Merchant Card Payment Engine

Gateway Freedom + IMA Integration Guide

Registered in England No: 2081330. VAT Reg. No: 618184140
Pay360 by Capita Gateway Hosted v 3.0

23rd March 2016

Table of Contents

Table of Figures

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 Pay360 by Capita **Merchant Extranet** is a web based back office system that provides detailed information and powerful tools to assist you in managing your Pay360 by Capita account and can be accessed at the following URL: https://secure.metacharge.com/extranet/

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)333 313 7161.

2 Payment Process Overview

"Gateway Freedom +IMA" MCPE (Merchant Card Payment Engine) is designed for large merchants employing experienced developers. To integrate, 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/.

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.

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.

Note that subject to providing sufficient data in your Payment Request, a **FraudGuard** check will be included in the transaction. However, should you wish to perform only a FraudGuard check without payment processing, this is also supported in MCPE.

What is FraudGuard?
FraudGuard is Pay360 by Capita's market leading integrated fraud and risk management solution. It is simple to configure yet powerful for both enterprise clients and small businesses. The service includes a comprehensive scoring engine which can be used to screen transactions, a full territory management application for managing country by country access, and blacklisting and whitelisting functions.

The service also includes an optional enterprise-grade rules engine which allows configuration of bespoke controls – both general and highly targeted – for managing specific risk profiles at a very detailed level. Taken together all of these services can control transaction outcome or for standalone checks (see section 15) advise of suggested outcome. More information can be found in the FraudGuard user guide.

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 "Freedom +IMA" installation number, refer to Merchant Extranet: *Account Management* > *Installations*)
- **strCartID** (a unique 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 MMYY)
- **intCV2** (the security code that appears on the signature strip of the card)
- **strCardType** (the type of card either VISA, DELTA for VISA DEBIT, MAESTRO, SOLO, MC for MASTERCARD, UKE for VISA ELECTRON)
- **fltAPIVersion** (the version of "Freedom +IMA" 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 range of test card numbers described in section 3.2. 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@example.com">
<input type="hidden" name="strCardNumber" value="1234123412341234">
<input type="hidden" name="intCV2" value="707">
<input type="hidden" name="strExpiryDate" value="0619">
<input type="hidden" name="strCardType" value="VISA">
<input type="hidden" name="strCountry" value="GB">
<input type="hidden" name="fltAPIVersion" value="1.3">
```

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@examp
le.com&strCardType=VISA&strCountry=GB&intTransID=12345678&intAccountID=123456&intStatu
s=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 Pay360 by Capita **FraudGuard** (see section 17) to perform additional look-ups and provide more accurate results. These fields are indicated

in Figure 1. Note that MCPE can process a FraudGuard check only without any payment processing. Please see section 15 for details.

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 3.4)
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(40)	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 DEBIT), SOLO, MAESTRO, MC (for MASTERCARD) or UKE (for VISA ELECTRON).
strUserIP	char(15)		Yes	The IP address of the purchaser. This is used to perform additional FraudGuard geolocation of the customer's actual location, through establishing the country from which a payment is actually made.
strTransType	char(30)	Yes		For this transaction type, the value of the field is *PAYMENT*.
fltAPIVersion	float(2,1)	Yes		The version of "Freedom +IMA" 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 which we can block.
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.
intRecurs	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.
strDigest	char(32)	Yes		An additional authentication of the transaction request; an MD5 hash of a string created as follows: *intInstID + strCardNumber + fltAmount [to 2 decimal places] + strCurrency + your shared key*. See section 3.3 for more details on implementing a digest.

3.2 Test Card Numbers

You can simulate both a successful or failed transaction using the **intTestMode** parameter in your API request and using any of the test card numbers below along with any valid credentials (expiry date, CV2 etc). Setting the value of intTestMode to 1 will result in a successful transaction response. Setting the value to 2 will result in a declined transaction response.

Figure 2: MCPE Test Card Numbers

Card Type	Card Number
VISA	1234123412341230
VISA	4007000000027120
VISA	4012888888881880
VISA	4485680502719430
VISA	4111111111111110
VISA	4444333322221110
MasterCard	5555555555554440
Solo	6312345678912340
Solo	6334580500000000
Maestro (UK)	6331101999990010
Maestro	6400051234567890
Maestro	6333333333331120

Setting the value of intTestMode to 0 or omitting it altogether will ensure a live transaction is requested.

3.3 Digest Security

You may optionally secure all your transaction requests by supplying us with a **Digest Key** to configure on your account. This is then used to prove that all requests are from you or a trusted source with access to your chosen key (password). This is possible because you supply one additional parameter in each request, strDigest, constructed as follows:

- **strDigest** (an MD5-hash of a single string concatenating these variables: *intInstID + strCardNumber + fltAmount [to 2 decimal places] + strCurrency + your shared key*)

By providing us with your key during setup of your account, we are able to construct the same digest using known parameters in each payment request and if this matches the digest which you also submit, we can establish that it is a trusted request and continue processing of it. To enable digest checking, please email completesupport@paypoint.net quoting your installation ID.

Please Note: Once use of a Digest Security has been enabled on your installation, in order to successfully initiate a transaction, the *strDigest* parameter must also be included in the API request. This key must be provided to us in advance of commencing processing of transactions (for example at the same time as requesting Digest Security), or all transactions will be declined.

As an example of constructing *strDigest*, if the shared key was 'TROP1CAL' then the *strDigest* for installation ID 123456, for a £500 payment to card 4111111111111111, would be the MD5-hash of '123456411111111111111111500.00GBPTROP1CAL'.

3.4 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. There is no chargeback liability while a payment is deferred.

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 9 and 11. You can also configure automatic capture or void on the Merchant Extranet via Account *Management* > *Installations*, or capture these manually via *Sales* > *Pre Auths*.

3.5 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

Other limits apply to accounts opened in other currencies. Please contact our support team if you need details of limits in other currencies. Please contact your account manager to request higher limits on your account, subject to approval by our risk team.

3.6 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*n*, strSchPeriod*n* (where *n* is 1, 2 or 3) and intRecurs in a **Payment Request**. intRecurs 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	fltSchAmount1 = 1.00, strSchPeriod1=7D fltSchAmount2 = 5.00, strSchPeriod2=1M intRecurs=1
£20.00 per week	fltSchAmount1=20.00, strSchPeriod1=1W intRecurs=1
£2.00 for the first 7 days £5.00 for the next 3 weeks £8.00 per month thereafter	fltSchAmount1=2.00, strSchPeriod1=7D fltSchAmount2=5.00, strSchPeriod2=3W fltSchAmount3=8.00, strSchPeriod3=1M intRecurs=1
£3.00 for the first 7 days £5.00 per month thereafter, automatic cancellation after 6 successful payments	fltSchAmount1 = 3.00, strSchPeriod1=7D intRecurs=1, intCancelAfter=6
Free for a week £10.00 per month thereafter	fltSchAmount1=0, strSchPeriod1=7D fltSchAmount2=10.00, strSchPeriod2=1M intRecurs=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.

3.7 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 3.

Figure 3: 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 *strCurrency* 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.
strCardType	char(6)	The purchaser's card type. Values: AMEX, VISA, MC, DELTA, SOLO, MAESTRO, UKE, UKMAESTRO, VISADEBIT.
intAVS	int(1)	The result of the AVS check performed for this transaction. **This field will be omitted if the check was not performed**. Values: 0=AVS check failed, 1=AVS check passed.
intCV2	int(1)	The result of the CV2 check performed for this transaction. **This field will be omitted if the check was not performed**. 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. **This field will be omitted if the check was not performed**. 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(40)	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 "Freedom +IMA" MCPE you are using. Currently, this is *1.3*
strTransType	char(30)	For this transaction type, the value of this field will be *PAYMENT*.
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.
str3DSResult	char (20)	For Merchants using 3D Secure via our MPI, you can optionally have us return the outcome of the 3D Secure check once decoded by our systems. Please first contact our support team via completesupport@paypoint.net to request that this is enabled. The possible values returned are: UNCHECKED, PASS, FAIL, BYPASS, ENROLLED, ENROLFAIL, INELIGIBLE. Please see the MCPE Bank Enterprise 3D Secure supplement for further details on 3D Secure and these outcomes.
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.
intScheduleID	int(11)	The MCPE unique identifier for any payment schedule associated with this transaction (if applicable).

**Returned only if supplied in your request*

Figure 4: Additional FraudGuard MCPE Payment Response Parameters

Group	Field	Type(Size)	Notes	Example
General	strFGBillingCountry	char(2)	Two-letter ISO 3166-1 alpha-2 code containing stated (billing address) country	GB
Card Issuer	strFGBINCountry	char(2)	Two-letter ISO 3166-1 alpha-2 code inidicating country of issue of the card	GB
	strFGBINIssuer	char(192)	Name of the card issuing bank	First National Bank

Geo-Location	strFGIPCountry	char(10)	Either:
			• Two-letter ISO 3166-1 alpha-2 code indicating geo-located country of consumer
			• One of the following special values:
			  • PROXY – the IP address is associated with an anonymous proxy
			  • SATELLITE – the IP address is associated with a satellite ISP
			  • EUROPE – the IP address is generally allocated for use across the European region
			  • APAC – the IP address is generally allocated for use across the Asia/Pacific region
				FR

	strFGIPCity	char(192)	Geo-located city of consumer	Paris
	strFGIPRegion	char(192)	Geo-located region of consumer	Ile-de-France
	intFGDistanceIPToBilling	int(11)	Distance (in kilometres) from the consumer's IP geo-location to their stated billing address	454

Channel Risk	intFGFreeEmailProvider	int(1)	Indicates whether or not the email address is from a free provider (e.g. Google, Hotmail etc):
			  • 1 – email address is from a free provider
			  • 0 – email address is not from a free provider
				0

	fltFGOpenProxyRisk	float(3,2)	Percentage risk (expressed as a decimal) that the IP address is an open proxy	0.12
Recent Activity	intFGLast24hIPAttempts	int(11)	Number of attempted transactions through the consumer's IP address in the last 24 hour period	2
	intFGLast24hCardAttempts	int(11)	Number of attempted transactions on the customer's card in the last 24 hour period	6
Identity Morphing	intFGMorphingAgainstAddress	int(11)	Sum of morphing on other factors against postal address; that is, the total count of distinct email addresses, IP addresses and card or payment account attempted against the given address	2
	intFGMorphingAgainstEmail	int(11)	Sum of morphing on other factors against the email address; that is, the total count of distinct postal addresses, IP addresses and card or payment account attempted against the given email address	3
	intFGMorphingAgainstCard	int(11)	Sum of morphing on other factors against the card or payment account; that is, the total count of distinct email addresses, postal addresses and IP addresses attempted against the given card or payment account	4
	intFGMorphingAgainstIP	int(11)	Sum of morphing on other factors against the IP address; that is, the total count of email addresses, postal addresses and card or payment account attempted against the given IP address	2

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 "Freedom +IMA" 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 4 for a complete set of fields that may be submitted in a Refund Request POST.

Figure 5: 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 "Gateway Freedom +IMA" you are using. Currently, this is *1.3*
strTransType	char(30)	Yes	For this transaction type, the value of this field must be *REFUND*.
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 5 below for a list of the fields returned.

Figure 6: 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 *REFUND*.
fltAPIVersion	float(2,1)	The version of MCPE "Gateway Freedom +IMA" 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 "Freedom +IMA" MCPE you are using)
- **strTransType** (the type of transaction to be performed)

See Figure 8 for a complete set of fields that may be submitted in a Repeat Payment Request POST.

Figure 7: 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 "Gateway Freedom +IMA" you are using. Currently, this is *1.3*
strTransType	char(30)	Yes	For this transaction type, the value of this field must be *REPEAT*.
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.
strUserIP	char(15)		The IP address of the purchaser. If we have enabled FraudGuard on Repeats for you, this is used to perform additional FraudGuard geolocation of the customer's actual location, through establishing the country from which a payment is actually made.

Note that for those Merchants using **FraudGuard**, if you've asked us to enable **FraudGuard on Repeat Transactions**, then by optionally supplying the **strUserIP** parameter with your repeat request, you will generate a fresh FraudGuard analysis. This can be useful for re-scoring and re-screening the transaction in order to track identity morphing and repeated card or IP use.

Please contact completesupport@paypoint.net should you wish to enable FraudGuard on Repeat Transactions.

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 9 below for a list of the fields returned.

Figure 8: 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 *REPEAT*.
fltAPIVersion	float(2,1)	The version of MCPE "Gateway Freedom +IMA" 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 "Freedom +IMA" MCPE you are using)
- **strTransType** (the type of transaction to be performed)

See Figure 10 for a complete set of fields that may be submitted in a PreAuth Capture Request POST.

Figure 9: 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 "Gateway Freedom +IMA" you are using. Currently, this is *1.3*
strTransType	char(30)	Yes	For this transaction type, the value of this field must be *CAPTURE*.
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 11 below for a list of the fields returned.

Figure 10: 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 *CAPTURE*.
fltAPIVersion	float(2,1)	The version of "Freedom +IMA" 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 "Freedom +IMA" MCPE you are using)
- **strTransType** (the type of transaction to be performed)

See Figure 12 for a complete set of fields that may be submitted in a PreAuth Void Request POST.

Figure 11: 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 "Gateway Freedom +IMA" 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 13 below for a list of the fields returned.

Figure 12: 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 "Gateway Freedom +IMA" 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 can use this API to remotely cancel any active subscription created previously (see section 3.6). You can even cancel those subscriptions created through your other installations (for example via Virtual Terminal or the hosted Fast-Track pages).

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 "Freedom +IMA" installation number – not the installation of the subscription)
- **intScheduleID** (the ID of the schedule that is to be cancelled)
- **fltAPIVersion** (the version of "Freedom +IMA" MCPE you are using)
- **strTransType** (the type of transaction to be performed)

See Figure 14 for a complete set of fields that may be submitted in a Schedule Cancellation Request POST.

Figure 13: Subscription Cancellation Request Parameters

Field	Type(Size)	Required?	Notes
intInstID	int(6)	Yes	The MCPE unique identifier for your "Gateway Freedom +IMA" installation.
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 "Gateway Freedom +IMA" you are using. Currently, this is *1.3*
strTransType	char(30)	Yes	For this transaction type, the value of this field must be *CANCEL*.
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 15 below for a list of the fields returned.

Figure 14: 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 *CANCEL*.
fltAPIVersion	float(2,1)	The version of "Freedom +IMA" MCPE you are using. Currently, this is *1.3*.
intTestMode	int(1)	Set to *1* if the schedule was performed in test mode.

15 Transaction Confirm Request Initiation

The API enables Merchants to verify the outcome and response of a previously completed Payment, Refund, PayOut, Repeat, PreAuth Capture or PreAuth Void transaction. This is useful should communications be interrupted, or for reconciliation.

To confirm the outcome of a previous transaction via MCPE, a request of type *CONFIRM* should be sent, accompanied by the Merchant transaction reference previously provided in *strCartID*, with the original transaction. Note that *intTransID*, a PayPoint reference used in other API functions, is not supported, as it is assumed not yet known to a Merchant performing this function.

You may optionally supply a *strConfirmType* parameter to limit the search to transactions of a given type. This is useful if you have persisted the same *strCartID* reference between a payment and its refund but only wish to confirm the outcome of one.

You must submit the following fields to initiate a **Transaction Confirm Request**. Your request must be submitted via HTTPS POST to https://secure.metacharge.com/mcpe/corporate. Those fields in italics are optional, as required.

- **strTransType** (the type of transaction you wish to perform, in this case CONFIRM)
- **intInstID** (your applicable installation number, see Merchant Extranet: *Account Management* > *Installations*)
- **strCartID** *(*your own prior unique order number or session by which you can identify the user/transaction*)*
- **fltAPIVersion** (the version of Bank Enterprise MCPE you are using)
- ***strConfirmType*** *(provides additional filtering based on the original transaction type – provided in strTransType in the original request – either PAYMENT, REFUND, PAYOUT, REPEAT, CAPTURE or VOID)*

The table below details the fields that can be submitted in a Transaction Confirm request to MCPE.

Figure 15: MCPE Transaction Confirm Request Parameters

Field	Type(Size)	Required?	Notes
intInstID	int(6)	Yes	The unique identifier for the MCPE Bank Enterprise installation.
strCartID	char(192)	Yes	Your unique identifier for the prior transaction to be searched.
strTransType	char(30)	Yes	For this transaction type, the value of the field is *CONFIRM*.
fltAPIVersion	float(2,1)	Yes	The version of Bank Enterprise MCPE you are using. Currently, *1.3*
strConfirmType	char(30)		The transaction type of the original transaction to be confirmed, either PAYMENT, REFUND, PAYOUT, REPEAT, CAPTURE or VOID.

An example request for a Transaction Confirm check on a prior PAYOUT transaction might be constructed as follows:

```
<form action="https://secure.metacharge.com/mcpe/corporate" method="POST">
<input type="hidden" name="intInstID" value="123456">
<input type="hidden" name="strCartID" value="816RONDOERON-5700">
<input type="hidden" name="fltAPIVersion" value="1.3">
<input type="hidden" name="strTransType" value="CONFIRM">
<input type="hidden" name="strConfirmType" value="PAYMENT">
</form>
```

If you process additional alternative payment methods which are not included by name in our discrete *strPaymentType* values shown in figure 17, then please contact completesupport@paypoint.net quoting the payment method required in FraudGuard.

Important Note: The ability to make 'CONFIRM' requests to MCPE isn't enabled on your account by default. Please contact your account manager or completesupport@paypoint.net to enable this. A limited number of checks are permitted each day.

16 Transaction Confirm Response

You will be notified of the outcome in the same session as your request. The response fields are sent as a URL-encoded query string and will consist of the original **Transaction Confirm Request** that you submitted to MCPE, as well as our additional **Transaction Confirm Response Parameters** shown in Figure 17.

Figure 16: MCPE Transaction Confirm Response Parameters

Field	Type(Size)	Notes
intStatus	int(1)	The status of this *CONFIRM* transaction. Values: 0=failed, 1=successful.
strMessage	char(255)	Any message returned by MCPE when this *CONFIRM* request was processed, typically returned only if no matching transaction is found with the reference provided.
strCartID	char(255)	Your unique identifier for the prior transaction to be searched.
fltAPIVersion	float(2,1)	The version of Bank Enterprise MCPE you are using. Currently, this is *1.3*
intTime	int(11)	The time at which the response was issued to this *CONFIRM* transaction request.
strTransType	char(30)	For this transaction type, the value of this field will be *CONFIRM*.
strConfirmType*	char(30)	The transaction type of the original transaction to be confirmed, either PAYMENT, REFUND, PAYOUT, REPEAT, CAPTURE or VOID.
strResponse	blob	If a valid transaction is found matching the unique identifier provided in *strCartID*, this parameter will return a URL-encoded string containing all relevant API response fields that would have been returned in response to the original transaction request. Please refer to each of the prior API response sections to review the format of these.

**Returned only if supplied in your request*

Given the unusual nature of this API method in returning a prior response, within a response, an example CONFIRM response which successfully locates a Payment, is shown below, with the URL encoded strResponse parameter clearly identified in red.

```
intTime=1308914337&fltAPIVersion=1.3&intStatus=1&strTransType=CONFIRM&strResponse=intTime%3D1308914335%26fltOriginalAmount%3D10.00%26strOriginalCurrency%3DGBP%26intTransID%3D12345678%26intInstID%3D123456%26strCartID%3D816RONDOERON-5700%26intAccountID%3D2%26intStatus%3D1%26strDesc%3DTest%2BTransaction%26fltAmount%3D10.000%26strMessage%3D%26intCountryIP%3D1%26strTransType%3DPAYMENT%26fltFraudScore%3D0.000%26intTestMode%3D%26strCardHolder%3DRon%2BDoe%26strAddress%3D23%2BTest%2BRoad%26strCountry%3DGB%26strEmail%3Dtester%2540localhost%26strCurrency%3DGBP%26strUserIP%3D81.93.226.45%26strCity%3DTestville%26strCardType%3DVISA
```

Please note that no new transaction is created by a CONFIRM request, so the only transaction ID applicable to such a request is that which is recovered and returned within the *strResponse* parameter in the case of a successfully matched transaction.

17 FraudGuard Check Request Initiation

Pay360 by Capita also enable Merchants to make 'non-authorisation' requests to MCPE, for example in order to simply obtain the output of a **FraudGuard** check for a planned future card transaction or for a payment which is to be processed elsewhere.

FraudGuard checks take place on all transactions which are submitted to MCPE where relevant data is supplied and therefore is supported in all previously documented payment request mechanisms; this method simply allows a standalone check.

You must submit the following fields to initiate a standalone **FraudGuard Check Request**. Your request must be submitted via HTTPS POST to https://secure.metacharge.com/mcpe/corporate. Those in italics are optional but recommended.

- **strTransType** (the type of transaction you wish to perform, in this case NONAUTH)
- **intInstID** (your "Freedom +IMA" installation number, refer to Merchant Extranet: *Account Management* > *Installations*)
- **strCity** (the card holder's billing city)
- **strCountry** (the card holder's billing country)
- **strEmail** (an e-mail address for the card holder)
- **strUserIP** (the IP address of the customer)
- **strCardNumber** (the card number)
- **strExpiryDate** (the expiry date that appears on the card, formatted as MMYY)
- **strCardType** (the type of card either VISA, DELTA for VISA DEBIT, MAESTRO, SOLO, MC for MASTERCARD, UKE for VISA ELECTRON, LASER, JCB or alternatively **strPaymentType** for an alternative payment method, see section 15.1)
- **fltAPIVersion** (the version of "Freedom +IMA" MCPE you are using)
- ***strCartID*** *(unique 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)*

The table below details the fields that can be submitted in a standalone FraudGuard request to MCPE. Fields not marked as required contain information that can be held in the MCPE system, is not necessary for FraudGuard but may be useful.

Figure 17: MCPE FraudGuard Check Request Parameters

Field	Type(Size)	Required?	Enhances FraudGuard?	Notes
intInstID	int(6)	Yes		The unique identifier for the MCPE "Freedom +IMA" installation.
strCartID	char(192)	Yes		Your unique identifier for this transaction, for your reconciliation.
strDesc	char(192)	Yes		Descriptive text for this transaction (defaults to 'Check Only').
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)	for *fltAmount*	Yes	The 3-letter ISO code for the currency of payment fltAmount
intTestMode	int(1)			If included, indicates a test transaction. A VISA with card number 1234123412341234 should be used. Values: 0=off, 1=on.
strCardHolder	char(40)		Yes	The name of the card holder, as it appears on the card.
strAddress	char(255)		Yes	The customer's postal billing address.
strCity	char(40)	Yes	Yes	The customer's city.
strState	char(40)		Yes	The customer's state, province or county.
strPostcode	char(15)		Yes	The postal code associated with the address in strAddress.
strCountry	char(2)	Yes	Yes	The 2-letter ISO code for the customer's country.
strTel	char(50)			The customer's telephone number.
strFax	char(50)			The customer's fax number.
strEmail	char(100)	Yes	Yes	The customer's e-mail address.
strCardNumber	char(20)	Yes or below	Yes	The card number.
strPaymentDetail	char(20)	Yes or above	Yes	The unique customer identity via an alternative payment method
strCardType	char(8)	Yes or below		The type of card - either VISA, DELTA (for VISA DEBIT), SOLO, MAESTRO, MC (for MASTERCARD) or UKE (for VISA ELECTRON).
strPaymentType	char(8)	Yes or above		The type of alternative payment method (see section 15.1)
strUserIP	char(15)	Yes	Yes	The IP address of the customer. This is used to perform additional FraudGuard geolocation of the customer's actual

				location.
strTransType	char(30)	Yes		For this transaction type, the value of the field is *NONAUTH*.
fltAPIVersion	float(2,1)	Yes		The version of "Freedom +IMA" MCPE you are using. Currently, *1.3*
intReference	int(11)			A numerical reference for this transaction, which must be unique. Can be used to alert us to duplicate requests which we can block.

17.1 Alternative Payment Methods

The **Merchant Card Payment Engine** supports **FraudGuard** checks in relation to a wide range of alternative payment methods that you might choose to process via another provider. This enables clients to use FraudGuard for all their customer payments.

To perform a FraudGuard check for an alternative payment method simply replace *strCardType* and *strCardNumber* parameters in the request described previously in section 15 and instead include values for *strPaymentType* and *strPaymentDetail*.

- **strPaymentType** (the type of alternative payment method used, see figure 18 below)
- **strPaymentDetail** (the unique customer identifier associated with this payment method – i.e. account ID, username)

The identifier for each alternative payment type is described below in figure 19. By identifying these separately we'll report on transactions for each type, allow you to build fraud rules specific to each, and monitor velocity from a unique *strPaymentDetail*.

Figure 18: MCPE FraudGuard Check Alternative Payment Methods

Alternative Payment Type	strPaymentType
MoneyBookers	MB
Neteller	NETELLER
Click2Pay	CLICKPAY
Click & Buy	CLICKBUY
Ukash	UKASH
PaySafe Card	PAYSAFE
EcoCard	ECOCARD
GiroPay	GIRO
ELV	ELV
instaDEBIT	INSTA
Nordea Solo	NORDEA
POLi	POLI
iDEAL	IDEAL
Bank Transfer	BANK
Local Wire Transfer	WIRE
Any Other Payment Method	OTHER

The *strPaymentDetail* for each method will vary depending on the unique identifier used by the customer to make payment, for example an email address with MoneyBookers, a back account number with a bank transfer and an account ID with Neteller.

An example request for a FraudGuard check on a Neteller customer might be constructed as follows:

```
<form action="https://secure.metacharge.com/mcpe/corporate" method="POST">
<input type="hidden" name="intInstID" value="123456">
<input type="hidden" name="strCartID" value="654321">
<input type="hidden" name="strDesc" value="Fraud Screening">
<input type="hidden" name="fltAmount" value="10.00">
<input type="hidden" name="strCurrency" value="GBP">
<input type="hidden" name="strPaymentType" value="NETELLER">
<input type="hidden" name="strPaymentDetail" value="450000000001">
<input type="hidden" name="strEmail" value="test@Pay360 by Capita">
<input type="hidden" name="strPostcode" value="BA12BU">
<input type="hidden" name="strCountry" value="GB">
<input type="hidden" name="fltAPIVersion" value="1.3">
<input type="hidden" name="strTransType" value="NONAUTH">
```

If you process additional alternative payment methods which are not included by name in our discrete *strPaymentType* values shown in figure 17, then please contact completesupport@paypoint.net quoting the payment method required in FraudGuard.

Important Note: The ability to make 'non-authorisation' requests to MCPE to obtain FraudGuard checks isn't enabled on your account by default. Please contact your account manager or completesupport@paypoint.net to add the service immediately.

18 FraudGuard Check Request Response

You will be notified of the outcome of a transaction in the same session as your Check Request. The response fields are sent as a URL-encoded query string and will consist of the original **FraudGuard Check Request** that you submitted to MCPE, as well as our additional **FraudGuard Check Response Parameters** shown in Figure 20.

The key additional parameters include the FraudGuard score *fltFraudScore* and the suggested outcome *intAdvisory*.

Figure 19: MCPE FraudGuard Check 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, if one was supplied (zero value if not), in the currency specified in the *strCurrency* field.
strCurrency	char(3)	The 3-letter ISO code for the currency associated with this transaction (uppercase).
strMessage	char(255)	Any message returned by FraudGuard when this transaction was processed.
strPaymentType	char(6)	The customer's card type. Values: VISA, MC, DELTA, SOLO, SWITCH, UKE.
intTestMode*	int(1)	Indicates whether this was a test transaction. Values: 0 or not present=live, 1=test transaction.
strCardHolder*	char(40)	The customer's name.
strAddress*	char(255)	The customer's street address.
strCity*	char(255)	The customer's city
strState*	char(255)	The customer's state/county
strPostcode*	char(255)	The customer's postcode.
strCountry*	char(255)	The customer's country.
strTel*	char(255)	The customer's telephone number.
strFax*	char(255)	The customer's fax number.
strEmail*	char(100)	The customer's email address.
strDesc	char(192)	The description of the transaction
strCartID	char(255)	The cartID of the transaction
intCountryIP	int(1)	The result of checking the customer's country as determined from their IP address against the country supplied as part of the billing address. This field will be omitted if the check was not performed. Values: 0=check failed, 1=check passed.
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.**
intAdvisory	**int(1)**	**The advised outcome of the transaction as dictated by your configuration of FraudGuard. Possible values are 0=suggest decline, 1=suggest approve, 2=suggest defer for review.**
intReference*	int(11)	The transaction reference you supplied in your transaction request, if you supplied one.
fltAPIVersion	float(2,1)	The version of "Freedom +IMA" MCPE you are using. Currently, this is *1.3*
strTransType	char(30)	For this transaction type, the value of this field will be *NONAUTH*.
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.

**Returned only if supplied in your request*

For information on configuring the **FraudGuard** controls which generate the *intAdvisory* value and handling of *fltFraudScore* – including FraudGuard score screening, FraudGuard territory management, FraudGuard blacklisting and whitelisting, and the FraudGuard rules engine – please consult the separate FraudGuard User Guide. Available from *Resources* > *Documentation*.

19 Preventing Online Credit Card Fraud

Pay360 by Capita recommend **FraudGuard**. This real-time fraud service operates on all “Freedom +IMA” transactions unless otherwise expressly agreed.

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 includes powerful metrics covering GeoIP location (the actual location of the consumer), card issuer location, stated location and discrepancies therein – plus a vast array of other measures including card and IP velocity (rate of transactions).

The service acts to screen all transactions by generating a FraudGuard score. It also provides detailed Territory Management an enterprise rules engine and a Blacklisting and Whitelisting function. A full description is outside the bounds of this document. Hwever, additional techniques can be used with “Freedom +IMA” MCPE to gain further confidence

19.1 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.

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.

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, Pay360 by Capita 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.

19.2 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.

20 Enabling 3D Secure

Pay360 by Capita “Freedom +IMA” 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 “Freedom +IMA” 3D Secure Integration Guide*.

Alternatively, for those Merchants wishing to use 3rd party 3D Secure software, the document also includes an appendix on the fields available in the “Freedom +IMA” MCPE API for integration of such a service.

Appendix A: Constructing HTTP requests over SSL

Communication to “Freedom +IMA” 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 20: 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&flt
APIVersion=1.3&…
…strTransType=REFUND
```

Figure 21: Example in PHP

```
<?php

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

// Create the HTTP Request
$postContent =
"intInstID=12345&intTransID=12345&strSecurityToken=12345&fltAmount=10.00&strDesc=Refund&fl
tAPIVersion=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";
}
?>
```