



Direct Payment Library  
UNIX SDK

Version 1.1  
April 2004

This manual and accompanying electronic media are proprietary products of Optimal Payments Inc. They are to be used only by licensed users of the product.

© 1999–2004 Optimal Payments Inc. All rights reserved.

The information within this document is subject to change without notice. The software described in this document is provided under a license agreement, and may be used or copied only in accordance with this agreement. No part of this manual may be reproduced or transferred in any form or by any means without the express written consent of Optimal Payments Inc.

FirePay and FireCash are registered trademarks of Optimal Payments Inc. All other names, trademarks and registered trademarks are the property of their respective owners.

Optimal Payments Inc. makes no warranty, either express or implied, with respect to this product, its merchantability or fitness for a particular purpose, other than as expressly provided in the license agreement of this product. For further information, please contact Optimal Payments Inc.

### **International Head Office**

Optimal Payments Inc.  
2 Place Alexis Nihon, Suite 700  
Westmount, Quebec H3Z 3C1  
Canada

Tel.: (514) 380-2700

Fax: (514) 380-2760

Email: [info@optimalpayments.com](mailto:info@optimalpayments.com)

Technical support: [support@optimalpayments.com](mailto:support@optimalpayments.com)

Web: [www.optimalpayments.com](http://www.optimalpayments.com)

### **U.K. Office**

Optimal Payments Ltd.  
Compass House  
Vision Park  
Histon, Cambridge CB4 9AD  
England

Email: [info@optimalpayments.co.uk](mailto:info@optimalpayments.co.uk)

Web: [www.optimalpayments.co.uk](http://www.optimalpayments.co.uk)

### **Hull Office**

Optimal Payments Inc.  
75 Promenade du Portage  
Gatineau, Quebec J8X 2J9  
Canada

### **U.S. Offices**

Optimal Payments Corp.  
1800 West Loop South, #770  
Houston, TX 77027

Optimal Payments Corp.  
225 Franklin, 26th Floor  
Boston, MA 02110

## 1 Installing Direct Payment

Software requirements .....	1-1
Merchant registration .....	1-1
Installing the Direct Payment library .....	1-2
Installing the Perl client .....	1-2
Installing the cgic library .....	1-3
Testing the Direct Payment library .....	1-3
On Perl .....	1-4
On cgic .....	1-6

## 2 Introduction to Direct Payment

What is Direct Payment? .....	2-1
Merchant transactions. ....	2-2
Submitting transaction requests .....	2-2
Failed transactions .....	2-3
Communication failure .....	2-4
Direct Payment library level .....	2-4
Payment service level .....	2-6
No response .....	2-6
Additional features .....	2-7
Secure communications .....	2-7
Configuring for Direct Payment. ....	2-7

## A Error Return Codes

Error messages .....	A-1
Direct Payment library error messages .....	A-1

Payment service error messages . . . . . A-5  
    Action codes . . . . . A-6  
    Error codes and strings . . . . . A-6  
    Suberror codes and strings . . . . . A-15  
    Unmapped suberror codes and strings . . . . . A-19

# Index

# Installing Direct Payment

---

The Direct Payment library for UNIX was developed in C language. It uses OpenSSL libraries to implement SSL communications – implementing strong encryption – with the Optimal Payments payment processor over the Internet. The installation package also provides also C and Perl client samples that use this library to send the transactions to the payment service.

## Software requirements

The Direct Payment library communicates with the payment service over the Internet using SSL. You need the following software:

- Sun Solaris 2.6 and higher; or
- Linux Red Hat 6.0 and higher

If you are using Perl on either of these two platforms, you need:

- Perl 5.004 and higher

## Merchant registration

Before using the Direct Payment library, the prospective merchant must register on the Optimal Payments Web site, by filling out an application form. Our sales department will then contact the merchant to finalize all necessary details. Once registration is complete, the merchant will receive an account ID, an account transaction processing ID and a password. These values are important, as they are used for merchant transaction requests with our transaction processor via the payment library. For more information, see the *Direct Payment Protocol Specification* manual.

## Installing the Direct Payment library



*While most Linux systems provide `/dev/random` and `/dev/urandom` devices – also called Entropy Pool – which are used to generate random numbers used by OpenSSL to generate encryption keys, Solaris does not provide those devices.*

*For Solaris, the Sun patch 105710-01 **must** be installed before you install the Direct Payment library. Then execute the following commands to create `/dev/random` and `/dev/urandom` devices:*

```
cryptorand -r /dev/random  
cryptorand -r /dev/urandom
```

You may need to log on as root or have enough permission to install the package.

### To install the Direct Payment library:

1. Untar the package:

```
tar -xvf sfPaymentSun.tar;  
or  
tar -xvf sfPaymentLinux.tar
```

2. Install the library as follows:

```
cd sfPayment/lib/  
chmod 755 libsfPayment.so  
ln -s /sfPayment/lib/libsfPayment.so /lib/libsfPayment.so
```

## Installing the Perl client

You can use Perl on the Solaris or Linux platform. If you do, you must install the Perl client.

**To install the Perl client:**

1. Place the *sfp.html* and *sf\_logo.jpg* files in your Web server's HTML folder (e.g., /opt/apache/htdocs).
2. Place the following files in your cgi-bin directory, where you can run executables from your Web server:
  - *receive.cgi*
  - *receiver.cgi*
  - *sample.cfg*
  - *sfClientPerl.so*
  - *sfClientPerl.pm*
  - *libsfPayment.so*
3. Make the *receive.cgi* and *receiver.cgi* files executable. You can do this by running the "chmod 755 <filename>" command on these two files.

## Installing the cgic library

**To install the cgic library:**

1. Type:  
cd ClientCgic
2. Change the path of the file *sample.cfg* on line 73 in *sfClientC.c*
3. Type:  
make -f sfClientC.mak

## Testing the Direct Payment library

**In order to test the Direct Payment library, type the following:**

1. cd ../ClientC
2. ./sfClient ../sample.cfg Request

You will see the following on your screen, indicating a successful transaction test:

```
The Request received from client is POST /tms-ts/payService/
merchant/paymentServiceM.cgi HTTP/1.1
host: https://test_site/tms-ts/payService/merchant/
paymentServiceM.cgi:443
Content-type: application/x-www-form-urlencoded
Content-length: 183
```

```
cardType=NEG-VI&cardNumber=4007000000027&amount=4000&
cardExp=12%2F04&operation=P
&merchantTxn=13465879901&clientVersion=1.03&account=401123&
merchantId=testaccoun
t1&merchantPwd=testpwd1
```

The request was processed by the Payment server.

The response returned by the server is:

```
status=SP&authCode=41TEST&authTime=961774635&avsInfo=
A YNA&curAmount=0&amount=
4000&txnNumber=104783&serviceVersion=1.1
```

## On Perl

### In order to use a Web application with Perl:

1. Open the *sample.cfg* file and modify the following parameters:
  - merchantId (“Merchant ID” in your activation email)
  - merchantPwd (“Merchant Password” in your activation email)
  - account (“Account Number” in your activation email)
2. In the *sfp.html* file, locate the action parameter of the form tag and change the path for the location of your *cgi-bin* (e.g., should be like “cgi-bin/receiver.cgi”).
3. Open your Web browser and enter the following URL in the address field: <http://localhost/sfp.html>. (Make sure Apache is running.) The following window opens:

REQUIRED INFORMATION:		ADDITIONAL INFORMATION:	
Credit Card Type:	Visa	Name:	
Credit Card Number:		Street Address:	
Expiry (MM/YY):		Address (cont'd):	
Total Amount (in cents):		City:	
Operation:		Prow/State:	
Merchant Transaction No:		Postal/Zip Code:	
Client Version:		Country:	
		Phone:	
		Email:	

Submit Form   Clear Form   Sample Data

4. Click Sample Data.
5. Click Submit Form.

If the Direct Payment component has been successfully installed, you should receive the results of an authorized transaction.

You will now have to edit the *sfp.html* file according to your store settings for your own transactions.

## On cgic

### In order to use a Web application with C:

1. Complete the following copy commands from the ClientCgic directory:
  - `cp *.html HTML_DIRECTORY`  
(where HTML\_DIRECTORY is the directory that contains the html files on your Web server)
  - `cp *.jpg ICONS_DIRECTORY`  
(where ICONS\_DIRECTORY is the directory that contains the image files on your Web server)
  - `cp sfClientC CGI-BIN_DIRECTORY`  
(where CGI-BIN\_DIRECTORY is the directory that contains the cgi files on your Web server)
2. Open your Web browser and enter the following URL in the address field: `http://your_server_name/sfClientC.html`. The following window opens:

REQUIRED INFORMATION:		ADDITIONAL INFORMATION:	
Credit Card Type:	<input type="text" value="Visa"/>	Name:	<input type="text"/>
Credit Card Number:	<input type="text"/>	Street Address:	<input type="text"/>
Expiry (MMYY):	<input type="text"/>	Address (cont'd):	<input type="text"/>
Total Amount (in cents):	<input type="text"/>	City:	<input type="text"/>
Operation:	<input type="text"/>	Prov/State:	<input type="text"/>
Merchant Transaction No.:	<input type="text"/>	Postal/Zip Code:	<input type="text"/>
Client Version:	<input type="text"/>	Country:	<input type="text"/>
		Phone:	<input type="text"/>
		Email:	<input type="text"/>

3. Click Sample Data.
4. Click Submit Form.

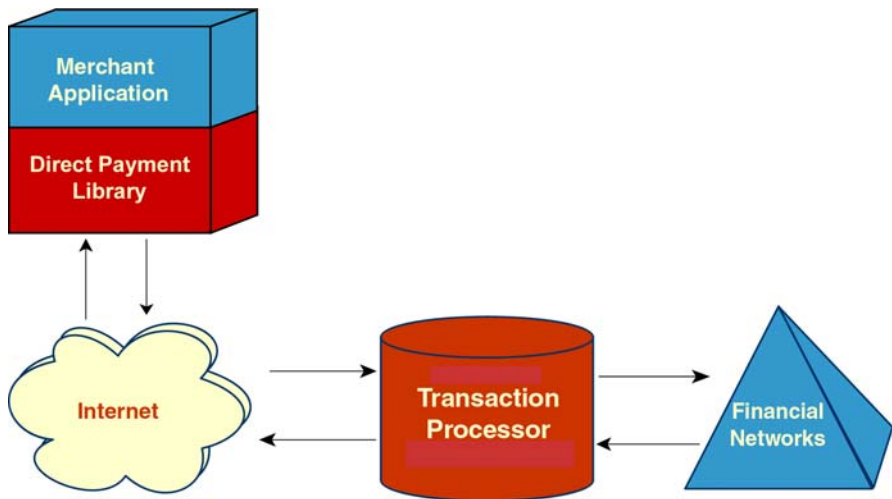
If the Direct Payment component has been successfully installed, you should receive the results of an authorized transaction.

# Introduction to Direct Payment

---

## What is Direct Payment?

A merchant typically needs to send transaction requests in support of business transactions to a payment service. The merchant sends transaction requests (e.g., Purchase) to the Optimal Payments transaction processor via the Direct Payment library, which acts as a transport medium.



The merchant uses sockets through SSL to communicate securely with the payment server over the Internet.

Transactions are validated at both the Direct Payment library and the payment service levels. For example, the transaction processor sends a response for both successful and failed transactions (e.g., if an invalid card type is supplied). The Direct Payment library sends an error message if information it requires is missing or incorrect (e.g., if your merchantId value is missing).



*In order to guarantee the security of customer information over the Internet, and in order to use the Direct Payment library and our transaction processor, the merchant's Web site must be secure. Optimal Payments ensures the security of information transmitted between the merchant application and the payment service, but the merchant is responsible for securing information transmitted between the customer and the merchant's Web site (by, for example, having a certificate from a recognized certificate authority such as VeriSign).*

## Merchant transactions

The merchant application sends a variety of information to the payment service via the Direct Payment library. The merchant application uses the *processRequest* function to pass the required parameters to the Direct Payment library, which performs the following five tasks:

1. Sends the required transaction parameters and the information from the configuration file to the Direct Payment library (see *Configuring for Direct Payment* on page 2-7)
2. Opens an SSL connection, connects, and sends the request to the payment service using sockets
3. Waits for the response from the payment service and communicates it to the Direct Payment library, which then passes it to the merchant application
4. Communicates the request status from the payment service to the merchant application
5. Communicates the error description, if applicable, to the merchant application

The Direct Payment library supports merchant transactions such as Purchase, Failure Lookup, and Query. For information on specific transaction requests see the *Direct Payment Protocol Specification* manual.

## Submitting transaction requests

The following is a brief overview of the process of submitting a transaction request via the Direct Payment library:

1. Register with the payment service. See *Merchant registration* on page 1-1 for more information.
2. Complete any fields in the configuration file that are either empty or that have values you want changed. See *Configuring for Direct Payment* on page 2-7 for more information.
3. Initiate a request by following the steps in *Testing the Direct Payment library* on page 1-3.
4. The Direct Payment library reads the parameters set in the merchant's configuration file and builds the request from the merchant application.
5. The Direct Payment library then establishes a secure connection with us over the Internet.
6. When the secure connection is established, the Direct Payment library sends the request, and waits for the result.
7. We process the transaction request and send the results back to the Direct Payment library, which in turn relays this information to the merchant application. For more information on the responses to each transaction type, see the *Direct Payment Protocol Specification* manual.

## Failed transactions

Transaction requests are made to the payment service via the Internet. Normally, the merchant receives a response indicating that the transaction was successful. If the transaction request fails, an error response is issued to the merchant. This response includes an error code and an error string.

There are four possible failure scenarios. The action to take varies with the type of error encountered.

- Communication failure
- Failure at the Direct Payment library level
- Failure at the payment service level
- No response is returned

## Communication failure

In some instances, a transaction request is made, but due to the nature of the Internet, the transaction gets interrupted as a result of a communication failure. If this should happen, the Direct Payment library supports an active, automatic method for recovery. In the case of a communication failure, the following occurs:

- If the communication error occurs before the transaction request has been sent, the Direct Payment library automatically reinitiates communication.
- If the communication error occurs after a request has been sent but before a response has been received, the Direct Payment library sends a Failure Lookup request to verify that the transaction request has been sent, and to resend the request if the transaction has not been completed.

## Direct Payment library level

If the Direct Payment library returns an error, it indicates one of two things:

- Information required to execute the transaction request is missing (e.g., the *merchantId* field has not been filled out). In this case, the transaction request is not sent to the payment service.
- The transaction request is sent to the payment server, but some further problem prevents it from being processed. In this case, both a Direct Payment library and a payment service error message are returned.

Two important pieces of information are returned to the merchant application:

1. A status level number, indicating the nature of the error:

Status Level	Description
0	Transaction was processed successfully.
-1	The configuration file is missing.
-2	Some parameters are incorrect or missing.
-3	A problem occurred with sockets.
-4	A connection error occurred.
-5	Communication problem – SSL read error.
-6	Communication problem – SSL write error.
-7	Communication problem – SSL connect error.
-8	Communication problem – SSL context error.
-9	Communication problem – SSL session error.
-10	Communication problem – SSL PRNG not seeded.
-11	Unrecoverable error.
-12	Error occurred during the transaction.
-13	Manual intervention is required to process the transaction.
-14	The transaction was aborted/rejected by the payment service.
-15	Invalid URL.
-16	Server encountered internal error when processing the request.
-17	HTTP version not supported.
-18	Bad HTTP request due to invalid syntax.
-19	Authorization failed – user authentication required.
-20	Web server refuses to fulfill the HTTP request.

Status Level	Description
-21	Proxy authentication required.
-22	Server is down.
-23	Web server cannot fulfill the request.
-24	Entropy pool device not found.
-25	Proxy server specified invalid.
-26	Communication error during send/receive

2. An error code and matching error string, describing the error more exactly. See *Direct Payment library error messages* on page A-1 for more information on the appropriate action to take.

## Payment service level

An error at the payment service level indicates that the communication between the Direct Payment library and the payment service was successful, and that the transaction request was posted. However, an error of some sort has occurred with the payment service. Should this occur, the merchant receives a response from the payment service that includes an error code and an error string.

See *Payment service error messages* on page A-5 for more information on the appropriate action to take.

## No response

Occasionally, the merchant receives no response at all. In this case the user does not know whether or not the request was processed and does not have the payment service transaction number associated with the request (which would have been assigned if the request had in fact been processed). In such a case, the merchant must make a request for more information about the failed transaction, using the failure recovery transactions. For more information, see the *Direct Payment Protocol Specification* manual.

## Additional features

In addition to transmitting merchant transaction requests, the Direct Payment library offers the following features:

- Secure communications (see *Secure communications* on page 2-7)
- Active recovery in case of communication failure (see *Communication failure* on page 2-4)

### Secure communications

SSL protocols provide secure data communications through data encryption and decryption. The communication between the merchant application and the Optimal Payments transaction processor is made with HTTP requests over the Internet, via TCP/IP, through a secure channel, using SSL.

## Configuring for Direct Payment

When you install the Direct Payment library, the configuration file is also installed. At this point you determine the pathname for your Direct Payment library configuration file. In order to build and send transaction requests, the Direct Payment library must read the merchant's configuration file. Ensure the following fields are completed:

Field	Description
merchantId	Account transaction processing ID See <i>Merchant registration</i> on page 1-1.
merchantPwd	Account transaction processing password See <i>Merchant registration</i> on page 1-1.
account	Merchant account ID. See <i>Merchant registration</i> on page 1-1.
Cipher	Specified cipher. Default=RC4-MD5

Field	Description
PaymentServerURL	The payment gateway URL. This is given in the configuration file at installation.
PaymentServerPort	Your payment server port number. Default=443
Timeout	Time, in seconds, the Direct Payment library waits for a response from the payment service before terminating communication. Default=120. Range=60–180.  NOTE: You will not receive an error message if you set the TimeOut value outside of this range. Optimal Payments recommends staying within this range, however.
Retries	Number of times the Direct Payment library tries to establish communication before returning an error message. Default=3. Range=2–5.  NOTE: You will not receive an error message if you set the Retries value outside of this range. Optimal Payments recommends staying within this range, however.
LogLevel	The level of information that is logged by the Direct Payment library in the UNIX syslog. Three levels are possible: <ul style="list-style-type: none"> <li>• Information – logs successful transactions.</li> <li>• Error – logs failed transactions.</li> <li>• Trace – logs all transactions.</li> </ul> Default=Trace
EntropyPool	The device that provides random numbers to build an encryption key.  /dev/random or /dev/urandom

<b>Field</b>	<b>Description</b>
ProxyServer	Your proxy server name (only if you use a proxy server).
ProxyPort	Your proxy server port number (only if you use a proxy server).
HTTPVersion	The HTTP version used.



# Error Return Codes

---

## Error messages

There are two classes of error messages associated with the use of the Direct Payment library:

- Direct Payment library errors
- Payment service errors, related to processing the request

The status, error code, and error string are returned to the merchant application, in the *processRequest* interface.

## Direct Payment library error messages

The following table lists all errors that can be returned by the Direct Payment library.

Error Code	Error String	Action
ERROR(50)	PaymentServerURL is missing.	Add our payment processor's URL to the configuration file.
ERROR(51)	PaymentServerPort is missing.	Add the payment server port to the configuration file.
ERROR(52)	merchantId is missing.	Add your merchant ID to the configuration file.
ERROR(53)	ProxyServer is missing.	Add the proxy server name to the configuration file.
ERROR(54)	ProxyPort is missing.	Add the proxy server port number to the configuration file.
ERROR(55)	Cipher is missing.	Add the cipher value to the configuration file.

<b>Error Code</b>	<b>Error String</b>	<b>Action</b>
ERROR(56)	HTTPVersion is missing.	Add the HTTPVersion parameter to the configuration file.
ERROR(57)	EntropyPool is missing.	Add the EntropyPool parameter to your configuration file.
ERROR(58)	PaymentServerPort is invalid.	Please verify the PaymentServerPort.
ERROR(59)	ProxyPort is invalid.	Please verify the ProxyPort.
ERROR(60)	HTTPVersion is invalid.	Please verify the HTTPVersion.
ERROR(61)	Configuration file not found.	Verify the path to the configuration file.
ERROR(62)	Reading error occurred in the configuration file.	Retry. If error persists, contact technical support.
ERROR(63)	Error opening the configuration file.	Retry. If error persists, please contact technical support.
ERROR(64)	Configuration file not supplied.	Specify the name of the configuration file as a parameter when calling the <b>init()</b> method.
ERROR(65)	merchantPwd is missing.	Add the merchant password parameter to the configuration file.
RROR(66)	account is missing.	Add the merchant account ID parameter to the configuration file.
ERROR(67)	Permission denied to create the socket.	Verify your authorization with your system administrator.
ERROR(68)	socket(): Insufficient resources (per-process table is full).	Retry later.
ERROR(69)	socket(): Insufficient resources (memory not available).	Retry later.
ERROR(70)	socket(): Insufficient resources (stream resources unavailale)	Retry later.

<b>Error Code</b>	<b>Error String</b>	<b>Action</b>
ERROR(71)	socket(): Protocol not supported.	If error persists, please contact technical support.
ERROR(72)	Cannot create socket.	Retry later.
ERROR(73)	connect(): Permission denied.	Verify your authorization with your system administrator.
ERROR(74)	connect(): Address already in use.	Retry later. If error persists, please contact technical support.
ERROR(75)	Connection refused by the server.	If error persists, please contact technical support.
ERROR(76)	connect(): Insufficient resources.	Retry later.
ERROR(77)	The server exited.	Retry later.
ERROR(78)	Connection establishment timed out.	Retry later.
ERROR(79)	Address not available on the remote server.	If error persists, please contact technical support.
ERROR(80)	The network is not reachable from this host.	If error persists, please contact technical support.
ERROR(81)	The connection was interrupted.	Retry later.
ERROR(82)	The socket is already connected.	Retry later.
ERROR(83)	Cannot connect to the server.	Retry later.
ERROR(84)	SSL_CTX_new(): Cannot create SSL context.	Retry later. If error persists, please contact technical support.
ERROR(85)	SSL_new(): Cannot create SSL session.	Retry later. If error persists, please contact technical support.
ERROR(86)	SSL_connect(): Cannot connect to SSL socket.	Retry later. If error persists, please contact technical support.

<b>Error Code</b>	<b>Error String</b>	<b>Action</b>
ERROR(87)	SSL_write(): Cannot write to the socket.	Retry later. If error persists, please contact technical support.
ERROR(87)	SSL_read(): Cannot read from the socket.	Retry later. If error persists, please contact technical support.
ERROR(89)	RAND_seed(): Random number generator not seeded with at least 128 bits of randomness.	Retry later.
ERROR(90)	URL specified is invalid	Verify that you have the correct URL in your configuration file.
ERROR(91)	Network is down.	Retry. If error persists, please contact technical support.
ERROR(92)	Request could not be understood by the server.	Verify your URL and/or HTTPHeader parameters.
ERROR(93)	Request requires user authentication.	Verify your proxy server parameters.
ERROR(94)	Server refuses to fulfill your request.	Retry. If error persists, please contact technical support.
ERROR(95)	Proxy authentication required.	Verify your proxy server parameters.
ERROR(96)	Server encountered internal error.	Retry. If error persists, please contact technical support.
ERROR(97)	HTTP protocol version not supported by the server.	Verify your HTTPVersion parameter in the configuration file.
ERROR(98)	Service unavailable.	Retry later.
ERROR(99)	Server encountered problem fulfilling your request.	Verify your request parameters.
ERROR(100)	Entropy pool file/device not found.	Verify your EntropyPool parameter.
ERROR(101)	The ProxyServer you specified is invalid.	Please verify your ProxyServer parameter.

<b>Error Code</b>	<b>Error String</b>	<b>Action</b>
ERROR(102)	Cannot communicate through the proxy server.	Please verify your ProxyServer parameter.
ERROR(103)	The server failed to process your transaction.	Verify the parameters in your request. Retry.
ERROR(104)	Request sent and no response received.	Please do a manual Failure Lookup/Query.
ERROR(105)	Transaction was aborted/rejected by server.	Verify your request parameters.
ERROR(106)	Manual intervention required to process your request.	Please contact technical support.

## Payment service error messages

If, after sending a transaction request, you receive an error message from the payment service, some or all of the following parameters are returned (in addition to some request-specific parameters):

<b>Parameter</b>	<b>Description</b>
status	<b>E</b> indicates that an error occurred.
errCode	An integer value associated with the error that occurred.
errString	String that describes the error that occurred.
subError	Lower level error that occurred. This value is only used when trying to resolve issues in co-operation with technical support.
subErrorString	String that describes the lower level error that occurred. This value is only used when trying to resolve issues in co-operation with technical support.
clientVersion	The version of the protocol that the payment service is running.

## Action codes

The payment service returns an error code and an error string for any error encountered. There is also an action code associated with each error (not returned by the payment service). In the table in Error codes and strings below, find the error code returned to you in order to find the action code associated with it.

The meanings for the action code abbreviations are as follows:

- **AR** = Authorization Refused. The card cannot be authorized. Ask the user to verify credit card information or to use a different credit card.
- **CP** = Customer Parameter. The customer has provided incorrect information. Ask the customer to correct the information.
- **IE** = Internal Error. There is a problem on the system that you should report to technical support. You should also determine the status of the transaction using a Transaction Lookup request.
- **MP** = Merchant Parameter. Your application has provided incorrect information. Verify your information.
- **SR** = Service is Restarting. Please retry later.

## Error codes and strings

The table immediately below contains all the error codes and error strings that might be returned while sending transaction requests, in addition to action codes (which are not returned by the payment service). The right-most column lists the action codes associated with each error.

Error Code	Error String	Description	Action Code
1	Error in HTTP environment.	HTTP level used not supported by server side. Should not occur.	IE

<b>Error Code</b>	<b>Error String</b>	<b>Description</b>	<b>Action Code</b>
2	No response from process within timeout settings. Please do a Transaction Lookup to determine the transaction status.	After the transaction was sent, no response was received, because the transaction was never processed. The clearing network could be down.	IE
3	Payment service is currently restarting. Retry later. If the problem persists, please contact technical support.	Our gateway process connecting to a clearing house is temporarily down – most likely due to a restart because of connectivity problems with the clearing house.	SR
4	Could not read configuration file. Please contact technical support.	Server side error. Should not occur.	IE
5	Request method Get not allowed.	Only POST method is supported.	MP
20	Remote validation error. Please verify request parameters.	One of the required parameters is not valid.	MP
21	Request validation failed. Please verify request parameters.	One of the required parameters is not valid.	MP
30	Request processing failure. Please contact technical support.	Server side error. Unlikely to occur, but could happen as a result of a configuration error.	IE
31	Request processing failure. Please contact technical support.	Server side error. Should not occur.	IE
32	Request not accepted. Please verify request parameters.	This error occurs if the request comes from an IP not configured for the merchant.	MP
33	Request processing failure. Please contact technical support.	Server side error. Should not occur.	IE

<b>Error Code</b>	<b>Error String</b>	<b>Description</b>	<b>Action Code</b>
34	Authorization refused.	This error usually results from a hard decline from the clearing house, or from declines due to fraud prevention measures. A sub-error code occurs in the latter case.	AR
56	Invalid amount. Please verify request parameters.	An amount greater than the range supported was entered.	IE (MP)
57	Invalid CVD indicator. Please verify request parameters.	An incorrect value was used for the <i>cvdIndicator</i> parameter.	IE (MP)
58	Invalid CVD value. Please verify request parameters.	An incorrect value was used for the <i>cvdValue</i> parameter.	IE (MP)
63	Invalid account ID. Please verify request parameters.	Some account ID values sent with the transaction do not correspond with the values stored in the our database (e.g., incorrect <i>merchantPwd</i> entered).	MP
91	Invalid payment information. Please verify request parameters	The card number, the brand, expiry date, or a combination thereof is incorrect. The suberror text describes the problem in more detail.	CP
92	Invalid payment method. Please verify request parameters.	A transaction was attempted with an incorrect value entered for the payment method ( <i>payMethod</i> ) parameter.	MP
93	Invalid card type. Please verify request parameters.	This error results when a transaction is attempted with a card type that is not supported.	CP
101	Internal error. Please contact technical support.	Server side error. Should not occur.	IE

<b>Error Code</b>	<b>Error String</b>	<b>Description</b>	<b>Action Code</b>
111	Could not assign name. Please verify request parameters.	A transaction was attempted with the wrong data type entered for the name parameter.	CP
113	Could not assign address. Please verify request parameters.	A transaction was attempted with the wrong data type entered for the address parameter.	CP
116	Could not assign province. Please verify request parameters.	A transaction was attempted with the wrong data type entered for the province parameter.	CP
117	Could not assign zip. Please verify request parameters.	A transaction was attempted with the wrong data type entered for the zip parameter.	CP
118	Could not assign country. Please verify request parameters.	A transaction was attempted with the wrong data type entered for the country parameter.	CP
119	Could not assign email. Please verify request parameters.	A transaction was attempted with the wrong data type entered for the email parameter.	CP
120	Could not assign phone number. Please verify request parameters.	A transaction was attempted with the wrong data type entered for the phone number parameter.	CP
121	Could not assign merchantTxn. Please verify request parameters.	A transaction was attempted with the wrong data type entered for the <i>merchantTxn</i> parameter.	MP
130	Invalid expiry date value. Please verify request parameters.	The expiry date is incorrect.	MP
131	Operation not supported. Please verify request parameters.	The transaction attempted is unknown (Purchase or Credit are examples of known transaction types), or the account is not configured for the transaction attempted.	MP

<b>Error Code</b>	<b>Error String</b>	<b>Description</b>	<b>Action Code</b>
132	Missing mandatory parameters for operation. Please verify request parameters.	A field that is mandatory for the transaction (e.g., <i>cardType</i> ) was not sent with the transaction.	MP
133	Invalid amount format. Should be integer. Please verify request parameters.	The amount of a transaction must be given with no decimal (e.g., \$4.95 = 495).	MP
134	Invalid client version. Please verify request parameters.	The <i>clientVersion</i> parameter must be set to 1.1 in order to use current functionality.	MP
137	Invalid zip code length. Please verify request parameters.	The <i>zip</i> parameter must be a maximum of 10 alphanumeric characters.	MP
138	Invalid zip code length. Please verify request parameters.	The <i>zip</i> parameter must be a maximum of 10 alphanumeric characters.	MP
139	Invalid expiry date format. Please verify request parameters.	The format for the <i>cardExp</i> parameter must be "MM/YY". E.g., September 2003 = 09/03	MP
161	Not authorized to make request. Please verify request parameters.	The user name and/or password included with the Settlement transaction request are not correct. These are the <i>merchantId</i> and <i>merchantPwd</i> parameters, respectively.	MP
163	Invalid txnNumber. Please verify request parameters.	The authorization number included with the Settlement transaction request is not correct or cannot be found.	MP
174	Request failed. Please contact technical support.	Server side error. Should not occur.	IE

<b>Error Code</b>	<b>Error String</b>	<b>Description</b>	<b>Action Code</b>
175	Requested Settlement exceeds remaining Authorization.	A Settlement transaction request must be equal to or less than the amount remaining to settle on an Authorization.	MP
176	Invalid settlement amount.	A Settlement transaction request must be equal to or less than the amount remaining to settle on an Authorization.	MP
178	Transaction already fully settled.	There is no amount remaining to settle on the original Authorization.	MP
209	Payment brand not in store list.	The transaction request was sent with a credit card type for which the account is not configured.	MP
210	Payment instrument error. Please verify request parameters.	Server side error. Should not occur.	CP
212	Authorization refused – AVS did not match.	This error occurs when AVS fails on a transaction that otherwise would have been successful.	AR
213	The Authorization was aborted.	Server side error. Should not occur.	AR
221	Authorization failed.	The transaction was not authorized. The suberror text describes the problem in more detail.	AR
222	Currency mismatch with store.	Server side error. Should not occur.	MP
234	Settlement refused because credit card did not pass negative database check.	A Settlement was attempted on a credit card that was entered into the negative database after the authorization that you are trying to settle was approved.	MP

<b>Error Code</b>	<b>Error String</b>	<b>Description</b>	<b>Action Code</b>
281	Not authorized to make request. Please verify request parameters.	The user name and/or password included with the Query transaction request are not correct. These are the <i>merchantId</i> and <i>merchantPwd</i> parameters, respectively.	MP
284	Invalid transaction number. Please verify request parameters.	The transaction number included with the Query transaction request cannot be found.	MP
311	Not authorized to make request. Please verify request parameters.	The user name and/or password included with the Transaction Lookup transaction request are not correct. These are the <i>merchantId</i> and <i>merchantPwd</i> parameters, respectively.	MP
321	Not authorized to make request. Please verify request parameters.	The user name and/or password included with the Authorization transaction request are not correct. These are the <i>merchantId</i> and <i>merchantPwd</i> parameters, respectively.	MP
331	Not authorized to make request. Please verify request parameters.	The user name and/or password included with the Credit transaction request are not correct. These are the <i>merchantId</i> and <i>merchantPwd</i> parameters, respectively.	MP
333	Invalid txnNumber. Please verify request parameters.	The authorization number included with the Credit transaction request is not correct or cannot be found.	MP
334	Credit refused because credit card did not pass negative database check.	A Credit was attempted to a credit card that was entered into the negative database after the Settlement that you are trying to credit was completed.	MP

<b>Error Code</b>	<b>Error String</b>	<b>Description</b>	<b>Action Code</b>
345	Requested Credit exceeds remaining funds settled.	A Credit transaction request must be equal to or less than the amount of funds available to credit (i.e., the amount settled for that credit card).	MP
346	Invalid credit amount.	A Credit transaction request must be equal to or less than the amount of funds available to credit (i.e., the amount settled for that credit card).	MP
347	Internal error. Please contact technical support.	Server side error. Should not occur.	IE
348	No settled funds available for credit.	A Credit transaction request can only be made on a credit card that has settled amounts remaining on it.	MP
353	Unknown txnNumber. Please verify request parameters.	The transaction number included with the Settlement transaction request is incorrect.	MP
356	Unknown merchant transaction, already fully credited, or no amount available for credit.	A Credit transaction request was attempted where there were no funds remaining to be settled.	MP
600	Invalid shipment method. Please verify request parameters.	A transaction was attempted with an incorrect value entered for the shipment method ( <i>shipMethod</i> ) parameter.	MP
601	Invalid carrier. Please verify request parameters.	A transaction was attempted with an incorrect value entered for the <i>carrier</i> parameter.	MP

<b>Error Code</b>	<b>Error String</b>	<b>Description</b>	<b>Action Code</b>
651	Invalid previous customer. Please verify request parameters.	A transaction was attempted with an incorrect value entered for the <i>previousCustomer</i> parameter, which indicates whether the customer has previously shopped online with this merchant.	MP
652	Invalid customer ID. Please verify request parameters.	A transaction was attempted with an incorrect value entered for the <i>customerId</i> parameter.	MP
653	Invalid customer IP. Please verify request parameters.	A transaction was attempted with an incorrect value entered for the customer's IP address ( <i>customerIP</i> ) parameter.	MP
701	Invalid product type. Please verify request parameters.	A transaction was attempted with an incorrect value entered for the product type ( <i>productType</i> ) parameter.	MP
702	Invalid product code. Please verify request parameters.	A transaction was attempted with an incorrect value entered for the product code ( <i>productCode</i> ) parameter.	MP
731	Invalid transaction category. Please verify request parameters.	A transaction was attempted with an incorrect value entered for the transaction category ( <i>txnCategory</i> ) parameter.	MP
751	Invalid merchant SIC code. Please verify request parameters.	A transaction was attempted with an incorrect value entered for the ISO Standard Industry Code ( <i>merchantSIC</i> ) parameter.	MP

<b>Error Code</b>	<b>Error String</b>	<b>Description</b>	<b>Action Code</b>
752	Invalid customer account open date. Please verify request parameters.	A transaction was attempted with an incorrect value entered for the parameter indicating the date the customer account was opened ( <i>custAcctOpenDate</i> ).	MP
771	Invalid user data. Please verify request parameters.	A transaction was attempted with an incorrect value entered for a user data (e.g., <i>userData04</i> ) parameter.	MP



*The merchant application should not encounter internal errors during normal operation of the payment service. If they are encountered, contact technical support.*

## Suberror codes and strings

<b>Suberror Code</b>	<b>Suberror String</b>	<b>Action Code</b>
1000	Approval	AR
1001	Unknown response from clearing network	AR
1002	Clearing network response is Reenter	AR
1003	Clearing network response is Referral	AR
1004	Clearing network response is Pickup	AR
1005	Clearing network response is Decline	AR
1006	Clearing network response is Timeout	AR
1007	Card in negative database	AR
1008	Invalid merchant number	AR

<b>Suberror Code</b>	<b>Suberror String</b>	<b>Action Code</b>
1010	CVV2 check failed	AR
1011	Approved with ID	AR
1012	Invalid request	AR
1013	Invalid amount	AR
1014	Invalid account	AR
1015	Retry	AR
1016	Invalid expiry date	AR
1017	PIN invalid	AR
1018	Unauthorized transaction	AR
1019	Max PIN retries	AR
1020	Duplicate transaction	AR
1021	Invalid account match	AR
1022	Invalid amount match	AR
1023	Invalid item number	AR
1024	Item voided	AR
1025	Must balance now	AR
1026	Use duplicate	AR
1027	No duplicate found	AR
1028	Invalid data	AR
1029	No transaction found	AR
1030	Approved but not captured	AR
1031	Approved auth only	AR
1032	Invalid bank ID	AR

<b>Suberror Code</b>	<b>Suberror String</b>	<b>Action Code</b>
1034	Transaction type invalid	AR
1035	Approved debit	AR
1036	DB unavailable2	AR
1037	DB unavailable3	AR
1038	DB unavailable4	AR
1039	Unauthorized user	AR
1040	Invalid card	AR
1041	DB issuer unavailable	AR
1042	Invalid pos card	AR
1043	Account type invalid	AR
1044	Invalid prefix	AR
1045	Invalid FIID	AR
1046	Verify	AR
1047	Invalid LIC	AR
1048	Invalid state	AR
1049	EDC unavailable	AR
1050	DB unavailable1	AR
1051	Scan unavailable	AR
1052	Exceeds max amount	AR
1053	Exceeds max uses	AR
1054	Unable to process	AR
1055	Invalid request for terminal	AR
1056	Invalid date	AR
1057	Invalid format	AR

<b>Suberror Code</b>	<b>Suberror String</b>	<b>Action Code</b>
1058	No pickup	AR
1059	No funds available	AR
1060	Exceed limit	AR
1061	Restricted card	AR
1062	Mac key incorrect	AR
1063	Exceed frequency limit	AR
1064	Retain card	AR
1065	Late response	AR
1067	No share arrangement	AR
1068	Function unavailable	AR
1069	Invalid key	AR
1070	Invalid lifecycle trans	AR
1071	Pin key error	AR
1072	Mac sync error	AR
1073	Security violation	AR
1074	IST unavailable	AR
1075	Invalid issuer	AR
1076	Invalid acquirer	AR
1077	Invalid originator	AR
1078	System error	AR
1079	Duplicate reversal	AR
1081	Credit card is blocked	AR
1082	Credit card is stolen	AR

<b>Suberror Code</b>	<b>Suberror String</b>	<b>Action Code</b>
1083	Credit card is forged	AR
4000	Declined by Risk Management	AR

## Unmapped suberror codes and strings

An unmapped suberror code and suberror string are returned in the event that a suberror response is not mapped to a standard 4-digit suberror code and string. All unmapped suberror codes are between 0 and 999, making them easy to differentiate from our suberror codes, which are all greater than 1000.



## A

---

account ID 2-7  
action codes A-6

## C

---

cgic library  
    installing 1-3  
    testing Direct Payment library 1-6  
Cipher 2-7  
codes  
    action A-6  
    error A-6  
communication failure 2-4  
configuration file 2-7  
    account 2-7  
    Cipher 2-7  
    EntropyPool 2-8  
    HTTPVersion 2-9  
    LogLevel 2-8  
    merchantId 2-7  
    merchantPwd 2-7  
    PaymentServerURL 2-8  
    ProxyPort 2-9  
    ProxyServer 2-9  
    Retries 2-8  
    Timeout 2-8  
configuring for Direct Payment 2-7

## D

---

Direct Payment configuration file 2-7  
Direct Payment library  
    installing 1-2  
    overview 2-1  
    testing 1-3  
    testing on cgic 1-6

testing on Perl 1-4  
transaction failure 2-4

## E

---

EntropyPool 2-8  
error messages A-1  
    action codes A-6  
    error codes and strings A-6  
    payment library A-1  
    payment service A-5  
    status level numbers 2-5

## F

---

Failure Lookup request 2-4  
failure recovery 2-3  
functions  
    processRequest 2-2

## H

---

HTTPVersion 2-9

## I

---

installing  
    cgic library 1-3  
    Direct Payment library 1-1, 1-2  
    Perl client 1-2  
interrupted transaction requests 2-4

## L

---

LogLevel 2-8

## M

---

merchant  
    application 2-2  
    registration 1-1  
    transactions 2-2  
merchantId 2-7  
merchantPwd 2-7

## P

---

parameters  
    error messages, from payment  
        services A-5  
payment service error messages A-5  
payment service failure 2-6  
PaymentServerURL 2-8  
Perl client  
    installing 1-2  
    testing Direct Payment library 1-4  
processRequest function 2-2  
ProxyPort 2-9  
ProxyServer 2-9

## R

---

registering Direct Payment library 1-1  
requirements  
    software 1-1  
response failure 2-6  
Retries 2-8

## S

---

security 2-2, 2-7  
sending transaction requests 2-2  
software requirements 1-1  
SSL protocols 2-7  
status levels, failure 2-5  
submitting transaction requests 2-2

## T

---

testing  
    Direct Payment library 1-3  
Timeout 2-8  
transaction failure  
    communication 2-4  
    Direct Payment library 2-4  
    no response 2-6  
    payment service 2-6  
    status levels 2-5  
transaction requests 2-1, 2-2  
    failed 2-3  
    submitting 2-2  
transaction validation 2-1