Real-time Implementation Guide

Contents

1 Overview and Summary Checklist

To implement the screening service, there are four simple steps:

Step 1. Identify the data on your systems

The more comprehensive the data feed, the more rules we will perform on it. The data-fields to be used are described in the Data Fields and Descriptions section of this document.

Step 2. Connection and Response

Connect to the The 3rd Man server, post transactions and receive responses.

Step 3. Release or Hold Transactions

Based on The 3rd Man response, release or hold transactions.

Step 4. Set up access to GateKeeper

Test that data displayed on GateKeeper is as input at your web site or by your call centre.

Update transaction status to release or rejected transactions.

Agree Schedules and SLAs and go live.

2 Transaction Posting

It is important that all transactions are presented to the service. That includes transactions that are declined by the bank as well as any that you may want to decline at your front-end (e.g. if you hold your own 'hot-list', or if you use SuperSearch data as a front-end check)

We undertake integration of this data into the 3rd Man screening engine.

3 Real-Time Engine

The real-time engine can be used in one of two ways.

The first method provides an initial risk assessment for transactions that are then forwarded to the off-line screening engine for more rigorous checks to be performed.

The second method is used for data collection and no initial risk assessment is made. The transactions are forwarded to the off-line screening engine.

Access to the servers is restricted via IP address and data is transmitted using at least 128 bit SSL.

The data should be submitted via an HTTPS POST command as detailed in the section entitled Data Fields and Descriptions.

All Real-Time transactions should be POSTed to the following URLs, depending on the service required:

URL	Description
https://cnprealtime.com/t3m/ThirdMan	Normal realtime call using HTTP attribute value pairs. Normal rules apply and values should be URL encoded.
https://cnprealtime.com/t3m/ThirdManXML	Normal realtime call a XML string. A valid XML string should be the sole content of the POST request. Compression is strongly recommended for these calls.
https://cnprealtime.com/t3m/RuleString	The rule string call (see below).
https://cnprealtime.com/t3m/RuleStringXML	The same but as XML see comments above.
https://cnprealtime.com/t3m/Response	Poll for the response string (see below).
https://cnprealtime.com/t3m/ResponseXML	The same but as XML see comments above.
https://cnprealtime.com/t3m/Blackbox	Black box call to identify client divice, note this can also be piggybacked on the main call. See below.
https://cnprealtime.com/t3m/BlackBoxXML	The same but as XML see comments above.
https://cnprealtime.com/t3m/Callback	Test call to ensure callbacks are working.
https://cnprealtime.com/t3m/CallbackXML	The same but as XML see comments above.
https://cnprealtime.com/t3m/SoapWrapper	XML calls (or the JSON equivalent) can be wrapped in a SOAP call. While not the most efficient way or doing things it is easy for some developers.

Unsupported calling examples are available at https://cnprealtime/t3m/examples.tar.gz or https://cnprealtime/t3m/examples.tar.gz. This is a good starting point for new integrators.

The 3rd Man will process the information you submit and reply in the response object of the same POST.

The results will be a series of value pairs separated by '&' as detailed in the section titled "Real-Time Response" or as an XML string.

4 Call Compression

Calls to the Real-Time service should be compressed if they are large, XML calls especially. This will significantly improve the responsiveness of the service. To compress calls use standard gzip http compression on the POST contents to the server. Do not forget to add the standard "Content-Encoding: gzip" header.

Similarly to receive the response compressed ensure the header "Accept-Encoding: gzip" is also included in the request and check for "Content-Encoding: gzip" in the response, in which case uncompress the response.

Note that as generally the calls sent to the server are much bigger than the responses compressing calls to us is much more useful.

5 Data Fields and Descriptions

Field Name	Description	Format / maximum length
transaction_reference	Your unique reference	50
transaction_date_time		YYYY-MM-DD HH:MM:SS
aggregator_identifier	Unique identifier for an aggregator (for example a PSP).	15
merchant_identifier	Used by PSPs to provide merchant identifier	15
merchant_order_ref	The reference number the merchant gives to the order	250
customer_ref	The reference number the Merchant gives to their customer	50
sales_channel	Type of sale, such as Internet or Call centre	1 = Mail order 2 = Telephone Order 3 = Internet 4 = other
cardholder_title	If Cardholder name cannot be split then put all details in Cardholder Surname field	10
cardholder_first_name		50
cardholder_surname		50
card_number	If you prefer not to provide then fields 12, 13 and 14 may be provided as an alternative	Numeric 19
card_bin	First 8 digits of card number	Numeric 8
card_last_4_digits		Numeric 4
card_sha1	May be provided as an alternative to field 11. N.B. please call the 3rd Man technical support for additional data to be added to each card prior to SHA-1 encryption.	Upper case Hex 50
card_issue_number		Numeric 2
card_expiry_date		MMYY
amount	Total transaction value	NNNNNNNNNNN.NN
currency	ISO currency code standard (ISO 4217)	(e.g. GBP, EUR, USD) (3)

Field Name	Description	Format / maximum length
transaction_type	The payment transaction type associated with a transaction. If the transaction is being passed to the 3rd man to notify them of a status (e.g. an order has been set to Rejected, OK, Chargeback or Chargeback Fraud – a transaction type of Update should be given. See Transaction Status Updates below.)	Authorisation type : Auth Refund Pre-auth Deferred Manual Repeat Repeat Deferred Void Cancel Update SuperSearch Account Registration Account Amendment Buyer Registration Seller Registration KYC Withdrawal Amendment Bank Response Update Other
payment_method		
authorisation_code	The bank authorisation code may be null if not applicable and is represented as the empty string. If the transaction type is Update, bank authorisation code should contain the status of the transaction (either OK, Rejected, Chargeback or Chargeback Fraud)	30
bank_response_code	The response code as received from the bank – 2 digit	Numeric 2
bank_response_message	The authorisation message as received from the bank	100
cv2_response	Security code response	0 = not given 1 = not checked 2 = pass 4 = fail
avs_address_response		0 = not given 1 = not checked 2 = pass 4 = fail
avs_postcode_response		0 = not given 1 = not checked 2 = pass 4 = fail
threed_secure_eci_indicator	Transaction ECI Status.	Numeric 2
threed_secure_cavv_avv	Visa transactions – CAVV Mastercard transactions – AVV	50
alt_cardholder_title	If Cardholder name cannot be split then put all details in Cardholder Surname field	10
alt_cardholder_first_name		50
alt_cardholder_surname		50
alt_card_number	If you prefer not to provide then fields 12, 13 and 14 may be provided	Numeric 19

	as an alternative	
alt_card_bin	First 8 digits of card number	Numeric 8
alt_card_last_4_digits		Numeric 4
alt_card_sha1	May be provided as an alternative to field 11. N.B. please call the 3rd Man technical support for additional data to be added to each card prior to SHA-1 encryption.	Upper case Hex 50
alt_card_issue_number		Numeric 2
alt_card_expiry_date		MMYY
alt_amount		NNNNNNNNNNN.NN
alt_currency	ISO currency code standard (ISO 4217)	(e.g. GBP, EUR, USD) (3)
alt_transaction_type	The payment transaction type associated with a transaction. If the transaction is being passed to the 3rd man to notify them of a status (e.g. an order has been set to Rejected, OK, Chargeback or Chargeback Fraud – a transaction type of Update should be given. See Transaction Status Updates below.)	Authorisation type : Auth Refund Pre-auth Deferred Manual Repeat Repeat Deferred Void Cancel Update SuperSearch Account Registration Buyer Registration Seller Registration KYC Withdrawal Amendment Bank Response Update Other
alt_payment_method		
alt_authorisation_code	The bank authorisation code may be null if not applicable and is represented as the empty string. If the transaction type is Update, bank authorisation code should contain the status of the transaction (either OK, Rejected, Chargeback or Chargeback Fraud)	30
alt_bank_response_code	The response code as received from the bank – 2 digit	Numeric 2
alt_bank_response_message	The authorisation message as received from the bank	100
alt_cv2_response	Security code response	0 = not given 1 = not checked 2 = pass 4 = fail
alt_avs_address_response		0 = not given 1 = not checked 2 = pass 4 = fail
alt_avs_postcode_response		0 = not given 1 = not checked 2 = pass 4 = fail
alt_threed_secure_eci_indicator	Transaction ECI Status.	Numeric 2

alt_threed_secure_cavv_avv	Visa transactions – CAVV Mastercard transactions –AVV	50
amount_1		NNNNNNNNNNN.NN
currency_1	ISO currency code standard (ISO 4217)	(e.g. GBP, EUR, USD) (3)
home_telephone_number		Numeric 20
delivery_telephone_number		Numeric 20
mobile_telephone_number		Numeric 20
customer_email_address		50
ip_address		NNN.NNN.NNN.NNN
customer_date_of_birth		YYYY-MM-DD
first_purchase_date	Has the customer purchased previously? Enter the first purchase date if available	YYYY-MM-DD
number_of_previous_purchases	Since first purchase date	Numeric 5
value_of_previous_purchases	Since first purchase date	NNNNNNNNNNN.NN
introduced_by	if introduced by another customer, enter that customer's reference	50
billing_street_address_1		100
billing_street_address_2		100
billing_city		30
billing_county		30
billing_country	Use numeric iso code eg 826 for UK	30
billing_postcode_zipcode		15
alt_billing_street_address_1		100
alt_billing_street_address_2		100
alt_billing_city		30
alt_billing_county		30
alt_billing_country	Use numeric iso code eg 826 for UK	30
alt_billing_postcode_zipcode		15
delivery_customer_title		10
delivery_customer_first_name		50
delivery_customer_surname		50
delivery_street_address_1		100
delivery_street_address_2		100
delivery_city		30
delivery_county		30
delivery_country	Use numeric iso code eg 826 for UK	30
delivery_postcode_zipcode		15
delivery_customer_title_1		10
delivery_customer_first_name_1		50
delivery_customer_surname_1		50
delivery_street_address_1_1		100
delivery_street_address_2_1		100
delivery_city_1		30
delivery_county_1		30
delivery_country_1	Use numeric iso code eg 826 for UK	30
delivery_postcode_zipcode_1		15
delivery_customer_title_2		10
delivery_customer_first_name_2		50
delivery_customer_surname_2		50
delivery_street_address_1_2		100
delivery_street_address_2_2		100
delivery_city_2		30
delivery_county_2		30
delivery_country_2	Use numeric iso code eg 826 for UK	30
delivery_postcode_zipcode_2		15

delivery_customer_title_3		10
delivery_customer_first_name_3		50
delivery_customer_surname_3		50
delivery_street_address_1_3		100
delivery_street_address_2_3		100
delivery_city_3		30
delivery_county_3		30
delivery_country_3	Use numeric iso code eg 826 for UK	30
delivery_postcode_zipcode_3		15
delivery_customer_title_4		10
delivery_customer_first_name_4		50
delivery_customer_surname_4		50
delivery_street_address_1_4		100
delivery_street_address_2_4		100
delivery_city_4		30
delivery_county_4		30
delivery_country_4	Use numeric iso code eg 826 for UK	30
delivery_postcode_zipcode_4		15
delivery_method	Where a special delivery is requested or next day etc. Also enter if a timed delivery is requested (include time)	e.g. next day delivery, 12.00 pm, etc. 30
acquirer	The merchant acquirers name	30
merchant_mid		
time_zone		GMT = 0 4
driving_license_number		30
operator_id		30
delivery_event_departure_date	Delivery date, or in the case of travel and entertainment, the date of travel or show	YYYY-MM-DD
event_departure_pick_up_location		50
destination_location	Used for travel and ticketing. Route codes may be used and defined as required with the 3rd Man support	50
route_via_location	Routing information	50
installation_requested	Where goods are to be supplied together with a physical installation	Yes or No 3
brand	Where a sales channel supports multiple different brands	30
real_time_score	Where a real-time scorecard is used, the score can be provided enabling tuning of real-time facility. This may be the merchant's own score card, a prior use of 3rd Man or any other scoring method	Numeric 5 -1000 to 1000
customer_real_time_score		Numeric 5 -1000 to 1000
real_time_response		Accept, Reject, Refer
real_time_callback_format	The interface used to call the realtime system. Defaults to HTTP. See interface section below.	HTTP XML SOAP (see below) Note: It is also possible to supply rules to format

		your own email using a template or XSLT. Refer to T3M. In which case the above are ignored.
real_time_callback	The callback URL used by this transaction if applicable.	200 Note: callback can be either http(s)//<url> or mailto:<email address>. Or a list of either or both.
real_time_callback_options	Use either/or both the response and the customer/aggregator registered callback.	0 – Immediate realtime callback Or 1 – Customer 2 - This callback (default) 3 – Both 4 – Polled not when available or combination of the two. i.e. 02 would be realtime callback and callback on backend response.
register_consumer_watch	Register the consumer associated with this transaction for the consumer product.	Y or N (or empty) 1
real_time_sha1	A SHA1 hash of the transaction_reference with the shared secret used for hashing the card number concatenated on the end	40
merchant_attributes	A comma delimited list of name=value pairs that have no meaning to 3rd Man and will be echoed back to the merchant. Note value must not contain commas.	200
http_header_fields	Original http header records, comma separated list of Headers, e.g. `Referer`: http://abc.com/. The headers of most interest are Referer, Host, Server, User-Agent, Via and X-Forwarded-For.	200
field_delimiter	The character or string used to delimit fields where appropriate for the interface being used. Default is '|', this character must not exist within the data stream.	10
authentication_method	KYC check – refer to 3rd Man	30
proposition_date		YYYY-MM-DD
vendor_id		30

alternative_email		50
discount_amount		NNNNNNNNNNN.NN
session_id		100
account_title		10
account_name		50
account_surname		50
account_address_1		100
account_address_2		100
account_city		30
account_county_state		30
account_country	Use numeric iso code eg 826 for UK	30
account_postcode_zipcode		15
account_phone		20
account_open_date		YYYY-MM-DD
account_current_use		30
account_previous_use		30
account_remaining		30
account_numbers		50
account_loyalty		50
account_balance		NNNNNNNNNNN.NN
account_balance_currency	ISO currency code standard (ISO 4217)	(e.g. GBP, EUR, USD) (3)
bank_customer_name		50
bank_account_number		30
bank_sort_code		20
bank_name		50
bank_branch_address		100
account_profit_loss		NNNNNNNNNNN.NN
account_status		50
number_cards_registered		Numeric
exposure_amount		NNNNNNNNNNN.NN
rake		30
total_account_withdrawal		NNNNNNNNNNN.NN
number_bonus_taken		Numeric
games_most_played		50
previous_guest		10
bank_country	Use numeric iso code eg 826 for UK	30
max_purchase		NNNNNNNNNNN.NN
min_purchase		NNNNNNNNNNN.NN
avg_purchase		NNNNNNNNNNN.NN
fraud_status		30
order_status		30
alias		50
user_id		50
usermachine_id		100
usermachine_id_type		100
user_profile_1		100
user_profile_2		100
user_profile_3		100
prev_details_1		100
prev_date_1		YYYY-MM-DD HH:MM:SS
prev_amount_in1		NNNNNNNNNNN.NN
prev_amount_out1		NNNNNNNNNNN.NN
prev_amount_currency_1	ISO currency code standard (ISO 4217)	(e.g. GBP, EUR, USD) (3)
prev_details_2		100
prev_date_2		YYYY-MM-DD HH:MM:SS
prev_amount_in2		NNNNNNNNNNN.NN
prev_amount_out2		NNNNNNNNNNN.NN

prev_amount_currency_2	ISO currency code standard (ISO 4217)	(e.g. GBP, EUR, USD) (3)
prev_details_3		100
prev_date_3		YYYY-MM-DD HH:MM:SS
prev_amount_in3		NNNNNNNNNNN.NN
prev_amount_out3		NNNNNNNNNNN.NN
prev_amount_currency_3	ISO currency code standard (ISO 4217)	(e.g. GBP, EUR, USD) (3)
prev_details_4		100
prev_date_4		YYYY-MM-DD HH:MM:SS
prev_amount_in4		NNNNNNNNNNN.NN
prev_amount_out4		NNNNNNNNNNN.NN
prev_amount_currency_4	ISO currency code standard (ISO 4217)	(e.g. GBP, EUR, USD) (3)
prev_details_5		100
prev_date_5		YYYY-MM-DD HH:MM:SS
prev_amount_in5		NNNNNNNNNNN.NN
prev_amount_out5		NNNNNNNNNNN.NN
prev_amount_currency_5	ISO currency code standard (ISO 4217)	(e.g. GBP, EUR, USD) (3)
browser_language		30
derived_language		30
count_1	To be agreed with 3rd Man	Numeric
count_2	To be agreed with 3rd Man	Numeric
count_3	To be agreed with 3rd Man	Numeric
generic_1	To be agreed with 3rd Man	100
generic_2	To be agreed with 3rd Man	100
generic_3	To be agreed with 3rd Man	100
generic_4	To be agreed with 3rd Man	100
generic_5	To be agreed with 3rd Man	100
generic_6	To be agreed with 3rd Man	100
generic_7	To be agreed with 3rd Man	100
generic_8	To be agreed with 3rd Man	100
generic_9	To be agreed with 3rd Man	100
generic_10	To be agreed with 3rd Man	100
generic_11	To be agreed with 3rd Man	100
generic_12	To be agreed with 3rd Man	100
generic_13	To be agreed with 3rd Man	100
generic_14	To be agreed with 3rd Man	100
generic_15	To be agreed with 3rd Man	100
generic_16	To be agreed with 3rd Man	100
generic_17	To be agreed with 3rd Man	100
generic_18	To be agreed with 3rd Man	100
generic_19	To be agreed with 3rd Man	100
generic_20	To be agreed with 3rd Man	100
no_of_products	This indicates how many occurrences of the following fields there are. **Please note that a "product_code" value must be provided for each product line.**	2
product_code	For travel please populate with passenger name	50
product_quantity		NNNNNNNNNN
product_price		NNNNNNNNNNN.NN
product_type	For airlines please use 3 character airport code (e.g. CDG)	50
product_category	For airlines please use 3 character airport code (e.g. LGW)	50

product_description	For travel use DD MMM YYYY HH:MM (e.g. 27 Feb 2008 17:50)	50
product_risk		50
id_number		50
id_type		50
passport_number_ssn		50
date_of_birth		YYYY-MM-DD
nationality		30
loyalty_number		30
loyalty_indicator		30
fare_class		30
carrier		30
flight_number		30
route_via	If multiple then separate with , (e.g. LHR,DUB)	50

Note: If a particular field is used in one product line, then it has to also be populated in all the corresponding product lines.

The Post will look as follows:

5.1 XML

URL: https://**test**.cnprealtime.com/t3m/ThirdManXML

```
<?xml version="1.0" encoding="UTF-8"?>
<RealTime>
	<transaction_reference>ABC123</transaction_reference>
	<transaction_date_time>2009-08-24 18:07:57</transaction_date_time>
	<merchant_identifier>30000</merchant_identifier>
	<merchant_order_ref>12345</merchant_order_ref>
	<customer_ref>customer</customer_ref>
…
<Products no_of_products="2">
	<Product>
		<product_code>prod 1</product_code>
		<product_quantity>1</product_quantity>
		<product_price>10.00</product_price>
	</Product>
	<Product>
		<product_code>prod 2</product_code>
		<product_quantity>2</product_quantity>
		<product_price>5.00</product_price>
	</Product>
…
</Products>
</RealTime>
```

5.2 Key Value Pair

URL: https://**test**.cnprealtime.com/t3m/ThirdMan

```
transaction_reference=KEY789&merchant_identifier=30041&merchant_order_ref=12
345&transaction_date_time=2010-04-07
11:55:00&card_number=4444333322221111&generic_10=generic_10&no_of_products=2
&product_code=1234&product_quantity=1&product_price=10.00&product_code=5678&
product_quantity=2&product_price=5.00
```

5.3 SOAP

URL : https://**test.**cnprealtime.com/t3m/SoapWrapper

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'
xmlns:SOAP-ENC='http://schemas.xmlsoap.org/soap/encoding/'
xmlns:apachesoap='http://xml.apache.org/xml-soap' xmlns:impl='urn:xxxx'
xmlns:intf='urn:xxxx' xmlns:soapenc='http://schemas.xmlsoap.org/soap/encoding/'
xmlns:wsdl='http://schemas.xmlsoap.org/wsdl/'
xmlns:wsdlsoap='http://schemas.xmlsoap.org/wsdl/soap/'
xmlns:xsd='http://www.w3.org/2001/XMLSchema'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'>
   <SOAP-ENV:Body>
      <ns1:ThirdManXML SOAP-
ENV:encodingStyle='http://schemas.xmlsoap.org/soap/encoding/' xmlns:ns1='urn:xxxx'>
         <xml xsi:type='xsd:string'>
            <![CDATA[ <RealTime>
              <transaction_reference>ABC123</transaction_reference>
              <transaction_date_time>2009-08-24 18:07:57</transaction_date_time>
              <merchant_identifier>30000</merchant_identifier>
              <merchant_order_ref>12345</merchant_order_ref>
              <customer_ref>customer</customer_ref>
              ...
                     <Products no_of_products="1">
                           <Product>
                                 <product_code>buncan jonathan</product_code>
                                 <product_quantity>10</product_quantity>
                                 <product_price>10.00</product_price>
                           </Product>
                           ...
                     </Products>
              </RealTime>           ]]>
         </xml>
      </ns1:ThirdManXML>
   </SOAP-ENV:Body>
     </SOAP-ENV:Envelope>
```

6 Real-Time Response

The response will contain the following fields as Name=Value fields separated by '&' or whatever the designated separator is defined to be above (non-attribute/value interfaces only). If the post is made containing XML data, the response will also contain XML data.

Field Name	Description	Format (maximum length)
score	The risk score	-999 to +999 for 3rd Man
recommendation	The risk action recommendation	0,1,2 (Release, Hold, Reject) for 3rd Man
merchant_attributes	A comma delimited list of name=value pairs that have no meaning to 3rd Man and will be echoed back to the merchant. Note values must not contain commas. NOTE : This field will need decoding. It will be URL encoded for standard attribute/values lists or XML Escaped for Xml/JSon.	200

Notes:

- Lengths are all maximums and for numeric fields leading zeros do not need to be supplied
- The Real-Time interface requires all fields to be named, it is never positional. The naming convention depends on the interface used and will be described in the Interfaces section.

Response will look as follows:

6.1 XML Response

```
<?xml version="1.0" encoding="UTF-8"?>
<RealTimeResponse xmlns="T3MInitialResponse">
       <score>0</score>
       <recommendation>1</recommendation>
       <merchant_attributes>merchantattrib</merchant_attributes>
</RealTimeResponse>
```

6.2 Key Value Pair Response

```
score=0&recommendation=1&merchant_attributes=
```

The response to which would be (no white space, score of 0 and recommendation of HOLD (1) indicates that a real-time score card is not available).

Note: this normally has content type text/plain as it should be parsed programmatically.

JSON can also be used instead of XML. This is recommended as it is more compact and hence more efficient.

A SOAP endpoint is available. The SOAP wrapper wraps the above XML or the JSON equivalent. The WSDL for SOAP is available at https://cnprealtime.com/t3m/in3m.wsdl. There is also a set of unsupported client examples available https://cnprealtime.com/t3m/examples.tar.gz or https://cnprealtime.com/t3m/examples.zip.

7 Transaction Status Updates

If you wish to notify us of a status given to a transaction that we have previously scored, e.g. where we have said it is a high risk transaction but you have reviewed it and found it to be ok, or if you are going to Reject a transaction or if you have subsequently received a chargeback etc, you can send the same post as above however you only need to send the following fields:

Field Name	Description	Format / maximum length
transaction_reference	Your unique reference	50
aggregator_identifier	Unique identifier for an aggregator (for example a PSP).	15
merchant_identifier	Used by PSPs to provide merchant identifier	15
merchant_order_ref	The reference number the merchant gives to the order	250
customer_ref	The reference number the Merchant gives to their customer	50
transaction_type	The payment transaction type associated with a transaction. **In this case the transaction type must be set to Update.**	**Update**
authorisation_code	The bank authorisation code may be null if not applicable and is represented as the empty string.	30
fraud_status	If the transaction type is Update, this should contain the fraud status of the transaction	OK Rejected Chargeback Chargeback Fraud Suspicious Bank Reported Fraud Suspect Delivery Confirmed Fraud Suspect Fraud Fraud Investigated Suspect Return Found on Hot Card File
bank_response_message	User comment relating to fraud_status update	100
order_status	If the transaction type is Update, this should contain the order status of the transaction	OK Rejected Refund Return Gesture of Goodwill ExchangeOrReplacement
alt_bank_response_message	User comment relating to order_status update	100

The Post will look as follows:

7.1 XML

URL: https://**test**.cnprealtime.com/t3m/ThirdManXML

```
<?xml version="1.0" encoding="UTF-8"?>
<RealTime>
	<transaction_reference>ABC123</transaction_reference>
	<merchant_identifier>30000</merchant_identifier>
	<merchant_order_ref>12345</merchant_order_ref>
	<customer_ref>customer</customer_ref>
	<transaction_type>Update</transaction_type>
	<authorisation_code>999999</authorisation_code>
	<fraud_status>Chargeback</fraud_status>
	<order_status>Rejected</order_status>
</RealTime>
```

7.2 Key Value Pair

URL: https://**test**.cnprealtime.com/t3m/ThirdMan

```
transaction_reference=KEY789&merchant_identifier=30041&merchant_order_ref=12
345&customer_ref=customer&transaction_type=Update&authorisation_code=999999&
fraud_status=Chargeback&order_status=Rejected
```

8 Off-line Screening Responses

Offline

The real time system in its simplest form just accumulates transactions and feeds them to the batch offline system. It is an interface to the offline system that is convenient for merchants to use for submitting transactions only. Therefore to normally collect the results from the system the offline file fetch over FTP or SFTP should be used in the normal way.

Online Callback

For those merchants that wish to receive responses as they become available over an HTTP(S) protocol and do not wish to use FTP or SFTP, a response for each transaction can be posted back to your server on a transaction-by-transaction basis. This is referred to as a callback, please see this parameter above.
This will be in the form of an HTTP POST which contains the following fields as Name=Value fields separated by '&'. Alternatively, XML can be supplied.

Field Name	Description	Format (maximum length)
aggregator_identifier	Aggregator reference as per field in data feed. See above.	15
merchant_identifier	Merchant reference as per field in data feed. See above	15
merchant_order_ref	The reference number the merchant gives to the order	250
t3m_id	Third Man internal identifier. Used for later calls such as the rules string request.	
score	Score	-999 to +999
recommendation	Recommendation	0=Release 1=Hold 2=Reject 9=Under Investigation
message_digest	A SHA1 hash of the t3m_id concatenated with the shared secret used for authenticating the caller concatenated on the end	Upper case hex 40

CallBack response will look as follows:

8.1 XML Response

```
<?xml version="1.0" encoding="UTF-8"?>
<RealTimeResponse xmlns="T3MCallback">
        <merchant_identifier>30000</merchant_identifier>
        <merchant_order_ref>12345</merchant_order_ref>
        <t3m_id>333333333</t3m_id>
        <score>0</score>
        <recommendation>1</recommendation>
        <message_digest></message_digest>
</RealTimeResponse>
```

8.2 Key Value Pair Response

```
merchant_identifier=30000&merchant_order_ref=12345&t3m_id=333333333&score=0&
recommendation=1&message_digest=
```

Considerations:

Only the XML payload will vary and will need to be parsed separately.

Customer systems must acknowledge the callback response within 1 second. This will then enable 3rd Man to record a successful post.

If the initial post does fail then the system will re-try 10 times with a 2 minute pause in-between. If the system still doesn't record a successful post after the 10th try, a report will be generated to the support team to try a manual re-send at a later date.

Additional fields can be added to the existing response – please request these before starting the implementation process or as soon into the process as possible.

9 Polling for a Response

As an alternative to, or as a back up to the callback method used in the previous section, it is possible to poll for either a synchronous response or for an asynchronous response (callback).

Field Name	Description	Format (maximum length)
aggregator_identifier	Aggregator reference as per field in data feed. See above.	15
merchant_identifier	Merchant reference as per field in data feed. See above	15
merchant_order_ref	The reference number the merchant gives to the order	250
response_type	Indicates how the response should be made.	0 - synchronously 1 - asynchronously (callback)
prev_details_1	Used as an identifier for amended transactions or where order number aren't unique.	100
real_time_sha1	A SHA1 hash of the merchant_order_ref concatenated with the shared secret used for authenticating the caller concatenated on the end	Upper case hex 40

POST will look as follows:

9.1 XML

URL: https://**test**.cnprealtime.com/t3m/ResponseXML

```
<?xml version="1.0" encoding="UTF-8"?>
<RealTime>
       <merchant_identifier>30000</merchant_identifier>
       <merchant_order_ref>12345</merchant_order_ref>
       <response_type>0</response_type>
</RealTime>
```

9.2 Key Value Pair

URL: https://**test**.cnprealtime.com/t3m/ThirdMan

```
merchant_identifier=30000&merchant_order_ref=12345&response_type=0
```

Note the important response_type parameter. This is mandatory and indicates how the response should be made, a value of 0 indicates synchronously and a value of 1 indicates asynchronously (callback).

Response will look as follows:

9.3 XML Response

```
<?xml version="1.0" encoding="UTF-8"?>
<RealTimeResponse xmlns="T3MPollResponse">
       <score>0</score>
       <t3m_id>xxxxxxxxx</t3m_id>
       <recommendation>1</recommendation>
       <merchant_attributes>merchantattrib</merchant_attributes>
</RealTimeResponse>
```

9.4 Key Value Pair Response

```
score=0&t3m_id=333333333&recommendation=1&message_digest=s3cr3t
```

10 Rule String Request

For those merchants and aggregators that do not wish to store additional information but do wish to view extended information about why particular transactions have succeeded or failed, a simple interface is available. After a transaction has been scored by the offline system and a callback has been made, a second call can be made to fetch the details.

Field Name	Description	Format (maximum length)
aggregator_identifier	Aggregator reference as per field in data feed. See above.	15
merchant_identifier	Merchant reference as per field in data feed. See above	15
t3m_id	Third Man internal identifier.	

With each callback, the 3rd Man **t3m_id** is supplied. This can be used to fetch a simple string from the realtime system which gives further details about the reasons why the transaction was rejected.

POST will look as follows:

10.1 XML

URL: https://**test**.cnprealtime.com/t3m/RuleStringXML

```
<?xml version="1.0" encoding="UTF-8"?>
<RealTime>
       <merchant_identifier>30000</merchant_identifier>
       <t3m_id>333333333</t3m_id>
</RealTime>
```

10.2 Key Value Pair

URL: https://**test**.cnprealtime.com/t3m/ThirdMan

```
merchant_identifier=30000&t3m_id=333333333
```

If the request is made containing XML data, the response will be in the same format.

Field Name	Description	Format (maximum length)
t3m_id	Third Man internal identifier.	
no_of_rules	The number of rule string values that will appear in the post	Numeric 0 to 999
rule_string	Description of the rule that the transaction hit	100

Response will look as follows:

10.3 XML Response

```
<?xml version="1.0" encoding="UTF-8"?>
<RealTimeResponse xmlns="T3MRuleDetail">
       <t3m_id>333333333</t3m_id>
       <no_of_rules>5</no_of_rules >
       <rule_string>
              Surname within e-mail address (-11)|Phone no is a UK landline
              (-3)|ER Fail Billing (10)|Postal address is recognised (-2)|CV2
              Pass (-10)
       </rule_string >
</RealTimeResponse>
```

10.4 Key Value Pair Response

```
t3m_id=333333333&no_of_rules=5&rule_string=Surname within e-mail address
(-11)|Phone no is a UK landline (-3)|ER Fail Billing (10)|Postal address
is recognised (-2)|CV2 Pass (-10)
```

11 Black Box client identification

An additional call is available to upload data to the system to identify client machines; these machines can then be later checked against a list of known fraudulent client machines. The data to identify the fraudulent client is contained within a 'blackbox' field. This makes the call very simple.

Field Name	Description	Format (maximum length)
aggregator_identifier	Aggregator reference as per field in data feed. See above.	15
merchant_identifier	Merchant reference as per field in data feed. See above	15
merchant_order_ref	The reference number the merchant gives to the order	250
blackbox		2000
real_time_sha1	A SHA1 hash of the merchant_order_ref concatenated with the shared secret used for authenticating the caller concatenated on the end	Upper case hex 40

POST will look as follows:

11.1 XML

URL: https://**test**.cnprealtime.com/t3m/BlackBoxXML

```
<?xml version="1.0" encoding="UTF-8"?>
<RealTime>
       <merchant_identifier>30000</merchant_identifier>
       <merchant_order_ref>12345</merchant_order_ref>
       <blackbox>ABC123</blackbox>
</RealTime>
```

11.2 Key Value Pair

URL: https://**test**.cnprealtime.com/t3m/ThirdMan

```
merchant_identifier=30000&merchant_order_ref=12345&black_box=ABC123
```

Response will look as follows:

11.3 XML Response

```
<?xml version="1.0" encoding="UTF-8"?>
<RealTimeResponse xmlns="T3MBlackBox">
       <response>Okay</response>
</RealTimeResponse>
```

11.4 Key Value Pair Response

```
response=Okay
```

12 Error Handling

Error Handling and Debugging

The real time interface like any networked system can suffer from a number of problems.

These are:

Connectivity or failure to respond errors.

These are normally indicated by the return of an HTTP 500 error message. These happen when there are networking problems such as system inaccessible, downtime or network/firewall misconfigured. It also may indicate that the realtime server failed to return any response whatsoever, this should not happen as it should tolerate almost any format of data sent to it and at least give a response. If an unanticipated format of data is sent to it, an HTTP 502 error may be returned. If this happens, email T3M with the exact message sent and the date/time it was sent. A 502 error normally indicates some thing was very wrong with the request sent to us so please check the documentation.

HTTP > 200. Generally this indicates that the file was not found or similar. Your request to us is not recognised, please check there are no typos.

A message is returned but the merchant-attributes field contains an error message.

The possible messages are:

- NoAggregatorOrMerchant - Either of (or both of) the merchant_identifier and/or the aggregator_identifier must be supplied.

- UnknownCustomer - The merchant or aggregator is not registered on our system.

- NoMerchantOrderRef - No merchant order reference was supplied, when required by the request, this is mandatory.

- BadSHA1 - The SHA1 hash does not match - is your password correct?

- NoT3mId - No t3m_id was supplied, when required by the request this is mandatory.

- NoRuleStrings - For a Rule string request there is no data.

- NoRecordFound - For a response poll request, there is no data. The backend scoring system has not yet completed and results are not yet available.

- NoResponseType - the response type field is missing.

- Error : <Some text describing the error in more detail> - This is a serious error and should not normally happen, it may be due to incorrect data supplied as the system is data driven or may be something wrong with the system. Please check the request sent to us and if that is definitely correct then please report the error, but before you do so try once more. It may well be a transient error.

No callback is received although you did not receive an error as detailed above.

This could be due to a misspelling of a parameter.

Note parameters are case sensitive and always in lower case.

This could also be as a result of additional pipes appearing in the data – the data collected from the real-time system is stored in pipe delimited files, additional pipes in the data do not cause issues at the front end but will prevent the order from being fully processed. Additionally if essential data is missing, this will prevent the orders from being imported into the screening engine –see below. In all these scenarios, NO error will be displayed in the initial response.

If the callback hasn't been received, try polling for a response (see below).

If the order has not been processed, the t3m_id field in the response will state NoRecordFound

13 Security

All transactions must be sent using at least 128 bit SSL.

The merchant will be verified as having supplied the information by checking that the realtime SHA1 supplied matches one calculated by the system.

The realtime response is synchronous so requires no verification or any identifying fields. However the callback, if used, does require identifying fields and a SHA1 hash is required to verify that it is actually 3rd Man that made the callback. Note that the shared secret would normally be associated with the aggregator when there is an aggregator or the merchant otherwise. As the callback is just a URL and contains no particularly sensitive data, the merchant does not need a secure server. However, in the case of an aggregator where the callback is directed to the merchant directly not via the aggregator, the shared secret of the aggregator will not normally be known by the merchant. Under these circumstances, the aggregator should make arrangements with the merchant to use an additional merchant attribute to verify the callback source.

14 Interfaces Available

A number of interfaces will be available for merchants or aggregators who wish to supply data to the realtime interface. All interfaces will supply the same data and use the same parameters and parameter names. Different interfaces will have different tradeoffs in terms of efficiency / simplicity and speed of development. It is for the merchant or aggregator to decide on the appropriate interface that best meets their requirements.

As a general guide, only supply data that is actually required (unused blank fields are a bandwidth and processing overhead). XML based interfaces are generally expensive in terms of overhead, SOAP in particular see below.

The default interface takes normal http POST (Content-Type: text/html) (GETs should not be used, but should work if data volumes are small) data and replies / callbacks are made in a similar way.

XML data can be supplied either as the only contents of the POST request or as the contents of a single text field called xml or SOAP (any other fields will be ignored completely). The realtime response will be returned in the same format but the callback response will be sent in the default format unless requested otherwise.

SOAP is also available as a wrapper for XML or JSON

15 Technical Considerations

- One transaction per POST
- The phrase "Real-Time" has two components, there is real-time collection of data which always happens and there is the application of rules. Rules may be applied either online or offline or both. Rules are not automatically included unless subscribed to
- Online scoring is available in as a real-time score. A real-time score is defined as a score that is processed synchronously with the particular POST to the system. The number and complexity of the rules processed will add processing time. In particular lookups against black and white lists and velocities take time. The more processed, the longer the time taken. For large rules sets extensive processing should be done offline. A two stage process is recommended, check for the most common problems in a relatively small rule set in real-time and defer more extensive checks to offline.
  Offline scoring is available subject to the 30 minute SLA
- The most common cause of POST failure is misconfiguration at the client end. If you are 'loosing' transactions please check firstly that they are in the Gatekeeper system. If they are, then they arrived, in which case was the response misparsed, lost, blocked by a firewall, timed out too aggressively or some other mishap?
  If there are no transactions in Gatekeeper, then they did not arrive, in which case there is either a problem with the client end or connectivity to T3M or misunderstand or mis-implementation of the API. It is very rare for T3M to not process transactions that arrive and are well formed, if we do not and there is a problem then we will keep you informed.
  If having considered all of the above in detail and there still appears to be a problem then please provide support with a explicit example of a failed transaction, the details of when it was sent and the text of the transaction according to the API you are using (Basic form POST as attribute/values, XML or SOAP).
- Customers systems need to be configured to a suitable timeout limit based on the expectations on the 3rd Mans systems, if an extensive rules base is in place then the timeout should be configured higher than if no rules base is used.
- Async or Sync real-time responses are available if systems can or can't handle the real-time response sync mechanism.

16 Testing

A test form is available at:

https://cnprealtime.com/realtimetest.html

It is also possible to test from the command line using wget. A Windows wget can be obtained from:

http://users.ugent.be/~bpuype/wget/

To test service using 'raw' XML:

*wget -O **out** --no-check-certificate --post-file=**xml.txt** http://test.cnprealtime.com/t3m/ThirdManXML*

To test service using simple attribute-value pairs (do not forget to URL

encode the field values) :

*wget -O **out** --no-check-certificate --post-file=**av.txt** http://test.cnprealtime.com/t3m/ThirdMan*

In both cases check the “out” file, it will contain the response. Add all the parameters you need.

Code your programmatic call to simulate the above.

Code your callback as per the spec. alternative scripts are supplied called avcallback.txt and xmlcallback.txt for testing callbacks.

There is a test cgi called 'echo' supplied which just echoes 'raw' i.e. without decoding parameters.

You should of course develop and supply your own callback

File contents are as follows. Please note that white space is not required for XML files and for attribute value files Base64 encoding must be done. Only parameters that actually have values need be supplied.

av.txt

transaction_reference=tran%20ref&aggregator_identifier=aggregator%20id&merchant_identifier=merchant%20id&merchant_order_ref=merchant%20order%20ref&no_of_products=2&product_code=prod%201&product_quantity=1&product_price=10.00&product_code=prod%202&product_quantity=2&product_price=5.00&merchant_attributes=att1%3Done%2Catt2%3Dtwo

avcallback.txt

transaction_reference=tran%20ref&aggregator_identifier=aggregator%20id&merchant_identifier=merchant%20id&merchant_order_ref=merchant%20order%20ref&no_of_products=2&product_code=prod%201&product_quantity=1&product_price=10.00&product_code=prod%202&product_quantity=2&product_price=5.00

xml.txt

```
<?xml version="1.0" encoding="UTF-8"?>
<RealTime>
	<transaction_reference>tran ref</transaction_reference>
	<aggregator_identifier>aggregator id</aggregator_identifier>
	<merchant_identifier>merchant id</merchant_identifier>
	<merchant_order_ref>merchant order ref</merchant_order_ref>
	<Products no_of_products="2">
		<Product>
			<product_code>prod 1</product_code>
			<product_quantity>1</product_quantity>
			<product_price>10.00</product_price>
		</Product>
		<Product>
			<product_code>prod 2</product_code>
			<product_quantity>2</product_quantity>
			<product_price>5.00</product_price>
		</Product>
	</Products>
	<merchant_attributes>att1=one,att2=two</merchant_attributes>
</RealTime>
```

xmlcallback.txt

```
<?xml version="1.0" encoding="UTF-8"?>
<RealTime>
	<transaction_reference>tran ref</transaction_reference>
	<aggregator_identifier>aggregator id</aggregator_identifier>
	<merchant_identifier>merchant id</merchant_identifier>
	<merchant_order_ref>merchant order ref</merchant_order_ref>
	<Products no_of_products="2">
		<Product>
			<product_code>prod 1</product_code>
			<product_quantity>1</product_quantity>
			<product_price>10.00</product_price>
		</Product>
		<Product>
			<product_code>prod 2</product_code>
			<product_quantity>2</product_quantity>
			<product_price>5.00</product_price>
		</Product>
	</Products>
	<real_time_response_format>XML</real_time_response_format>
	<real_time_callback>http://www.intelligent-net.co.uk/inet/cgi-
bin/echo</real_time_callback>
	<merchant_attributes>test_callback=true</merchant_attributes>
</RealTime>
```

SOAP Call

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'
xmlns:SOAP-ENC='http://schemas.xmlsoap.org/soap/encoding/'
xmlns:apachesoap='http://xml.apache.org/xml-soap' xmlns:impl='urn:xxxx'
xmlns:intf='urn:xxxx' xmlns:soapenc='http://schemas.xmlsoap.org/soap/encoding/'
xmlns:wsdl='http://schemas.xmlsoap.org/wsdl/'
xmlns:wsdlsoap='http://schemas.xmlsoap.org/wsdl/soap/'
xmlns:xsd='http://www.w3.org/2001/XMLSchema'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'>
   <SOAP-ENV:Body>
      <ns1:ThirdManXML SOAP-
ENV:encodingStyle='http://schemas.xmlsoap.org/soap/encoding/' xmlns:ns1='urn:xxxx'>
         <xml xsi:type='xsd:string'>
            <![CDATA[ <RealTime>
              <transaction_reference>test123</transaction_reference>
              <transaction_date_time>2009-08-24 18:07:57</transaction_date_time>
              <merchant_identifier>30000</merchant_identifier>
              <merchant_order_ref>12345</merchant_order_ref>
              <ip_address>81.93.226.35</ip_address>
              <card_number>4444333322221111</card_number>
              <card_bin>444433</card_bin>
              <merchant_attributes>att1=one,att2=two</merchant_attributes>
              <real_time_callback_format>XML</real_time_callback_format>
              <real_time_response_format>HTML</real_time_response_format>
              <customer_email_address>fred@hotmail.com</customer_email_address>
              <threed_secure_eci_indicator>1</threed_secure_eci_indicator>
                     <Products no_of_products="1">
                     <Product>
                            <product_code>prod_code</product_code>
                            <product_quantity>10</product_quantity>
                            <product_price>10.00</product_price>
                     </Product>
                     </Products>
              </RealTime>           ]]>
         </xml>
      </ns1:ThirdManXML>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```