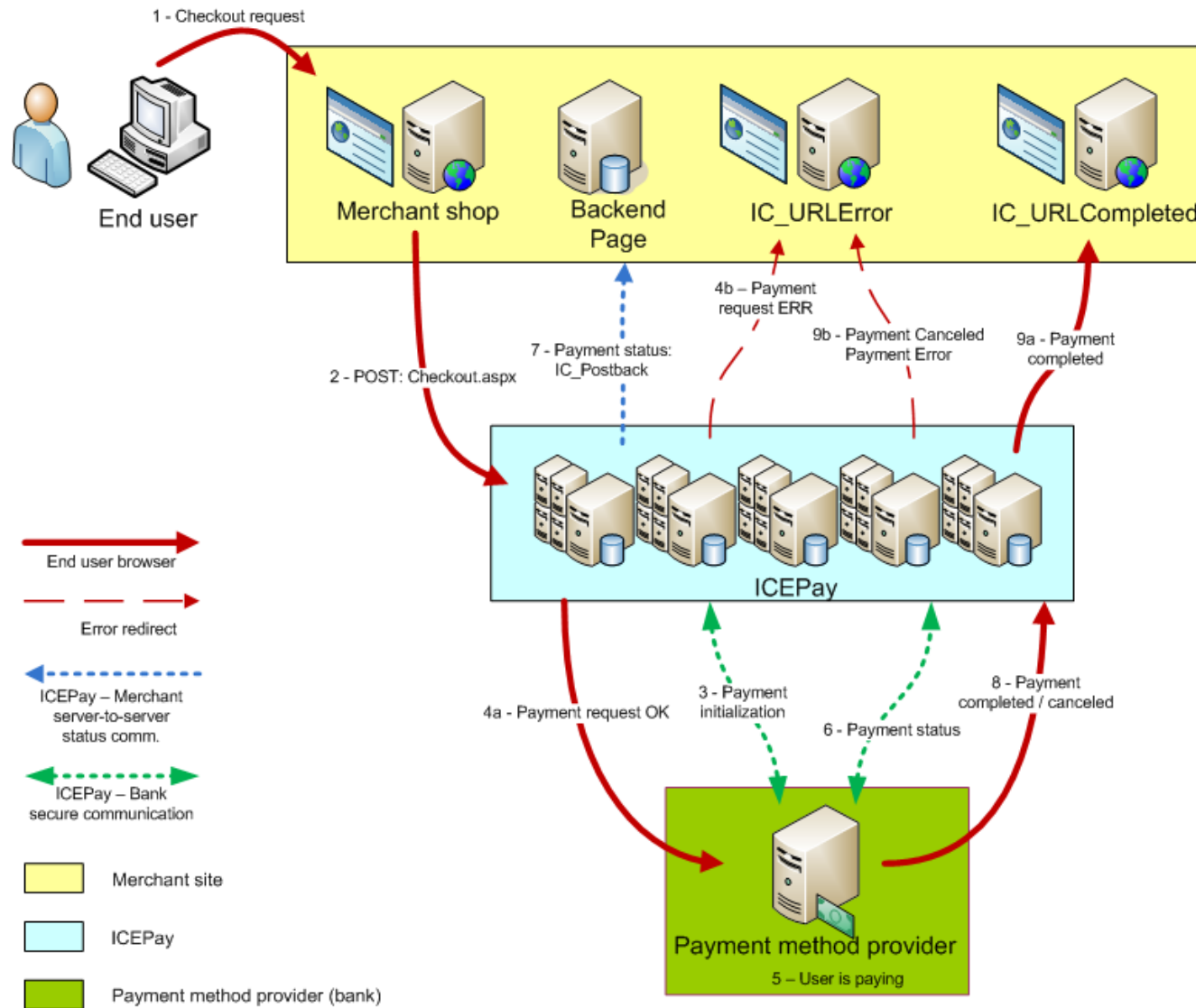




Figure 1: The dataflow of pathway 1

4.2 PATHWAY 2: SERVER-TO-SERVER COMMUNICATION OR SERVER-TO-FLASH COMMUNICATION.

This method is adapted to complicated sites, for example those that entirely work in flash or use XML.

Advantage is that end users will see minimal amount of ICEPAY URLs because payment initialization is done in background between ICEPAY and merchant server.

You can find information related to this option in section 5.3.

Be advised that credit card companies and banks demand that their full URL is visible at all times, so even though the user does not need to see ICEPAY URLs, having the payment taking place outside of your own site is unavoidable.

Contact us or check available samples if you plan to use this method.



5 PARAMETER DEFINITIONS

This chapter describes the parameters that your website must provide when communicating with an ICEPAY page. For the other pathway (FLASH / XML) you need to follow a slightly different sequence of events.

5.1 DOMAIN AND SERVER IP RANGE

5.1.1 DOMAIN

The domain for all communication with ICEPAY is

pay.icepay.eu

While we support both HTTP and HTTPS protocols, HTTPS is only recommended option. We might disable HTTP without previous notice!

5.1.2 SERVER IP RANGE

Our servers are in following IP range

*194.30.175.0 – 194.30.175.255
and
95.211.51.64 - 95.211.51.95*

5.2 CHECKOUT.ASPX

Use the following URL for initiating the payment:



<http://pay.icepay.eu/Checkout.aspx>
or
<https://pay.icepay.eu/Checkout.aspx>

Checkout.aspx handles the start of the payment. It does the handling of the merchant request (step 2 in payment flowchart diagram) the initialization to the payment method provider (step 3 in payment flowchart diagram) and the redirect to the payment method provider (step 4a in payment flowchart diagram) or the error page (step 4b in payment flowchart diagram).

Note: You can only POST to this page. GET requests are not handled to prevent query string limitation problems.

While we support both HTTP and HTTPS protocols, HTTPS is only recommended option. We might disable HTTP without previous notice!

5.2.1 PARAMETERS

The parameters handled by Checkout.aspx are:

Parameter	Description	Data type	Max length / range	Example	Default Value	Req.
IC_Merchant	Your merchant ID received from our administration	Numeric	1000-1000000	1000		Yes
IC_Amount	Value in eurocents from 30 up to 1000000 eurocent. If you have to handle more than 10000 EUR per transaction please contact your account manager. Check Appendix A for more details.	Numeric	30-1000000	100 (equivalent of 1 euro)		Yes
IC_Currency	Currency 3 letter symbol in ISO 4217 format. Use one of these: EUR, GBP, USD, KRO	String	3	EUR		Yes
IC_Language	2 letter language in ISO 639-1 format such as NL, DE, EN	String	2	EN	NL	No
IC_Country	2 letter country code in ISO 3166-1 format such as NL, DE, UK, BE. Use 00 for global coverage if payment method is not country dependent	String	2	UK	NL	No
IC_OrderID	Your unique order number. This can be auto incremental number from your payments table	String	10	1001		Yes
IC_Reference	Your reference number. This might be your Shopping cart ID. It does not have to be unique	String	50	Z1234567		No
IC_PaymentMethod	Payment method such as IDEAL, PHONE. See Appendix Error! Reference source not found. for complete list	String	20	IDEAL		Yes
IC_Issuer	Payment method issuer. See Appendix Error! Reference source not found. for complete list	String	20	ABN		Yes
IC_Description	Text appears on transaction lists	String	100	Your online download for...		No
IC_CheckSum	Generated SHA1 signature for request validation. The generation of the checksum is explained in 5.2.2	String	40	da39a3ee5e6b4b0d3255bfef95601890afd80709		Yes



IC_URLCompleted	When the payment is confirmed the user is redirected to this page. This parameter overrides the default setting defined in our system. The last known transaction status will be passed along, so you can process the result instantly. See 5.4 for returned parameter values	String	500	http://localhost/success.aspx	Value set in the control panel	No
IC_URL_Error	In case of any error during the payment procedure, the user is redirected here. This parameter overrides the default setting defined in our system. The last available error code/description is provided as a parameter. Note: In the initial payment request the user will NOT be redirected to the error page in case IC_ResponseType is different than REDIRECT. In that case you will receive the error code which you have to handle properly in own application (in that case check 5.3). See 5.4 for returned parameter values	String	500	http://localhost/error.aspx	Value set in the control panel	No
IC_ResponseType	Specifies how the return value is handled: REDIRECT, XML, FLASH	String	10	REDIRECT	REDIRECT	No
IC_Style	HTML style to be used. Use only if instructed by ICEPAY technical support	String	20	DEFAULT		No
IC_PINCode	PIN/Activation Code. This parameter is needed for some payment methods, and in that case this is a required field.	String	100	456789		**

Table 1

** Only for PPC payments, then required.

5.2.2 CHECKSUM

In order to generate the IC_CheckSum you will need the Merchant ID and the Encryption code. You should receive it when you have an ICEPAY account.

The checksum is generated using the following formula:

$$IC_Checksum = SHA1(Encryptioncode + | + IC_Merchant + | + IC_Amount + | + IC_Currency + | + IC_OrderID + | + IC_PaymentMethod + | + IC_Issuer)$$

As you may have noticed the “pipe” sign (|) is used as the delimiter between values. You may need to put the delimiters between single quotes (') or double quotes (") depending on the programming language that you will be using.

The value returned by the SHA1 function is a string of 40 characters representing a hexadecimal value.

IMPORTANT NOTE

The ICEPAY system will always cut off the parameter values up to its corresponding maximum allowable length as defined in Table 1. This may cause the ICEPAY checksum not to match your own checksum. Example: if you provide “EURO” for the currency parameter, then it will be recognized by ICEPAY as “EUR”



because the maximum length of the currency parameter is 3. Thus, the provided value “EURO” is valid and ICEPAY will not complain. However, ICEPAY uses “EUR” to generate the checksum whereas you may have used “EURO” for the checksum generation. This causes the two checksums not to match and therefore the request fails.

Please keep this in mind when generating the checksum, especially if you get this error: “IC_ERR: Not valid Checksum value”

5.2.2.1 CHECKSUM GENERATION SAMPLE CODE

5.2.2.1.1 PHP

In PHP you can generate IC_CheckSum as:

```
$IC_CheckSum = SHA1( $Encryptioncode . "|" . $IC_Merchant . "|" . $IC_Amount . "|" . $IC_Currency . "|" . $IC_OrderID . "|" . $IC_PaymentMethod . "|" . $IC_Issuer );
```

5.2.2.1.2 C#

In C# you can generate IC_CheckSum as:

```
class Program
{
    string IC_Checksum(string Encryptioncode, string IC_Merchant, string IC_Amount, string IC_Currency, string IC_OrderID, string IC_PaymentMethod,
    string IC_Issuer)
    {
        byte[] message = Encoding.ASCII.GetBytes(Encryptioncode + "|" + IC_Merchant + "|" + IC_Amount + "|" + IC_Currency + "|" + IC_OrderID + "|" +
        IC_PaymentMethod + "|" + IC_Issuer);
        return (new System.Security.Cryptography.SHA1CryptoServiceProvider()).ComputeHash(message).ToString();
    }
}
```




```
Return sb.ToString  
End Function  
End Module
```

5.2.3 PAYMENT REQUEST FORM SAMPLE

In this sample we will use following merchant information:

MerchantID = 10000

Merchant Secret = bvjdhIAS82hdiue13hkna08hd63bdiabc823hd

Payment Method = Credit card / Visa

Amount = 1.30 €

and other obvious values

You can use this sample to match your checksum value and to get an idea what a payment initialization form can look like.

```
<form method='POST' action='https://pay.icepay.eu/checkout.aspx'>  
<input type='hidden' name='ic_paymentMethod' value='CREDITCARD'>  
<input type='hidden' name='ic_issuer' value='VISA'>  
<input type='hidden' name='ic_merchant' value='10000'>  
<input type='hidden' name='ic_amount' value='130'>  
<input type='hidden' name='ic_currency' value='EUR'>  
<input type='hidden' name='ic_language' value='NL'>  
<input type='hidden' name='ic_country' value='NL'>  
<input type='hidden' name='ic_orderid' value='TESTk6QAr'>  
<input type='hidden' name='ic_reference' value='RefTESTk6QAr'>  
<input type='hidden' name='ic_description' value='Test payment'>  
<input type='hidden' name='ic_checksum' value='76b872945a5640991e5ee50649cda1d56bf7cc7f'>  
<input type='hidden' name='ic_urlcompleted' value='http://localhost/payment/completed.aspx?done=t1'>  
<input type='hidden' name='ic_urlerror' value='http://localhost/payment/error.aspx?err=t1'>  
<input type='hidden' name='ic_responsetype' value='REDIRECT'>
```



```
<input type="submit" value="Pay" />  
</form>
```

5.3 PARAMETERS RETURNED IN STEP 4 ONLY FOR FLASH AND XML RESPONSE TYPE

If in step 2 you have used FLASH or XML for *IC_ResponseType* parameter then in step 4 you will receive, as return value, parameters which you have to process in your own site.

5.3.1 PARAMETERS

Parameters returned in step 4 in the payment flowchart diagram:

Parameter	Description	Data type	Sample	Can be empty
Status	Return OK or ERR as payment request status. OK means that payment initialization was successful and that payment provider is ready to process the payment ERR is signal for error in request, in our system or payment provider is not ready to handle the payment. In this case you should consult ErrCode for more information	String(10)	ERR	N
ErrCode	The error code and a short description which may be displayed on screen are returned. In any case you should treat the current order as failed and do not retry to send the same order request again because our system will then return "Duplicate OrderID"!	String (50)	Duplicate OrderID	N
OrderID	Value of IC_OrderID which was sent to Checkout.aspx	String (10)	1234567	Y
PaymentID	The unique numeric value that identifies this payment in our system.	Numeric	100000012	Y
URL	If the response type is XML/Flash this value will have the payment URL (step 4a). In case of error this parameter should be ignored.	String(2000)	https://localhost/paywithphone	Y
Checksum	If possible, a checksum is generated (see 5.3.2). In some cases it is not possible to identify the Merchant, for example when there is an error in the received parameters. In that case the checksum will be empty.	String (40)	da39a3ee5e6b4b0d3255bfe95601890 afd80709	Y



5.3.2 CHECKSUM

The Checksum is generated using the following formula (if possible):

```
SHA1( Encryptioncode + | + IC_Merchant + | + OrderID + | + PaymentID + | + URL)
```

5.3.3 SAMPLE

Sample of a returned string for FLASH format in case of error:

```
Status=ERR&ErrCode=IC_ERR%3a+Checksum+is+not+valid&URL=&OrderID=TESTDx87Z&PaymentID=&Checksum=
```

Below you can see returned string for valid request in FLASH format. Note that in this document the text is wrapped, while in a normal situation you would see response as one line

```
Status=OK&ErrCode=&URL=https%3a%2f%2fipay.ICEPay.eu%2fCreditCard%2fCreditCard_Checkout.aspx%3fu%3dbYUA2ZNPjRWk8cGfYP0f3mtXPL5Hbg2wYEs7KPOzZaAbIN7N0iG%252fPZRZSf2yGT1ZIF6MexLIOBS5%252fdK3RaGz6yYBJxM3ID1bZwHcSA98ClgTlePBjWX%252f%252b9Hznf1%252bxPI4JfLEz5j8ppBUI4QxWvZn3na5tshm7NKcqtS3q7o0x8c8bkw4nk134lrKXkP0%252fxwBBSWEJZc1SyMUMLM6OZeUtZTfoiL98QUCDsbg588mrwq1SQIhwf%252beM07ba13J%252fg8Jd%252blZhfG%252fsxnk7wHTy4crt6vP5hLTp4MuPBtLil76Pk4NJ%252bjLxjZjiEJKpVnzp6LL1VBv2JPXgeL%252b02eAVVdaBzKrYxIBrx%252bwQYqUV3y7I0aHi527AS7SFXfxPCEC74yjhpPUkxDVrSQQW0sU4EkxO1TYTWLSWHQE4Ub81L2hIIEnvArZZYB7Py1AzaZxXC5Z6Td8Qg9Dhd%252f5sH4YZAD1nVvm3gbCnD9vPO1ehHuiXAJG Y9Ty9SXqUsEDRN%252bzJ3DIjXQ5Eo1zT6zlx8HUNgxvNv9c9qaJTS5HsVy3GXR01dSLvV6W%252fv6M4I7C1oi%252be%252ftX&OrderID=TESTDk7p8&PaymentID=1058754&Checksum=b6d41402231053d5c0ba7356e03dd4988c4ff8a7
```



Sample of a returned string for XML format in case of error:

```
<?xml version="1.0" encoding="utf-16"?>
<Checkout>
<Status>ERR</Status>
<ErrCode>IC_ERR: Checksum is not valid</ErrCode>
<URL />
<OrderID>TESTDx87Z</OrderID>
<PaymentID />
<Checksum />
</Checkout>
```

Sample of a returned string for XML format in case of valid request:

```
<?xml version="1.0" encoding="utf-16"?>
<Checkout>
<Status>OK</Status>
<ErrCode />
<URL>https://pay.ICEPay.eu/CreditCard/CreditCard_Checkout.aspx?u=bYUA2ZNPjRWk8cGfYP0f3mtXPL5Hbg2wYEs7KPOzZaAbIN7N0iG%2fPZRZSf2y
GT1ZKDjOnNmHul%2fInv4aGZGP3Y0fXwoSxdP7HV1fXmZEUB6dWyC3I65Jme6n4HL1WimiwGdjA5BL7u7S4e%2fQO7cHPNtAM1EZSw9uahMQ1wltD%
2b4qXav2%2bS3leKmNBdG8mTmthgE0SZsdgMge6HvyBrKKYYbZhVXNfGp9QfIIW1vG3FYopxllturjcoQtqINikTNg5VZy%2bcV66fwETJfNkUOcwDcimuc
MZ8FhUnd%2bSWJBBzdVDwemz6nfjPpaKSMmdbc7FiDrlVOVnW29x6Almj%2bL7qRIOGQ%2btPpO5t7HziPi4sZbqGITBYSHh4IH56%2fDkGI%2fBHICKh
N1WAAtXqFg2o791qqqMKVbpD8QN6mi37gRAECzFzTgfo96qMEEIB5yTinGkdUxdXFj6YMo4YXn%2bWcg2zZhw0hgNxtw5UKQsq1LyguhrL0VtGZnlks5nC
M4IODOdKvVfhjPGZ%2fqNyAqhcsOFqg%2bSdjjqN3h9HabmXdfSXRC%2bLvGmlFvHIJUbkiilmwt</URL>
<OrderID>TESTp3W8C</OrderID>
<PaymentID>1058753</PaymentID>
<Checksum>dd99c33e3e32449b87540b6b690224cf83d8641e</Checksum>
</Checkout>
```



5.4 PARAMETERS SENT TO IC_URLCOMPLETED AND IC_URLERROR

When everything goes as it should, after a successful payment, the user will be redirected to the Merchant defined page that was sent in the IC_URLCompleted parameter to Checkout.aspx, or the value in the ICEPAY web interface.

When an error occurs, a payment is canceled or interrupted, a timeout occurs, parameters are sent incorrectly or something else is wrong, the user will be redirected to the merchant defined page that was sent in the IC_URLError parameter to Checkout.aspx, or the value in the ICEPAY web interface.

All values below will be appended as GET parameters. The append process takes into account whether there already is a “?” or not in the IC_URLCompleted / IC_URLError page, so it is possible to construct this URL with some parameters of your own. Do make sure however that there are no name-clashes.

5.4.1 PARAMETERS

The return parameters (step 9a/9b/4b in the payment flowchart diagram) are:

Parameter	Description	Data type	Sample
Status	Transaction status. Possible values are OK, OPEN, ERR	String(10)	OK
StatusCode	A short description of the transaction status	String(100)	Completed
Merchant	Your MerchantID	Numeric	10000
OrderID	Value of IC_OrderID which was sent to Checkout.aspx	String(10)	1234567
PaymentID	The unique numeric value that identifies this payment in our system.	Numeric	12345
Reference	Value of IC_Reference which was sent to Checkout.aspx	String(50)	Z1234567
TransactionID	This value is created by the payment method provider / bank and showed on the users bank statement	String(50)	3001233213132
Checksum	A checksum is generated (see 5.4.2) over the return parameters, so that you can verify the authenticity of the returned values.	String(40)	da39a3ee5e6b4b0d3255bfe95601890afd80709



5.4.2 CHECKSUM

The Checksum is generated as:

```
SHA1( Encryptioncode + | + IC_Merchant + | + Status + | + StatusCode + | + OrderID + | + PaymentID + | + Reference + | + TransactionID )
```

5.4.3 SAMPLE

Sample of a success URL:

```
http://localhost/success.aspx?Status=OK&StatusCode=Completed&Merchant=10000&OrderID=1000000920&PaymentID=1058262&Reference=XYZ123&TransactionID=0030825521452120&Checksum=da39a3ee5e6b4b0d3255bfef95601890afd80709
```

```
http://localhost/success.aspx?Status=OK&StatusCode=AUTHORISED&Merchant=10000&OrderID=1000000940&PaymentID=1058287&Reference=&TransactionID=8512210510933341&Checksum=c60e045f9d2a988caea94429acf82ad86d67528f
```

Sample of an error URL:

```
http://localhost/error.aspx?Status=ERR&ErrCode=IC_ERR%3a+Checksum+is+not+valid&URL=&OrderID=TESTx4QYd&PaymentID=&Checksum=
```

```
http://localhost/error.aspx?Status=ERR&StatusCode=Cancelled&Merchant=10000&OrderID=1000000971&PaymentID=1058549&Reference=&TransactionID=0030824521482120&Checksum=e82dc18cbb1839c99356b834d0bd45d08420a973
```



5.5 HANDLING THE POSTBACK NOTIFICATIONS, PARAMETERS SENT TO IC_POSTBACK

While the user is doing his payment, ICEPAY will report all status changes back to the Merchant with a **server-to-server** POST to the URL as defined in IC_Postback. You can set this URL in the web administration at <https://www.icepay.eu/> This section provides you with guidelines on how your Postback Script should handle incoming Postback Notifications

Note: The script in URLPostback should not generate any output or errors. It is very important that this script works well, otherwise payments will be aborted!

We would advise you to always return HTTP 200 (The default “OK” response of a web server. An empty page does exactly that.), if you have processed the request to prevent our server to repeat the request. Only return HTTP error if you have (good) reason for it (like db connection error or update error).

Note: Our postback uses the UTF-8 character encoding format.

5.5.1 PARAMETERS

The parameters (step 7 in payment flowchart diagram) are:

Parameter	Description	Data type	Sample	Can be empty
Status	Payment status as OK, OPEN, ERR, REFUND, CBACK	String(10)	OPEN	N
StatusCode	A short description of the status. We will use the codes as received from the payment method provider. Check Appendix A for most common values	String(100)	Completed	N
Merchant	Your MerchantID	Numeric		
OrderID	IC_OrderID as passed to Checkout.aspx	String(10)	1234567	N
PaymentID	The unique numeric value that identifies this payment in our system.	Numeric	12345	N
Reference	IC_Reference as passed to Checkout.aspx	String(50)	Z1234567	Y
TransactionID	This value is created by the payment method provider / bank and showed on the users bank statement	String(50)		Y
ConsumerName	Name of the bank account owner	String(100)		Y
ConsumerAccountNumber	Last 4 digits of account number from which payment was done, if received from the bank	String(100)		Y
ConsumerAddress	Consumer address/street as filled in payment form	String(100)		Y
ConsumerHouseNumber	Consumer house number as filled in payment form	String(10)		Y
ConsumerCity	Consumer city as filled in payment form	String(100)		Y
ConsumerCountry	Consumer country as filled in payment form	String(100)		Y
ConsumerEmail	Consumer email value as filled in payment form	String(200)		Y
ConsumerPhoneNumber	If available phone number from which payment was made or used in payment form.	String(50)		Y



	In international format as: 31703242323. If CID is hidden you will get {PRIVE}			
ConsumerIPAddress	IP address from which payment form was filled	String(50)	1.2.3.4	Y
Amount	The final paid amount value in whole cents	Numeric	550	Y
Currency	The currency in which the amount is represented.	String(3)	EUR	Y
Duration	Represents the call duration (if available), in whole seconds, in phone payment methods.	Numeric	0	Y
PaymentMethod	Which payment method was used	String(20)	CREDITCARD	N
Checksum	A checksum is generated over the return parameters, so that you can verify the authenticity of the returned values.	String(40)		N

5.5.2 POSSIBLE STATUSES

The Postback Notification contains a parameter called Status. You will most likely want to use this parameter to update the status of your payment in your local database. The Status that is returned by ICEPAY can be only one of the following codes:

Status	Description
OPEN	The payment is not yet completed. After some time you will receive a Postback Notification which contains the OK or ERR status. The time varies depending on the payment method that was used.
OK	The payment has been completed.
ERR	The payment was not completed successfully or expired. It cannot change into anything else.
RE-FUND	A payment has been successfully refunded. You will receive a different PaymentID parameter but all the other parameters remain the same.
CBACK	The consumer has filed a chargeback via their issuing bank.
VALID-DATE	The payment is awaiting validation by the consumer by means of a validation code returned by ICEPAY. Currently, this status is only used by SMS payments. You can safely ignore postbacks with this status if you have integrated ICEPAY using the Checkout.aspx method.

You should ignore all other statuses. If a new status is introduced, you will be notified by your account manager.

5.5.3 DETAILED STATUS DESCRIPTION



The Postback Notification also contains a parameter called `StatusCode`. This is an additional parameter which gives you a more detailed description regarding the status of a payment. Your Postback Script should **NOT** rely on the content of this parameter to decide what to do as it may change from time to time. It is purely informational. Instead, you should always use the status parameter as described in paragraph 5.5.2

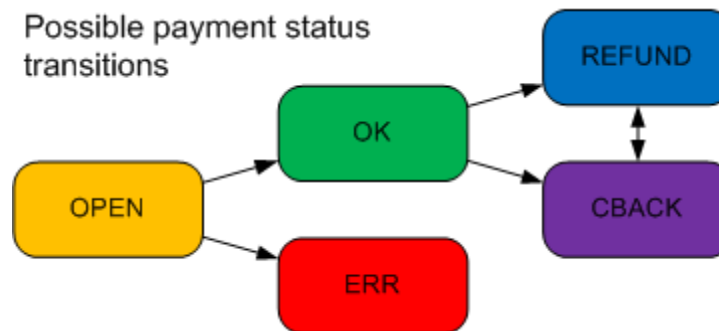


Examples of StatusCode content:

- Canceled
- Completed with user hangup
- Money received. Bank statement ID: 12345
- Payment aborted by user
- Success

5.5.4 POSSIBLE STATUS TRANSITIONS

If your Postback Script synchronizes the information from Postback Notifications with your local data storage, then you must only do that according to following diagram:



If your transaction is already flagged as OK, it will **NEVER** transition into ERR. If this ever occurs then you must ignore it. You might consider logging this event and triggering internal alarm for intrusion inspection, but never generate HTTP error.

5.5.5 CHECKSUM

The Checksum is generated as:



```
SHA1( Encryptioncode + | + IC_Merchant + | + Status + | + StatusCode + | + OrderID + | + PaymentID + | + Reference + | + TransactionID + | + Amount + |  
+  
Currency + | + Duration + | + ConsumerIPAddress )
```

5.5.6 SAMPLE

Here we will show the sample of POST data which is “pushed” to your IC_Postback page during a payment status change event.

```
Status=OK  
StatusCode=AUTHORISATION  
Merchant=10000  
OrderID=10761  
PaymentID=1057135  
Reference=  
TransactionID=1215209043726587  
ConsumerName=  
ConsumerAccountNumber=7211  
ConsumerIPAddress=127.0.0.1  
Amount=740  
Currency=EUR  
Duration=0  
Checksum=a32d1886b763ef83126615cd344d4c9d4a9dac6b
```



APPENDIX

A. CODE SAMPLES, DOCUMENT UPDATES AND OTHER DOCUMENTS

To get an up to date version of this document you can visit following URL

http://developer.icepay.eu/files/pdf/icepay_implementation_guide.pdf

Document with supported parameter combinations and amount ranges per transaction:

http://developer.icepay.eu/files/pdf/ICEPAY_Supported_Parameters_Sheet.pdf

Error codes and troubleshooting:

http://developer.icepay.eu/files/ICEPAY_Error_Codes.xls

Ready to use libraries (PHP and .NET) you can find here:

<http://developer.icepay.eu/>

Knowledge base and support site:

<http://support.icepay.eu/>

Code samples in PHP, C# and VB.NET at this location:

<http://developer.icepay.eu/files/CodeSamples.zip>



Up to date version of terms and conditions you find at:

http://www.icepay.eu/documents/ICEPAY_algemene_voorwaarden.pdf

B. CHANGELOG

Version	Date	By	Changes
1.3.0	20-02-2012	NH	Updated document for MISTERCASH and removed Appendix A
1.2.6	06-06-2011	ST	Updated document for PayPal
1.2.5	05-31-2011	ST	New Layout
1.2.4	01-03-2011	AB	Updated iDEAL issuers list Updated server IP range
1.2.3	03-02-2011	ST	Added PaySafeCard to issuers list.
1.2.1	30-11-2009	HH	Added description about the new VALIDATE payment status.
1.2.0	26-10-2009	AB	Added ASNBANK, SNSREGIOBANK to the iDEAL issuers list Some content moved to external documents
1.1.8	23-09-2009	AB	Added Friesland bank to the iDEAL issuers list
1.1.7	16-09-2009	AB	PPM rates fixed
1.1.4	11-05-2009	HH	SMS allowed amounts updated
1.1.3	04-05-2009	AB	Issuers table updated with new payment methods and allowed amount ranges
1.1.2	23-02-2009	AB	POSTBANK renamed to ING according to new iDEAL specifications
1.1.0	20-01-2009	AB	New payment statuses are introduced: REFUND, CBACK All previous text changes
1.0	12-09-2008	AB	Document restyled, reformatted and completely revised.
0.1	14-12-2007	AB	Initial document

C. CONTACT

ICEPAY Global Payment Solutions
Wassenaarseweg 7
2596 CD Den Haag
Netherlands

Tel: +31 70 3242323
Fax: +31 70 3264617
Sales: sales@icepay.eu
General: info@icepay.eu

Web: www.icepay.eu
Support: <http://support.icepay.eu>
<http://developer.icepay.eu>
<http://forum.icepay.eu>