cybersource
A Visa Solution

Network Tokenization

Direct Model Integration Guidance

cybersource
A Visa Solution

Contents

1. Introduction

Summary

This is a help and guidance document for merchants wishing to enroll for Network Tokenization for one or more card networks such as the Visa Token Service, Mastercard Digital Enablement Service or American Express Tokenization service.

This guide is designed to support merchants understanding the requirements for Network Tokenization, the process of enrolling a TMS token for Network Tokenization and supporting the different use cases that exist when managing Network Tokens under Token Management Service.

Audience and Purpose

The intended audience for this document is merchants who are currently using the Token Management Service and wish to utilize the Network Tokenization service that this offers. This specific guide is designed for merchants on the Direct Network Tokenization model who are also using Cybersource as a gateway for payment processing. For more information on Token Management Service and its other tokenization features, please see the Token Management Service Developer Guide.

Pre-requisites

In order for merchants to be enabled for Network Tokenization, they must be configured for Token Management Service and have an understanding of the Token Management Service functionalities.

2. Network Tokenization

What is a Network Token?

A Network Token is network scheme generated token, that represents customer card information for secure transactions that references a customer's actual PAN. The Network Token identifier will be provisioned by the card network and provided to the merchant to use in place of the customers Payment Account Number. When processing an authorization using a Network Token, the scheme generated token will be used in place of the customer's actual PAN which only the card scheme (VISA, Mastercard, Amex) will have access to. To find more information on Network Tokens work, please see the *Arrival of Network tokens* chapter of our Cybersource Network Tokenization guide.

Why use Network Tokens?

Using Network Tokens has a number of benefits, including more secure transactions, improved authorization rates and also lower interchange fees in certain regions. Using a Network Token is more secure than sending the customers PAN details, since the merchant nor Payment Service Provider are exposed to the sensitive customer information that exists when processing an Authorization. Only the card network that provisioned the Network Token will have insight into the actual customers PAN, reducing the risk of this data being exposed elsewhere. For more detailed information on the benefits of using a Network Token, please see the *Why Tokenization* chapter of our Cybersource Network Tokenization guide.

3. Network Tokenization Enablement

Process of enabling a MID for Network Tokenization

Once a MID has been successfully configured for Token Management Service as outlined in the Prerequisites chapter of this document, merchants can then request a Network Tokenization Enablement for one or more MIDs through their Account Managers or through their normal support channels.

Enabling a MID for Network Tokenization is a 2-step process, with each step outlined as below.

Enrolling for a Network Tokenization TRID (Token Requestor ID) for each card scheme

Before a MID can be enabled for Network Tokenization, it must have a Token Requestor ID provisioned for each card scheme that the MID will be enabled for.

A Token Requestor ID or TRID is a unique identifier that will tie each MID to a record on the card scheme's side used to identify which entity is using a Network Token. Each Network Tokenization request will use the TRID provisioned by the card scheme to help authenticate the owner of the Network Token and the merchant account that is requesting the use of this credential.

Enrolling for a Token Requestor ID

To begin the process of Network Tokenization enablement, please contact your Account Manager or Cybersource support representative to request a TRID provision and Network Token enablement.

The TRID enrollment will be done at a TMS vault level, so please indicate the specific TMS vault that you would like to provision a Token Requestor ID for.

Requesting a Network Token enablement

Once a Token Requestor ID has been successfully provisioned for a TMS vault, merchants can then request a Network Token enablement using the TRID that was returned. The TRID provisioning and Network Token enablement can be requested together if the intention is to immediately go live with Network Tokenization provisioning or transacting.

4. Provisioning a Network Token

Once a MID has been successfully enabled for Network Tokenization, merchants can then begin provisioning Network Tokens for their existing Token Management Service tokens or begin creating new Token Management Service tokens which will be auto enrolled for a Network Token.

Depending on the merchants existing token store and use case, the Network Token provisioning will fall into one of the below three scenarios.

Provisioning a Network Token when creating a new Token Management Service token

When creating a Token Management Service token that will contain a new Instrument Identifier, merchants can request that this Instrument Identifier be enrolled for a Network Token at the same time the Instrument Identifier is created.

Using the sample request: *"Create Instrument Identifier (Card & Enroll for Network Token)"*

URL: https://apitest.cybersource.com/tms/v1/instrumentidentifiers

Method: POST

Request:

```
{
  "type": "enrollable card",
  "card": {
    "number": "489537XXXXXX5029",
    "expirationMonth": "12",
    "expirationYear": "2030"
  }
}
```

Including the above mandatory request body contents, such as the enrollable card element along with the customer PAN, expiration data and security code.

Response:

A successful provision request on the above endpoint with the required body contents will return a HTTP **200** response to confirm the request has been fulfilled.

For a full list of the expected responses and API data, please see the Cybersource Developer Center for this endpoint

Test PANs for simulating Network Token Provisioning in CAS

In the Cybersource CAS environment, all Network Token provisions and events are simulated to give an indication of the available API responses that will be seen in production. To simulate the available Network Token responses and events, please use the values in the below table.

Successful Provisioning:

Card Scheme	PAN	Expiration Date	Card Verification Value
VISA	FOUR622943123037161	12/25	319
VISA	FOUR622943123037179	12/25	626
VISA	FOUR622943123037187	12/25	292
VISA	FOUR622943123037195	12/25	972
MASTERCARD	FIVE204245750003216	ANY	ANY
MASTERCARD	FIVE204245750003224	ANY	ANY
MASTERCARD	FIVE204245750003232	ANY	ANY
MASTERCARD	FIVE204245750003240	ANY	ANY
AMEX	THREE78282246310005	ANY	ANY

For additional Network Token provisioning test cards, please contact your Cybersource representative.

Provisioning a Network Token for an existing Token Management Service token

Merchants can provision a Network Token for an existing TMS token using the following API endpoint, passing the Instrument Identifier element of the existing token.

Using the sample request *"Enroll an Instrument Identifier for a Payment Network Token":*

URL: https://apitest.cybersource.com/tms/v1/instrumentidentifiers/{instrumentIdentifierId}/enrollment

Method: POST

Appending the Instrument Identifier value to the above URL and including the required request body contents as below:

Request:

```
{
  "type": "enrollable card",
  "card": {
    "expirationMonth": "12",
    "expirationYear": "2031"
  }
}
```

Response:

A successful provision request on the above endpoint with the required body contents will return a HTTP 204 response to confirm the request has been fulfilled but does not require a return body.

For a full list of the expected responses and API data, please see the Cybersource Developer Center for this endpoint.

Provisioning a Network Token whilst processing an Authorization + Creating a new TMS token

Merchants can also request the provision of a Network Token whilst processing an Authorization if the Authorization being processed is combined with a new token creation.

Using the sample request *"Authorization with Customer Token Creation"*

URL: https://apitest.cybersource.com/pts/v2/payments

Method: POST

Request:

```
{
  "processingInformation": {
    "actionList": [
      "TOKEN_CREATE"
    ],
    "actionTokenTypes": [
          "customer",
      "paymentInstrument",
      "shippingAddress"
    ],
    "commerceIndicator": "internet"
  },
  "paymentInformation": {
    "card": {
      "number": "41111111XXXX1111",
      "expirationMonth": "12",
      "expirationYear": "2031",
      "securityCode": "123"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "100.00",
      "currency": "USD"
    },
  }
}
```

Response:


```
  {
  "clientReferenceInformation": {
    "code": "MerchantRefCode"
  },
  "id": "680617146XXXXXXXX04953",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "100.00",
      "currency": "USD"
    }
  },
,
  "paymentInformation": {
    "tokenizedCard": {
      "type": "001"
    },
  },
  "reconciliationId": "XXXXXXXXXXXXXXXXX",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2025-01-01T14:05:46Z",
  "tokenInformation": {
    "instrumentidentifierNew": true,
    "instrumentIdentifier": {
      "state": "ACTIVE",
      "id": "701XXXXXXXX16241111"
    }
  }
}
```


A successful provision request on the above endpoint with the required body contents will return a HTTP 200 response to confirm the request has been fulfilled.

Failure responses when provisioning a Network Token

When a Network Token provision fails for one or more reasons, the Cybersource API endpoint will respond with the tokenized card state of *UNPROVISIONED* and one of the below reason descriptions:

Failure Response	Reason
INVALID_REQUEST	The Network Token provision request contained invalid data.
CARD_VERIFICATION_FAILED	Card information could not be verified with the issuing bank
CARD_NOT_ELIGIBLE	This card currently cannot be used for Network Tokenization.
CARD_NOT_ALLOWED	This card currently cannot be used for Network Tokenization.
DECLINED	The Issuer does not support Network Tokenization for this PAN.
SYSTEM_ERROR	Potential configuration issue with the merchant account.
SERVICE_UNAVAILABLE	The card network encountered an unexpected error.

5. Retrieving Network Token information

Retrieving Network Token information via API

Once a Network Token has been provisioned for an Instrument Identifier, merchants can validate the Network Token has been successfully provisioned and retrieve its associated information by doing a TMS token retrieval.

Using the sample request *"Retrieve an Instrument Identifier"*

URL: https://apitest.cybersource.com/tms/v1/instrumentidentifiers/{instrumentIdentifierId}

Method: GET

Request:

Appending the Instrument Identifier value to the end of the above URL and submitting the API request will return all information for the Instrument Identifier as below.

Response:


```
"id": "703000000XXXX623232",
"object": "instrumentIdentifier",
"state": "ACTIVE",
"tokenizedCard": {
  "state": "ACTIVE",
  "number": "520424XXXXXX7686",
  "expirationMonth": "06",
  "expirationYear": "2025",
  "type": "mastercard",
  "card": {
    "suffix": "3232",
    "expirationMonth": "12",
    "expirationYear": "2025"
  }
},
"card": {
  "number": "520424XXXXXX3232"
},
"issuer": {
  "paymentAccountReference":
"50013DZ8M2A454Y5OXXXXXX44BHP1"
}
```

The *"tokenizedCard"* element returned in the API response will contain all the provisioned information for an existing Network Token such as the *number*, *expiration data* and *card type*. For Instrument Identifiers that do not return a *"tokenizedCard"* element referenced above, this means there is no existing Network Token provisioned for that Instrument Identifier.

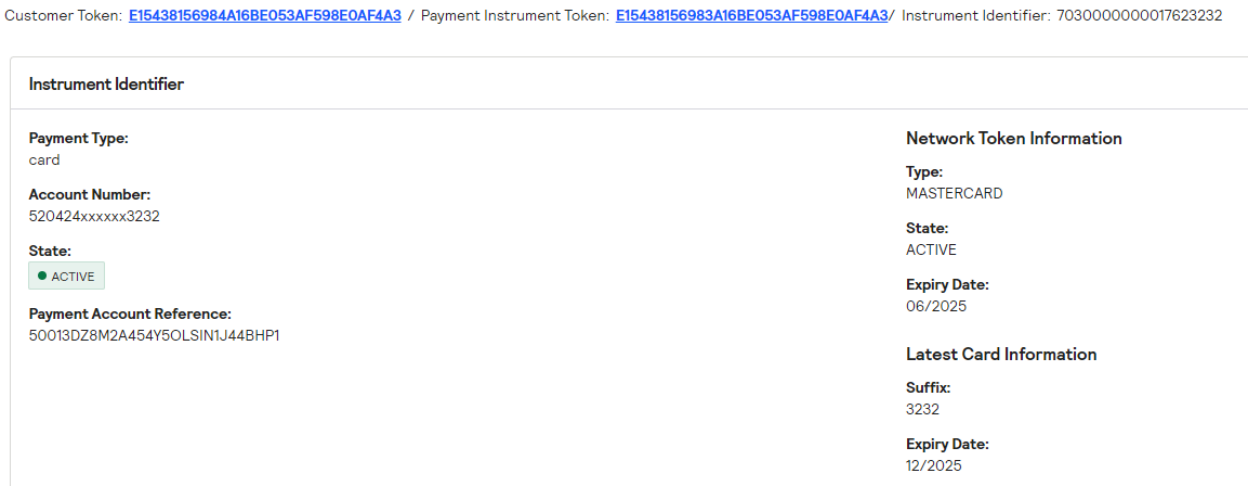
The Instrument Identifier and Network Token information will also be returned when retrieving a Customer, Customer Payment Instrument or Payment Instrument token.

For a full list of the expect responses and API data, please see the Cybersource Developer center for this API endpoint.

Validating Network Token information via EBC

As well as retrieving the Network Token information via API, merchants can also log into the Enterprise Business Center to validate the Network Token information from the front-end UI. This can be done by logging into EBC and navigating to Token Management > Customers, then searching for the Instrument Identifier which the Network Token was provisioned for.

Example:



NOTE: The full Network Token PAN cannot be retrieved from the EBC UI, this must be retrieved when doing an Instrument Identifier retrieval via API as explained in the previous section of this document.

6. Webhooks and Lifecycle Management updates

Introduction to Webhooks and Lifecycle Management

Merchants utilizing Network Tokenization can subscribe to Webhook notifications to keep up to date with the Lifecycle Management (LCM) updates that occur naturally as part of the Network Tokenization service. One of the key benefits of Network Tokenization is that customer stored credentials can be automatically updated when a customer PAN expires or when the card scheme has updated card information for a particular customer.

The two existing Lifecycle Management updates supported by Visa Acceptance Solutions and Issuers who are participating in Lifecycle Management are the *"Token Provisioned"* event confirming a Network Token has been successfully provisioned for an Instrument Identifier and the *"Token Updated"* event confirming a card scheme has updated information for an existing Network Token.

Creating a Webhook Subscription

The first step in creating a webhook subscription will be to create Webhook "Security Keys" that will be used to sign the payloads sent containing the Network Token event updates. These same security keys can be used to validate the signature of the notification which is an optional step when receiving webhooks from Visa Acceptance Solutions. Validating the signature of the payload can be done to ensure the notification being received is from Visa Acceptance Solutions and not a malicious actor.

Whitelisting required to receive Webhook Notifications

Before receiving webhook notifications, clients must ensure the following 2 IP addressed have been whitelisted in their application server before Token Management Service will be authorized to send notification updates:

- 198.241.206.21
- 198.241.207.21

Creating Webhook Security Keys

Using the sample request: Create Webhook Symmetry Key

URL: https://apitest.cybersource.com/kms/egress/v2/keys-sym

Method: POST

Request:

```
{
  "clientRequestAction": "CREATE",
  "keyInformation": {
    "provider": "nrtd",
    "tenant": "MerchantAccount",
    "keyType": "sharedSecret",
    "organizationId": "MerchantAccount"
  }
}
```

Replacing the *"MerchantAccount"* variable in both the *"tenant"* and *"organizationId"* fields with the Merchant Account Organization ID boarded will configure the webhook events to be sent for this Merchant.

Response:


```
{
  "submitTimeUtc": "2023-11-17T12:01:10Z",
  "status": "SUCCESS",
  "keyInformation": {
    "provider": "nrtd",
    "tenant": "MerchantAccount",
    "organizationId": "MerchantAccount",
    "keyId": "0a58e467-xxxx -4a5b-xxxx-5c588e0a5430",
    "key": "9L1MJfwxZZZPgTvRSrYk18E6eng82JPkxeWC5nyGC9c=",
    "keyType": "sharedSecret",
    "status": "active",
    "expirationDate": "2026-11-16T12:01:10Z"
  }
}
```


When valid organization information is submitted in the Create Webhook Keys API call, Visa Acceptance Solutions will respond with the **SUCCESS** status including the Key ID and Key Secret value that can be used to validate the signature of future webhook notifications that are received. Please ensure to save these keys if validating the signature of the notification is a step that you would like to complete.

For detailed guidance on how to validate the signature of a Webhook Notification, please see our Notification Validation Developer guide.

Finding Products to Subscribe to

Once valid webhook keys have been created for an organization, merchants can then begin subscribing to specific product events, one of the options being updates for the Token Management Service and its supported Network Token notifications.

Merchants can find the list of available products to subscribe to, using the below API call and examples.

Using the Sample Request: Find Products you can Subscribe to

URL: https://apitest.cybersource.com/notification-subscriptions/v1/products/{organizationId}

Method: GET

Request:

```
This request has no body contents.
```

Appending the *organizationId* that is provisioning the Network Token to the end of the above URL and submitting the API call will return the list of available Products that can be subscribed to.

Response


```
[
  {
    "productId": "tokenManagement",
    "eventTypes": [
      {
        "eventName": "tms.token.created",
        "payloadEncryption": false
      },
      {
        "eventName": "tms.token.updated",
        "payloadEncryption": false
      },
      {
        "eventName": "tms.token.pan_updated",
        "payloadEncryption": false
      },
      {
        "eventName": "tms.networktoken.updated",
        "payloadEncryption": false
      },
      {
        "eventName": "tms.networktoken.provisioned",
        "payloadEncryption": false
      }
]
```


When Token Management Service is enabled for an organization and a valid organization ID is provided in the request URL, the above response will indicate that Token Management Service events are available to be subscribed to.

Subscribing to Token Management Service events

Once Token Management Service is returned as a product that can be subscribed to, merchants can then create a subscription for this product to begin receiving the available notifications.

Using the Request Sample: Create a Token Management Webhook

URL: https://apitest.cybersource.com/notification-subscriptions/v1/webhooks

Method: POST

Request:

```
{
  "name": "Custom Webhook Subscription 1",
  "description": "Sample Webhook for Testing",
  "organizationId": "organizationId",
  "productId": "tokenManagement",
  "eventTypes": [
    "tms.networktoken.provisioned",
    "tms.networktoken.updated",
    "tms.token.pan_updated",
    "tms.token.created",
    "tms.token.updated"
  ],
  "webhookUrl": "https://ClientWebhookURL.com/test",
  "healthCheckUrl": " https://ClientWebhookURL.com/test",
  "notificationScope": "DESCENDANTS",
  "retryPolicy": {
    "algorithm": "ARITHMETIC",
    "firstRetry": 1,
    "interval": 1,
    "numberOfRetries": 3,
    "deactivateFlag": "false",
    "repeatSequenceCount": 0,
    "repeatSequenceWaitTime": 0
  },
  "securityPolicy": {
    "securityType": "KEY",
    "proxyType": "external"
  }
}
```

In the above example, a new webhook is being created which can be given a name and description for identification. Clients will need to provide a "webhookURL" which will be destination URL where the notification is sent, as well as a "healthCheckURL", which the webhook service will send a test notification to every 5 minutes to validate the endpoint is available and can be accessed.
If the webhook service cannot connect to that endpoint, the webhook status will be updated to SUSPENDED indicating the endpoint cannot be used for webhook notifications.

NOTE: When creating a Webhook Subscription at the Merchant Account level, please ensure the "notificationScope" value is set to "**DESCENDANTS**". This will ensure that notifications are processed for all Transacting MIDs beneath the Merchant Account level organization.

Response:

```
{
  "organizationId": " organizationId ",
  "productId": "tokenManagement",
  "eventTypes": [
    "tms.networktoken.provisioned",
    "tms.networktoken.updated",
    "tms.token.pan_updated",
    "tms.token.created",
    "tms.token.updated"
  ],
  "webhookId": "0a5d16b3-5ec8-XXXX-XXXX-a0588e0abdc0",
  "name": "Custom Webhook Subscription 1",
  "webhookUrl": " https://ClientWebhookURL.com:443/test",
  "healthCheckUrl": " https://ClientWebhookURL.com:443/test",
  "createdOn": "2023-11-17T17:01:34.373Z",
  "status": "ACTIVE",
  "description": "Sample Webhook from Developer Center",
  "retryPolicy": {
    "algorithm": "ARITHMETIC",
    "firstRetry": 1,
    "interval": 1,
    "numberOfRetries": 3,
    "deactivateFlag": false,
    "repeatSequenceCount": 0,
    "repeatSequenceWaitTime": 0
  },
  "securityPolicy": {
    "securityType": "KEY",
    "proxyType": "external",
    "digitalSignatureEnabled": "yes"
  },
  "version": "3",
  "deliveryType": "nrtdCentral",
  "notificationScope": "DESCENDANTS"
}
```

In the above example response, a Webhook ID has been returned which can be used to retrieve the details of a created webhook, as well as the webhook status along with the product events that have been subscribed to. The status of the created webhook has also been returned as ACTIVE, indicating that the healthcheck URL can be reached and now used for future webhook notifications.

Retrieving the information of a valid Webhook

Once a Webhook has been successfully created, clients can retrieve the details of a webhook to confirm its status, URL and the list of product events that have been subscribed to.

Using the sample request: Get Details on a Single Webhook

URL: https://apitest.cybersource.com/notification-subscriptions/v1/webhooks/{webhookId}

METHOD: GET

Request:

```
This request has no body contents.
```

Appending the Webhook ID that was returned in the "Create a Webhook" API call to the end of the above URL and submitting the API call will return all the information for the webhook subscription.

Response:


```
{
  "organizationId": "organizationID",
  "productId": "tokenManagement",
  "eventTypes": [
    "tms.networktoken.provisioned",
    "tms.networktoken.updated",
    "tms.token.pan_updated",
    "tms.token.created",
    "tms.token.updated"
  ],
  "webhookId": "0a5d16b3-5ec8-XXXX-XXXX-a0588e0abdc0",
  "webhookUrl": "https://ClientWebhookURL.com:443/test",
  "healthCheckUrl": " https://ClientWebhookURL.com:443/test",
  "createdOn": "2023-11-17T17:01:34.376",
  "status": "ACTIVE",
  "retryPolicy": {
    "algorithm": "ARITHMETIC",
    "firstRetry": 1,
    "interval": 1,
    "numberOfRetries": 3,
    "deactivateFlag": false,
    "repeatSequenceCount": 0,
    "repeatSequenceWaitTime": 0
  },
  "securityPolicy": {
    "securityType": "KEY",
    "proxyType": "external",
    "digitalSignatureEnabled": "yes"
  },
  "version": "3",
  "deliveryType": "nrtdCentral",
  "notificationScope": "DESCENDANTS"
}
```

Webhook Notification Examples

Webhook Health-Check Notification

The below shows an example of the Webhook Health Check notification that is sent to the clients server every 5 minutes to validate the endpoint is up and can be accessed.

```
GET /test?healthCheck=true HTTP/1.1
Host: ClientWebhookURL.com
Content-Type: application/json
User-Agent: Vert.x-WebClient/4.4.4
V-C-Event-Type: ping
V-C-Organization-Id: organizationId
V-C-Product-Name: tokenManagement
V-C-Retry-Count: 0
V-C-Transaction-Trace-Id: 03c7b4e6-666b-4355-ba26-f943c2648b76
V-C-Webhook-Id: 0abe69c5-XXXX-5032-XXXX-a2588e0a7dd5
```

Network Token Provisioned:

The below shows an example of the Webhook Notification that is triggered when there is a new successful Network Token provisioned for a TMS token.


```
{
  "eventType": "tms.networktoken.provisioned",
  "webhookId": "0abe69c5-XXXX-5032-XXXX-a2588e0a7dd5",
  "productId": "tokenManagement",
  "organizationId": "organizationId",
  "eventDate": "2023-11-22T14:19:42",
  "transactionTraceId": "5c9d2640-XXXX-4086-XXXX-e817054c4c78-0",
  "retryNumber": 0,
  "payload": {
    "data": {
      "id": "7030200XXXXXX174957",
      "type": "tokenizedCardEnrollments",
      "version": "1.0",
      "_links": {
        "instrumentIdentifiers": [
          {
            "href": "/tms/v1/instrumentidentifiers/7030200XXXXXX174957"
          }
        ]
      }
    },
    "organizationId": "organizationId"
  },
  "requestType": "NEW"
}
```

Network Token Updated:

The below shows an example of the Webhook Notification that is triggered when there is an update to an existing Network Token and TMS Token. This update can occur when a customer's PAN behind a Network Token has changed, and the issuer has generated a new Network Token to be used.


```
{
  "eventType": "tms.networktoken.updated",
  "webhookId": "0abe69c5-XXXX-5032-XXXX-a2588e0a7dd5",
  "productId": "tokenManagement",
  "organizationId": "organizationID",
  "eventDate": "2023-11-22T14:19:42",
  "transactionTraceId": "5c9d2640-XXXX-4086-XXXX-e817054c4c78-0",
  "retryNumber": 0,
  "payload": {
    "data": {
      "id": "7030200XXXXXX174957",
      "type": "tokenizedCardUpdates",
      "version": "1.0",
      "_links": {
        "instrumentIdentifiers": [
          {
            "href": "/tms/v1/instrumentidentifiers/7030200XXXXXX174957"
          }
        ]
      }
    },
    "organizationId": "organizationId"
  },
  "requestType": "NEW"
}
```

7. Processing an Authorization using a Network Token

Using a Token Management Service token that contains a Network Token

Once a Network Token has been provisioned for a Token Management Service token, merchants can then begin processing authorizations that use the Network Token instead of the customers PAN. When a Token Management Service token is used in an authorization, Cybersource will automatically present the Network Token to be used for payment without merchants having to specify this in the request message.

The below example contains an Authorization request on the Cybersource *PAYMENTS* endpoint, where a Token Management Service token is presented containing an Instrument Identifier with a Network Token provisioned.

Using the sample request: *"Authorization with Customer Token Creation"*

URL: https://apitest.cybersource.com/pts/v2/payments

Method: POST

Request:

```
{
  "clientReferenceInformation": {
    "code": "Network Token Authorization"
  },
  "paymentInformation": {
    "customer": {
      "id": "E15438156984A16BE053XXXXXX0AF4A3"
    }
  },
  "orderInformation": {
    "amountDetails": {
      "totalAmount": "100.00",
      "currency": "USD"
    }
  }
```

Response:


```
{
,
  "clientReferenceInformation": {
    "code": "Network Token Authorization"
  }
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "002"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "002"
    },
    "card": {
      "type": "002"
    },
    "customer": {
      "id": "E15438156984A16BE053XXXXXX0AF4A3"
    }
  },
  "processingInformation": {
    "paymentSolution": "014"
  }
```

In the above API response example, Merchants can validate a Network Token was used for the transaction using the payment solution value of **014** for MDES transactions, **015** for VTS transactions or **016** for AETS transactions.

Each scheme's Network Token option will return a unique value in this element of the API response to confirm the Network Token scheme which was used.

Bypassing the use of a Network Token for a specific transaction

Merchants wanting to opt-out of the use of a Network Token for a specific transaction can include the below element in the request body contents, to bypass the use of the Network Token and default to using the TMS token that exists for a particular customer.

```
"tokenInformation": {
  "networkTokenOption": "ignore"
}
```

When the *networkTokenOption* value is *ignore*, the customer's PAN will be used in the authorization lifecycle even when there is a Network Token associated. When the *networkTokenOption* is not supplied, or set to *prefer* in the below example, a Network Token will be used for the transaction if one exists for the customer credential.

```
"tokenInformation": {
  "networkTokenOption": "prefer"
}
```