

Programmer Manual

V6.1



**Level 3, 267 St Georges Terrace
Perth, Western Australia, 6000
P: (08) 9261 7773
F: (08) 9261 7700**

Table of Contents

Hosted Payment Page -----	3
<u>Header or Background Image Loader</u> -----	6
<u>Hosted Payment Page Return URL</u> -----	7
iPayBy Tax Invoice Receipts -----	8
Web Service for Card Capture and Pre Auth -----	9
Web Service for Query -----	11
Web Service for Settlement Advice -----	12
Web Service for Order Capture -----	13
Web Service Examples -----	14
<u>Generic Capture SOAP Call</u> -----	14
<u>Generic Query SOAP Call</u> -----	16
<u>PHP SOAP Call</u> -----	18
<u>.NET (C#) Call</u> -----	20
AJAX enabled XML Payment Servlet -----	21
XML Background Servlet -----	23
Batch and Recurring Payments -----	25
Zen Cart/osCommerce Payment Module Implementation -----	28
AntiFraud Pack -----	29
<u>AntiFraud BRONZE</u> -----	29
<u>AntiFraud GOLD</u> -----	31
Summary Response Codes -----	32
Response Codes -----	33

iPayBy Programmer Manual

If SCNet is hosting your payment page or processing background requests, the following information will show how to program the system to capture card transactions. This manual assumes the user has a medium level of HTML and/or XML experience which is required when customising the parameters for the call to the gateway.

Hosted Payment Page

From your web application, use the following URL to call the payment page. You can use either a GET Method or POST Method. A custom http form using POST Method to call the payment page with the variables as hidden is the preferred method as it is more secure and the parameters cannot be read or altered easily by the user.

The Test URL is:

<https://www.scnet.com.au/ipayby/hostedccbuy>

Please use the Test URL for all testing and then switch to the production URL once going Live.

Note: All parameters names are in lower case

Parameter	Required or Optional	Default Value	Description
gate	Required		Your Merchant ID . This Merchant ID is your tie in to your nominated Merchant Account. For trial purposes use SCNet_Cert6
adminemail	Required		Usually the Merchant's Administration Email account. This email address will receive a copy of all successful receipts.
process	Optional	LATER	Either PROCESS (realtime) or LATER . For realtime credit card processing use "process=process". Later mode is for those sites which do not ship immediately or need to vet the transaction first before shipping. The customer will receive an approval immediately and you can then use the admin screen when satisfied to actually process the transaction.
amount	Required		Amount in dollars.cents or currency.decimal (i.e. 11.45)
returl	Required		This is the URL which returns the customer back to your site. Described below are a number of parameters returned including the response code which you can use in your shopping cart to verify the status of the order. The URL the payment page returns to must be prefixed by HTTP://> - DO NOT put a trailing "/" in the URL
invoice_id	Optional		The Invoice ID comes from YOUR system. It is used as a "tie in" to SCNet if you have a separate shopping cart which generates its own Invoice Ids. Your SCNet iPayBy order number will suffix this number allowing a direct number link from your own system to our automatically generated receipt number.

Parameter	Required or Optional	Default Value	Description
items	Optional		A list of purchased items. This will also be sent to the customer via email if the sendmail option is set.
cust	Optional		Optional. Customer name if different from the "card holder" name.
street	Optional		Street address.
suburb	Optional		Name of the suburb.
city	Optional		Name of City.
country	Optional		Name of Country.
postcode	Optional		Postal code.
phone	Optional		Phone no. at the shipping destination.
mobile	Optional		Mobile phone no.
info	Optional		Additional shipping information.
shop	Optional		Name of the shop to be displayed on the payment page
image	Optional		Thumbprint image to be displayed on the Payment Page. The image can be uploaded using the Administration Panel.
message	Optional		For a customized message on the payment screen. Use this to display any type of message including security messages and so on.
emailcapture	Optional		Use this field if you need to capture the email address of the customer before passing it to the SCNet gateway. All the emails will be sent to this address provided the sendmail option is set.
sendemail	Optional	N	Y or N . This is the master sendmail option if you want to send a receipt from our system. Note that for SPAM Filtering reasons the email will appear from our system – not from yours. The "reply-to" address will be set using the "adminemail" option. If you specify N, you will be responsible for sending out a receipt email to your customer.
currency	Optional	AUD	If not given defaults to AUD. Use this field if you have an account with multiple currency facilities. For example currency=USD
abn	Optional		Your ABN . If using this option, a TAX INVOICE header will appear on the email sent and the GST will be calculated automatically on the email. Use this if you do not want to send a separate invoice to the customer.
allowretry	Optional	Y	If a customer transaction is declined, you may specify whether another retry transaction is allowed. This parameter is to stop repeating retry failures.
allowcomments	Optional	Y	This option controls whether a customer can enter a comment as part of the payment. Set to "N" if you do not want this option enabled.
tablebgcolour	Optional	57577A	Set the table background caption colour. If you want to change the colour set the 6 digit hexadecimal code here <i>without</i> the preceding # ie. 000000
backgroundimage	Optional		If you want your own background water mark gif/jpg you can upload it to our server using our Administration Panel. You can then specify the gif/jpg filename in the call ie. logo.gif

A simple example of a TEST URL for GET Method call is below:

https://www.scnets.com.au/ipayby/hostedccbuy?gate=SCNet_Cert6&amount=11.00&process=process&sendemail=Y&returl=http://www.scnets.com.au&adminemail=you@company.com.au&message=Thankyou%20For%20Your%20Business&shop=SCNet%20eStore%20&items=1x50pk%20Princo%20DVDs

A simple example of a Form for the POST Method call is below:

```
<form action="https://www.scnets.com.au/ipayby/hostedccbuy" method="post">
<div><label class="left" for="Name">Name</label>
<div><input id="Name" maxlength="32" name="cust" size="30" type="text" /></div></div>
<div><label class="left" for="Email">Email address</label>
<div><input id="Email" name="emailcapture" type="text" /></div>
<div><label>Select your donation amount</label>
<select id="Donation" name="amount">
<option value="10.00">$10</option>
<option value="50.00">$50</option></select>
<input name="shop" type="hidden" value="Online Donations" />
<input name="message" type="hidden" value="Thanks again for thinking of us!" />
<input name="image" type="hidden" value="drh-form-header.gif" />
<input name="tablebgcolour" type="hidden" value="839912" />
<input name="info" type="hidden" value="DONATION" />
<input name="requiredfields" type="hidden" value="emailcapture" />
<input name="adminemail" type="hidden" value="enquiries@test.org.au" />
<input name="gate" type="hidden" value="SCNet_Cert6" />
<input name="items" type="hidden" value="Donation - Thank you for your gift. All
donations $2.00 and over are tax deductible. Please retain this receipt." />
<input name="process" type="hidden" value="PROCESS" />
<input name="shop" type="hidden" value="Donation (ABN 123456789)" />
<input name="message" type="hidden" value="Many thanks from Us." /> <input
name="returl" type="hidden" value="http://www.test.org/thanks-for-your-donation/" />
<input name="sendemail" type="hidden" value="Y" />
<div class="formbutt">
<input class="action" name="Proceed" type="submit" value="Proceed to Secure Payment
Site" /></div>
</form>
```

There is a variety of the parameters you can use to customise your site as above.

Header or Background Image Loader

If you are using a hosted payment page, you may want to load your own graphic for top of the payment page, or a background water mark. You can then use the "image=" or "backgroundimage=" option in your call to the payment page. Please note that once you load your image, the name will always be prefixed by the merchant ID to avoid name clashes in our system. Please note that image sizes are restricted to 200K.

For example, if my image is called logo.gif and my merchant id is SCNet_Cert, then my image name when it's loaded will become SCNet_Certlogo.gif. In the call to our page you will then specify "image= SCNet_Certlogo.gif".

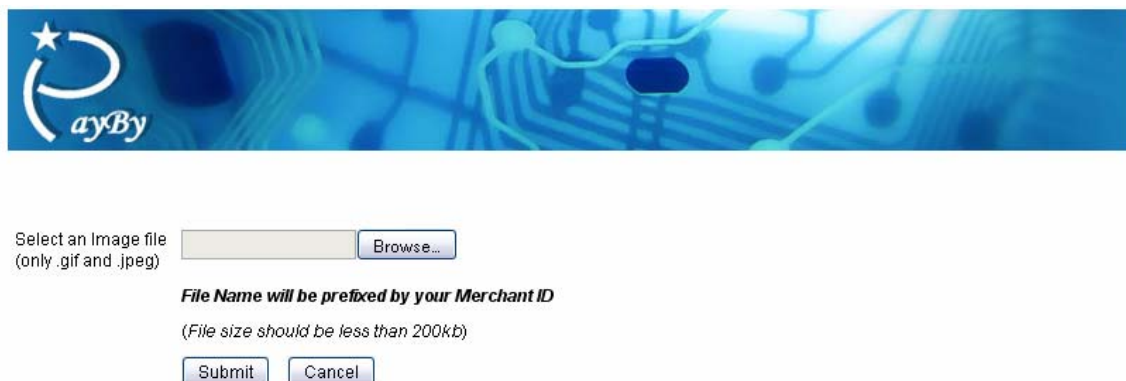
To load the image, go to our Merchant Login page, located at:

<https://www.schnet.com.au/ipayby/index.htm>

Login using your SCNet Gateway ID and password and

Navigate to **Payment Page Graphics Loader**

The loader screen will look like this:



Select an Image file
(only .gif and .jpeg)

File Name will be prefixed by your Merchant ID

(File size should be less than 200kb)

Once loaded a success screen should appear like this:



Image Successfully uploaded.

Please [click here](#) to upload another image.

An error screen will appear if there were any errors encountered.

Hosted Payment Page Return URL

iPayBy will return the following variables via the Return URL (returl) variable specified in the calling page.

Parameter	Description
bank_id	The iPayBy order number
summarycode	The Summary Code as specified in the Code manual.
responsecode	The Response Code as specified in the Code manual.
invoice_id	Your Invoice Id. This is specified when calling the payment page. It is your tie in to the iPayBy transaction. You can use this to interface back into your shopping cart.



iPayBy Tax Invoice Receipts

This feature allows (using the abn=yourABN option) a TAX Invoice receipt to be sent out via the email confirmation procedure. This alleviates the need to send the customer a separate invoice/receipt. The GST is automatically created on the email and this conforms to a legal Tax Invoice.

SCNet Demonstration Payment Page	
TAX INVOICE	
ABN: 123456789	
Date:	13/01/2009 03:02:18 PM
Receipt No:	090113150217251
Email Address:	rmascaro@scnet.com.au
Card No:	4242#####4242
Items:	1x50pk DVDs
GST:	1.00
Amount:	11.00

Please keep of copy of this receipt for your records

SCNet Certification
Logged IP Address: 203.5.34.1

Web Service for Card Capture and Pre Auth

Provided we have pre-approved your system, you can call our Web Service from your own cart or system in the background using a Web Service enabled SOAP call. The main advantage of this type of call is that the user doesn't have to leave your own shopping cart and jump to an external page. You will need to have an SSL certificate installed at your own site to use this option.

The SOAP Parameters with Descriptions are listed below:

Parameter	Required or Optional	Format	Description
arg0	Required	String	Your Merchant ID . This Merchant ID is your tie in to your nominated Merchant Account. For trial purposes use SCNet_Cert6
arg1	Required	String	Your issued Web Services password to the gateway.
arg2	Required	String	Credit Card Number, note that this is not checked and you will need to provide your own "Luhn" check.
arg3	Required	String	Credit Card Expiry date in MMY format.
arg4	Optional	String	Credit Card CVC or CVV- we strongly recommend you provide this.
arg5	Optional	Decimal	Amount in dollars.cents or currency.decimal (i.e. 11.45)
arg6	Optional	String	Credit Card Holder Name.
arg7	Required	String	Card Currency ie. AUD or USD etc.
arg8	Optional	String	Invoice ID. If you have an internal invoice reference, you can specify it here. The reference will be prefixed onto our generated number to give you a tie-in to your internal system. Note the number must not be larger than 16 characters and is alpha numeric.
arg9	Optional	String	Email address for the administrator copy of the receipt email sent. Mandatory if arg10 is set as this sets the "reply-to" address.
arg10	Optional	String	Email address to email the customer a receipt directly from our system.
arg11	Optional	String	Shop Name to be displayed in the email subject and body.
arg12	Optional	String	Items to be displayed on the email.
arg13	Optional	String	Customer Name to be displayed on the email.
arg14	Optional	String	Street to be displayed on the email.
arg15	Optional	String	Suburb to be displayed on the email.
arg16	Optional	String	Postcode to be displayed on the email.
arg17	Optional	String	Phone to be displayed on the email.
arg18	Optional	String	Mobile to be displayed on the email.
arg19	Optional	String	Comments to be displayed on the email.
arg20	Optional	String	Your ABN . If using this option, a TAX INVOICE header will appear on the email sent and the GST will be calculated automatically on the email. Use this if you do not want to send a separate invoice to the customer.
arg21	Optional	String	"Y" or "y" or null, please use if you want to specify your own unique orderNumber/Invoice ID (arg8). Please note that the banking network will reject duplicates.
arg22	Optional	String	City to be displayed in the email
arg23	Optional	String	Country to be displayed in the email.
arg24	Optional	String	3D Secure only. Set to base64 value returned by MPI
arg25	Optional	String	3D Secure only. Set to base64 value returned by MPI
arg26	Optional	String	3D Secure only. Set to appropriate value returned by MPI
arg27	Optional	String	Anti Fraud Forwarded IP address from your customer to enable more accurate Geo IP address detection

The TEST WSDL and URL for the service is listed below:

Capture: <https://www.scnet.com.au/ipayby/ipaybyws?WSDL>
Auth: <https://www.scnet.com.au/ipayby/ipaybyauthws?WSDL>

Web Service for Query

Once you have captured your card details, you may want to use this service to query the status of the transaction, or alternatively provide bespoke reporting. You will need to have an SSL certificate installed at your own site to use this option.

The SOAP Parameters with Descriptions are listed below:

Parameter	Required or Optional	Format	Description
arg0	Required	String	Your Merchant ID . This Merchant ID is your tie in to your nominated Merchant Account. For trial/test purposes use SCNet_Cert6
arg1	Required	String	Your issued Web Services password to the gateway.
arg2	Optional	String	An orderNumber or wildcard (min 4 chars) i.e. 123456-%
arg3	Optional	String	FromDate in the format yyyy-mm-dd i.e. 2009-01-01
arg4	Optional	String	ToDate in the format yyyy-mm-dd i.e. 2009-01-01

NB. The Query design is that once an orderNumber is supplied, the date parameters are ignored. Also if you supply a fromDate you must supply a toDate - no blind queries are allowed.

The TEST WSDL and URL for the service is listed below:

QueryStatus: <https://www.scnet.com.au/ipayby/ipaybyqueryws?WSDL>

Web Service for Settlement Advice

Once you have obtained and Authid from an Auth Capture to transfer the funds you will need to make a Settlement Advice Web Service using the previous Order Number from the Auth.

The SOAP Parameters with Descriptions are listed below:

Parameter	Required or Optional	Format	Description
arg0	Required	String	Your Gateway ID . You will get this one once you have your Merchant ID. This gateway ID is your tie in to your nominated Merchant Account. For trial/test purposes use SCNet_Cert6
arg1	Required	String	Your issued password to the gateway.
arg2	Required	String	Order Number from previous Auth
arg3	Required	String	Authid from previous Auth
arg4	Required	Decimal	Amount in dollars.cents or currency.decimal (i.e. 11.45)

The TEST WSDL and URL for the service is listed below:

Settlement Advice: <https://www.scnet.com.au/ipayby/ipaybysettlementadvicews?WSDL>

Web Service for Order Capture

This web service can only be used if you have a **previous Capture, Auth or Refund** transaction. This transaction is then used to trigger a Capture or Refund on the same card without knowing the credit card details of the original order.

The SOAP Parameters with Descriptions are listed below:

Parameter	Required or Optional	Format	Description
arg0	Required	String	Your Gateway ID . You will get this once you have your Merchant ID. This Gateway ID is your tie in to your nominated Merchant Account. For trial/test purposes use SCNet_Cert6
arg1	Required	String	Your issued password to the gateway.
arg2	Required	String	Transaction Type ie. 'Capture' or 'Refund'
arg3	Required	String	Original Order Number
arg4	Required	Decimal	Amount in dollars.cents or currency.decimal (i.e. 11.45)
arg5	Optional	String	'Y', 'N' or null (see arg6)
arg6	Required/Optional	String	New Order Number for order being generated <ul style="list-style-type: none"> If arg5 = 'Y' the new Order Number will be set to the value in arg6. Note: arg6 is mandatory in this case and <u>must</u> be supplied. If arg5 = 'N' or is not set (null) then the new Order Number will be system generated and then prefixed with the value in arg6

The TEST WSDL and URL for the service is listed below:

Order Capture: <https://www.scnet.com.au/ipayby/ipaybyordercapturews?WSDL>

Web Service Examples

Generic Capture SOAP Call

Use the below example as a guide to formatting a SOAP Envelope should you be coming in from a system not mentioned in this manual. Note that the function action should be "performTransaction".

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://service.ipay.scnet/"
xmlns:scn="https://www.scnet.com.au/ipayby/ipaybyws">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:performTransaction>
      <arg0>SCNet_Cert6</arg0>          <!-- Merchant ID Mandatory-->
      <arg1>testSCNet</arg1>          <!-- Password Mandatory -->
      <arg2>4242424242424242</arg2>    <!-- Card No Mandatory -->
      <arg3>1111</arg3>                <!-- Expiry Date Mandatory-->
      <arg4>666</arg4>                <!-- CVC Mandatory-->
      <arg5>12.00</arg5>              <!-- Amount Mandatory-->
      <arg6>Mr. John Citizen</arg6>    <!-- Card Holder Name Optional-->
      <arg7>AUD</arg7>                <!-- Currency Mandatory-->
      <arg8>123</arg8>                <!-- Invoice ID Optional -->
      <arg9>admin@yourdomain.com</arg9> <!-- Admin email address -->
      <arg10>cust@theirmail.com</arg10> <!-- Customer email address -->
      <arg11>My Online Shop</arg11>    <!-- Shop Name Optional -->
      <arg12>Items Purchased</arg12>   <!-- Items Optional -->
      <arg13>Customer Name</arg13>     <!-- Customer Name Optional -->
      <arg14>1 Any Street</arg14>      <!-- Street Address Optional -->
      <arg15>Any Suburb</arg15>        <!-- Suburb Optional-->
      <arg16>6000</arg16>              <!-- Post Code Optional-->
      <arg17>0891234567</arg17>       <!-- Phone Optional-->
      <arg18>0410000000</arg18>       <!-- Mobile Phone Optional -->
      <arg19>MyComments</arg19>       <!-- Comments Optional -->
      <arg20>123456789</arg20>        <!-- Shop ABN optional -->
      <arg21>N</arg21>                 <!-- Supply own unique invoiceId -->
      <arg22>N</arg22>                 <!-- City -->
      <arg23>N</arg23>                 <!-- Country -->
    </ser:performTransaction>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header/>
  <env:Body>
    <ns2:performTransactionResponse xmlns:ns2="http://service.ipay.scnnet/">
      <return>
        <amount>44.00</amount>
        <authId>9k6K</authId>
        <orderNumber>090727173432082</orderNumber>
        <RRN>k3XZP0ie</RRN>
        <responseCode>00</responseCode>
        <responseText>Approved</responseText>
        <settlementDate>2009-07-27T00:00:00+08:00</settlementDate>
        <summaryResponseCode>Approved</summaryResponseCode>
        <transactionType>Capture</transactionType>
        <XID>6185167</XID>
      </return>
    </ns2:performTransactionResponse>
  </env:Body>
</env:Envelope>
```

Generic Query SOAP Call

Use the below example as a guide to formatting a SOAP Envelope should you be coming in from a system not mentioned in this manual. Note that the function action should be "performQuery".

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://service.ipay.scnet/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:performQuery>
      <arg0>SCNet_Cert6</arg0>           <!-- Merchant ID Mandatory-->
      <arg1>testSCNet</arg1>           <!-- Password Mandatory -->
      <arg2>124-091006073528758</arg2> <!-- OrderNumber Optional -->
      <arg3>2009-01-01</arg3>         <!-- FromDate Optional -->
      <arg4>2009-01-01</arg4>         <!-- ToDate Optional -->
    </ser:performQuery>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header/>
  <env:Body>
    <ns2:performQueryResponse xmlns:ns2="http://service.ipay.scnet/">
      <return>
        <amount>12.00</amount>
        <cardHolder>Mr. John Citizen</cardHolder>
        <cardNumber>4242#####4242</cardNumber>
        <comments>My Comments</comments>
        <currency>AUD</currency>
        <custemail>john@company.com.au</custemail>
        <custname>John Citizen</custname>
        <expiryDate>1112</expiryDate>
        <ipAddress>203.3.222.109</ipAddress>
        <items>Items Purchased</items>
        <orderNumber>124-091006073528758</orderNumber>
        <postCode>2000</postCode>
        <RRN>Rfd4Q0TaxmK</RRN>
        <responseCode>00</responseCode>
        <responseText>Approved</responseText>
        <street>2 Any Street</street>
        <suburb>Sydney</suburb>
        <summaryResponseCode>Approved</summaryResponseCode>
        <transactionType>Capture</transactionType>
      </return>
    </ns2:performQueryResponse>
  </env:Body>
</env:Envelope>
```

Note that actual SummaryResponse Code statuses are returned from the banking network. If for some reason we cannot present your transaction to the banking network i.e. invalid Card Number, you will receive a "Declined" SummaryResponse (which is stored at our server) and of course as the banking network has never received the transaction a response like the one listed below:

```
<summaryResponseCode>Declined</summaryResponseCode>
<responseCode>RE</responseCode>
<responseText>No transaction available for Merchant ID '278' and Order Number '123-
091005172302062' .
</responseText>
```

PHP SOAP Call

Note: The soapClient functionality is implemented only in PHP 5.1 and above and will only work properly if you have enabled the php_soap and php_openssl extensions.

Below is an example of a simple "Helloworld" type of example to make a call to the gateway. Simply save this file into your PHP run area on your web server and execute it.

The Test WSDL and URL (php SOAPCLIENT call) for the service is listed below:

<https://www.scnet.com.au/ipayby/ipaybyws?WSDL>

```
<?php
//A Simple Helloworld Example to perform a
//PHP SOAP enabled call to the Payment Gateway

$params->arg0 = 'SCNet_Cert6'; //Merchant Id
$params->arg1 = 'testSCNet'; //password
$params->arg2 = '4444333322221111'; //Card Number
$params->arg3 = '1111'; //Expiry
$params->arg4 = '111'; //CVC
$params->arg5 = '12.00'; //Amount
$params->arg6 = 'Mr John Citizen'; //Cardholder optional
$params->arg7 = 'AUD'; //Currency
$params->arg8 = '100'; //InvoiceId optional

$client = new
soapClient("https://www.scnet.com.au/ipayby/ipaybyws?WSDL",array('location' =>
'https://www.scnet.com.au/ipayby/ipaybyws'));

$soap_response = $client->performTransaction($params);
print("\n<br><pre>");
    print_r("The Transaction Summary is listed below....");
    print("</pre><br>\n");

echo("\n<br><pre>");
echo("Order Number: " . $soap_response->return->orderNumber);
echo("<br>");
echo("Summary Response: " . $soap_response->return->summaryResponseCode);
echo("<br>");
echo("Response Code: " . $soap_response->return->responseCode);
echo("<br>");
echo("Response Text: " . $soap_response->return->responseText);
echo("<br>");
echo("Amount: " . $soap_response->return->amount);
echo("<br>");
echo("Auth ID: " . $soap_response->return->authId);
echo("<br>");
```

```
echo("RRN: " . $soap_response->return->RRN);  
echo("<br>");  
echo("Settlement Date: " . $soap_response->return->settlementDate);  
echo("<br>");  
echo("Transaction Type: " . $soap_response->return->transactionType);  
echo("</pre><br>\n");  
?>
```

Note: When using this service, it is important to provide a credit card luhn check and to check card expiry dates, amounts and so on to prevent erroneous data being presented to our servers and onto the banking network.

.NET (C#) Call

When defining the Web Services call, the appropriate Web Reference must be setup in your project. Please input the following URLs into your project.

The Test WSDL and URL (Web Reference)) for the service is listed below:

<https://www.scnet.com.au/ipayby/IpayByWS?WSDL>

```
protected void btnSubmit_Click(object sender, EventArgs e)
{
    au.com.scnet.www.IpayByWS ipayByWS = new au.com.scnet.www.IpayByWS();
    au.com.scnet.www.performTransaction performTransactionDetails = new
    au.com.scnet.www.performTransaction();
    performTransactionDetails.arg0 = "SCNet_Cert6";
    performTransactionDetails.arg1 = "testSCNet";
    performTransactionDetails.arg2 = "4444333322221111";
    performTransactionDetails.arg3 = "1111";
    performTransactionDetails.arg4 = "666";
    performTransactionDetails.arg5 = Convert.ToDecimal("12.00");
    performTransactionDetails.arg5Specified = true;
    performTransactionDetails.arg6 = "Mr John Citizen";
    performTransactionDetails.arg7 = "AUD";
    performTransactionDetails.arg8 = "1234567";
    performTransactionDetails.arg9 = "Joe.Bloggs@company.com";
    performTransactionDetails.arg10 = "John.Citizen@personalmail.com";
    performTransactionDetails.arg11 = "Purchase of Sporting Equipment";
    performTransactionDetails.arg12 = "2 x Golf Balls, 3 x tennis racquets";
    performTransactionDetails.arg13 = "John Citizen";
    performTransactionDetails.arg14 = "123 Smith St, Lygon";
    performTransactionDetails.arg15 = "North Shore";
    performTransactionDetails.arg17 = "234234234";
    au.com.scnet.www.performTransactionResponse response =
    ipayByWS.performTransaction(performTransactionDetails);

    if (response.@return.summaryResponseCode.Equals("Approved"))
    {
        message.Text = "Thank you. Your order has been submitted. You will receive a
        confirmation email shortly.";
    }
}
```

Note: When using this service, it is important to provide a credit card luhn check and to check card expiry dates, amounts and so on to prevent erroneous data being presented to our servers and onto the banking network.

AJAX enabled XML Payment Servlet

Provided we have pre-approved your system for Servlet usage, you can call our Payment servlet from your own cart or system in the background using an AJAX call. You receive the following XML document back which you can then process as you wish in your own system. The main advantage of this type of call is that the user doesn't have to leave your own shopping cart and jump to an external page. You will need to have an SSL certificate installed at your own site to use this option.

The supported Servlet parameter calls are listed below.

Note: The call must be made from an SSL enabled server.

Servlet Parameter	Description
gate	Your SCNet Merchant ID
ccnumber	The card number
ccexpiry	The card expiry in MMY
cccvc	The CVC on the card
amount	The amount in dollars.cents ie. 11.00
adminemail	The administration email to receive the receipt.
invoice_id	The invoice number from your system – we prefix this to the receipt number - limit of 9 chars
currency	Currency = "AUD" (for other currencies your bank must support it).
cardholder	Card Holder name.
street	Street Address
suburb	Suburb Address
city	City
country	Country
postcode	Postal Code
phone	Phone no
items	Items purchased
process	Only set to "LATER" (case sensitive) if you do not want the transaction presented, rather deferred and manually processed on our admin screen later.
trxtype	Not needed if you only require a Capture. Only use "AUTH" for Pre Auth or "SETTLEMENTADVICE" for Settlement Advice calls.
xid	3D Secure only. Set to base64 value returned by MPI.
cav	3D Secure only. Set to base64 value returned by MPI.
sli	3D Secure only. Set to appropriate value returned by MPI.

The response back from the Servlet is as follows:

XML Element	Description
RECEIPTNO	Automatically generated banking receipt no for reconciliation.
SUMMARYCODE	The Summary Code as specified in the Code manual.
RESPONSECODE	The Response Code as specified in the Code manual.
RESPONSETEXT	The Response Text as specified in the Code manual.
AMOUNT	The amount of the transaction.
RRN	The Retrieval Reference number as specified by the bank.
SETTLEMENT_DATE	The settlement clearing date.
AUTHID	The Authorisation ID for Auth and Settlement Advices
TYPE	The transaction type ie. Capture or Refund.
XID	The XID is another number issued by the bank for the transaction.

Calling the servlet can be accomplished by the following html code, as an example.
(The responses are in the “xmldoc” elements.)

```
-Cut-----  
<HTML><h1>Example SCNet Payment Servlet AJAX Call</h1>  
<head>  
<META HTTP-EQUIV="Pragma" CONTENT="no-cache">  
<script src="https://www.schnet.com.au/ipayby/psjs.js"></script>  
<script>  
function returnXML(xmlDoc) {  
    alert("ReceiptNo " + xmlDoc.getElementsByTagName('RECEIPTNO')[0].firstChild.data +  
    "\nResponseCode " + xmlDoc.getElementsByTagName('RESPONSECODE')[0].firstChild.data +  
    "\nResponseText " + xmlDoc.getElementsByTagName('RESPONSETEXT')[0].firstChild.data +  
    "\nAmount " + xmlDoc.getElementsByTagName('AMOUNT')[0].firstChild.data + "\nRRN " +  
    xmlDoc.getElementsByTagName('RRN')[0].firstChild.data + "\nSettlement_date " +  
    xmlDoc.getElementsByTagName('SETTLEMENT_DATE')[0].firstChild.data + "\nAuthID " +  
    xmlDoc.getElementsByTagName('AUTHID')[0].firstChild.data + "\nType " +  
    xmlDoc.getElementsByTagName('TYPE')[0].firstChild.data + "\nXID " +  
    xmlDoc.getElementsByTagName('XID')[0].firstChild.data  
    );  
}  
</script>  
</head>  
<body>  
<input type="button" value="Click"  
onClick="doPayment('amount=1.00&ccnumber=4444333322221111&ccexpiry=0807&cccvc=0737&ad  
minemail=you@some.com.au');">  
</body>  
</HTML>  
-Cut-----
```

When testing the above code, the Production URL (highlighted in **bold**) should be replaced with the Test URL below:

<https://www.schnet.com.au/ipayby/psjs.js>

XML Background Servlet

To use the XML servlet you must set up an XML stream to the servlet. This option enables you to seamlessly call our payment engine without leaving your own secure site. You can also imbed AJAX calls with this option. Note only Capture is supported with this option.

The TEST URL is:

<https://www.scnet.com.au/ipayby/PaymentServletXML>

XML Element	Required (Y/N)	Description
TYPE	Y	Must be "Capture"
GATE	Y	Your Merchant ID
CCNUMBER	Y	The card number
CCEXPYRY	Y	The card expiry in MMY
CCCVC	Y	The CVC on the card
INVOICEID	Y	The invoice number from your system – we prefix this to the receipt number - limit of 9 chars
CURRENCY	Y	Currency (note: your bank must be multicurrency enabled)
AMOUNT	Y	The amount in dollars.cents ie. 11.00
ADMINEMAIL	Y	The administration email to receive the receipt.
COMMENTS	N	Any Comments
CARDHOLDER	N	Card Holder Name
ITEMS	N	Items
STREET	N	Street Address
SUBURB	N	Suburb
CITY	N	City
COUNTRY	N	Country
POSTCODE	N	Postcode
PHONE	N	Phone No
MOBILE	N	Mobile Phone No

The XML document should be formatted like the example below.

Ensure that **invoiceid** and **adminemail** are populated.

```
<?xml version="1.0"?>
<PAYMENTREQUESTS>
<REQUEST>
<TYPE>Capture</TYPE>
<GATE>SCNet_Cert6</GATE>
<CCNUMBER>44443332221111</CCNUMBER>
<CCEXPIRY>1111</CCEXPIRY>
<CCCVC>666</CCCVC>
<INVOICEID>266</INVOICEID>
<CURRENCY>AUD</CURRENCY>
<AMOUNT>1.00</AMOUNT>
<ADMINEMAIL>you@company.com.au</ADMINEMAIL>
</REQUEST>
</PAYMENTREQUESTS>
```

Note: Several payments can be performed at once.

The XML response from the servlet will be as such:

```
<PAYMENTRESPONSES>
<PAYMENTRESPONSE>
<RECEIPTNO>266-630086388279572</RECEIPTNO>
<SUMMARYCODE>Approved</SUMMARYCODE>
<RESPONSECODE>00</RESPONSECODE>
<RESPONSETEXT>Approved</RESPONSETEXT>
<AMOUNT>100</AMOUNT>
<RRN>FVrvIouVf3</RRN>
<SETTLEMENT_DATE>Mon Jun 30 00:00:00 WST 2008</SETTLEMENT_DATE>
<AUTHID>Rmd0</AUTHID>
<TYPE>1</TYPE>
<XID>5383640</XID>
</PAYMENTRESPONSE>
</PAYMENTRESPONSES>
```

Response and summary codes meanings are documented later in this manual.

Note: When using this service, it is important to provide a credit card luhn check and to check card expiry dates, amounts and so on to prevent erroneous data being presented to our servers and onto the banking network.

Batch and Recurring Payments

The recurring and batch systems can be used for both one off batch payments and for periodical payments. An XML TAG called <RECURRING_PERIOD> designates the frequency of the payment.

A feature of the system is that a previous payment order number can be used instead of keeping credit card numbers and expiry dates in the file, which could violate your merchant agreement. If a valid order number is given (from a previous payment) the credit card details are retrieved from our database.

When the file is reloaded, any “<order_number>” tag values are automatically matched to existing order numbers that it finds. These values are then updated so if an adjustment needs to be made - a reload of the changed file is all that needs to happen. If an order number is not found, a new record is inserted. If the value "" (blank) is found in the recurring period then the record is completely deleted.

The XML fields to be populated are given below:

XML TAG	Required	Description
<RECURRING_TRXS>	Required	Root Tag and denotes the start of the file and is enclosed at the end of the file.
<RECURRING_TRX>	Required	Denotes the start of the transaction itself and is enclosed by the end of the record (see example).
<MERCHANT_ID>	Required	Your Merchant ID as given to you by SCNet
<ORDER_NUMBER>	Optional	This can be an existing receipt number, thereby the system will use the credit card details in the database. If you want to supply the credit card details then you must supply a unique reference in this field.
<INVOICE_ID>	Optional	The value here will be appended to the receipt number generated by iPayBy. This value can be used as a “tie in” to your own system.
<CC_NUMBER>	Optional	A valid credit card number, can be left blank if a previous order number is used.
<CC_EXPIRY>	Optional	A valid expiry date, can be left blank if a previous order number is used.
<RECURRING_PERIOD>	Required	A recurring period, please see the below table for valid values.
<NEXT_TRX_DATE>	Required	The first date the transaction is to be performed regardless of the recurring period. This value cannot be changed after the upload as it is used by the system onwards. If it needs to be changed, delete the record and reload it. Note that this value will be incremented to the next date according to the recurring period specified earlier. Date Format = 01/01/2000
<FINISH_TRX_DATE>	Required	The finishing date for the periodic transaction, note that no transactions will be performed if this date is earlier than your perceived last payment. Date Format = 01/01/2000
<CUST_EMAIL>	Optional	If you want to send your customer an email, please fill this in with a valid email address.
<ADMIN_EMAIL>	Optional	If the administrator of the payments wants to receive an email please fill this in with a valid email address.
<AMOUNT>	Required	The amount in dollars and cents. Example 11.00
<CURRENCY>	Required	A valid currency. Example AUD
<CARD_HOLDER_NAME>	Optional	The card holder name for your reference.
<ITEMS>	Optional	Optional items for your reference.

Recurring Period values are defined below:

Period Value	Description
B	One off batch payment, will not be performed again.
Y	Annual payments to be performed once per year, on the anniversary.
Q	Quarterly payments to be performed once per quarter.
6M	Payments to be performed every 6 months
2W	Payments to be performed every 14 days or fortnightly
M	Monthly payments to be performed once per month.
	A blank value will delete the payment from the system completely. For example <RECURRING_PERIOD></RECURRING_PERIOD>

An example file when using an existing order number is detailed below:

Note that the generated receipt will be prefixed with 2222-

```
<RECURRING_TRXS>
  <RECURRING_TRX>
    <MERCHANT_ID>SCNet_Cert6</MERCHANT_ID>
    <ORDER_NUMBER>0911111100000</ORDER_NUMBER>
    <INVOICE_ID>2222</INVOICE_ID>
    <CC_NUMBER></CC_NUMBER>
    <CC_EXPIRY></CC_EXPIRY>
    <RECURRING_PERIOD>M</RECURRING_PERIOD>
    <NEXT_TRX_DATE>23/06/2009</NEXT_TRX_DATE>
    <FINISH_TRX_DATE>01/01/2012</FINISH_TRX_DATE>
    <CUST_EMAIL>customeremail@someaddress.com.au</CUST_EMAIL>
    <ADMIN_EMAIL>youradmin@yourdomain.com.au</ADMIN_EMAIL>
    <AMOUNT>1.00</AMOUNT>
    <CURRENCY>AUD</CURRENCY>
    <CARD_HOLDER_NAME>JOHN CITIZEN</CARD_HOLDER_NAME>
    <ITEMS>Annual Membership</ITEMS>
  </RECURRING_TRX>
</RECURRING_TRXS>
```

An example file when using an existing order number is detailed below:

Note that (in this example) the generated receipt will be prefixed with "1111-" as per the <invoice_id> setting. The order number "123456" is your reference when updating or deleting the entry.

```
<RECURRING_TRXS>
  <RECURRING_TRX>
    <MERCHANT_ID>SCNet_Cert6</MERCHANT_ID>
    <ORDER_NUMBER>123456</ORDER_NUMBER>
    <INVOICE_ID>1111</INVOICE_ID>
    <CC_NUMBER></CC_NUMBER>
    <CC_EXPIRY></CC_EXPIRY>
    <RECURRING_PERIOD>M</RECURRING_PERIOD>
    <NEXT_TRX_DATE>23/06/2009</NEXT_TRX_DATE>
    <FINISH_TRX_DATE>01/01/2012</FINISH_TRX_DATE>
    <CUST_EMAIL>customeremail@someaddress.com.au</CUST_EMAIL>
    <ADMIN_EMAIL>youradmin@yourdomain.com.au</ADMIN_EMAIL>
    <AMOUNT>1.00</AMOUNT>
    <CURRENCY>AUD</CURRENCY>
    <CARD_HOLDER_NAME>JOHN CITIZEN</CARD_HOLDER_NAME>
    <ITEMS>Annual Membership</ITEMS>
  </RECURRING_TRX>
</RECURRING_TRXS>
```

Zen Cart/osCommerce Payment Module Implementation

SCNet will issue you will a zipped file containing the module. The module should be unzipped into the Zen Cart or osCommerce home directory (the “includes” directory should be in this directory). Then “install” the SCNet Payment Module in the list.

The following lists the options available:

<p>Enable iPayBy Module Do you want to accept payment via iPayBy?</p> <p><input checked="" type="radio"/> True <input type="radio"/> False</p>	
<p>Payment Zone If a zone is selected, only enable this payment method for that zone.</p> <p>--none--</p>	Use this to distinguish which payment zone is allowed to use this
<p>Sort order of display. Sort order of display. Lowest is displayed first.</p> <p>0</p>	Display order on administration screen
<p>Set Order Status Set the status of orders made with this payment module to this value</p> <p>default</p>	Set the order status on Credit Card payment
<p>iPayBy Gateway ID: Your iPayBy Gateway ID:</p> <p></p>	Your SCNet MERCHANT ID as provided
<p>Production or Testing Mode Testing Mode will truncate Cents Amount</p> <p><input checked="" type="radio"/> True <input type="radio"/> False</p>	Production = True (Live Mode) Production = False (Testing mode)
<p>iPayBy Handler: URL of the iPayBy Handler: (default: http://www.scnet.com.au/zc/ipayby_handler.php http://www.scnet.com.au</p>	The handler can usually be defaulted as we pick this up in the install.
<p>Merchant Notifications Should iPayBy e-mail a receipt to the store owner?</p> <p><input checked="" type="radio"/> True <input type="radio"/> False</p>	Set to “True” if a receipt should be emailed
<p>Merchants Email Address The Merchants email address?</p> <p></p>	Administrator email address where receipt should be sent
<p>Accepted Credit Cards Check which credit cards your store accepts.</p> <p><input checked="" type="checkbox"/> Visa <input checked="" type="checkbox"/> MasterCard <input checked="" type="checkbox"/> American Express <input type="checkbox"/> Diners Club</p>	
<p>Request CVC Number Do you want to ask the customer for the card's CVC number</p> <p><input checked="" type="radio"/> True</p>	CVC is mandatory

AntiFraud Pack

If your account has the **AntiFraud Pack** enabled (Levels **BRONZE** and **GOLD** are available), iPayBy provides you with a number of options to determine the aggressiveness of fraud monitoring.

To view the **AntiFraud Pack** levels and options, go to our Merchant Login page, located at:

<https://www.schnet.com.au/ipayby/index.htm>

Login using your SCNet Gateway ID and password and

Navigate to **AntiFraud Control Panel**

When these options are set every single transaction will be assessed and a risk score attached, based on a sophisticated heuristics and database system including card issuer origin and internet IP address amongst other measures. These scores can be viewed either in the reporting screens and/or displayed in the administrator emails. **The administrator can also decide whether to accept the transaction in the “Administration and Processing Screen” by processing it or rejecting the order entirely.**

AntiFraud BRONZE

iPayBy’s **Anti Fraud BRONZE** option provides basic fraud protection and has the following monitoring levels:

- **Low** Just monitoring information, no action taken
- **Medium** Any Risk Analysis Score above 5 will be rejected
- **High** Any Risk Analysis Score above 3 will be rejected
- **Aggressive** Any Risk Analysis Score above 1 will be rejected
- **None** Deactivate **AntiFraud BRONZE**

Depending on the level selected, if the Risk Analysis Score for the transaction is above the level, then the credit card will be rejected and a message indicating this sent to the administrator via an email (see below). **The administrator can then decide whether to accept the transaction in the “Administration and Processing Screen” by processing it or rejecting the order entirely.** This keeps fraud and your transaction costs to a minimum.

Risk Analysis Scores

The Risk Analysis Scores are detailed below.

- **0 The Risk could not be determined accurately**
 - No action will be taken however the transaction should be scrutinised.
- **1 LOW Risk**
 - The transaction appears to contain an Australian issued card and Australian IP Address.
- **2 MEDIUM Foreign Credit Card**
 - The transaction appears to contain a Foreign issued card, but from an Australian IP Address.
- **3 MEDIUM Foreign IP Address**
 - The transaction appears to contain an Australian issued card and a foreign, but not high risk, country IP Address.
- **4 MEDIUM No Landline Phone**
 - The transaction does not contain a landline phone to verify the authenticity of the customer.
- **5 MEDIUM Foreign Credit Card Foreign IP Address**
 - The transaction appears to contain a Foreign issued card and a foreign, but not high, risk country IP Address. This has been moved to number 7 as very high risk and it not being used.
- **6 HIGH Risk Foreign Credit Card**
 - The transaction appears to contain a Foreign issued card and a foreign, but not high risk, country IP Address.
- **7 HIGH Risk IP Address**
 - The transaction appears to contain an Australian issued card and a foreign AND high risk, country IP Address.
- **8 HIGH Risk IP Address No Landline Phone**
 - The transaction appears to contain an Australian issued card and a foreign AND high risk, country IP Address with no landline to verify customer authenticity.
- **9 HIGH Risk Billing Address to Foreign Credit Card**
 - The transaction appears to contain a Foreign issued card and an Australian issued Credit Card.
- **10 VERY HIGH Risk Foreign Credit Card and Foreign IP Address**
 - The transaction appears to contain a Foreign issued card and a foreign, but not high risk, country IP Address.
- **11 EXTREME Risk Foreign Credit Card and High Risk IP Address**
 - The transaction appears to contain a Foreign issued card and a foreign AND high risk, country IP Address.
- Based on the above scores, the administrator has the option to set the aggressiveness of the system. For instance, if "Medium" is selected, then any score above medium i.e. 4 will be rejected and a message sent in the risk assessment email like this:
The AntiFraud system has rejected the transaction.

AntiFraud GOLD

iPayBy's **AntiFraud GOLD** functionality gives the Merchant complete control over approving suspicious transactions and can be switched on and off by the Merchant.

The advanced Risk Analysis system allows the Merchant to decide in what circumstances transactions are to be rejected based on the following options:

- **Same Card Attempts** Multiple same card number attempts
- **Same IP Attempts** Multiple same IP attempts
- **Risk Score Threshold** Transaction risk score breaches set Risk Score threshold
- **Card Matches Shipping Country** Card Issuer country does not match the country to where the goods will be shipped
- **Reject High Risk Countries** Transaction is from a high risk country
- **Reject Anonymous Proxies** Transaction is from an open proxy

The Merchant controls which of the above Risk Analysis checks are enabled by setting the various **AntiFraud GOLD** options available in iPayBy's "**AntiFraud Control Panel**".

Note 1: When AntiFraud GOLD functionality is activated, AntiFraud BRONZE is deactivated automatically.

The following Warning message is displayed and the AntiFraud BRONZE option is set to "None" ...



Note 2: A charge per transaction applies when switching on AntiFraud GOLD.

Full risk analysis information of each transaction is available as a popup from iPayBy's reporting screens. You can also sort based on the risk score so can quickly scan the more dubious transactions. **This allows the administrator to scrutinize the transaction fully before deciding whether to accept or reject the order. (This functionality is available in the "Administration and Processing Screen")**

Summary Response Codes

Summary Response Codes are provided by the financial institution network and the payment system to provide the result of the requested transaction based on the response code. A listing of the summary codes can be found on Table 1.

The summary response code should be the main mechanism used by a Merchant to determine the outcome of a transaction.

A summary code 2 (see E* response codes in Table 2) indicates an intermittent problem with processing the transaction.

Table 1: Summary Response Codes

Result Summary	Code	Description
APPROVED	0	The transaction has been approved by the acquiring network.
DECLINED	1	The transaction has been declined by the acquiring network.
ERROR	2	The transaction request has encountered an unrecoverable and unexpected error in its processing.

Summary: The only response the Merchant wants to see is 0 (approved). Any other response means failure.

Response Codes

Response Codes are provided by the financial institution network and the payment system and provide detail of the result of the requested transaction.

The Response Code field is a variable length alphanumeric field with a maximum length of three. Response codes returned by the financial institution are in the range '00' to '99'.

Note: the Summary Code should be used to determine the result of the transaction as different financial institutions may return the same Response Code with different results.

Table 2: Response Codes

Response Code	Description	Summary Response
00	Approved or completed successfully	0
01	Refer to card issuer	1
02	Refer to card issuers special conditions	1
03	Invalid merchant	1
04	Pick-up card	1
05	Do not honour	1
06	Error	1
07	Pick-up card, special condition	1
08	Honour with identification	0
09	Request in progress	1
10	Approved for partial amount	0
11	Approved VIP	0
12	Invalid transaction	1
13	Invalid amount	1
14	Invalid card number (no such number)	1
15	No such issuer	1
16	Approved, update Track 3	0
17	Customer cancellation	1
18	Customer dispute	1
19	Re-enter transaction	1
20	Invalid response	1
21	No action taken	1
22	Suspected malfunction	1
23	Unacceptable transaction fee	1
24	File update not supported by receiver	1
25	Unable to locate record on file	1
26	Duplicate file update record, old record replaced	1
27	File update field edit error	1
28	File update file locked out	1
29	File update not successful, contact acquirer	1
30	Format error	1
31	Bank not supported by switch	1
32	Completed partially	1
33	Expired card	1
34	Suspected fraud	1
35	Card acceptor contact acquirer	1
36	Restricted card	1

Response Code	Description	Summary Response
37	Card acceptor call acquirer security	1
38	Allowable PIN tries exceeded	1
39	No credit account	1
40	Request function not supported	1
41	Lost card	1
42	No universal account	1
43	Stolen card, pick up	1
44	No investment account	1
45-50	Reserved for ISO use	1
51	Not sufficient funds	1
52	No cheque account	1
53	No savings account	1
54	Expired card	1
55	Incorrect PIN	1
56	No card record	1
57	Transaction not permitted to cardholder	1
58	Transaction not permitted to terminal	1
59	Suspected fraud	1
60	Card acceptor contact acquirer	1
61	Exceeds withdrawal amount limits	1
62	Restricted card	1
63	Security violation	1
64	Original amount incorrect	1
65	Exceeds withdrawal frequency limit	1
66	Card acceptor call acquirers security department	1
67	Hard capture (requires that card be picked up at ATM)	1
68	Response received too late	1
69-74	Reserved for ISO use	1
75	Allowable number of PIN tries exceeded	1
76-89	Reserved for private use	1
90	Cutoff is in process (Switch ending a days business and starting the next. The transaction can be sent again in a few minutes).	1
91	Issuer or switch is inoperative	1
92	Financial institution or intermediate network facility cannot be found for routing	1
93	Transaction cannot be completed. Violation of law	1
94	Duplicate transmission	1
95	Reconcile error	1
96	System malfunction	1
97	Advises that reconciliation totals have been reset	1
98	MAC error	1
99	Reserved for national use	1
EM	Error at the Merchant Server level	2
RM	Rejected at the Merchant Server level	2
EG	<i>Errored at Gateway - response text varies depending on reason</i>	2
RG	<i>Rejected at Gateway - response text varies depending on reason</i>	2
EA	<i>Errored at Acquirer - response text varies depending on reason</i>	2
RA	<i>Rejected at Acquirer - response text varies depending on reason</i>	2
RS	<i>Rejected at Gateway - Usually a card type not configured for the Merchant</i>	2