



# **Merchant Card Payment Engine**

## **BANK ENTERPRISE INTEGRATION GUIDE**

Copyright © PayPoint.net 2008

This document contains the proprietary information of PayPoint.net and may not be reproduced in any form or disclosed to any third party without the expressed written permission of a duly authorised representative of PayPoint.net Limited.

Registered in England No: 3539217. VAT Reg. No: 680 1343 55

## PayPoint.net Bank Enterprise v 2.0

1<sup>st</sup> July 2008

# Table of Contents

<b>1</b>	<b>Getting Started</b>	<b>4</b>
<b>2</b>	<b>Payment Process Overview</b>	<b>4</b>
<b>3</b>	<b>Payment Request Initiation</b>	<b>5</b>
3.1	MCPE Payment Request Parameters	6
3.2	Deferred Payments	7
3.3	Limits Per Transaction	7
3.4	Subscriptions (Scheduled Payments)	7
3.5	Additional Parameters (Pass-Thru Data)	7
<b>4</b>	<b>Payment Request Response</b>	<b>8</b>
<b>5</b>	<b>Refund Request Initiation</b>	<b>9</b>
<b>6</b>	<b>Refund Request Response</b>	<b>9</b>
<b>7</b>	<b>Repeat Payment Request Initiation</b>	<b>10</b>
<b>8</b>	<b>Repeat Payment Request Response</b>	<b>10</b>
<b>9</b>	<b>PreAuth Capture Request Initiation</b>	<b>11</b>
<b>10</b>	<b>PreAuth Capture Request Response</b>	<b>11</b>
<b>11</b>	<b>PreAuth Void request Initiation</b>	<b>12</b>
<b>12</b>	<b>PreAuth Void Request Response</b>	<b>12</b>
<b>13</b>	<b>Subscription Cancellation Request Parameters</b>	<b>13</b>
<b>14</b>	<b>Subscription Cancellation Request Response</b>	<b>13</b>
<b>15</b>	<b>Preventing Online Credit Card Fraud</b>	<b>14</b>
15.1	Manual Checks	14
15.2	Automated Processes	14
<b>16</b>	<b>Enabling 3D Secure</b>	<b>14</b>
<b>Appendix A: Constructing HTTP requests over SSL</b>		<b>15</b>

# Table of Figures

<b>Figure 1: MCPE Payment Request Parameters</b>	<b>6</b>
<b>Figure 2: MCPE Payment Response Parameters</b>	<b>8</b>
<b>Figure 3: Refund Request Parameters</b>	<b>9</b>
<b>Figure 4: Refund Request Response Parameters</b>	<b>9</b>
<b>Figure 5: Repeat Payment Request Parameters</b>	<b>10</b>
<b>Figure 6: Repeat Payment Request Response Parameters</b>	<b>10</b>
<b>Figure 7: PreAuth Capture Request Parameters</b>	<b>11</b>
<b>Figure 8: PreAuth Capture Request Response Parameters</b>	<b>11</b>

<b>Figure 9: PreAuth Void Request Parameters .....</b>	<b>12</b>
<b>Figure 10: PreAuth Void Request Response Parameters .....</b>	<b>12</b>
<b>Figure 11: Subscription Cancellation Request Parameters .....</b>	<b>13</b>
<b>Figure 12: Subscription Cancellation Request Response Parameters .....</b>	<b>13</b>
<b>Figure 13: Example HTTP Request.....</b>	<b>15</b>
<b>Figure 14: Example in PHP .....</b>	<b>15</b>

# 1 Getting Started

You will receive a welcome pack via email confirming your application has been accepted and that your account is setup and ready for you to begin integration.

The PayPoint.net **Merchant Extranet** is a web based back office system that provides detailed information and powerful tools to assist you in managing your PayPoint.net account and can be accessed at the following URL:  
<https://secure.metacharge.com/extranet/>

Your welcome pack will contain primary login details for your account. Once logged in you can change passwords and add new users. Please refer to the Merchant Extranet User Guide for more information.

If you have any questions or require further information, please contact our dedicated **Merchant Helpdesk**. All contact details and contact options are available via the Support tab of the Merchant Extranet. Alternatively please call +44 (0)8701 904 126

# 2 Payment Process Overview

**Bank Enterprise MCPE** (Merchant Card Payment Engine) is designed for large merchants employing experienced developers. To perform integration you must be familiar with server side scripting and operate a secure domain on your web server.

You must also observe card association guidelines on the safe handling of sensitive data. For further information please refer to <https://www.pcisecuritystandards.org/>. You may also download our Guide to Card Types and Card Handling, available from the *Resources* section of the Merchant Extranet.

The payment process starts with a secure form on your web site. The consumer enters their card details and any other required information and submits the transaction. Your server receives this POST data directly, can perform any other arbitrary processes (store to database, change order status to processing) and then initiates its own POST to our payment gateway.

This POST, made as part of your server-side processing, is known as a **Payment Request**; although as this document shows, we accept many types of request including refund and repeat billing instructions. MCPE responds immediately by returning a copy of the request back to you and authorising or declining the payment, providing a status and transaction ID, where applicable.

If the payment is authorised, a transaction receipt is sent to the consumer. This may be disabled upon request to our Merchant Support team, provided that you send your own transaction receipt email to aid customer recognition.

From a technical perspective, the handler script on your server (the target of the payment page form POST) collects data from your secure form and relays it to our payment gateway via a background HTTPS POST. This POST is entirely independent from the cardholder experience in their web browser. The cardholder is a client of your script. Your script is a client of MCPE.

Your handler must then process our instant HTTPS response and, as well as performing any further arbitrary processes (update database, send fulfilment instructions) it completes the integration process by generating the HTML response to the cardholder in their web browser, confirming the transaction outcome and offering any associated instructions.

Please refer to Appendix A for examples showing how to construct HTTP requests over SSL. You can download sample handler code (in PHP, Perl or ASP format) from the Merchant Extranet under the *Resources* tab.

Please notify us of the IP address (or block of IP addresses) from which we can expect to receive your POST's. You can enter this data via the Merchant Extranet. Select *Account Management* then click the *MCPE Firewall* sub tab.

### 3 Payment Request Initiation

You must submit the following 14 fields to initiate a **Payment Request**. Your request must be submitted via HTTPS POST to <https://secure.metacharge.com/mcpe/corporate>

- **intInstID** (your installation number, refer to Merchant Extranet: *Account Management > Installations*)
- **strCartID** (order number or session by which you can identify the user/transaction)
- **strDesc** (description of the goods or service associated with the payment)
- **fltAmount** (transaction amount)
- **strCurrency** (3-character ISO code for the currency you wish to transact in)
- **strCardHolder** (card holder's name as it appears on their card)
- **strPostcode** (the postal code associated with the card's billing address)
- **strEmail** (an e-mail address for the card holder)
- **strCardNumber** (the card number)
- **strExpiryDate** (the expiry date that appears on the card, formatted as MMY)
- **intCV2** (the security code that appears on the signature strip of the card)
- **strCardType** (the type of card either VISA, DELTA - for VISA DELTA, SOLO, SWITCH, MC - for MASTERCARD or UKE - for VISA ELECTRON)
- **fltAPIVersion** (the version of Bank Enterprise MCPE you are using)
- **strTransType** (the type of transaction you wish to perform)

Depending upon the card type, it may be necessary to supply additional information in order for a transaction to be successfully authorised.

To avoid duplicate transactions, please ensure that your secure form cannot be resubmitted by the consumer.

You can submit test transactions (to validate the behaviour and response of your handler) using our dummy VISA card number: 1234123412341234. This will automatically be authorised provided you include the field **intTestMode** set to 1.

Here is an example POST:

```
<form action="https://secure.metacharge.com/mcpe/corporate" method="POST">
<input type="hidden" name="intTestMode" value="1">
<input type="hidden" name="intInstID" value="123456">
<input type="hidden" name="strCartID" value="654321">
<input type="hidden" name="strDesc" value="description of goods">
<input type="hidden" name="fltAmount" value="10.00">
<input type="hidden" name="strCurrency" value="GBP">
<input type="hidden" name="strCardHolder" value="Joe Bloggs">
<input type="hidden" name="strPostcode" value="BA12BU">
<input type="hidden" name="strEmail" value="test@paypoint.net">
<input type="hidden" name="strCardNumber" value="1234123412341234">
<input type="hidden" name="intCV2" value="707">
<input type="hidden" name="strExpiryDate" value="0609">
<input type="hidden" name="strCardType" value="VISA">
<input type="hidden" name="strCountry" value="GB">
<input type="hidden" name="fltAPIVersion" value="1.3">
<input type="hidden" name="strTransType" value="PAYMENT">
</form>
```

For the example above, the response would be:

```
intTestMode=1&intInstID=123456&strCartID=654321&strDesc=description+of+goods&fltAmount=10.00&strCurrency=GBP&strCardHolder=Joe+Bloggs&strPostcode=BA12BU&strEmail=test@paypoint.net&strCardType=VISA&strCountry=GB&intTransID=12345678&intAccountID=123456&intStatus=1&intTime=1070332412
```

Simply remove **intTestMode** from your POST when you are ready to proceed with live transactions. You will only be able to accept real transactions once your Merchant Account is enabled. You will be notified by email when this is available.

There are several optional fields which can be submitted to MCPE. Many of these will enable PayPoint.net Fraud Solutions (see section 13) to perform additional look-ups and provide more accurate results. These fields are indicated in Figure 1.

### 3.1 MCPE Payment Request Parameters

The table below details the fields that can be submitted in a Payment Request to MCPE. Fields that are not marked as required contain information that can be held in the MCPE system but is not necessary for authorisation, or that enables additional security checks prior to authorisation.

**Figure 1: MCPE Payment Request Parameters**

Field	Type(Size)	Required?	Enhances FraudGuard?	Notes
intInstID	int(6)	Yes		The unique identifier for the MCPE installation that will process this payment.
strCartID	char(192)	Yes		Your unique identifier for this purchase, for your reconciliation.
intAccountID	int(6)			The MCPE unique identifier for the account to receive funds for this purchase. If this field is omitted or invalid, a decision is made based on the currency specified in strCurrency field and the value of the intTestMode field (if included).
strDesc	char(192)	Yes		Descriptive text for this purchase.
fltAmount	float(8,3)	Yes		A decimal value representing the transaction amount in the currency specified in the strCurrency field, using a point (.) as the separator. Include no other separators, or non-numeric characters.
strCurrency	char(3)	Yes		The 3-letter ISO code for the currency of payment fltAmount
intAuthMode	int(1)			A value to indicate the type of authorisation to use. If this field is omitted, full authorisation with capture is assumed. Values: 0=equivalent to field omitted, 1=authorisation with capture, 2=pre-authorisation only (see section 3.2)
intTestMode	int(1)			If included, indicates a test purchase. A VISA card with card number 1234123412341234 should be used. Values: 0=equivalent to field omitted (payment is live), 1=all payments are successful, 2=all payments fail. Banks are not involved in test payments.
strCardHolder	char(20)	Yes	Yes	The name of the card holder, as it appears on the card.
strAddress	char(255)		Yes	The purchaser's postal billing address.
strCity	char(40)		Yes	The purchaser's city.
strState	char(40)		Yes	The purchaser's state, province or county.
strPostcode	char(15)	Yes	Yes	The postal code associated with the address in strAddress.
strCountry	char(2)		Yes	The 2-letter ISO code for the purchaser's country.
strTel	char(50)			The purchaser's telephone number.
strFax	char(50)			The purchaser's fax number.
strEmail	char(100)	Yes	Yes	The purchaser's e-mail address.
strCardNumber	char(20)	Yes	Yes	The card number.
strStartDate	char(4)			The card valid from date if available, formatted as MMYY.
strExpiryDate	char(4)	Yes		The expiry date that appears on the card, formatted as MMYY.
intCV2	char(4)	Yes	Yes	The security code that appears on the card signature strip.
strIssueNo	char(2)			The issue number of the card, if it has one, including a leading zero if one appears on the card.
strCardType	char(8)	Yes		The type of card - either VISA, DELTA (for VISA DELTA), SOLO, SWITCH, MC (for MASTERCARD) or UKE (for VISA ELECTRON).
strUserIP	char(15)		Yes	The IP address of the purchaser. This is used to perform additional security checking by establishing the country from which a payment is being made.
strTransType	char(30)	Yes		For this transaction type, the value of the field is <i>PAYMENT</i> .
fltAPIVersion	float(2,1)	Yes		The version of Bank Enterprise MCPE you are using. Currently, this is 1.3
intReference	int(11)			A numerical reference for this transaction, which must be unique. Can be used to alert us to duplicate requests.
datFulfillment	char(10)	As Advised		A date the customer order will be fulfilled, in the format DD/MM/YYYY.
fltSchAmount	float(8,3)			For scheduled payments based upon this transaction, the amount associated with each scheduled payment, in the currency specified in the strCurrency field, formatted as for the fltAmount field.
strSchPeriod	char(4)			For scheduled payments based upon this transaction, the interval between payments, given as XY where X is a number (1-999) and Y is "D" for days, "W" for weeks or "M" for months.
intRecur	int(1)			For scheduled payments, indicates if scheduled payments should recur. Values: 0=no, 1=yes.
intCancelAfter	int(1)			Cancel a subscription after this many successful payments.

## 3.2 Deferred Payments

Your account has full support as standard for Deferred Payments. This enables you to authorise a charge from a card without capturing funds, ideal for checking the validity of the card, allowing time to review orders after authorisation, or for preparing orders before committing charges to the customer card.

Deferred Payments can be performed by supplying *intAuthMode* set to 2. You have up to 7 days to optionally capture funds which you have authorised. This can be done automatically via a further capture request, or via the Merchant Extranet. Auth capture and void requests are detailed in sections 8 and 10. You can also configure automatic capture or void on the Merchant Extranet via *Account Management > Installations*, or capture these manually via *Sales > Pre Auths*.

## 3.3 Limits Per Transaction

By default, the minimum and maximum values of individual transactions on your account are as shown below:

Limits	GBP	USD	EUR
Minimum	£1	\$1	€1
Maximum	£1,000	\$1,500	€1,500

Please contact your account manager to request higher limits on your account, subject to approval by our risk team.

## 3.4 Subscriptions (Scheduled Payments)

The **Merchant Card Payment Engine** supports advanced subscription management. For example, you may set up a subscription offering your customers a trial period with a special introductory rate followed by a regular payment each month. The engine manages scheduling and bills customers automatically. It supports up to 3 levels for each subscription. Each level has an associated amount and period.

This functionality is not enabled by default – please contact your account manager if you would like this feature enabled on your account. To create subscriptions include *fltSchAmount<sub>n</sub>*, *strSchPeriod<sub>n</sub>* (where *n* is 1, 2 or 3) and *intRekurs* in a **Payment Request**. *intRekurs* specifies whether the subscription should continue indefinitely and always applies to the last level specified.

If you would like a subscription to automatically cancel after '*n*' payments, *intCancelAfter* determines after how many payments the schedule should be cancelled by the engine.

Here are some examples:

Consumer Proposition	POST to MCPE
£1.00 for the first 7 days, £5.00 per month thereafter	<i>fltSchAmount1</i> = 1.00, <i>strSchPeriod1</i> =7D <i>fltSchAmount2</i> = 5.00, <i>strSchPeriod2</i> =1M <i>intRekurs</i> =1
£20.00 per week	<i>fltSchAmount1</i> =20.00, <i>strSchPeriod1</i> =1W <i>intRekurs</i> =1
£2.00 for the first 7 days £5.00 for the next 3 weeks £8.00 per month thereafter	<i>fltSchAmount1</i> =2.00, <i>strSchPeriod1</i> =7D <i>fltSchAmount2</i> =5.00, <i>strSchPeriod2</i> =3W <i>fltSchAmount3</i> =8.00, <i>strSchPeriod3</i> =1M <i>intRekurs</i> =1
£3.00 for the first 7 days £5.00 per month thereafter, automatic cancellation after 6 successful payments	<i>fltSchAmount1</i> = 3.00, <i>strSchPeriod1</i> =7D <i>intRekurs</i> =1, <i>intCancelAfter</i> =6
Free for a week £10.00 per month thereafter	<i>fltSchAmount1</i> =0, <i>strSchPeriod1</i> =7D <i>fltSchAmount2</i> =10.00, <i>strSchPeriod2</i> =1M <i>intRekurs</i> =1

Please note that these fields are in addition to the standard (mandatory) fields sent to MCPE, however *fltSchAmount1* replaces *fltAmount*. *fltSchAmount1* is the initial payment (first level) of the subscription.

## 3.5 Additional Parameters (Pass-Thru Data)

You have the option of sending additional parameters in your Payment Request POST that you wish MCPE to return back to you. This is called **Pass-Thru Data**. Any field which starts with the characters "PT\_" will be returned to you in the contents of the **Payment Request Response**.

## 4 Payment Request Response

You will be notified of the outcome of a transaction in the same session as your Payment Request. The response fields will be sent as a URL-encoded query string and will consist of the original **Payment Request** that you submitted to MCPE, as well as our additional **Payment Response Parameters** shown in Figure 2.

**Figure 2: MCPE Payment Response Parameters**

Field	Type(Size)	Notes
intTransID	int(11)	The MCPE unique identifier for this transaction.
intAccountID	int(11)	The account used for the transaction.
intStatus	int(1)	The status of this transaction. Values: 0=failed, 1=successful.
intTime	int(11)	The time at which this transaction was authorised, given as the number of seconds since the start of 1970 GMT. This is omitted in the event of a cancelled transaction.
fltAmount	float(8,3)	The amount associated with this transaction, in the currency specified in the <i>strCurrency</i> field.
strCurrency	char(3)	The 3-letter ISO code for the currency associated with this transaction (uppercase).
strMessage	char(255)	Any message returned by the bank when this transaction was processed.
strPaymentType	char(6)	The purchaser's card type. Values: VISA, MC, DELTA, SOLO, SWITCH, UKE.
intAVS	int(1)	The result of the AVS check performed for this transaction. <b>This field will be omitted if the check was not performed.</b> Values: 0=AVS check failed, 1=AVS check passed.
intCV2	int(1)	The result of the CV2 check performed for this transaction. <b>This field will be omitted if the check was not performed.</b> Values: 0=CV2 check failed, 1=CV2 check passed.
intCountryIP	int(1)	The result of checking the purchaser's country as determined from their IP address against the country supplied as part of the billing address. <b>This field will be omitted if the check was not performed.</b> Values: 0=check failed, 1=check passed.
intTestMode	int(1)	Indicates whether this was a test transaction. Values: 0 or not present=live transaction, 1=test transaction.
strCardHolder	char(255)	The card holder's name.
strAddress	char(255)	The card holder's street address.
strCity	char(255)	The card holder's city
strState	char(255)	The card holder's state/county
strPostcode	char(255)	The card holder's postcode.
strCountry	char(255)	The card holder's country.
strTel	char(255)	The card holder's telephone number.
strFax	char(255)	The card holder's fax number.
strEmail	char(100)	The card holder's email address.
strSecurityToken	char(255)	A value that must be stored alongside each transaction in your system. This value will be required when performing an operation that references a previous transaction – a refund, for example.
strDesc	char(192)	The description of the transaction
strCartID	char(255)	The cartID of the transaction
fltFraudScore	float(2,3)	Likelihood of the transaction being fraudulent. A value between 0.000 and 10.000, 0.000 being the most unlikely and 10.000 being the most likely.
intReference	int(11)	The transaction reference you supplied in your Payment Request, if you supplied one.
fltAPIVersion	float(2,1)	The version of Bank Enterprise MCPE you are using. Currently, this is 1.3
strTransType	char(30)	For this transaction type, the value of this field will be <i>PAYMENT</i> .
fltOriginalAmount	float(8,3)	Included to reflect the original amount of the Payment Request, in case currency conversion was performed during authorisation.
strOriginalCurrency	char(3)	Included to reflect the original currency of the Payment Request, in case currency conversion was performed during authorisation.
fltSchAmount	float(8,3)	For scheduled payments based upon this transaction, the amount associated with each scheduled payment, in the currency specified in the <i>strCurrency</i> field, formatted as for the <i>fltAmount</i> field.
strSchPeriod	char(4)	For scheduled payments based upon this transaction, the interval between payments, given as XY where X is a number (1-999) and Y is "D" for days, "W" for weeks or "M" for months.
intScheduleID	int(11)	The MCPE unique identifier for any payment schedule associated with this transaction (if applicable).



## 5 Refund Request Initiation

You must submit the following six fields to initiate a Refund Request. Your request must be submitted via HTTPS POST to <https://secure.metacharge.com/mcpe/corporate>

- **intInstID** (your installation number)
- **intTransID** (the ID of the transaction that is to be refunded)
- **fltAmount** (the amount to be refunded, in the currency of the original transaction)
- **strSecurityToken** (the value that was returned in the strSecurityToken field of the Payment Request Response for the transaction that is to be refunded)
- **fltAPIVersion** (the version of Bank Enterprise MCPE you are using)
- **strTransType** (the type of transaction to be performed)

Partial refunds are allowed. MCPE will decline any refund that would cause the sum of all refunds performed against a particular transaction to exceed the value of that transaction.

See Figure 3 for a complete set of fields that may be submitted in a Refund Request POST.

**Figure 3: Refund Request Parameters**

Field	Type(Size)	Required?	Notes
intInstID	int(6)	Yes	The MCPE unique identifier for the installation performing this refund.
intTransID	int(11)	Yes	The MCPE unique identifier for the transaction that is to be refunded.
strSecurityToken	char(32)	Yes	The value that was sent in the strSecurityToken field of the Payment Request Response for the transaction that is to be refunded.
fltAmount	float(8,3)	Yes	The amount to be refunded, in the currency of the original transaction.
strDesc	char(192)		A description of the refund transaction.
fltAPIVersion	float(2,1)	Yes	The version of MCPE Bank Enterprise you are using. Currently, this is 1.3
strTransType	char(30)	Yes	For this transaction type, the value of this field must be <i>REFUND</i> .
intReference	int(11)		A numerical reference for this transaction. This value must be unique.
intTestMode	int(1)		Set to 1 if you wish to refund a transaction performed in test mode.

## 6 Refund Request Response

You will be notified of the outcome of a refund in the same session as the Refund Request was made. The response fields will be sent as a URL-encoded query string. See Figure 4 below for a list of the fields returned.

**Figure 4: Refund Request Response Parameters**

Field	Type(Size)	Notes
intTransID	int(11)	The MCPE unique identifier for this refund transaction.
intStatus	int(1)	1 for a successful refund, or 0 for failure.
strMessage	char(255)	Any message returned by the bank performing this refund.
intTime	int(11)	The time at which this refund transaction was authorised, given as the number of seconds since the start of 1970 GMT.
strTransType	char(30)	For this transaction type, the value of this field will be <i>REFUND</i> .
fltAPIVersion	float(2,1)	The version of MCPE Bank Enterprise you are using. Currently, this is 1.3.
intReference	int(11)	Your numerical reference for this transaction, if you supplied one.

## 7 Repeat Payment Request Initiation

You must submit the following six fields to initiate a Repeat Payment Request. Your request must be submitted via HTTPS POST to <https://secure.metacharge.com/mcpe/corporate>

- **intInstID** (your installation number)
- **intTransID** (the ID of the transaction that is to be repeated)
- **fltAmount** (the repeat payment amount, in the currency of the original transaction)
- **strSecurityToken** (the value that was returned in the strSecurityToken field of the Payment Request Response for the transaction that is to be repeated)
- **fltAPIVersion** (the version of Bank Enterprise MCPE you are using)
- **strTransType** (the type of transaction to be performed)

See Figure 5 for a complete set of fields that may be submitted in a Repeat Payment Request POST.

**Figure 5: Repeat Payment Request Parameters**

Field	Type(Size)	Required?	Notes
intInstID	int(6)	Yes	The MCPE unique identifier for the installation performing this repeat payment.
intTransID	int(11)	Yes	The MCPE unique identifier for the transaction that is to be repeated.
strSecurityToken	char(32)	Yes	The value that was sent in the strSecurityToken field of the Payment Request Response for the transaction that is to be repeated.
fltAmount	float(8,3)	Yes	The amount of this payment, in the currency of the original transaction.
strDesc	char(192)		A description of the repeat payment transaction.
fltAPIVersion	float(2,1)	Yes	The version of MCPE Bank Enterprise you are using. Currently, this is 1.3
strTransType	char(30)	Yes	For this transaction type, the value of this field must be <i>REPEAT</i> .
intReference	int(11)		A numerical reference for this transaction. This value must be unique.
intTestMode	int(1)		Set to 1 if you wish to take an additional payment against a transaction performed in test mode.
intCV2	int(4)		Used to initiate a repeat request if your risk agreements mandate that security code must be sent. Must be collected from the customer for each repeat payment request, and never stored on your systems.

## 8 Repeat Payment Request Response

You will be notified of the outcome of a repeat payment in the same session as the Repeat Payment Request was made. The response fields will be sent as a URL-encoded query string. See Figure 6 below for a list of the fields returned.

**Figure 6: Repeat Payment Request Response Parameters**

Field	Type(Size)	Notes
intTransID	int(11)	The MCPE unique identifier for this repeat payment transaction.
intStatus	int(1)	1 for a successful repeat payment, or 0 for failure.
strMessage	char(255)	Any message returned by the bank performing this repeat payment.
intTime	int(11)	The time at which this repeat payment transaction was authorised, given as the number of seconds since the start of 1970 GMT.
strTransType	char(30)	For this transaction type, the value of this field will be <i>REPEAT</i> .
fltAPIVersion	float(2,1)	The version of MCPE Bank Enterprise you are using. Currently, this is 1.3.
intReference	int(11)	Your numerical reference for this transaction, if you supplied one.

## 9 PreAuth Capture Request Initiation

You must submit the following five fields to initiate a PreAuth (Deferred Payment) Capture Request. Your request must be submitted via HTTPS POST to <https://secure.metacharge.com/mcpe/corporate>

- **intInstID** (your installation number)
- **intTransID** (the ID of the pre-auth transaction that is to be captured)
- **strSecurityToken** (the value that was returned in the strSecurityToken field of the Payment Request Response for the pre-auth transaction that is to be captured)
- **fltAPIVersion** (the version of Bank Enterprise MCPE you are using)
- **strTransType** (the type of transaction to be performed)

See Figure 7 for a complete set of fields that may be submitted in a PreAuth Capture Request POST.

**Figure 7: PreAuth Capture Request Parameters**

Field	Type(Size)	Required?	Notes
intInstID	int(6)	Yes	The MCPE unique identifier for the installation performing this pre-auth capture.
intTransID	int(11)	Yes	The MCPE unique identifier for the pre-auth transaction that is to be captured.
strSecurityToken	char(32)	Yes	The value that was sent in the strSecurityToken field of the Payment Request Response for the pre-auth transaction that is to be captured.
strDesc	char(192)		A description of the pre-auth capture transaction.
fltAPIVersion	float(2,1)	Yes	The version of MCPE Bank Enterprise you are using. Currently, this is 1.3
strTransType	char(30)	Yes	For this transaction type, the value of this field must be <i>CAPTURE</i> .
intReference	int(11)		A numerical reference for this transaction. This value must be unique.
intTestMode	int(1)		Set to 1 if you wish to capture a pre-auth performed in test mode.

## 10 PreAuth Capture Request Response

You will be notified of the outcome of a pre-auth capture in the same session as the PreAuth Capture Request was made. The response fields will be sent as a URL-encoded query string. See Figure 8 below for a list of the fields returned.

**Figure 8: PreAuth Capture Request Response Parameters**

Field	Type(Size)	Notes
intTransID	int(11)	The MCPE unique identifier for this pre-auth capture transaction.
intStatus	int(1)	1 for a successful pre-auth capture, or 0 for failure.
strMessage	char(255)	Any message returned by the bank performing this pre-auth capture.
intTime	int(11)	The time at which this pre-auth capture transaction was authorised, given as the number of seconds since the start of 1970 GMT.
strTransType	char(30)	For this transaction type, the value of this field will be <i>CAPTURE</i> .
fltAPIVersion	float(2,1)	The version of Bank Enterprise MCPE you are using. Currently, this is 1.3.
intReference	int(11)	Your numerical reference for this transaction, if you supplied one.

## 11 PreAuth Void Request Initiation

You must submit the following five fields to initiate a PreAuth Void Request. Your request must be submitted via HTTPS POST to <https://secure.metacharge.com/mcpe/corporate>

- **intInstID** (your installation number)
- **intTransID** (the ID of the pre-auth transaction that is to be void)
- **strSecurityToken** (the value that was returned in the strSecurityToken field of the Payment Request Response for the pre-auth transaction that is to be void)
- **fltAPIVersion** (the version of Bank Enterprise MCPE you are using)
- **strTransType** (the type of transaction to be performed)

See Figure 9 for a complete set of fields that may be submitted in a PreAuth Void Request POST.

**Figure 9: PreAuth Void Request Parameters**

Field	Type(Size)	Required?	Notes
intInstID	int(6)	Yes	The MCPE unique identifier for the installation performing this pre-auth void.
intTransID	int(11)	Yes	The MCPE unique identifier for the pre-auth transaction that is to be void.
strSecurityToken	char(32)	Yes	The value that was sent in the strSecurityToken field of the Payment Request Response for the pre-auth transaction that is to be void.
strDesc	char(192)		A description of the pre-auth void transaction.
fltAPIVersion	float(2,1)	Yes	The version of Bank Enterprise MCPE you are using. Currently, this is 1.3
strTransType	char(30)	Yes	For this transaction type, the value of this field must be VOID.
intReference	int(11)		A numerical reference for this transaction. This value must be unique.
intTestMode	int(1)		Set to 1 if you wish to void a pre-auth performed in test mode.

## 12 PreAuth Void Request Response

You will be notified of the outcome of a pre-auth void in the same session as the PreAuth Void Request was made. The response fields will be sent as a URL-encoded query string. See Figure 10 below for a list of the fields returned.

**Figure 10: PreAuth Void Request Response Parameters**

Field	Type(Size)	Notes
intTransID	int(11)	The MCPE unique identifier for this pre-auth void transaction.
intStatus	int(1)	1 for a successful pre-auth void, or 0 for failure.
strMessage	char(255)	Any message returned by the bank performing this void.
intTime	int(11)	The time at which this pre-auth void transaction was authorised, given as the number of seconds since the start of 1970 GMT.
strTransType	char(30)	For this transaction type, the value of this field will be VOID.
fltAPIVersion	float(2,1)	The version of MCPE Bank Enterprise you are using. Currently, this is 1.3.
intReference	int(11)	Your numerical reference for this transaction, if you supplied one.

## 13 Subscription Cancellation Request Parameters

You must submit the following four fields to initiate a Subscription (Schedule) Cancellation Request. Your request must be submitted via HTTPS POST to <https://secure.metacharge.com/mcpe/corporate>

- **intInstID** (your installation number)
- **intScheduleID** (the ID of the schedule that is to be cancelled)
- **fltAPIVersion** (the version of Bank Enterprise MCPE you are using)
- **strTransType** (the type of transaction to be performed)

See Figure 11 for a complete set of fields that may be submitted in a Schedule Cancellation Request POST.

**Figure 11: Subscription Cancellation Request Parameters**

Field	Type(Size)	Required?	Notes
intInstID	int(6)	Yes	The MCPE unique identifier for the installation that the schedule was performed on.
intScheduleID	int(11)	Yes	The MCPE unique identifier for the schedule that is to be cancelled.
fltAPIVersion	float(2,1)	Yes	The version of MCPE Bank Enterprise you are using. Currently, this is 1.3
strTransType	char(30)	Yes	For this transaction type, the value of this field must be <i>CANCEL</i> .
intTestMode	int(1)		If this schedule was performed in Test Mode, then this must be set to 1.

\* Please note that intTestMode this will not test a cancellation, it is used when you wish to cancel a subscription which was performed in test mode.

## 14 Subscription Cancellation Request Response

You will be notified of the outcome of a cancellation in the same session as the Cancellation Request was made. The response fields will be sent as a URL-encoded query string. See Figure 12 below for a list of the fields returned.

**Figure 12: Subscription Cancellation Request Response Parameters**

Field	Type(Size)	Notes
intScheduleID	int(11)	The MCPE unique identifier for this schedule.
intStatus	int(1)	1 for a successful cancellation, or 0 for failure.
strMessage	char(255)	Any message relevant to the cancellation request.
intTime	int(11)	The time at which this schedule was cancelled, given as the number of seconds since the start of 1970 GMT.
strTransType	char(30)	For this transaction type, the value of this field will be <i>CANCEL</i> .
fltAPIVersion	float(2,1)	The version of Bank Enterprise MCPE you are using. Currently, this is 1.3.
intTestMode	int(1)	Set to 1 if the schedule was performed in test mode.

## 15 Preventing Online Credit Card Fraud

PayPoint.net recommend **FraudGuard**. This real-time fraud service operates on all Bank Enterprise transactions unless otherwise expressly agreed. For more information on FraudGuard please consult the **FraudGuard User Guide**.

Transactions processed via MCPE with FraudGuard result in a FraudGuard score response and detailed transaction information available via the Merchant Extranet. These enable you to make more qualified judgements about each of your consumers.

FraudGuard data includes powerful metrics covering GeoIP location (the actual location of the consumer), card issuer location, and discrepancies therein – plus a vast array of other measures including number of recent similar requests.

The service acts to screen all transactions by generating a FraudGuard score. It also provides detailed Territory Management and a Blacklisting and Whitelisting function. A full description of FraudGuard is outside the bounds of this document.

However, additional techniques can be used with Bank Enterprise MCPE to gain further confidence

### 15.1 Manual Checks

#### Know Your Customer (KYC)

KYC is a process many businesses use to establish more trusting and hence more profitable relationships with consumers. It can be used to evaluate suspicious consumers or before acceptance and/or fulfilment of higher value transactions.

There is no single process attached to KYC but it would typically involve validating the cardholder identity. This might be achieved by collecting telephone number at payment and contacting the consumer to establish their identity, or any other cross-check.

It might involve querying the issuer of their card and checking that against information published on Transaction Detail via the Merchant Extranet. PayPoint.net provide a service called **Verify Your Customer** (VYC) which also validates identities instantly.

#### Manual Authorisation with Signature

This is an excellent way of verifying the card holder as part of a formal KYC process. It also serves as strong mitigation against the risk of Chargebacks. For more information on chargebacks please see the **Chargebacks Guide**.

This manual authorisation process involves sending a consumer a transaction confirmation document to sign and return with copies of their card. The trade-off is that it makes the customer do more work, but serves as excellent fraud mitigation.

For your convenience, PayPoint.net provides a part-completed manual authorisation form via a link on each transaction detail screen on the Merchant Extranet. Simply launch the form from alongside the cardholder name and email/fax to the consumer.

### 15.2 Automated Processes

#### Deferred (Pre-Authorised) Transactions

Any **Payment Request** can be submitted as a Pre-Authorisation. This means the card is not debited immediately but instead the transaction can be completed within 7 days, using an additional API request, or by setting an automatic delayed capture.

Automated capture is configured for your account via *Account Management > Installations*. Deferring transactions gives you the opportunity to evaluate risk before accepting liability. It gives you a risk free period in which to conduct KYC processes.

#### Geo-IP Location & Limits

As well as contributing to our FraudGuard service, we indicate via the payment response whether our GeoIP check detected if a cardholder web request was from their stated country or not. This allows you to make immediate decisions on legitimacy.

Many additional controls exist in MCPE. As well as minimum and maximum transaction limits described on page 7, we also control maximum number of attempts from unique card or IP within a 24 hour period. Our Risk team can adjust this for you.

## 16 Enabling 3D Secure

PayPoint.net Bank Enterprise MCPE has full support for 3D Secure via our own dedicated **Merchant Plug-In (MPI)**. This solution involves simple extensions to the API detailed in this document. You will find full details on how to integrate 3D Secure via our separate 3D Secure supplement, *MCPE Bank Enterprise 3D Secure Integration Guide*.

Alternatively, for those Merchants wishing to use 3<sup>rd</sup> party 3D Secure software, the document also includes an appendix on the fields available in the Bank Enterprise MCPE API for integration of such a service.

## Appendix A: Constructing HTTP requests over SSL

Communication to Bank Enterprise MCPE is performed via HTTP over an SSL connection. Below is an example HTTP request and an example of how to perform the request using PHP. Please note that all headers must be sent. It is essential that the Content-Type is present and set correctly.

**Figure 13: Example HTTP Request**

```
POST /mcpe/corporate HTTP/1.0
Host: secure.metacharge.com
Content-Type: application/x-www-form-urlencoded
Content-Length: 86
Connection: Close

intInstID=12345&intTransID=12345&strSecurityToken=12345&fltAmount=10.00&strDesc=Refund&fltAPIVersion=1.3&...strTransType=REFUND
```

**Figure 14: Example in PHP**

```
<?php

$httpRequest = Array();
$errors = Array();

// Create the HTTP Request
$postContent =
"intInstID=12345&intTransID=12345&strSecurityToken=12345&fltAmount=10.00&strDesc=Refund&fltAPIVersion=1.3&...strTransType=REFUND";

$httpRequest[] = "POST /mcpe/corporate HTTP/1.0";
$httpRequest[] = "Host: secure.metacharge.com";
$httpRequest[] = "Content-Type: application/x-www-form-urlencoded";
$httpRequest[] = "Content-Length: ". strlen($postContent) ;
$httpRequest[] = "Connection: Close";
$httpRequest[] = "";
$httpRequest[] = $postContent;

// Open socket connection
// Send HTTP Request
// Read HTTP Response
$httpResponse = "";

$secureSocket = fsockopen("ssl://secure.metacharge.com",443,$errno, $errstr, 3);

if ( !$secureSocket || ! is_resource($secureSocket) ) {
    $errors[] = "Could not establish connection [$errno : $errstr]";
} elseif( fwrite( $secureSocket , join("\n",$httpRequest) ) ) {
    while (!feof($secureSocket)){
        $httpResponse .= fgets($secureSocket,128);
    }
    fclose($secureSocket);
} else {
    $errors[] = "Could not write to secure connection";
}
?>
```