Scam Transaction Monitoring Guide

Powered by Featurespace

Version: 1.0
26 March 2025
Publication number: STMP-1.0-3/26/2025

For the latest technical documentation, see the Documentation Portal.

Thredd, Kingsbourne House, 229-231 High Holborn, London, WC1V 7DA
Support Email: occ@thredd.com
Support Phone: +44 (0) 203 740 9682

Copyright

To the maximum extent permitted by law, Thredd Ltd shall not be liable to any person or organisation, in any manner whatsoever from the use, construction or interpretation of, or the reliance upon, all or any of the information or materials contained in this guide.

The information in these materials is subject to change without notice and Thredd Ltd. assumes no responsibility for any errors.

About This Document

This document describes how to use the Scam Transaction Monitoring to:

- Setting up the schema to use the Scam Transaction Monitoring
- Understanding the API endpoints used to communicate potential scams with Thredd
- Managing scams in the Fraud Transaction Monitoring Portal

Note: The Thredd Scam Transaction Monitoring is based on the Featurespace ARIC Risk Hub product, which has been customised for use by Thredd customers. In this guide, we refer to the Featurespace ARIC Risk Hub User Interface as the Scam Transaction Monitoring or 'the portal'.

Target Audience

This document is intended for Thredd clients (Program Managers) who are using the Scam Transaction Monitoring.

What’s Changed?

If you want to find out what's changed since the previous release, see the Document History page.

Related Documents

Refer to the table below for other documents which should be used in conjunction with this guide.

Document	Description
Fraud AMDL Rules Configuration Guide	Explains how to configure the rules used by the ThreddFraud Transaction Monitoring System.
Fraud Transaction Monitoring System Access Configuration Guide	Describes how to set up user access and user access role available.
Fraud Transaction Monitoring Portal	Describes how to use the Fraud Transaction Monitoring Portal.

Other Guides

Refer to the table below for other relevant documents.

Document	Description
Payments Dispute Management Guide	Describes how to manage chargebacks and the disputes management process using Thredd.
Smart Client Guide	Describes how to use the Thredd Smart Client to manage your account.
Thredd Portal	Describes how to use the Thredd Portal, Thredd's new web application for managing your cards and transactions on the Thredd Platform.

Tip: For the latest technical documentation, see the Documentation Portal.

1 Introduction to Scam Transaction Monitoring

Scam Transaction Monitoring is powered by a Machine-Learning model developed by FeatureSpace that provides clients with a risk score against each payment (inbound or outbound), showing how likely it is to be originating from a scam.

Scam Transaction Monitoring enables businesses to assess potential inbound and outbound scams in real-time. For potential outbound scams, Scam Transaction Monitoring checks if a payer is making a payment under the false pretence, for example if a payer got scammed. For potential inbound scams, Scam Transaction Monitoring checks if the funds are coming from a victim of a scam into the scammer's account.

What is a Scam?

A scam is an authorised push payment fraud where the account holder authorised payment to be sent to a "scammer"/ "fraudster" (For example, romance scam, investment scam, purchase scam). Scams differ from fraud, which is where an account holder did not authorise the payment themselves (For example, in cases of account takeover, stolen account details).

How it Works

Scam Transaction Monitoring is intended to be used to assess risk at the level of an individual payment. It is recommended that Scam Transaction Monitoring is positioned as the last step before the payer's Funding Institution (FI) sends a payment through to the payee's Funding Institution. Any other checks done by the FI (such as policy checks) should be done by the FI prior to sending the payment message to Scam Transaction Monitoring, with only approved payments being sent to Scam Transaction Monitoring.

Note: It is expected that Scam Transaction Monitoring will receive approved payments (such as payments that pass a balance check). However, for certain use cases, an FI may wish to send in Failed or Cancelled payments for risk-scoring and further analysis, or to provide an update on the payment after it was made.

The figure below describes the suggested outbound scam transaction monitoring flow.

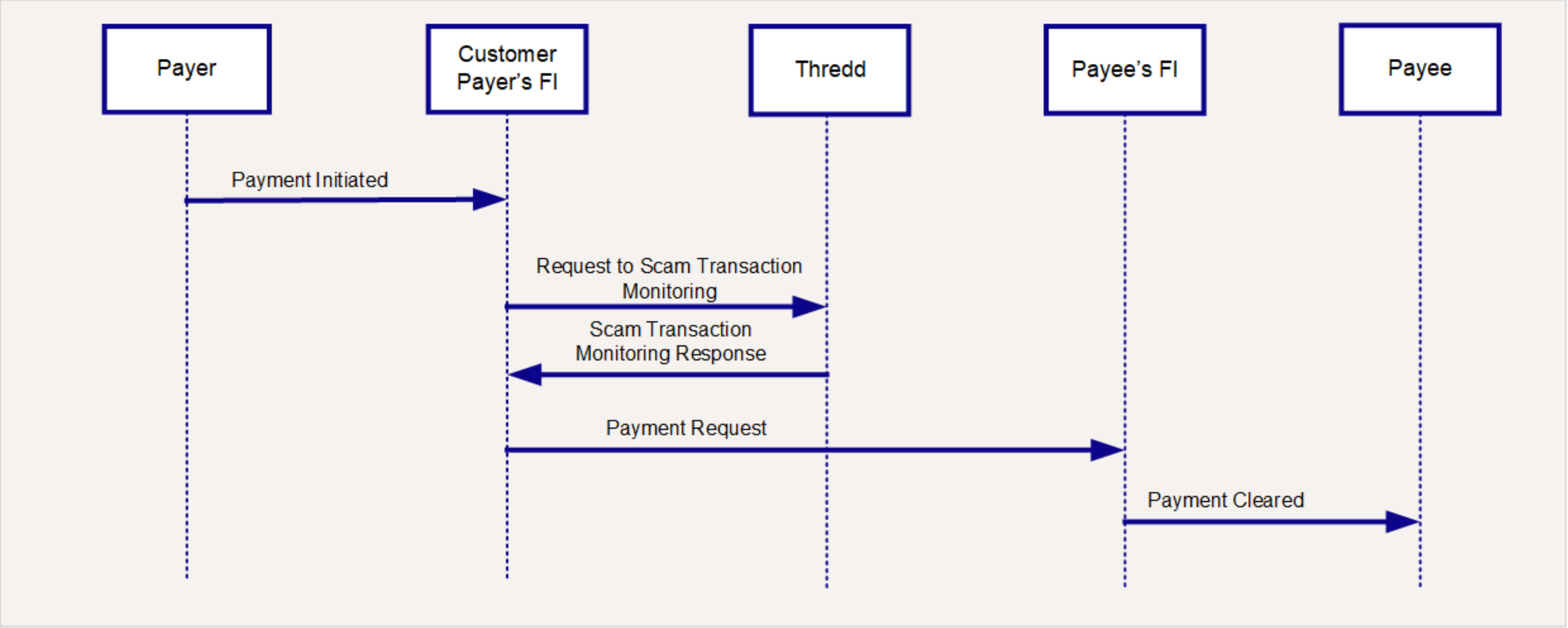


Figure 1: Outbound Scam Transaction Monitoring Flow

The figure below describes the suggested inbound scam transaction monitoring flow.

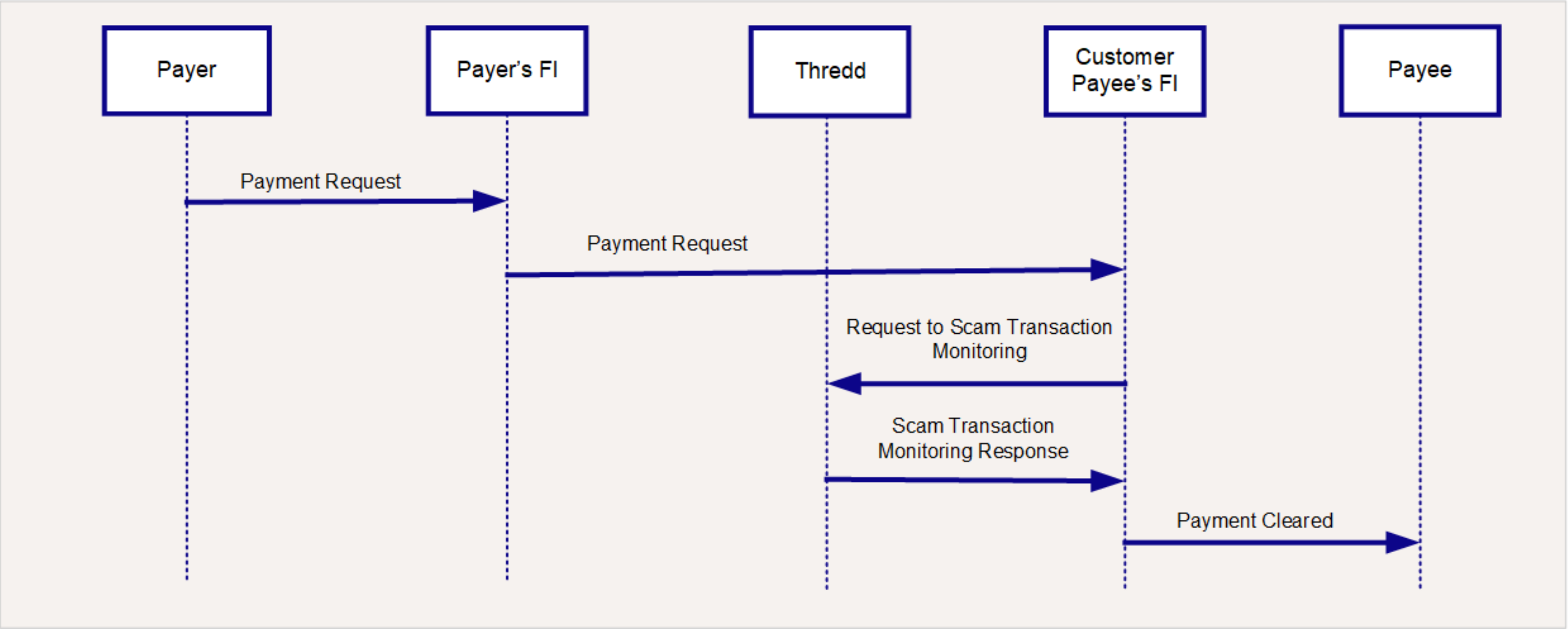


Figure 2: Inbound Scam Transaction Monitoring Flow

Note: Historical payment data should not be uploaded to Scam Transaction Monitoring as it does not improve Scam Transaction Monitoring performance and can affect system performance.

To support a migration from another Transaction Monitoring tool, we recommend that Scam Transaction Monitoring is run silently for a period of time. For more information, contact your Implementation Manager.

Interpreting the Score

The real-time response from the Scam Transaction Monitoring API includes a risk score in the range of 0.0 to 1.0. For example, a score of 0.2 represents a lower scam risk and 0.8 represents a higher scam risk. Scam Transaction Monitoring is designed to offer an Exponential score calibration by default. The Exponential score calibration approximately halves the decline rate every 0.071 increase in the score, at least for decline rates smaller than 0.1%.

Examples of thresholds are provided in the table below. You can choose the score threshold, how to use it in your existing analytics, and associated downstream actions that are appropriate.

For example, if payment is declined at a score threshold of 0.706, the approximate expected decline rate across all payments for the FI that are risk scored by Scam Transaction Monitoring is 0.10% or 1 in 1000 payments. The model is regularly calibrated to reflect the expected decline rates as well as the expected value detection rates.

Decline Rate (basis points)	Score Threshold
1	0.900
5	0.771
10	0.706
25	0.615
50	0.545
100	0.474

Setting up Scam Transaction Monitoring

Before you can use Scam Transaction Monitoring in the Fraud Transaction Monitoring Portal, you must first complete the following:

- Set up client authentication to connect to Thredd. For more information, see the Key Concepts guide.
- Provide details to Thredd in the pre-agreed schema format in order to receive the risk score and related output action tags. For more information, see Scam Transaction Monitoring Schema.
- Consider how your business will process the real-time response from the Scam Transaction Monitoring API and how the score will be used to augment your decisioning. This includes automated or manual treatments you employ depending on your risk appetite thresholds, such as:
  - Blocking or holding the payment
  - Freezing the account
  - Manual review in a case management system
  - Customer contact

Feedback Requirements for Scam Transaction Monitoring

For Scam Transaction Monitoring to remain performant and provide the best results, we recommend that at least 90% of confirmations of fraud or scam (For example, fraud labels) are returned to Scam Transaction Monitoring within 30-40 days.

There are two ways to send feedback:

- Using the paymentTransactionReturn API endpoint
- Marking an event as a Risk in the Fraud Transaction Monitoring Portal. If there is an alert generated, then mark it as Risk, including information if it's a Fraud or Scam. If there is no alert generated, click on the Eye icon for payment event, create an Alert and then mark it as Risk, including information if it's a Fraud or Scam.

It is a requirement to send feedback using one of these channels whenever a scam or fraud occurs. However, it is important that the feedback is not shared using both channels, as double counting can occur and this can impair the system's performance.

Scam Transaction Monitoring only expects feedback in cases of fraud or scam. There is no need to send feedback if the payment was genuine.

Scam Transaction Monitoring expects the feedback to be sent even if the payment was declined or cancelled after it has been risk-scored.

2 Scam Transaction Monitoring Schema

The Scam Transaction Monitoring data schema is a collection of definitions that establish the format of payment events and how these will be linked to their associated entities so the platform can build behavioural profiles. The schema defines the types of payment events the system can process, as well as the attributes contained within them. There are three different types of payment events that can be sent into Scam Transaction Monitoring:

- Real-time payments (paymentRT): Real-time payments receive a scam score in the synchronous Payments response
- Non-real-time payments (paymentNRT): Used for non-real-time payment decisioning and will not receive a scam score in the synchronous Payments response. However, it can be used for information purposes in the Fraud Transaction Monitoring Portal and/or generate silent alerts for further review. These events will also be risk scored if the required parameters are provided.
- Scam and fraud labels (paymentTransactionReturn): Informational events confirming that a scam or fraud has taken place

IMPORTANT: Each of these types have a corresponding REST API endpoint. You must ensure the data you share with Thredd matches the values in the endpoints.

It is expected that all end-customer payments are sent through as paymentRT or paymentNRT events for the Scam Transaction Monitoring model to be performant.

All payments share common attributes that are required for processing, as well as mandatory, recommended, and non-mandatory attributes specific to each transaction type. All attributes in the data are expected to be of a certain data type such as string, number, integer, Boolean, date-time, or an array. Additional validation may also be applied to restrict the permitted attribute to a specific set of expected values. If it is more efficient to design a bundle of data types that can be re-used as a JSON object instead of defining each multiple times across transactions, a derived type is defined (for example, type money). For more information on derived types, see Appendix A: Derived Types.

Parties Involved in a Payment

Several parties can be involved in the lifecycle of a payment. The parties that are most important to Scam Transaction Monitoring are the accounts where payments are sent from and to, the Funding Institutions (FIs), and the branches of the payer and payee.

The following table shows how these parties are represented in the Scam Transaction Monitoring schema. Account is the account belonging to your customer (whether the payer or payee), while counterparty relates to the account on the other side of the payment.

Account Fields	Counterparty Fields	Description
accountId	counterpartyId	A unique identifier for the accounts that the payment is sent from and to.
accountAgentId	counterpartyAgentId	A globally unique identifier for the FIs at which the accounts are held, usually the Bank Identification Code (BIC).
accountBranchId	counterpartyBranchId	A unique identifier for the FIs (at a more granular level than the *AgentId fields) at which the accounts are held. This is usually a sort code or routing number.

These fields are populated depending on the direction of the transaction. The account fields always represent the customer that belongs to the FI scoring data using Scam Transaction Monitoring.

Steering Fields

In the different messages there are several key attributes that describe the scenario that the message is representing. These attributes are known as steering fields and are key to getting optimum analytical performance from Scam Transaction Monitoring. These steering fields are shown in the following table:

Event	Field	Description
paymentRT	direction	Reflects the direction of the payment from the perspective of the institution. This is either inbound when account holder is the payee, or outbound when the account holder is the payer.
paymenttransactionReturn	originalPaymentDirection	Must match the direction of the original payment message.
	confirmedRisk	Should always be set to **true** where this is sent.
	returnType	Indicates whether this is reporting a confirmed fraud event of scam event. This is either fraud or scam. See What is a Scam? for more information on how these differ.

3 Integrate with Scam Transaction Monitoring API

To send transactions to the Scam Transaction Monitoringt API, a Thredd representative provides the client certificates and connection details as required.

There are three endpoints supported by Scam Transaction Monitoring.

Name	Description	URL
paymentRT	Send real-time payments to receive a scam score in the synchronous payments response.	{{baseURL}}/v1/risk/payment-rt
paymentNRT	Used for non-real-time payment decisioning and information only.	{{baseURL}}/v1/risk/payment-nrt
paymentTransactionReturn	Informational events confirming that a scam or fraud has taken place	{{baseURL}}/v1/risk/payment-transaction-return

Note: For each API request, the average transaction event size should be no more than 2kb, with a maximum event size of 10kb for any event type.

All transactions sent to the API must be in the form of an HTTP POST request. The message body must contain JSON-formatted data describing the event. The message header for all events must contain a content-type header with the appropriate MIME type. For example, JSON event data must contain content-type:application/json.

Scam Transaction Monitoring is available over the Internet as well as via existing IPsec tunnels (VPN). To access the service via Internet, clients have to use enhanced security mechanism supported by Thredd based on private_key_jwt and mTLS. If you are connecting either through the Internet or VPN, it is recommended to access Scam Transaction Monitoring endpoints using hostnames, following DNS resolution and time to live(TTL) settings.

To pass validation, the event data must conform to the following criteria as defined by Scam Transaction Monitoring's data schema:

- All mandatory and recommended data fields are present.
- There are no field names present that are not defined in the schema.
- All field values are of the correct data type and match any additional validation requirements (for example, date-time fields match the ISO 8601 format requirements).
- String types can have a maximum of 255 characters
- Empty fields are stripped from the transaction data and are not sent to Scam Transaction Monitoring.

The following is an example body of a JSON-formatted POST request. The eventType field indicates this is for the paymentRT endpoint.

```
{
 "accountBranchId": "001122",
 "accountId": "00112212345678",
 "amount": {
   "value": 1000,
   "currency": "GBP"
 },
 "channel": "online",
 "counterpartyBranchId": "223344",
 "counterpartyId": "22334412345678",
 "customerId": "CUST123456789",
 "direction": "outbound",
"localDateTime": "2024-09-02T06:34:56",
"eventTime": "2024-09-02T07:34:56-01:00",
 "msgStatus": "New",
 "paymentClearingSpeed": "LessThanTwoHours",
 "paymentMethod": "Faster Payment",
 "programManagerCode": "THR",
 "transactionId": "31c0a76f-7351-43c2-876c-fc1323c7d042"
 }
```

Note: Event time is in Coordinated Universal Time (UTC).

When the Scam Transaction Monitoring API receives the transaction, it enriches and validates the message before processing. If successful, a 200 response is returned and the body of the response includes the risk score and transaction identifiers.

An example response from the PaymentRT can be found below.

Note: The PaymentNRT and paymentTransactionReturn endpoints has no response body and only returns a 204 No Content response.

```
{
  "jsonVersion": 4,
  "processorId": "proc",
  "originatingEvent": {
    "eventId": "540c5bf1-bd81-424d-83d2-0d5d6c3c4d05",
    "decorationId": [
      "payments"
    ],
    "accountBranchId": "001122",
    "accountId": "00112212345678",
    "amount": {
      "baseCurrency": "GBP",
      "baseValue": 1000,
      "currency": "GBP",
      "value": 1000
    },
    "channel": "online",
    "counterpartyBranchId": "223344",
    "counterpartyId": "22334412345678",
    "eventTime": "2024-10-16T09:34:56.000",
    "customerId": "CUST123456789",
    "direction": "outbound",
    "localDateTime": "2024-10-16T08:34:56.000",
    "msgStatus": "New",
    "paymentClearingSpeed": "LessThanTwoHours",
    "paymentMethod": "Faster Payment",
    "transactionId": "THR31c0a76f-7351-43c2-876c-fc1323c7d042",
    "vitalFourScreeningRequired": false,
    "toScamDetect": true,
    "tenantId": "THR",
    "eventType": "paymentRT",
    "schemaVersion": 1,
    "accountEntityId": "00112212345678",
    "customerEntityId": "THR-CUST123456789",
    "counterpartyEntityId": "22334412345678",
    "_metadata": {
      "systemEventId": "212234-0X3oPVf9",
      "receivedTime": "2024-10-16T09:55:06Z",
      "eventTime": "2024-10-16T09:34:56Z",
      "eventId": "540c5bf1-bd81-424d-83d2-0d5d6c3c4d05",
      "eventType": "paymentRT",
      "tenantId": [
        "THR"
      ],
      "execution": {
        "expectResponse": true
      },
      "searchable": {
        "channel": "online",
        "amount": {
          "baseValue": 1000,
          "value": 1000
        },
        "eventType": "paymentRT",
        "counterpartyEntityId": "22334412345678",
        "counterpartyId": "22334412345678",
        "paymentMethod": "Faster Payment",
        "accountId": "00112212345678",
        "accountEntityId": "00112212345678",
        "eventTime": "2024-10-16T09:34:56.000",
        "customerEntityId": "THR-CUST123456789",
        "transactionId": "THR31c0a76f-7351-43c2-876c-fc1323c7d042",
        "customerId": "CUST123456789",
        "eventId": "540c5bf1-bd81-424d-83d2-0d5d6c3c4d05",
        "accountBranchId": "001122",
        "msgStatus": "New"
      },
      "financialValue": 1000,
      "scamDetect": {
        "model": {
          "score": 0.05255
        },
        "_metadata": {
          "httpcode": 200,
          "latency": 123,
          "headers": {
            "Date": "Wed, 16 Oct 2024 09:55:07 GMT",
            "Content-Type": "application/json;charset=utf-8",
            "Content-Length": "1202",
            "Connection": "keep-alive",
            "Access-Control-Allow-Origin": "*",
            "Access-Control-Allow-Headers": "Origin, X-Requested-With, Content-Type, Content-Length, Accept",
            "Server": "webserver",
            "X-XSS-Protection": "1; mode=block",
            "X-Content-Type-Options": "nosniff",
            "Cache-Control": "no-store, no-cache, must-revalidate, proxy-revalidate, max-age=0"
          },
          "outputTime": "2024-10-16T09:55:07.109Z"
        }
      },
      "entities": {
        "ACCOUNT": [
          "00112212345678"
        ],
        "COUNTERPARTY": [
          "22334412345678"
        ],
        "CUSTOMER": [
          "THR-CUST123456789"
        ]
      }
    }
  },
  "outputTime": "2024-10-16T09:55:07.128Z",
  "entities": [
    {
      "entityType": "COUNTERPARTY",
      "entityId": "22334412345678",
      "tenantId": "THR",
      "overallScore": {
        "overallScore": null
      },
      "models": [],
      "outputTags": [],
      "riskStatus": "no-risk",
      "configGroups": []
    },
    {
      "entityType": "CUSTOMER",
      "entityId": "THR-CUST123456789",
      "tenantId": "THR",
      "overallScore": {
        "overallScore": null
      },
      "models": [],
      "outputTags": [],
      "riskStatus": "no-risk",
      "configGroups": []
    },
    {
      "entityType": "ACCOUNT",
      "entityId": "00112212345678",
      "tenantId": "THR",
      "overallScore": {
        "overallScore": null
      },
      "models": [],
      "outputTags": [],
      "riskStatus": "no-risk",
      "configGroups": []
    }
  ],
  "versions": {
    "modelGraph": 2,
    "configGroups": [
      {
        "type": "analytical",
        "id": "Thredd_ACG",
        "version": "0"
      },
      {
        "type": "global",
        "version": "0"
      }
    ]
  },
  "statusCode": "success"
}
```

Other HTTP response codes are returned in the event of an error or other issues. The following table describes the possible response codes and what they mean:

HTTP Response Code	Meaning
200 OK	Real-time response (response data in body).
204 No Contect	Real-time response (no data in body).
400 Bad Request	Transaction failed schema validation or no certificate was presented.
403 Forbidden	Not authorised to access that endpoint. This can be due to a missing scope or role added by Thredd Identity Provider.
404 Not Found	Invalid URI.
405 Method Not Allowed	Invalid HTTP method.
500 Server Error	Error occurred during real-time processing. Details of the error are returned in the response body.
502 Bad Gateway	API is offline or otherwise unable to receive the event.
504 Gateway Timeout	Timed out waiting for a response from the API.

4 PaymentRT

This page describes each of the fields that can be included in the request for the PaymentRT endpoint.

The following table describes each of the fields available in the request for the PaymentRT endpoint. The Options column shows the available options that are validated by Scam Transaction Monitoring for that field where applicable. Only one of these options can be selected.

> **Note:** To ensure the system performs correctly, a PaymentRT request must include accountId, counterpartyId, customerId, and then a maximum of two of the following id fields:
> - cardId
> - deviceId
> - initiatingPartyId
> - merchantId

Attribute	Type	Description	Mandatory	Options	Example Values
accountAddress	address (for more information, see Appendix A: Derived Types)	Postal address the account specified in accountId is registered to.	N	N/A	
accountAgentId	string	Unique identifier of the financial institution (at the global level) providing the account in accountId. accountAgentId will relate to accountBranchId, but may be less granular. For example, if accountBranchId is passed as a sort code, this code will not unqiuely identify a financial institution. accountAgentId should identify the financial institution as a whole. For example, in the UK you would have an id for Barclays, and another id for NatWest. **Note:** Providing this data can improve Scam Transaction Monitoring Machine Learning model performance.	N	N/A	
accountAgentName	string	Name of the financial institution in accountAgentId.	N	N/A	
accountBalanceBefore	money (for more information, see Appendix A: Derived Types)	The balance before the transaction of the account in accountId. **Note:** Providing this data can improve Scam Transaction Monitoring Machine Learning model performance.	N	N/A	
accountBranchAddress	address (for more information, see Appendix A:	Postal address of the branch at which the account is held.	N	N/A	

Attribute	Type	Description	Mandatory	Options	Example Values
	Derived Types)				
accountBranchId	string	Unique identifier for the branch the account is associated with at a more granular level than the accountAgentId field. In the UK this would typically be populated by a sort code, in the US this would typically be populated by a routing number. For examples on how the accountBranchId is defined for a region, see Appendix B: Unique Identification Definitions.	Y	N/A	
accountFlag	array	Any flags on the account that aren't covered by other attributes. This is an array that can include free text values that might indicate VIP accounts, or accounts that are suspected to have been compromised.	N	N/A	VIP
accountId	string	Unique identification of the account involved in the event. If this is an outbound transaction, accountId will be the debtor's account. If this is an inbound transaction, accountId will be the creditor's account. accountId must be consistent across different event types, and it must also be consistent over time. For clarity, this is not an internally generated ID. accountId is a unique account number which is recognised by the payment scheme. For example, in the UK, if the sort code is 112233, and the account number is 12345678, then accountId is 11223312345678. For examples on how the accountId is defined for a region, see Appendix B: Unique Identification Definitions. In order to ensure optimal model performance, the accountId must be consistent over time. For example, it shouldn't be represented with IBAN as well as account+sortcode. Customers should check that the following relationships exist in their data when mapping to accountId: • For personal banking, one accountId should be associated with a small number of customerIds For personal banking, one customerId should be associated with a small number of accountIds • One accountId should belong to a single accountAgentId. For example, in a card transaction we expect accountAgentId to be an issuing financial institution, and one accountId should never be seen with two different issuers.	Y	N/A	11223312345678
accountIdFormat	string	Account numbers can take multiple formats, this is a free text field that describes the format of the accountId.	N	• IBAN • UK account • US account	IBAN
accou	date	The date the account was opened.	N	N/A	

Attribute	Type	Description	Mandatory	Options	Example Values
ntOpenDate		**Note:** Providing this data can improve Scam Transaction Monitoring Machine Learning model performance.			
accountSubType	string	The sub type of the accountType. For example, accountType could be a current account and the subType would determine the specific product of the account type offered by the FI.	N	N/A	reward
accountType	string	The type of account. **Note:** Providing this data can improve Scam Transaction Monitoring Machine Learning model performance.	N	• Personal • Business • Current • Savings	Personal
amount	money (for more information, see Appendix A: Derived Types)	Actual amount of the transaction in the transaction currency.	Y	N/A	
approverId	array	An array containing identifiers of any approvers required to approve this action (such as making a payment or adding a payee). approverId is mainly used for commercial accounts, and identifies who approved the payment. **Note:** Do not populate this attribute if additional approval is not needed.	N	N/A	
batchPaymentDetails	batchPaymentDetails (for more information, see Appendix A: Derived Types)	Extra information that may be relevant to processing ACH transactions or other batch-type corporate payments. **Note:** Each payment in a batch payment is expected to have its own event. Batch information is provided in batchPaymentDetails to identify specific batch payments.	N	N/A	
brand	string	Indicates the group brand this payment belongs to. For use in financial institutions that operate under multiple brands, such as consumer and commercial.	N	N/A	
cardId	string	A unique identifier for the card, which is a tokenised/masked PAN number and not the original PAN. For example, if populated, this must be a public card token. Unmasked and untokenised PANs are not permitted. **Note:** When using masked PANs there must be a direct 1:1 mapping back to the original PAN.	N	N/A	1254 6813 1465 4290

Attribute	Type	Description	Mandatory	Options	Example Values
		This field is expected to be populated in order to match card and non-card transactions. Alternatively, it is also expected to be populated in cases when card rails are used to make a peer-to-peer transfer. We would expect the following relationships to be true. Customers should verify that this is the case in their own data before mapping to cardId: • One cardId appears with only a small number of customerIds • One cardId appears with only a small number of accountIds			
channel	string	Channel through which the system was accessed. For example, it should be mapped to the channel through which the payment request was made. If the payer used a mobile app to make a payment into account then the channel should be set to `mobile`. If this information is not available, for example in case of inbound payments, channel should be set to `unknown`. **Note:** Providing this data can improve Scam Transaction Monitoring Machine Learning model performance.	Y	• online • mobile • atm • branch • lockbox • mailed check • post • telephone • agent • unknown	atm
checkDetails	checkDetails (for more information, see Appendix A: Derived Types)	Extra information relevant to processing checks/cheques.	N	N/A	
counterpartyAddress	address (for more information, see Appendix A: Derived Types)	Postal address of the customer associated with `counterpartyId`.	N	N/A	
counterpartyAgentId	string	Unique identifier of the financial institution (at the global level), providing the account for the counterparty where applicable. counterpartyAgentId relates to counterpartyBranchId, but may be less granular. For example, if counterpartyBranchId is passed as a sort code, it will not unqiuely identify a financial institution. counterpartyAgentId should identify the financial institution as a whole.	N	N/A	

Attribute	Type	Description	Mandatory	Options	Example Values
		Note: Providing this data can improve Scam Transaction Monitoring Machine Learning model performance.			
counterpartyAgentName	string	Name of the financial institution in counterpartyAgentId.	N	N/A	
counterpartyBranchAddress	address (for more information, see Appendix A: Derived Types)	The postal address for the branch at which the counterparty's account is held. Note that, for FI processing outbound transactions, this may only be available for "on-us" transactions, or in cases where the customer is made to supply this as part of the payment message, or using a lookup (outside of Scam Transaction Monitoring) from the counterpartyBranchId.	N	N/A	
counterpartyBranchId	string	Unique identifier for the specific FI's branch (at a more granular level than the agentId fields) at which the counterparty's account is held. In the UK this would typically be populated by a sort code. For examples on how the counterpartyBranchId is defined for a region, see Appendix B: Unique Identification Definitions.	Y	N/A	
counterpartyId	string	Identification of the counterparty. If this is an outbound transaction, counterpartyId will be the creditor's account. If this is an inbound transaction, counterpartyId will be the debitor's account. This is not an internally generated ID. This is a unique account number which is recognised by the payment scheme. For example, in the UK, if sort code is 112233 and account number is 12345678 then counterpartyId is 11223312345678. **Note:** For more information, see Appendix B: Unique Identification Definitions.	Y	N/A	
counterpartyIdFormat	string	Account numbers can take multiple formats. This is a free text field that describes the format of the counterpartyId.	N	• IBAN • UK Account • US Account	IBAN
counterpartyName	string	Name of the counterparty associated with counterpartyId. If this supplied by the customer as payee name and/or looked up if "on us".	N	N/A	Susan Smith
counterpartyType	string	The type of account for the countepartyId.	N	• Business • Personal	Personal

Attribute	Type	Description	Mandatory	Options	Example Values
customerAddress	address (for more information, see Appendix A: Derived Types)	Postal address the customer in customerId is registered to.	N	N/A	
customerFlag	array	Field used to specify customer flags that determine treatment strategies. This is an array that can include free text values. For retail customers this may be a vulnerability or a VIP marker. For business customers it should be related to the company.	N	N/A	VIP
customerId	string	A unique identifier for the customer. In an event with multiple customers, this must be the customer associated with the primary entity. If this is an outbound transaction, customerId will be the debtor. If this is an inbound transaction, customerId will be the creditor (which will be typically only available for "on-us" transactions). For retail banking type use cases, this should be an individual person. However, for business banking the customerId represents the business. customerId must be consistent across different event types. For example, the same customer should have the same customerId in paymentRT, paymentNRT, and paymentTransactionReturn. It is also essential that customerId is consistent over time. customerId should provide a link between cards, accounts, and devices. For example, if a customer replaces a card and gets a new card number, the customerId must stay the same.	Y	N/A	CA13771612318
customerName	string	For retail banking use cases this should be the name of the customer. For business banking use cases this would be the name of the company.	N	N/A	John Doe
customerType	string	The customer type.	N	• Retail • Commercial	retail
destinationCountry	string	The destination country of the funds, given as a 3 letter ISO 3166-1 Alpha-3 country code, such as GBR, AUS or CAN. **Note:** This can differ from the counterpartyAddress.country attribute, when the counterparty lives in a different country to where the funds are being sent. It will typically be the same as the counterpartyBranchAddress.country attribute, where available.	N	N/A	
device	deviceDetails	Details of the device used, excluding the ID.	N	N/A	

Attribute	Type	Description	Mandatory	Options	Example Values
	(for more information, see Appendix A: Derived Types)	**Note:** Providing this data can improve Scam Transaction Monitoring Machine Learning model performance.			
deviceId	string	A unique identifier of the device performing the event, such as the laptop/mobile phone used to log into the online banking. If this is an outbound transaction, deviceId will be the debtor's device. If this is an inbound transaction, deviceId will be the creditor's device. deviceId should be as consistent as possible over time. For example, a MAC address is a much better way to identify a device than an IP address. If this is used for profiling (as an entity), it is essential that deviceId satisfies several criteria. We recommend customers check their own data to ensure the following are true before using deviceId as an entity: • One deviceId should refer to one physical device in the world • One deviceId should be seen with a small number of customerIds, most often 1 • deviceId cannot change too often. If the median number of events per deviceId is very small, deviceId will perform poorly as an entity.	N	N/A	FG9834YY82
direction	string	The direction of the message. Valid values are outbound (for transactions being sent from the FI) or inbound (for transactions received by the FI). This should match the direction of the flow of funds.	Y	This field has validation, and must have one of the following options in the request: • outbound • inbound	outbound
eventTime	date-time	The date-time the event happened in the real world, defined by ISO 8601 and validated against RFC 3339. The eventTime should be prior (or equal to) the time the event was sent to Scam Transaction Monitoring, for example the date and time when the payment is sent or received. In the scenario where events are being backfilled into Scam Transaction Monitoring after an outage (to update profiles) the eventTime should reflect the time the event happened, not the time it was sent to Scam Transaction Monitoring. This field must contain a timezone designator, as specified in ISO 8601, which is either Z for UTC, or an offset from UTC in +HH:MM. eventTime should capture the time at which the event occurred in the real world. For example, the eventTime for an authorisation request should be the date and time when a request message was initiatiated. Preference should be given to transactions, rather than to internal processes. For example, suppose a batch file is	Y	N/A	2021-08-21T14:41:23Z

Attribute	Type	Description	Mandatory	Options	Example Values
		produced on October 2, containing transactions from October 1. eventTime should contain date-times from October 1.			
finalPaymentDate	date	The last date on which the payment will be sent. This is applicable for payments scheduled to be sent in the future.	N	N/A	
firstPaymentDate	date	The first date on which the payment will be sent. This is applicable for payments scheduled to be sent in the future.	N	N/A	
fraudLiability	string	If known, this is a free text field that enables you to denote who would be liable to provide the refund if the transaction turns out to be fraudulent. Typical values are `account`, `counterparty`, or `shared`. **Note:** This field determines which side of the transaction bears the liability (rather than distinguishing between whether it is the individual or the bank).	N	N/A	
initiatingPartyId	string	ID of the initiating party. It is expected to be mandatory for commercial use cases. The initiating party is the user initiating the payment on behalf of the business (not applicable to inbound payments).	N	N/A	
initiatingPartyName	string	Name of the initiator as identified in the `intiatiatingPartyId` attribute where applicable.	N	N/A	
initiatingPartyType	string	This attribute details how the initiating party is being used.The field can be used to either represent that it is an individual user (in commercial banking setting) or it's open banking or other entity that initiates the payment. Example values are `User` or `Open Banking`.	N	• User • Open Banking	User
localDateTime	date-time	Local date and time the transaction takes place at the payer location. Unlike eventTime, this should not have any associated timezone.	Y	N/A	2021-10-12T10:30:00
locationId	string	Identifier for staff location, site, or service centre. Should only populate when the event takes place in a branch, on the phone or by mail.	N	N/A	
merchantCatego	string	Merchant Category Code (MCC) related to the type of services or goods the merchant provides for the transaction. It is strongly recommended that this conforms to an	N	N/A	7011

Attribute	Type	Description	Mandatory	Options	Example Values
ryCode		international standard such as ISO18245. The merchantCategoryCode should only be populated when using this event to capture alternative payment methods, where a person pays a merchant direct from their bank account without needing a card (for example QR code based payments).			
merchantId	string	Identifier of the merchant in a transaction. This should be fully unique. It is also essential that this merchantId is consistent over time. The merchantId should only be populated when using this event to capture alternative payment methods, where a person pays a merchant direct from their bank account without needing a card (for example, QR code based payments). To ensure its uniqueness, consider concatenation with other card processing field(s) prior to sending this field through.	N	N/A	CamH33528
msgStatus	string	Identifies the status of a transaction during its flow. This should be set to New for a payment that needs to be risk-assessed by Scam Transaction Monitoring. The Setup value can be used to denote setting up a future-dated-payment, however it will not be risk assessed.	Y	This field has validation, and must have one of the following options in the request: • Setup • New	New
msgStatusReason	string	The reason for the msgStatus field. This is a free text field. It should contain useful information for an investigator, should investigation be necessary. For example, if a transaction was reversed, why this happened.	N	• New Payee • Trusted Beneficiary	New payee
msgType	string	This attribute should be set to Request unless it is specified to be another value due to a specific scenario that's been agreed with Thredd separately.	N	This field has validation, and must have one of the following options in the request: • Request	Request
numberOfTransactions	integer	Number of individual transactions contained in the message. This is relevant if it's related to a specific batch of payments. It would indicate a number of payments in the batch. For clarity, it is not expected that there is more than 1 payment in a single paymentRT message.	N	N/A	
paymentClearin	string	This attribute determines the analytics in the payments solution and determines the speed at which the payments is expected to clear. paymentClearingSpeed is needed for	Y	This field has validation, and must have one of	LessThanTwoHours

Attribute	Type	Description	Mandatory	Options	Example Values
gSpeed		payment rules as it signals what is used in payment model(s). This is related to payment clearing speed as set by the payment scheme. For example, Faster Payments would be LessThanTwoHours.		the following options in the request: • LessThanTwoHours • TwoHoursToOneDay • MoreThanOneDay	
paymentFrequency	string	If the payment is recurring, this attribute defines how often it occurs. **Note:** This field will be ignored, but should be blank, for incoming transactions. The fields are: • Year (YEAR) • Month (MNTH) • Quarterly (QURT) • Every six months (MIAN) • Weekly (WEEK) • Daily (DAIL) • Ad-hoc (ADHO) • Event takes place several times a day (INDA) • Fortnightly (FRTN)	N	This field has validation, and must have one of the following options in the request: • YEAR • MNTH • QURT • MIAN • WEEK • DAIL • ADHO • INDA • FRTN	YEAR
paymentGroupId	string	When the individual payment event is part of a group of payments, this identifier can be used to determine payments which belong to the same group (For example, a single order to split a payment between multiple beneficiaries). **Note:** Each individual transaction in the group should still have its own unique transactionId. If just sending individual payments, do not populate this attribute.	N	N/A	
paymentMethod	string	The method being used to make the payments. Note that both `Cash` and `Check` are protected values that the solution uses to uniquely determine these types of payments. For example, in the UK, if Faster Payment payment is used, then populate this field as Faster Payment. Apart from those values this attribute should be used as a free text field that helps analysts using the UI to determine the method of payment, and to write custom rules against. **Note:** The `Check` value is the same as cheque. `Cheque` is not supported as a value for paymentMethod.	Y	• Faster Payment • BACS • SEPA • CHAPS • RTP • ACH • FedNow • Check • Wire • Cash • Swift	Faster Payment

Attribute	Type	Description	Mandatory	Options	Example Values
				- On Us	
paymentPurpose	string	Free text field that indicates the intended purpose of the payment (if available). This differs from the payment reference since customers typically enter the payment reference themselves, while with paymentPurpose they are asked to select a purpose from a range of options provided by their FI.	N	N/A	
paymentReference	string	Reference for the payment added by the sender.	N	N/A	
paymentSubMethod	string	Used to provide more detailed information about the paymentMethod, if required. For example if paymentMethod is Check, this attribute describes the specific type of check such as Bearer Check.	N	- Standing Order - Future Dated Payment - Direct Debit - Real-time Payment - Foreign Exchange - Bearer Check - Order Check - Travellers Check - Other	Other
productId	string	Product ID code if applicable (this applies in re-seller use case). It needs to be the same one used by Thredd for billing and API validation purposes.	N	N/A	
programManagerCode	string	Name of the program. This should be the same one used by Thredd for billing and API validation purposes. Contact Thredd if you are unsure what this attribute should be.	Y	N/A	
requestExecutionDateTime	date-time	The time at which the payment is due to execute. For single immediate payments this should be close to the eventTime. However, for future dated payments this will be further in the future than the eventTime.	N	N/A	
tellerI	string	The unique identifier of the individual involved in the	N	N/A	

Attribute	Type	Description	Mandatory	Options	Example Values
d		transaction when it occurs at a Branch.			
totalAmount	money (for more information, see Appendix A: Derived Types)	The total amount of money moved as part of this payment request, including any additional fees.	N	N/A	
transactionId	string	A unique identifier for the transaction. It can be used to link different messages as part of the same transaction (for example, payment request and confirmation messages). **Note:** transactionId is used to link transactions to confirmed fraud events sent through the paymentTransactionReturn event.	Y	N/A	
transactionOnUsFlag	boolean	Indicates if the transaction is on-us, meaning when a payment's payer and payee belong to the same FI. Valid values are True or False.	N	N/A	
verificationResult	string	This is the final result of the verification, indicating whether the customer was successful in verifiying themselves. The verificationType determines whether any specific verifications were failed or were successful (see description of verificationType). For example, if the customer is successfully verified and can proceed with the action, the value would be `"verificationResult": "SUCC"`.	N	This field has validation, and must have one of the following options in the request: • SUCC • FAIL	SUCC
verificationType	verificationType (for more information, see Appendix A: Derived Types)	Type of the verification or authentication. The overall result of this is recorded in verificationResult. **Note:** oneTimePassword is a protected field meaning that there is a related UI functionality related to it, if it's provided. Providing this data can improve Scam Transaction Monitoring Machine Learning model performance.	N	N/A	
wireDetails	wireDetails (for more information, see Appendix A: Derived Types)	Extra information that may be relevant to processing wires.	N	N/A	

5 PaymentNRT

The PaymentNRT endpoint enables you to send non-real-time payment decisioning information to Scam Transaction Monitoring.

> **Note:** To ensure the system performs correctly, a PaymentNRT request must include accountId, counterpartyId, customerId, and then a maximum of two of the following id fields:
> - cardId
> - deviceId
> - initiatingPartyId
> - merchantId

PaymentNRT is not a mandatory event and can be used for On-Us payments (payments where both the payer and payee belong to the same FI), or Failed/Cancelled/Returned payments or any other payments which do not require real-time action. These events can be viewed in the Fraud Transaction Monitoring Portal and rules can be created to raise silent alerts for further investigation. The use of paymentNRT depends on FI requirements, but it is not expected to send Failed/Cancelled/Returned payments for Scam Transaction Monitoring model to be performant.

The following table describes each of the fields that can be included in the request body for the PaymentNRT endpoint. The Options column shows the available options that are validated by Scam Transaction Monitoring for that field where applicable. Only one of these options can be selected.

Attribute	Type	Description	Mandatory	Options	Example Values
accountAddress	address (for more information, see Appendix A: Derived Types)	Postal address the account specified in accountId is registered to.	N	N/A	
accountAgentId	string	Unique identifier of the financial institution (at the global level) providing the account in accountId. accountAgentId will relate to accountBranchId, but may be less granular. For example, if accountBranchId is passed as a sort code, this code will not unqiuely identify a financial institution. accountAgentId should identify the financial institution as a whole. For example, in the UK you would have an id for Barclays, and another id for NatWest. **Note:** Providing this data can improve Scam Transaction Monitoring Machine Learning model performance.	N	N/A	
accountAgentName	string	Name of the financial institution in accountAgentId.	N	N/A	
accountBalanceBefore	money (for more information, see Appendix A: Derived Types)	The balance before the transaction of the account in accountId. **Note:** Providing this data can improve Scam Transaction Monitoring Machine Learning model performance.	N	N/A	
accountBranchAddress	address (for more information, see Appendix A: Derived Types)	Postal address of the branch at which the account is held.	N	N/A	

Attribute	Type	Description	Mandatory	Options	Example Values
accountBranchId	string	Unique identifier for the branch the account is associated with at a more granular level than the accountAgentId field. In the UK this would typically be populated by a sort code, in the US this would typically be populated by a routing number. For examples on how the accountBranchId is defined for a region, see Appendix B: Unique Identification Definitions.	Y	N/A	
accountFlag	array	Any flags on the account that aren't covered by other attributes. This is an array that can include free text values that might indicate VIP accounts, or accounts that are suspected to have been compromised.	N	N/A	VIP
accountId	string	Unique identification of the account involved in the event. If this is an outbound transaction, accountId will be the debtor's account. If this is an inbound transaction, accountId will be the creditor's account. accountId must be consistent across different event types, and it must also be consistent over time. For clarity, this is not an internally generated ID. accountId is a unique account number which is recognised by the payment scheme. For example, in the UK, if the sort code is 112233, and the account number is 12345678, then accountId is 11223312345678. For examples on how the accountId is defined for a region, see Appendix B: Unique Identification Definitions. In order to ensure optimal model performance, the accountId must be consistent over time. For example, it shouldn't be represented with IBAN as well as account+sortcode. Customers should check that the following relationships exist in their data when mapping to accountId: • For personal banking, one accountId should be associated with a small number of customerIds For personal banking, one customerId should be associated with a small number of accountIds • One accountId should belong to a single accountAgentId. For example, in a card transaction we expect accountAgentId to be an issuing financial institution, and one accountId should never be seen with two different issuers.	Y	N/A	11223312345678
accountIdFormat	string	Account numbers can take multiple formats, this is a free text field that describes the format of the accountId.	N	• IBAN • UK account • US account	IBAN
accountOpenDate	date	The date the account was opened.	N	N/A	

Attribute	Type	Description	Mandatory	Options	Example Values
		Note: Providing this data can improve Scam Transaction Monitoring Machine Learning model performance.			
accountSubType	string	The sub type of the accountType. For example, accountType could be a current account and the subType would determine the specific product of the account type offered by the FI.	N	N/A	reward
accountType	string	The type of account. **Note:** Providing this data can improve Scam Transaction Monitoring Machine Learning model performance.	N	• Personal • Business • Current • Savings	Personal
amount	money (for more information, see Appendix A: Derived Types)	Actual amount of the transaction in the transaction currency.	Y	N/A	
approverId	array	An array containing identifiers of any approvers required to approve this action (such as making a payment or adding a payee). approverId is mainly used for commercial accounts, and identifies who approved the payment. **Note:** Do not populate this attribute if additional approval is not needed.	N	N/A	
batchPaymentDetails	batchPaymentDetails (for more information, see Appendix A: Derived Types)	Extra information that may be relevant to processing ACH transactions or other batch-type corporate payments. **Note:** Each payment in a batch payment is expected to have its own event. Batch information is provided in batchPaymentDetails to identify specific batch payments.	N	N/A	
brand	string	Indicates the group brand this payment belongs to. For use in financial institutions that operate under multiple brands, such as consumer and commercial.	N	N/A	
cardId	string	A unique identifier for the card, which is a tokenised/masked PAN number and not the original PAN. For example, if populated, this must be a public card token. Unmasked and untokenised PANs are not permitted. **Note:** When using masked PANs there must be a direct 1:1 mapping back to the original PAN. This field is expected to be populated in order to match card and non-card transactions. Alternatively, it is also expected to be populated in cases when card rails are used to make a peer-to-peer transfer. We would expect the following relationships to be	N	N/A	1254681314654290

Attribute	Type	Description	Mandatory	Options	Example Values
		true. Customers should verify that this is the case in their own data before mapping to cardId: • One cardId appears with only a small number of customerIds • One cardId appears with only a small number of accountIds			
channel	string	Channel through which the system was accessed. For example, it should be mapped to the channel through which the payment request was made. If the payer used a mobile app to make a payment into account then the channel should be set to `mobile`. If this information is not available, for example in case of inbound payments, channel should be set to `unknown`. **Note:** Providing this data can improve Scam Transaction Monitoring Machine Learning model performance.	Y	• online • mobile • atm • branch • lockbox • mailed check • post • telephone • agent • unknown	atm
checkDetails	checkDetails (for more information, see Appendix A: Derived Types)	Extra information relevant to processing checks/cheques.	N	N/A	
counterpartyAddress	address (for more information, see Appendix A: Derived Types)	Postal address of the customer associated with `counterpartyId`.	N	N/A	
counterpartyAgentId	string	Unique identifier of the financial institution (at the global level), providing the account for the counterparty where applicable. counterpartyAgentId relates to counterpartyBranchId, but may be less granular. For example, if counterpartyBranchId is passed as a sort code, it will not unqiuely identify a financial institution. counterpartyAgentId should identify the financial institution as a whole. **Note:** Providing this data can improve Scam Transaction Monitoring Machine Learning model performance.	N	N/A	
counterpartyAgentName	string	Name of the financial institution in `counterpartyAgentId`.	N	N/A	
counterpartyBranchAddress	address (for more information, see Appendix A: Derived	The postal address for the branch at which the counterparty's account is held. Note that, for FI processing outbound transactions, this may only be available for "on-us" transactions, or in cases where the customer is made to supply this as part of the	N	N/A	

Attribute	Type	Description	Mandatory	Options	Example Values
	Types)	payment message, or using a lookup (outside of ARIC) from the counterpartyBranchId.			
counterpartyBranchAddress	address (for more information, see Appendix A: Derived Types)	The postal address for the branch at which the counterparty's account is held. Note that, for FI processing outbound transactions, this may only be available for "on-us" transactions, or in cases where the customer is made to supply this as part of the payment message, or using a lookup (outside of Scam Transaction Monitoring) from the counterpartyBranchId.	N	N/A	
counterpartyId	string	Identification of the counterparty. If this is an outbound transaction, counterpartyId will be the creditor's account. If this is an inbound transaction, counterpartyId will be the debitor's account. This is not an internally generated ID. This is a unique account number which is recognised by the payment scheme. For example, in the UK, if sort code is 112233 and account number is 12345678 then counterpartyId is 11223312345678. **Note:** For more information, see Appendix B: Unique Identification Definitions.	Y	N/A	
counterpartyIdFormat	string	Account numbers can take multiple formats. This is a free text field that describes the format of the counterpartyId.	N	• IBAN • UK Account • US Account	IBAN
counterpartyName	string	Name of the counterparty associated with counterpartyId. If this supplied by the customer as payee name and/or looked up if "on us".	N	N/A	
counterpartyType	string	The type of account for the countepartyId.	N	• Business • Personal	Personal
customerAddress	address (for more information, see Appendix A: Derived Types)	Postal address the customer in customerId is registered to.	N	N/A	
customerFlag	array	Field used to specify customer flags that determine treatment strategies. This is an array that can include free text values. For retail customers this may be a vulnerability or a VIP marker. For business customers it should be related to the company.	N	N/A	VIP
customerId	string	A unique identifier for the customer. In an event with multiple customers, this must be the customer associated with the primary entity. If this is an outbound transaction, customerId will be the debtor. If this is an inbound transaction, customerId will be the creditor (which will be typically only available for "on-us" transactions).	Y	N/A	CA13771612318

Attribute	Type	Description	Mandatory	Options	Example Values
		For retail banking type use cases, this should be an individual person. However, for business banking the customerId represents the business. customerId must be consistent across different event types. For example, the same customer should have the same customerId in paymentRT, paymentNRT, and paymentTransactionReturn. It is also essential that customerId is consistent over time. customerId should provide a link between cards, accounts, and devices. For example, if a customer replaces a card and gets a new card number, the customerId must stay the same.			
customerName	string	For retail banking use cases this should be the name of the customer. For business banking use cases this would be the name of the company.	N	N/A	John Doe
customerType	string	The customer type.	N	• Retail • Commercial	retail
declinePhase	string	Indicates which system was responsible for declining a transaction.	N	N/A	
destinationCountry	string	The destination country of the funds, given as a 3 letter ISO 3166-1 Alpha-3 country code, such as GBR, AUS or CAN. **Note:** This can differ from the counterpartyAddress.country attribute, when the counterparty lives in a different country to where the funds are being sent. It will typically be the same as the counterpartyBranchAddress.country attribute, where available.	N	N/A	
device	deviceDetails (for more information, see Appendix A: Derived Types)	Details of the device used, excluding the ID. **Note:** Providing this data can improve Scam Transaction Monitoring Machine Learning model performance.	N	N/A	
deviceEntityId	string	Entity field for the device. This can contain the concatenation of the deviceId and the identifier of their bank (fromId/toId), or just the deviceId, depending on whether uniqueness can be guaranteed.	N	N/A	
deviceId	string	A unique identifier of the device performing the event, such as the laptop/mobile phone used to log into the online banking. If this is an outbound transaction, deviceId will be the debtor's device. If this is an inbound transaction, deviceId will be the creditor's device. deviceId should be as consistent as possible over time. For example, a MAC address is a much better way to identify a device than an IP address. If this is	N	N/A	FG9834YY82

Attribute	Type	Description	Mandatory	Options	Example Values
		used for profiling (as an entity), it is essential that deviceId satisfies several criteria. We recommend customers check their own data to ensure the following are true before using deviceId as an entity: • One deviceId should refer to one physical device in the world • One deviceId should be seen with a small number of customerIds, most often 1 • deviceId cannot change too often. If the median number of events per deviceId is very small, deviceId will perform poorly as an entity.			
direction	string	The direction of the message. Valid values are `outbound` (for transactions being sent from the FI) or `inbound` (for transactions received by the FI). This should match the direction of the flow of funds.	Y	This field has validation, and must have one of the following options in the request: • outbound • inbound	outbound
eventTime	date-time	The date-time the event happened in the real world, defined by ISO 8601 and validated against RFC 3339. The eventTime should be prior (or equal to) the time the event was sent to Scam Transaction Monitoring, for example the date and time when the payment is sent or received. In the scenario where events are being backfilled into Scam Transaction Monitoring after an outage (to update profiles) the eventTime should reflect the time the event happened, not the time it was sent to Scam Transaction Monitoring. This field must contain a timezone designator, as specified in ISO 8601, which is either Z for UTC, or an offset from UTC in +HH:MM. eventTime should capture the time at which the event occurred in the real world. For example, the eventTime for an authorisation request should be the date and time when a request message was initiated. Preference should be given to transactions, rather than to internal processes. For example, suppose a batch file is produced on October 2, containing transactions from October 1. eventTime should contain date-times from October 1.	Y	N/A	2021-08-21T14:41:23Z
finalPaymentDate	date	The last date on which the payment will be sent. This is applicable for payments scheduled to be sent in the future.	N	N/A	
firstPaymentDate	date	The first date on which the payment will be sent. This is applicable for payments scheduled to be sent in the future.	N	N/A	
fraudLiability	string	If known, this is a free text field that enables you to	N	N/A	shared

Attribute	Type	Description	Mandatory	Options	Example Values
		denote who would be liable to provide the refund if the transaction turns out to be fraudulent. Typical values are account, counterparty, or shared. **Note:** This field determines which side of the transaction bears the liability (rather than distinguishing between whether it is the individual or the bank).			
initiatingPartyName	string	Name of the initiator as identified in the intiatiatingPartyId attribute where applicable.	N	N/A	
initiatingPartyType	string	This attribute details how the initiating party is being used.The field can be used to either represent that it is an individual user (in commercial banking setting) or it's open banking or other entity that initiates the payment. Example values are User or Open Banking.	N	• User • Open Banking	User
localDateTime	date-time	Local date and time the transaction takes place at the payer location. Unlike eventTime, this should not have any associated timezone.	Y	N/A	2021-10-12T10:30:00
locationId	string	Identifier for staff location, site, or service centre. Should only populate when the event takes place in a branch, on the phone or by mail.	N	N/A	
merchantCategoryCode	string	Merchant Category Code (MCC) related to the type of services or goods the merchant provides for the transaction. It is strongly recommended that this conforms to an international standard such as ISO18245. The merchantCategoryCode should only be populated when using this event to capture alternative payment methods, where a person pays a merchant direct from their bank account without needing a card (for example QR code based payments).	N	N/A	7011
merchantId	string	Identifier of the merchant in a transaction. This should be fully unique. It is also essential that this merchantId is consistent over time. The merchantId should only be populated when using this event to capture alternative payment methods, where a person pays a merchant direct from their bank account without needing a card (for example, QR code based payments). To ensure its uniqueness, consider concatenation with other card processing field(s) prior to sending this field through.	N	N/A	CamH33528
msgStatus	string	Identifies the status of a transaction during its flow. This should be set to New for a payment that needs to be risk-assessed by Scam Transaction Monitoring. Failed is for when a payment failed before it got sent to the receiving FI. Cancelled is for scheduled payments like a standing order, specifically when a customer cancelled a payment after it was risk scored by Scam Transaction Monitoring but before the next scheduled payment	Y	This field has validation, and must have one of the following options in the request: • Failed • Cancelled • Returned	New

Attribute	Type	Description	Mandatory	Options	Example Values
		was executed. Returned is used if the receiving FI returns the funds for the initial payment to the payer's FI.		- New	
msgStatusReason	string	The reason for the msgStatus field. This is a free text field. It should contain useful information for an investigator, should investigation be necessary. For example, if a transaction was reversed, why this happened.	N	- New Payee - Trusted Beneficiary	New Payee
msgType	string	This attribute is used to determine the message type. It can be used to denote decline stages or settlement messages. If it's set as Request, this determines it as a request to send a payment. In example values, Pre-decline means decline before payment was sent to Scam Transaction Monitoring. Post-decline means decline after payment was sent to Scam Transaction Monitoring.	N	- Request - Post-decline - Pre-decline	Request
numberOfTransactions	integer	Number of individual transactions contained in the message. This is relevant if it's related to a specific batch of payments. It would indicate a number of payments in the batch. For clarity, it is not expected that there is more than 1 payment in a single paymentRT message.	N	N/A	
paymentClearingSpeed	string	This attribute determines the analytics in the payments solution and determines the speed at which the payments is expected to clear. paymentClearingSpeed is needed for payment rules as it signals what is used in payment model(s). This is related to payment clearing speed as set by the payment scheme. For example, Faster Payments would be LessThanTwoHours.	Y	This field has validation, and must have one of the following options in the request: - LessThanTwoHours - TwoHoursToOneDay - MoreThanOneDay	LessThanTwoHours
paymentFrequency	string	If the payment is recurring, this attribute defines how often it occurs. **Note:** This field will be ignored, but should be blank, for incoming transactions. The fields are: - Year (YEAR) - Month (MNTH) - Quarterly (QURT) - Every six months (MIAN) - Weekly (WEEK) - Daily (DAIL) - Ad-hoc (ADHO) - Event takes place several times a day (INDA) - Fortnightly (FRTN)	N	This field has validation, and must have one of the following options in the request: - YEAR - MNTH - QURT - MIAN - WEEK - DAIL - ADHO - INDA - FRTN	YEAR

Attribute	Type	Description	Mandatory	Options	Example Values
paymentGroupId	string	When the individual payment event is part of a group of payments, this identifier can be used to determine payments which belong to the same group (For example, a single order to split a payment between multiple beneficiaries). **Note:** Each individual transaction in the group should still have its own unique transactionId. If just sending individual payments, do not populate this attribute.	N	N/A	
paymentMethod	string	The method being used to make the payments. Note that both Cash and Check are protected values that the solution uses to uniquely determine these types of payments. For example, in the UK, if Faster Payment payment is used, then populate this field as Faster Payment. Apart from those values this attribute should be used as a free text field that helps analysts using the UI to determine the method of payment, and to write custom rules against. **Note:** The Check value is the same as cheque. Cheque is not supported as a value for paymentMethod.	Y	• Faster Payment • BACS • SEPA • CHAPS • RTP • ACH • FedNow • Check • Wire • Cash • Swift • On Us	Faster Payment
paymentPurpose	string	Free text field that indicates the intended purpose of the payment (if available). This differs from the payment reference since customers typically enter the payment reference themselves, while with paymentPurpose they are asked to select a purpose from a range of options provided by their FI.	N	N/A	
paymentReferenc e	string	Reference for the payment added by the sender.	N	N/A	
paymentSubMeth od	string	Used to provide more detailed information about the paymentMethod, if required. For example if paymentMethod is Check, this attribute describes the specific type of check such as Bearer Check.	N	• Standing Order • Future Dated Payment • Direct Debit • Real-time Payment • Foreign Exchange • Bearer Check • Order Check • Travellers Check • Other	Other
productId	string	Product ID code if applicable (this applies in re-seller use case). It needs to be the same one used by	N	N/A	

Attribute	Type	Description	Mandatory	Options	Example Values
		Thredd for billing and API validation purposes.			
programManagerCode	string	Name of the program. This should be the same one used by Thredd for billing and API validation purposes. Contact Thredd if you are unsure what this attribute should be.	Y	N/A	
requestExecutionDateTime	date-time	The time at which the payment is due to execute. For single immediate payments this should be close to the `eventTime`. However, for future dated payments this will be further in the future than the `eventTime`.	N	N/A	2021-08-21T14:41:23Z
tellerId	string	The unique identifier of the individual involved in the transaction when it occurs at a Branch.	N	N/A	
totalAmount	money (for more information, see Appendix A: Derived Types)	The total amount of money moved as part of this payment request, including any additional fees.	N	N/A	
transactionId	string	A unique identifier for the transaction. It can be used to link different messages as part of the same transaction (for example, payment request and confirmation messages). **Note:** transactionId is used to link transactions to confirmed fraud events sent through the paymentTransactionReturn event.	Y	N/A	
transactionOnUsFlag	boolean	Indicates if the transaction is on-us, meaning when a payment's payer and payee belong to the same FI. Valid values are True or False.	N	N/A	
verificationResult	string	This is the final result of the verification, indicating whether the customer was successful in verifiying themselves. The verificationType determines whether any specific verifications were failed or were successful (see description of verificationType). For example, if the customer is successfully verified and can proceed with the action, the value would be `"verificationResult": "SUCC"`.	N	This field has validation, and must have one of the following options in the request: • SUCC • FAIL	SUCC
verificationType	verificationType (for more information, see Appendix A: Derived Types)	Type of the verification or authentication. The overall result of this is recorded in verificationResult. **Note:** oneTimePassword is a protected field meaning that there is a related UI functionality related to it, if it's provided. Providing this data can improve Scam Transaction Monitoring Machine Learning model performance.	N	N/A	
wireDetails	wireDetails (for more information, see Appendix A:	Extra information that may be relevant to processing wires.	N	N/A	

Attribute	Type	Description	Mandatory	Options	Example Values
	Derived Types)				

6 paymentTransactionReturn

The paymentTransactionReturn endpoint enables you to send information confirming that a scam or fraud has taken place. Data attributes that relate to the original payment message must be populated as they were populated in the payment message.

Note: To ensure the system performs correctly, a paymentTransactionReturn request must include accountId, counterpartyId, customerId, and then a maximum of two of the following id fields:
- cardId
- deviceId
- initiatingPartyId
- merchantId

The following table describes the fields that can be included in the request body. The Options column shows the available options that are validated by Scam Transaction Monitoring for that field where applicable. Only one of these options can be selected.

Attribute	Type	Description	Mandatory	Options	Example Values
accountAgentId	string	Unique identifier of the financial institution (at the global level) providing the account in accountId. accountAgentId will relate to accountBranchId, but may be less granular. For example, if accountBranchId is passed as a sort code, this code will not uniquely identify a financial institution. accountAgentId should identify the financial institution as a whole. For example, in the UK you would have an id for Barclays, and another id for NatWest. **Note:** Providing this data can improve Scam Transaction Monitoring Machine Learning model performance.	N	N/A	
accountAgentName	string	Name of the financial institution in accountAgentId.	N	N/A	
accountBranchId	string	Unique identifier for the branch the account is associated with at a more granular level than the accountAgentId field. In the UK this would typically be populated by a sort code, in the US this would typically be populated by a routing number. For examples on how the accountBranchId is defined for a region, see Appendix B: Unique Identification Definitions.	Y	N/A	
accountId	string	Unique identification of the account involved in the event. If this is an outbound transaction, accountId will be the debtor's account.	Y	N/A	11223312345678

Attribute	Type	Description	Mandatory	Options	Example Values
		If this is an inbound transaction, accountId will be the creditor's account. accountId must be consistent across different event types, and it must also be consistent over time. For clarity, this is not an internally generated ID. accountId is a unique account number which is recognised by the payment scheme. For example, in the UK, if sort code is 112233, and the account number is 12345678, then accountId is 11223312345678. For examples on how the accountId is defined for a region, see Appendix B: Unique Identification Definitions. In order to ensure optimal model performance, the accountId must be consistent over time. For example, it shouldn't be represented with IBAN as well as account and sort code. Customers should check that the following relationships exist in their data when mapping to accountId: • For personal banking, one accountId should be associated with a small number of customerIds • For personal banking, one customerId should be associated with a small number of accountIds • One accountId should belong to a single accountAgentId. For example, in a card transaction we expect accountAgentId to be an issuing financial institution, and one accountId should never be seen with two different issuers. **Note:** accountId must match the one used in the original transaction.			
accountIdFormat	string	Account numbers can take multiple formats, this is a free text field that describes the format of the accountId.	N	• IBAN • UK account • US account	IBAN
authorizationIndicator	boolean	Indicates whether the original transaction was successfully authorised or not. False indicates the authorisation was not authorised, while true indicates the transaction was authorised.	N	N/A	TRUE
cardId	string	A unique identifier for the card, which is	N	N/A	1254681

Attribute	Type	Description	Mandatory	Options	Example Values
		a tokenised/masked PAN number and not the original PAN. For example, if populated, this must be a public card token. Unmasked and untokenised PANs are not permitted. **Note:** When using masked PANs there must be a direct 1:1 mapping back to the original PAN. cardId refers to the card involved in the original transaction and not the one involved in the reporting. This may be required for Alternative Payment Methods. We would expect the following relationships to be true. Customers should verify that this is the case in their own data before mapping to cardId: • One cardId appears with only a small number of customerIds • One cardId appears with only a small number of accountIds			314654290
confirmedRisk	boolean	If true, confirms the original event is a risk (marked as a fraud or scam). This should be set to true even if a chargeback or reimbursement wasn't raised (for example, if the transaction was declined before it was sent or authorised). In cases where non-fradulent chargebacks are also sent toScam Transaction Monitoring, set this to false so that Scam Transaction Monitoring can determine what is and isn't risk.	Y	N/A	
counterpartyAgentId	string	Unique identifier of the financial institution (at the global level), providing the account for the counterparty where applicable. counterpartyAgentId relates to counterpartyBranchId, but may be less granular. For example, if counterpartyBranchId is passed as a sort code, it will not unqiuely identify a financial institution. counterpartyAgentId should identify the financial institution as a whole. **Note:** Providing this data can improve Scam Transaction Monitoring Machine Learning model performance.	N	N/A	

Attribute	Type	Description	Mandatory	Options	Example Values
counterpartyBranchId	string	Unique identifier for the specific FI's branch (at a more granular level than the agentId fields) at which the counterparty's account is held. In the UK this would typically be populated by a sort code. For examples on how the counterpartyBranchId is defined for a region, see Appendix B: Unique Identification Definitions.	Y	N/A	
counterpartyId	string	Identification of the counterparty. If this is an outbound transaction, counterpartyId will be the creditor's account. If this is an inbound transaction, counterpartyId will be the debitor's account. This is not an internally generated ID. This is a unique account number which is recognised by the payment scheme. For example, in the UK, if sort code is 112233 and account number is 12345678 then counterpartyId is 11223312345678. counterpartyId must match the one used in the original transaction. **Note:** For more information, see Appendix B: Unique Identification Definitions.	Y	N/A	11223312345678
counterpartyIdFormat	string	Account numbers can take multiple formats. This is a free text field that describes the format of the counterpartyId.	N	• IBAN • UK Account • US Account	IBAN
customerId	string	A unique identifier for the customer. In an event with multiple customers, this must be the customer associated with the primary entity. If this is an outbound transaction, customerId will be the debtor. If this is an inbound transaction, customerId will be the creditor (which will be typically only available for "on-us" transactions). For retail banking type use cases, this should be an individual person. However, for business banking the customerId represents the business. customerId must be consistent across different event types. For example, the same customer should have the same customerId in paymentRT,	Y	N/A	CA13771612318

Attribute	Type	Description	Mandatory	Options	Example Values
		paymentNRT, and paymentTransactionReturn. It is also essential that customerId is consistent over time. customerId should provide a link between cards, accounts, and devices. For example, if a customer replaces a card and gets a new card number, the customerId must stay the same.			
deviceId	string	A unique identifier of the device performing the event, such as the laptop/mobile phone used to log into the online banking. If this is an outbound transaction, deviceId will be the debtor's device. If this is an inbound transaction, deviceId will be the creditor's device. deviceId should be as consistent as possible over time. For example, a MAC address is a much better way to identify a device than an IP address. If this is used for profiling (as an entity), it is essential that deviceId satisfies several criteria. We recommend customers check their own data to ensure the following are true before using deviceId as an entity: • One deviceId should refer to one physical device in the world • One deviceId should be seen with a small number of customerIds, most often 1 • deviceId cannot change too often. If the median number of events per deviceId is very small, deviceId will perform poorly as an entity. **Note:** Providing this data can improve Scam Transaction Monitoring Machine Learning model performance.	N	N/A	FG9834 YY82
eventTime	date-time	The date-time the event happened in the real world, defined by ISO 8601 and validated against RFC3339. The eventTime should be prior (or equal to) the time the event was sent to Scam Transaction Monitoring i.e. the date-time when the confirmation of fraud / scam was sent. In the scenario where events are being	Y	N/A	2021-08-21T14:41:23Z

Attribute	Type	Description	Mandatory	Options	Example Values
		backfilled into Scam Transaction Monitoring after an outage (to update profiles) the eventTime should reflect the time the event happened, not the time it was sent to Scam Transaction Monitoring. This field must contain a timezone designator, as specified in ISO 8601, which is either Z for UTC, or an offset from UTC in +HH:MM. eventTime should capture the time at which the event occurred in the real world. For example, the eventTime for an authorisation request should be the date and time when a request message was initiatiated.			
initiatingPartyId	string	ID of the initiating party. It is expected to be mandatory for commercial use cases. The initiating party is the user initiating the payment on behalf of the business (not applicable to inbound payments). **Note:** initiatingPartyId refers to the initiatingParty involved in the original transaction and not the one involved in the reporting.	N	N/A	
merchantCategoryCode	string	Merchant Category Code (MCC) related to the type of services or goods the merchant provides for the transaction. It is strongly recommended that this conforms to an international standard such as ISO18245. The merchantCategoryCode should only be populated when using this event to capture alternative payment methods, where a person pays a merchant direct from their bank account without needing a card (for example QR code based payments). **Note:** merchantCategoryCode refers to the MCC involved in the original transaction and not the one involved in the reporting.	N	N/A	7011
merchantId	string	Identifier of the merchant in a transaction. This should be fully unique. It is also essential that this merchantId is consistent over time. The merchantId should only be	N	N/A	CamH33528

Attribute	Type	Description	Mandatory	Options	Example Values
		populated when using this event to capture alternative payment methods, where a person pays a merchant direct from their bank account without needing a card (for example, QR code based payments). To ensure its uniqueness, consider concatenation with other card processing field(s) prior to sending this field through. **Note:** merchantId refers to the merchantId involved in the original transaction and not the one involved in the reporting.			
msgStatus	string	This attribute identifies the status of the action. Set this value to `Risk` when there is a confirmed fraud or scam report.	Y	This field has validation, and must have `Risk` as the value.	Risk
msgStatusReason	string	Explains the reason for the msgStatus field. Any free text captured as part of working the case, for example, customer sentiment reported by the analyst.	N	N/A	
originalAmount	money (for more information, see Appendix A: Derived Types)	Amount of money, as provided in the original transaction.	Y	N/A	
originalEventTime	date-time	Indicates the time at which the original transaction took place (and not the time at which the event was reported by the customer or risk analyst).	Y	N/A	2021-08-21T14:41:23Z
originalTransactionDirection	string	The direction of the original message. Valid values are `outbound` (for transactions being sent from the FI) or `inbound` (for transactions received by the FI).	Y	This field has validation, and must have one of the following options in the request: • inbound • outbound	outbound
originalTransactionId	string	A unique identifier for the transaction, used to link confirmed fraud or scam events sent through the paymentTransactionReturn event back to the original transaction.	Y	N/A	

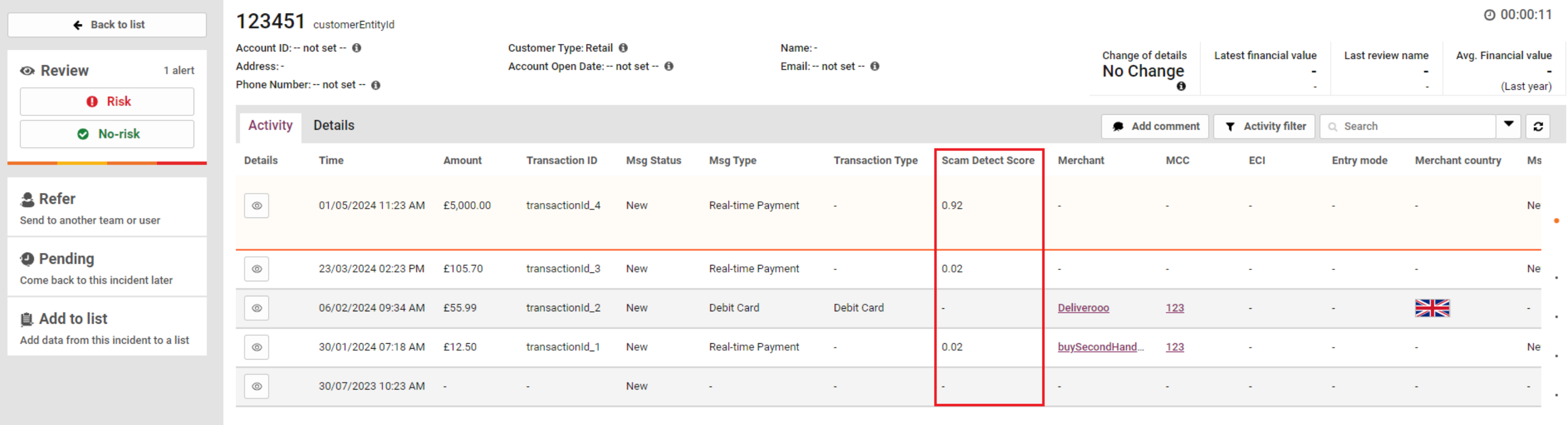
Attribute	Type	Description	Mandatory	Options	Example Values
productId	string	Product ID code if applicable (this applies in re-seller use case). It needs to be the same one used by Thredd for billing and API validation purposes.	N	N/A	
programManagerCode	string	Name of the program. This should be the same one used by Thredd for billing and API validation purposes. Contact Thredd if you are unsure what this attribute should be.	Y	N/A	
reportedBy	string	This is the entity reporting the originating event.	N	- Customer - Fraud Analyst	Customer
returnedAmount	money	Total amount refunded to the customer in this event. This can be different to the original transaction amount, specificied in the originalAmount field.	N	N/A	
returnSubType	string	The subtype of risk label. For payments, examples are advance fee scam, investment scam, invoice and mandate scam, phising/smishing, purchase scam, romance scam, vishing. **Note:** Providing this data can improve Scam Transaction Monitoring Machine Learning model performance.	N	- Account Takeover - Counterfeit With Unassigned IIN - Card Not Received - Fraudulent Account Use - Fraudulent Application - Counterfeit with Existing Account - Lost Card - Miscellaneous - Other National - Other Private - Stolen Card - Card Not Present - Account Use - Multiple Use Fraud - Collusion - 1st Party Fraud - 2nd Party Fraud - 3rd Party Fraud - Card Cloned - Advance Fee Scam - Investment Scam - Invoice and Mandate Scam - Phishing/Smishing	Account Takeover

Attribute	Type	Description	Mandatory	Options	Example Values
				• Purchase Scam • Romance Scam • Vishing	
returnType	string	This attribute indicates if paymentTransactionReturn report is for a Fraud or a Scam. Scam is authorised push payment fraud when a customer is tricked into approving a transaction. Fraud is unauthorised payment fraud, for example an account takeover.	Y	This field has validation, and must have one of the following options in the request: • Fraud • Scam	Fraud

7 Managing Scams

When Scam Transaction Monitoring has been set up successfully, any suspected scams can be sent to Fraud Portal to be assessed. Scams appear in Fraud Portal in the same way as fraud events do, and are managed the same way too.

If the incident is a scam, the Scam Detect Score displays on the activity. Navigate to the Incidents page in Fraud Transaction Monitoring Portal by clicking on the Incidents tab.



123451 customerEntityId

Account ID: -- not set --
Address: -
Phone Number: -- not set --

Customer Type: Retail
Account Open Date: -- not set --

Name: -
Email: -- not set --

Change of details	Latest financial value	Last review name	Avg. Financial value
No Change	- -	- -	- (Last year)

Details	Time	Amount	Transaction ID	Msg Status	Msg Type	Transaction Type	Scam Detect Score	Merchant	MCC	ECI	Entry mode	Merchant country	Ms
	01/05/2024 11:23 AM	£5,000.00	transactionId_4	New	Real-time Payment	-	0.92	-	-	-	-	-	Ne
	23/03/2024 02:23 PM	£105.70	transactionId_3	New	Real-time Payment	-	0.02	-	-	-	-	-	Ne
	06/02/2024 09:34 AM	£55.99	transactionId_2	New	Debit Card	Debit Card	-	Deliverooo	123	-	-	🇬🇧	-
	30/01/2024 07:18 AM	£12.50	transactionId_1	New	Real-time Payment	-	0.02	buySecondHand...	123	-	-	-	Ne
	30/07/2023 10:23 AM	-	-	New	-	-	-	-	-	-	-	-	-

Figure 3: Viewing the Scam Detect Score field for a potential scam incident

Scam activities work in the same way as fraud activities, where a risk analyst should review the activity and decide whether there is a risk there. For more information on how to manage risks, see the Fraud Transaction Monitoring Portal Guide.

Scam Transaction Monitoring Webhook Notification

The Scam Transaction Monitoring webhook notification enables you to action your pending payments based on the outcome of the manual review. When the webhook service has been successfully set up, notifications will be sent when:

- There is a payment event only. For example, a non-card event
- An incident is reviewed (one message per alert reviewed) and set to Risk or No Risk

To use these notifications, you must first set up a webhook using the webhook event that allows for scam alerts to be sent. Scam alerts are sent using the **105** event code. When an event is processed in Scam Transaction Monitoring, a message is sent in near real time using the webhook.

See the following example of an API request to create a webhook to support scam notifications.

```
{
  "programManagerCode" : "TRD",
  "productId": 12345,
  "events": [105],
  "webhookStatus": "active",
  "config": {
    "url": "https://client_domain.com/webhook",
    "customHeaders": {
      "header1": "value_1",
      "header2": "value_2"
    }
  }
}
```

When the webhook is created, the notification is sent with details of the transaction, the event id, the review status and review date. The following example shows the response of a 105 event code.

```
{
  "context": {
    "programManagerId": 16,
    "eventCode": 105,
    "eventVersion": "v1",
    "notificationTime": "2024-11-24T11:20:28Z",
  },
  "payload":
   {
      "transactionId": "123412341234",
      "eventid": "55221979-a4cd-4d7a-bb0c-1b3f0b7cb92c",
      "reviewStatus": "no-risk",
      "reviewDate": "2024-11-24T09:15:10.086Z"
   },
  "messageHeaders": {
    "schemaId": 1
  }
```

Note: For more information on creating and managing webhooks, see Introduction to Webhooks.

8 Appendix A: Derived Types

The following table describes the different derived types used by the Scam Transaction Monitoring schema. A derived type is a group of nested attributes that share some commonality. For example, ways to express elements of an address.

There are eight derived types:

- address
- batchPaymentDetails
- checkDetails
- deviceDetails
- duration
- money
- verificationType
- wireDetails

Address Derived Type

The following table describes the attributes for the **address** Derived Type. The address type is used for the following attributes:

- accountAddress
- accountBranchAddress
- counterpartyAddress
- customerAddress

Attribute	Description	Type	Mandatory
addressLine1	Address line 1 of entity.	string	Yes
addressLine2	Address line 2 of entity.	string	No
addressLine3	Address line 3 of entity.	string	No
addressLineType	Identifies the nature of the postal address.	string	No
country	Nation with its own government, country code in ISO 3166 alpha-3 format	string	Yes
countrySubDivision	Identifies a subdivision of a country such as state, region, county.	string	No
latitude	Latitude of the address.	number	No
longitude	Longitude of the address.	number	No
postalCode	Identifier consisting of a group of letters and/or numbers that is added to a postal address to assist the sorting of mail.	string	Yes
townName	Name of a built-up area, with defined boundaries, and a local government.	string	No
fullAddress	Full string containing all submitted address components.	string	No
residentAtAddressFrom	The date the applicant started to live at the address.	date	No
residentAtAddressTo	The date the applicant left the address.	date	No
timeAtAddress	How long the applicant has spent at the address.	duration (for more information, see duration Derived Type)	No

batchPaymentDetails Derived Type

The following table describes the attributes for the **batchPaymentDetails** Derived Type. The batchPaymentDetails type is used for the batchPaymentDetails attribute only.

Attribute	Description	Type	Mandatory
batchNumber	Assign batch numbers in ascending order in each file.	string	No
categoryPurposeDescription	Description of payment entries, such as Payroll or eCheckPay.	string	No
endOfBatchIndicator	Indicates that all records within the batch have been processed. This should be updated on the last entry record processed to enable analytics on "totals" fields for each batch.	boolean	No
endOfFileIndicator	Indicates that all batches within the file have been processed. This should be updated on the last entry record processed to enable analytics on "totals" fields.	boolean	No
entryDetailRecNum	Incremental number of each detail record within the batch.	number	No
fileIdModifier	Code to distinguish between multiple files sent per day.	string	No
numberOfAddendaRecords	The number of Addenda records included.	number	No
serviceClassCode	Identifies the type of entries in the batch. 200 (credits/debits mixed) 220 (credits) and 225 (debits).	string	No
terminalAddress	For point of purchase (PoP) ACHs only, this is the address where the electronic terminal is located.	address	No
totalBatchCountInFile	Total number of batches in the file.	number	No
totalBatchCreditsAmount	Total value amount of Credits in the file.	money	No
totalBatchDebitsAmount	Total value amount of Debits in the batch.	money	No
totalBatchEntries	Total number of Entry detail	number	No

Attribute	Description	Type	Mandatory
	records plus addenda records.		
totalEntryCountInFile	Total number of entries and addenda records in the file.	number	No
totalEntryHash	Total number of Transit routing/ama numbers in the batch.	number	No
totalFileCredits	Total value amount of Credits in the File.	money	No
totalFileDebits	Total value amount of Debits in the File.	money	No
totalTransitCountInFile	Total number of routing and transit numbers in the file (Across all entry detail records).	number	No

checkDetails Derived Type

The following table describes the attributes for the **checkDetails** Derived Type. The checkDetails type is used for the checkDetails attribute only.

Attribute	Description	Type	Mandatory
checkNumber	Number printed on each check to show its position in a series of issued or printed checks. Note this may not be unique for a given account.	string	No
depositedCashAmount	The amount of cash which was deposited along with check items.	money	No
depositSlipId	The deposit slip item related to the transaction.	string	No
depositLocation	Identifies the address at which the check was deposited.	address	No
mICRAccountNumber	The account number as recorded on the bottom of the check.	string	No
routingTransitNumber	The routing and transit code imprinted on the check MICR line.	string	No
splitDepositFlag	Identifies if the deposit is into a single account or split between 2 or more accounts. true for a single account, false otherwise.	boolean	No
splitAcctId2	Identifies the account Id of the split deposit, only updated when split deposit flag is true.	string	No
splitAcctId3	Identifies the account Id of the split deposit, only updated when split deposit flag is true.	string	No
splitAcctId4	Identifies the account Id of the split deposit, only updated when split deposit flag is true.	string	No

deviceDetails Derived Type

The following table describes the attributes for the **deviceDetails** Derived Type. The deviceDetails type is used for the device attribute only.

Attribute	Description	Type	Mandatory
anonymizerInUseFlag	A flag indicating if the anonymizer was in use during the session. This is true if the anonymizer was in use.	boolean	No
areaCode	The device area code captured during the customer session.	string	No
browserType	The device browser type captured during the customer session.	string	No
browserVersion	The device browser version captured during the customer session.	string	No
city	The device city name captured during the customer session.	string	No
clientTimezone	The clientTimeZone captured during the customer session.	string	No
continentCode	The device continent code captured during the customer session.	string	No
cookieId	The cookie ID used during the customer session.	string	No
countryCode	The device country code captured during the customer session.	string	No
countryName	The device country name captured during the customer session.	string	No
deviceFingerprint	The device fingerprint captured during the customer session by a third party.	string	No
deviceIMEI	The IMEI of the device used during the customer session.	string	No
deviceName	The device name given at point of registration.	string	No
flashPluginPresent	The Flash Plugin captured during the customer session.	string	No

Attribute	Description	Type	Mandatory
httpHeader	The HTTPHeader captured during the customer session.	string	No
ipAddress	The Ipaddress captured during the customer session.	string	No
ipAddressV4	The Ipaddress v4 captured during the customer session.	string	No
ipAddressV6	The Ipaddress v6 captured during the customer session.	string	No
metroCode	The device metropolitan code captured during the customer session.	string	No
mimeTypesPresent	The Mime-Type(s) captured during the customer session.	string	No
mobileNumberDeviceLink	A concatenated string of the mobile number and device ID to establish the link.	string	No
networkCarrier	The Network/carrier captured during the customer session.	string	No
oS	The device Operating System captured during the customer session.	string	No
postalCode	The device postal code captured during the customer session.	string	No
proxyDescription	The Proxy type description captured during the customer session.	string	No
proxyType	The Proxy type captured during the customer session.	string	No
region	The device region code captured during the customer session.	string	No
screenResolution	The screen resolution captured during the customer session.	string	No
sessionLatitude	Thelatitude captured during the customer session.	number	No
sessionLongitude	The longitude captured during the customer session.	number	No

Attribute	Description	Type	Mandatory
timestamp	The timestamp captured during the customer session.	date-time	No
type	The Device Type captured during the customer session.	string	No
userAgentString	The UserAgentString captured during the customer session.	string	No

duration Derived Type

The following table describes the attributes for the **duration** Derived Type. The money type is used for the timeAtAddress in the address derived type.

Attribute	Description	Type	Mandatory
unit	The time unit of the duration. For example, MONTH.	string	Yes
value	The value of the duration. For example, 11.	number	Yes

money Derived Type

The following table describes the attributes for the **money** Derived Type. The money type is used for the following attributes:

- accountBalanceBefore
- amount
- originalAmount
- returnedAmount
- totalAmount

Attribute	Description	Type	Mandatory
currency	3 letter ISO 4217 currency code, such as GBP, USD or EUR.	string	Yes
value	Value of transaction.	number	Yes

verificationType Derived Type

The following table describes the attributes for the **verificationType** Derived Type. The verificationType type is used for the verificationType attribute only.

Attribute	Description	Type	Mandatory
aa	The Accountholder Authentication Value {AAV} is a MasterCard SecureCode specific token that uses the Universal Cardholder Authentication Field (UCAF) field for transport in MasterCard authorization messages.	string	No
accountDigitalSignature	Account-based digital signature authentication.	string	No
authenticationToken	A token is used to verify an already performed authentication.	string	No
avs	Address Verification Service (AVS) verification.	string	No
biometry	Biometric authentication of the cardholder.	string	No
cardholderIdentificationData	Cardholder data provided for verification, for instance social security number, driver license number, passport number.	string	No
cryptogramVerification	Verification of a cryptogram generated by a chip card or another device, for instance ARQC (Authorisation Request Cryptogram).	string	No
cscVerification	Verification of Card Security Code.	string	No
cvv	A 3 or 4 digit code to provide a check of the card's authenticity.	string	No
offlinePIN	Off-line PIN authentication (Personal Identification Number).	string	No
oneTimePassword	Verification of a one-time password provided by the issuer.	string	No
onlinePIN	On-line PIN authentication (Personal Identification Number).	string	No
other	Other type of verification.	string	No

Attribute	Description	Type	Mandatory
paperSignature	Handwritten paper signature.	string	No
passiveAuthentication	Authentication based on statistical cardholder behaviour.	string	No
password	Authentication by a password.	string	No
threeDS	Authentication performed during a secure electronic commerce transaction.	string	No
tokenAuthentication	Cryptogram generated by the token requestor or a customer device to validate the authorised use of a token.	string	No

wireDetails Derived Type

The following table describes the attributes for the **wireDetails** Derived Type. The wireDetails type is used for the wireDetails attribute only.

Attribute	Description	Type	Mandatory
addenda	Any addenda information included alongside the wire.	string	No
agentToAgentMsg	Final insitution to Financial Insitution Information message.	string	No
businessFunctionCode	Used for Fedwire in the US. Fed-defined 3-character code that identifies the business purpose of the funds transfer message. Example values include: • CTR (Customer Transfer) • CTP (Customer Transfer Plus) • BTR (Bank Transfer) • FFR (Fed Funds Return) • FFS (Fed Funds Sold) • CKS (Check Same Day Settlement Wire) • DEP (Deposit to Sender's Account) • DRW (Drawdown Response) • DRB (Bank to Bank Drawdown) • DRC (Customer or Corporate Drawdown) • SVC (Service Message)	string	No
debtorToCreditorMsg	Message from the originator to the beneficiary to convey details of the payments, such as including invoice numbers and amounts.	string	No
iMADInputCycleDate	Input cycle date extracted from the IMAD.	date	No
iMADInputSequenceNumber	Input sequence number (6 digits) extracted from the IMAD.	string	No
iMADInputSource	LTERM Id extracted from the IMAD.	string	No
oFACCheckCompletedFlag	Indicates whether OFAC checking was completed prior to sending the wire for fraud scanning.	string	No

Attribute	Description	Type	Mandatory
oMADOutputCycleDate	Output cycle date extracted from the OMAD.	date	No
oMADOutputDate	Output date extracted from the OMAD.	date	No
oMADOutputDestinationId	LTERM Id of the endpoint Id of the receiving financial institution extracted from the OMAD.	string	No
oMADOutputSequencer	Output sequence number (6 digits) extracted from the OMAD.	string	No
oMADOutputTime	Output time extracted from the OMAD.	string	No
supervisorOverrideFlag	Indicates if a supervisor overrode the wire status in order to release the wire with immediate effect.	boolean	No

9 Appendix B: Unique Identification Definitions

The following table describes examples of accountBranch, counterPartyId, accountBranchId and counterpartyBranchId for different geographies and payment rails.

Country/Region	Payment Rail	accountID	counterpartyId	accountBranchId and counterpartyBranchId
EU	SEPA	Full IBAN	Full IBAN	BIC11 (Business Identifier Codes) or Swift code (used to identify branch of institution).
UK	-	Sort code and account ID.	Sort code and account ID.	Sort code.
US	-	Routing number and bank account number.	Routing number and bank account number.	US Routing Number (ABA routing number).
Singapore	-	Bank, branch code, and bank account number.	Bank, branch code, and bank account number.	Singapore Bank & Branch Code.
Peru	-	20-digit Codigo Cuenta Interbancario (CCI) number 5.	20-digit Codigo Cuenta Interbancario (CCI) number 5.	First 8 digits of the Codigo Cuenta Interbancario (CCI).
Mexico	-	18-digit Clave Bancaria Estandarizada (CLABE) number 4.	18-digit Clave Bancaria Estandarizada (CLABE) number 4.	First 6 digits of the Clave Bancaria Estandarizada (CLABE).
Colombia	-	BIC8/11 and bank account number.	BIC8/11 and bank account number.	BIC11 (if available, otherwise BIC8), also known as SWIFT code.
Australia	-	Bank State Branch (BSB) and bank account number.	Bank State Branch (BSB) and bank account number.	Bank State Branch (BSB) code.
Argentina	-	22-digit Clave Bancaria Uniforme (CBU) number 1.	22-digit Clave Bancaria Uniforme (CBU) number 1.	First 7 digits of the Clave Bancaria Uniforme (CBU).

Note: For Alternative Payment Methods (APMs), the counterpartyId field can sometimes be populated with the email or phone number identifiers.

10 FAQs

Q. How do I see incidents that need reviewing?

When you first go to the Incidents page, it shows a list of all incidents visible to you that haven't yet been reviewed. You can choose to view only certain incidents, using filters. For more information, see the Fraud Transaction Monitoring Portal Guide.

Q. How do I see more information on an incident or alert?

The Incident Review page contains detailed information on:

- The event that triggered an alert
- The entity that event happened to
- Related user activity, including how previous alerts for the same entity were reviewed
- Notes and comments from analysts who reviewed previous alerts

For more information, see the Fraud Transaction Monitoring Portal Guide.

Q. How do I Integrate with the Scam Transaction Monitoring API?

First, you will need to contact @Thredd, who can provide client certificates and connection details as required.

For more information, see Integrate with Scam Transaction Monitoring API.

Q. What are the differences between the APIs?

There are three APIs used by Scam Transaction Monitoring:

- paymentRT, which sends real-time payments to receive a scam score in the synchronous payments.
- paymentNRT, which sends non-real-time payment decisioning and information only.
- paymentTransactionReturn, which sends informational events confirming that a scam or fraud has taken place.

Glossary

This page provides a list of glossary terms used in this guide.

A

Aggregator

An aggregator is a type of analytic that can combine and use the outputs of multiple rules and models to generate alerts.

Alert

The Fraud Transaction Monitoring System can flag up high-risk events for alert reviews. A flagged event is said to have generated an alert. The system's analytics rules, models and aggregators) can all generate alerts.

Alert Review

This is where analysts review alerts generated by the Thredd Fraud Transaction Monioring System. They can classify alerts as ‘Risk’ or ‘No Risk’, refer them to other users, or put them aside for further monitoring or to await additional information.

AMDL

AMDL (ARIC Modelling Data Language) is a language for specifying rules and logic within the Fraud Transaction Monitoring System. It is a declarative language for specifying state updates and executions on each event that passes through the system. An example of an event is an account registration or a transaction. Every event contains a reference (for example, an ID field) to one or more entities of different types, such as a merchant and a consumer. You can use AMDL to create Business Rules for the detection of fraud.

B

BIN Attack

An act of guessing an accurate combination of a debit or credit card number, Card Verification Value (CVV), and expiry date using brute-force computing. When this has been completed and the fraudster acquires the right information, they use the card to commit fraudulent transactions.

C

Chargeback

Where a cardholder disputes a transaction on their account and is unable to resolve directly with the merchant, they can raise a chargeback with their card issuer. The chargeback must be for a legitimate reason, such as goods and services not received, faulty goods, or a fraudulent transaction. For more information, see the Payments Dispute Management Guide.

E

Entity

Events happen to entities. An entity represents a unique individual or object, and every event is associated with at least one entity. For example, if a customer makes a card transaction, that event can be associated with the customer entity, the card entity, or both.

Entity ID

Each entity is identified by a unique entity ID in the event data for example, a 16-digit token.

Entity State

Every entity has a state – a combination of information about the entity that the system has accumulated over time. This is also called a behavioral profile. Every event processed by the system has the potential to update an entity’s state, adding more information or updating information that the system can use to build a behavioral profile of a customer or card for example.

Event

The Fraud Transaction Monitoring System recognizes potential fraud and financial crime by monitoring events. An event could be a customer transaction, a new customer application, or a merchant attempting to process a payment – these are all examples of event types. Each event is associated with one or more entities and one or more solutions.

I

Incident

In the Fraud Transaction Monitoring System, alerts are grouped into incidents. Each incident contains all the unreviewed alerts related to a particular entity.

Issuer (BIN sponsor)

The card issuer, typically a financial organisation authorised to issue cards. The issuer has a direct relationship with the relevant card scheme (payment network). For more information, see the Key Concepts Guide.

L

Label Events

Label events are types of event that contain ground truth information. They are used to label other events as 'risk' (i.e. confirmed fraud, financial crime, etc.) or 'no risk' (i.e. genuine). Alert reviews are one common form of label event, but your portal may also use other kinds of label event, such as chargebacks or manual fraud reports. Labels are used by Adaptive Behavioral Analytics models to learn to better identify high-risk events. They are also used to quantify and report on the performance of models.

M

Mastercard Fraud and Loss Database

A Mastercard repository of fraud transactions submitted by issuers. It is used for reporting, monitoring, and combating card fraud. Previously know as: System to Avoid Fraud Effectively (SAFE).

MasterCom API

MasterCom API offers Mastercard customers the ability to create and manage dispute claims in MasterCom. MasterCom is a system for dispute management. All activities for any given dispute can be tracked within a single claim using Mastercom, including Retrieval Request and Fulfilment, First Chargeback, Second Presentment, Fraud reporting, Case Filing, and Fee Collection requests. All activities for any given dispute throughout its lifecycle can be tracked within a single claim.

Model

A model in the Thredd Fraud Transaction Monitoring System is a predictive model that processes events and generates a risk score for certain event types, for example, authorisations.

P

PAN

The Primary Account Number (PAN) is the card identifier found on payment cards, such as credit/debit/prepaid cards, as well as stored-value cards, gift cards and other similar cards. The card’s 16-digit PAN is typically embossed on a physical card. For more information, see the Key Concepts Guide.

R

Real time/near-real time events

Every event is processed by analytics in the Featurespace fraud monitoring system engine. This processing happens in strict chronological order, so that no event is ever processed out of sequence. This is asynchronous processing, and happens to all events. However, some events, such as authorisations, require a real-time response (within a few hundredths of a second). These must be processed in a way that prioritizes low latency (such as a fast response), rather than chronological order. This kind of event is called a real-time event, and is processed by the portal synchronous response generator (as well as the portal Engine). Events that do not require a real-time response (asynchronous events), are only processed by the engine, for example, chargebacks, address or phone number updates

Rule

A rule defines some simple logic – rules take in information from events, entity states, and other data, and output a simple true/false response. Rules are written in the business logic definition language, AMDL.

Rule Set

Each Analytical Workflow is divided into a series of Rule Sets. Each Rule Set contains a number of expressions written in AMDL, and one or more Scorecards which contain conditions that determine what effects the Workflow triggers (e.g. generating an alert, adding a tag, outputting a risk score). Each Rule Set may also have a condition that determines whether or not that Rule Set is executed for an event.

S

Single Sign-On (SSO)

An identification method that enables users to log in to multiple applications and websites with one set of credentials.

Smart Client

Smart Client is Thredd's legacy desktop application for managing your account on the Thredd Platform.

Solution

Multiple product Solutions may be configured in your portal deployment. Each Solution provides a combination of UI configurations, data enrichment and analytics for detecting a specific type of risk. For example, you may have a Solution for application fraud and another for inbound/outbound payments, subject to your programs set up with Thredd and Featurespace. The same event may trigger separate alerts in different Solutions.

Solution ID

Each Solution is uniquely identified by a Solution ID in the event data.

Solution UI

The fraud system user interface that users access when they open the relevant Solution. The Solution UI is mainly used for reviewing incidents that are specific to that Solution, and can be customized for detecting the relevant type of financial risk.

State

Every entity has a state – a combination of information about the entity that the systems has accumulated over time. This is also called a behavioral profile. Every event processed by the system has the potential to update an entity’s state, adding more information or updating information that the system can use to build a behavioral profile of a customer or card for example.

T

Tag

Rules, aggregators and models can add tags to alerts, to give analysts more information or to automate a response in a downstream system, such as declining a transaction.

Thredd Portal

Thredd Portal is Thredd's new web application for managing your cards and transactions on the Thredd Platform.

Token

Displays the unique token linked to the card PAN on which the transaction was made.

Document History

Version	Date	Description	Revised by
1.0	26/03/2025	Added Scam Transaction Monitoring Webhook Notification section to Managing Scams.	JB
	11/02/2025	Added references to Thredd Portal, our new web application for managing your cards and transactions.	JB
	11/11/2024	Corrected the types for several fields for the wireDetails derived type. See Appendix A: Derived Types.	JB
	17/10/2024	First version.	JB

Contact Us

Please contact us if you have queries relating to this document. Our contact details are provided below.

Thredd UK Ltd.

Company registration number 09926803

Support Email: occ@thredd.com

Telephone: +44 (0) 203 740 9682

Our Head Office

Kingsbourne House
229-231 High Holborn
London
WC1V 7DA

Technical Publications

If you want to contact our technical publications team directly, for queries or feedback related to this guide, you can email us at: docs@thredd.com.