Credentialed Transactions

REST API

Visa Platform Connect

cybersource

A Visa Solution

Developer Guide

Demo Draft

Cybersource Contact Information

For general information about our company, products, and services, go to https://www.cybersource.com.

For sales questions about any Cybersource service, email sales@cybersource.com or call 650-432-7350 or 888-330-2300 (toll free in the United States).

For support information about any Cybersource service, visit the Support Center: https://www.cybersource.com/support

Copyright

Restricted Rights Legends

For Government or defense agencies: Use,duplication, or disclosure by the Government or defense agencies is subject to restrictions as set forth the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 and in similar clauses in the FAR and NASA FAR Supplement.

For civilian agencies: Use, reproduction, or disclosure is subject to restrictions set forth in suparagraphs (a) through (d) of the Commercial Computer Software Restricted Rights clause at 52.227-19 and the limitations set forth in Cybersource Corporation's standard commercial agreement for this software. Unpublished rights reserved under the copyright laws of the United States.

Trademarks

Authorize.Net, eCheck.Net, and The Power of Payment are registered trademarks of Cybersource Corporation. Cybersource, Cybersource Payment Manager, Cybersource Risk Manager, Cybersource Decision Manager, and Cybersource Connect are trademarks and/or service marks of Cybersource Corporation. Visa, Visa International, Cybersource, the Visa logo, and the Cybersource logo are the registered trademarks of Visa International in the United States and other countries. All other trademarks, service marks, registered marks, or registered service marks are the property of their respective owners.

Confidentiality Notice

This document is furnished to you solely in your capacity as a client of Cybersource and as a participant in the Visa payments system.

By accepting this document, you acknowledge that the information contained herein (the "Information") is confidential and subject to the confidentiality restrictions contained in Visa's operating regulations and/or other confidentiality agreements, which limity our use of the Information. You agree to keep the Information confidential and not to use the Information for any purpose other than its intended purpose and in your capacity as a customer of Cybersource or as a participant in the Visa payments system. The Information may only be disseminated within your organization on a need-to-know basis to enable your participation in the Visa payments system. Please be advised that the Information may constitute material non-public information under U.S. federal securities laws and that purchasing or selling securities of Visa Inc. while being aware of material non-public information would constitute a violation of applicable U.S. federal securities laws.

Revision

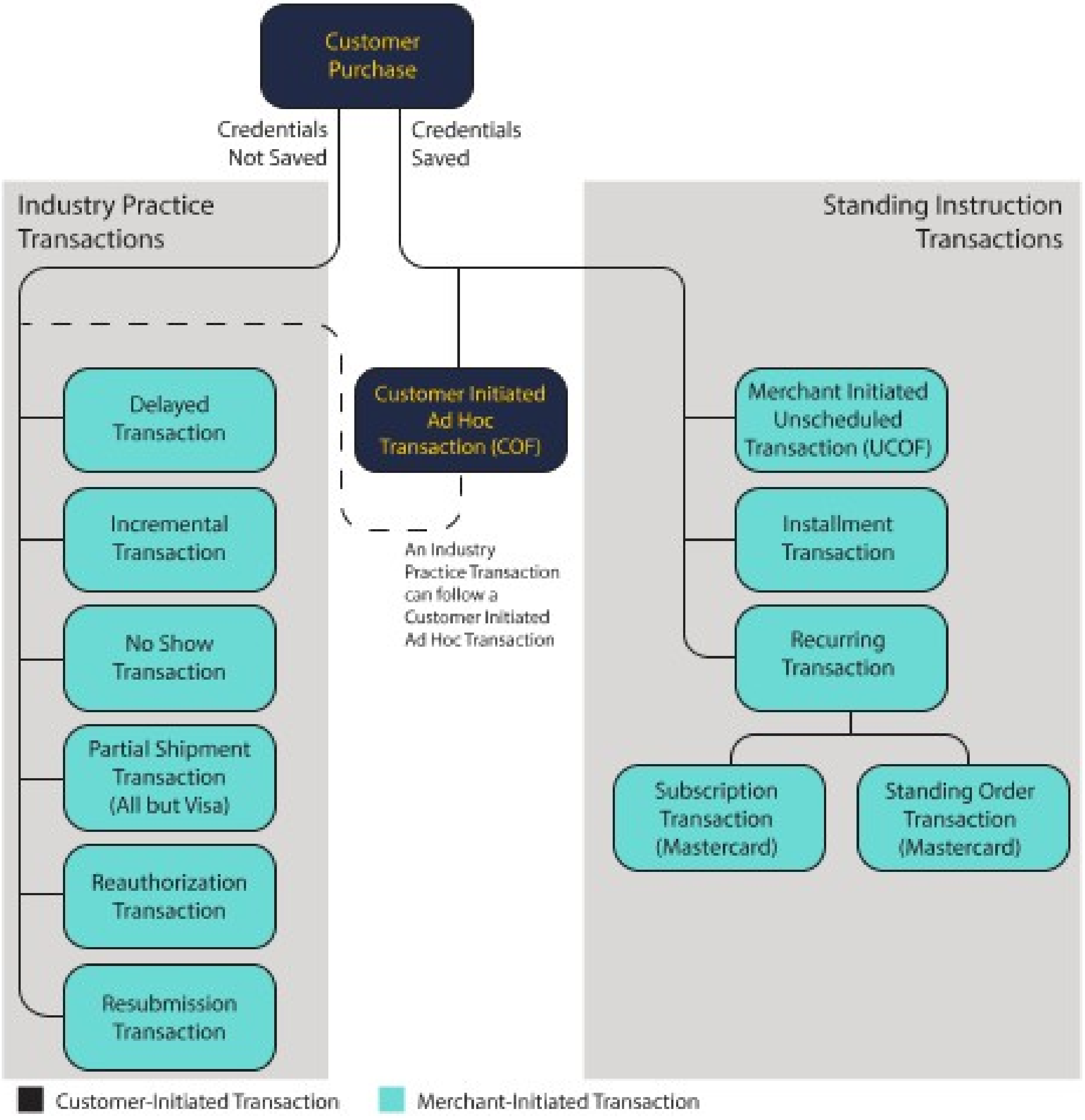
Version: 22.01

Contents

Stored Credential Processing

There are several types of credentialed transactions:

- Customer-Initiated Transactions: Ad hoc customer-initiated transactions that use stored credentials.
- Merchant-Initiated Transactions:
    - Industry Practice Transactions: Merchant-initiated transactions that do not require stored credentials.
    - Standing Instruction Transactions: Merchant-initiated transactions that require stored credentials and customer consent to execute.


Customer Purchase
Credentials Not Saved
Credentials Saved
Industry Practice Transactions
Standing Instruction Transactions
Delayed Transaction
Incremental Transaction
No Show Transaction
Partial Shipment Transaction (All but Visa)
Reauthorization Transaction
Resubmission Transaction
Customer Initiated Ad Hoc Transaction (COF)
An Industry Practice Transaction can follow a Customer Initiated Ad Hoc Transaction
Merchant Initiated Unscheduled Transaction (UCOF)
Installment Transaction
Recurring Transaction
Subscription Transaction (Mastercard)
Standing Order Transaction (Mastercard)
Customer-Initiated Transaction
Merchant-Initiated Transaction

Industry Practice Transactions

Industry practice transactions do not require stored customer credentials in order to execute. The transaction ID of the original transaction is used to track these follow-on transactions. Industry practice transaction types are:

- Delayed transactions
- Incremental transactions
- No-show transactions
- Partial shipment transactions

- Reauthorization transactions
- Resubmission transactions

Standing Instructions Transactions

Standing instructions require stored credentials. The stored credentials framework sets standards for the storage and subsequent use of stored customer credentials by merchants. Transactions that use stored customer credentials are called credentials-on-file (COF) transactions.

By following the stored credentials framework, merchants are expected to:

- Gain greater visibility of transaction risk levels
- Achieve higher authorization approval rates and completed sales
- Improve the customer experience
- Reduce customer complaints
- Participate in the Real Time Visa Account Updater service

For more information on the stored credentials framework, see *Improving Authorization Management for Transactions with Stored Credentials*.

Standing instructions include these types of transactions:

- Merchant initiated unscheduled transactions (UCOF)
- Installment transactions
- Recurring transactions
- Subscription transactions (Mastercard only)
- Standing order transactions (Mastercard only)

All COF transactions start with customer-initiated transactions during which customers elect to store their credentials for future use.

Requirements for Standing Instruction Transactions

Merchants who offer stored credentials must:

- Disclose to cardholders how those credentials will be used
- Obtain the customer's consent to store credentials
- Notify cardholders when changes are made to stored credential terms of use
- Inform the card issuer during an authorization that the payment credentials are now stored on file
- Identify transactions that use stored credentials

Recurring Billing for Recurring Payments

If you are using the Cybersource Recurring Billing service, do not use this document. Cybersource saves and stores payment credentials for recurring transactions, ensuring compliance with COF best practices.

For more information on Recurring Billing, see *Recurring Billing*.

Transaction Matrix for Industry Practice and Standing Instructions

This section provides the various field values used to configure the different initial and follow-on transactions available.

Initial Transactions

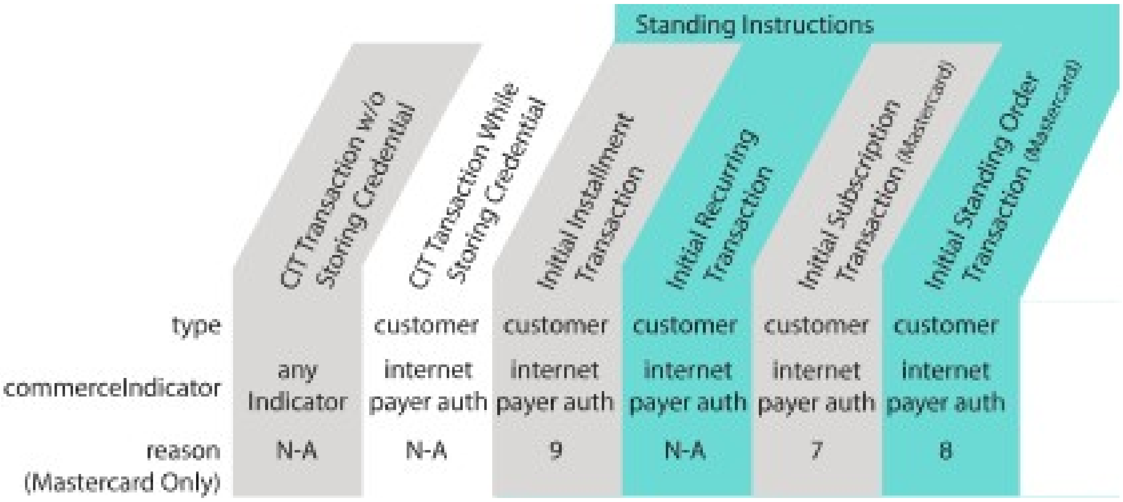
When you process an initial transaction, Mastercard requires a reason code. When you process subsequent transactions, all card types require a reason code.

For an initial transaction, include these fields with a standard authorization request:

```
"processingInformation": {
  "authorizationOptions": {
    "initiator": {
      "type": "customer",
      "merchantInitiatedTransaction": {
        "reason": "7" // Mastercard only
      }
    }
  },
  "commerceIndicator": "internet"
}
```

When you receive the initial transaction response, save the transaction identifier.

You need the transaction identifier for subsequent transactions. If you're using token management service (TMS), the transaction identifier will be stored on your behalf.

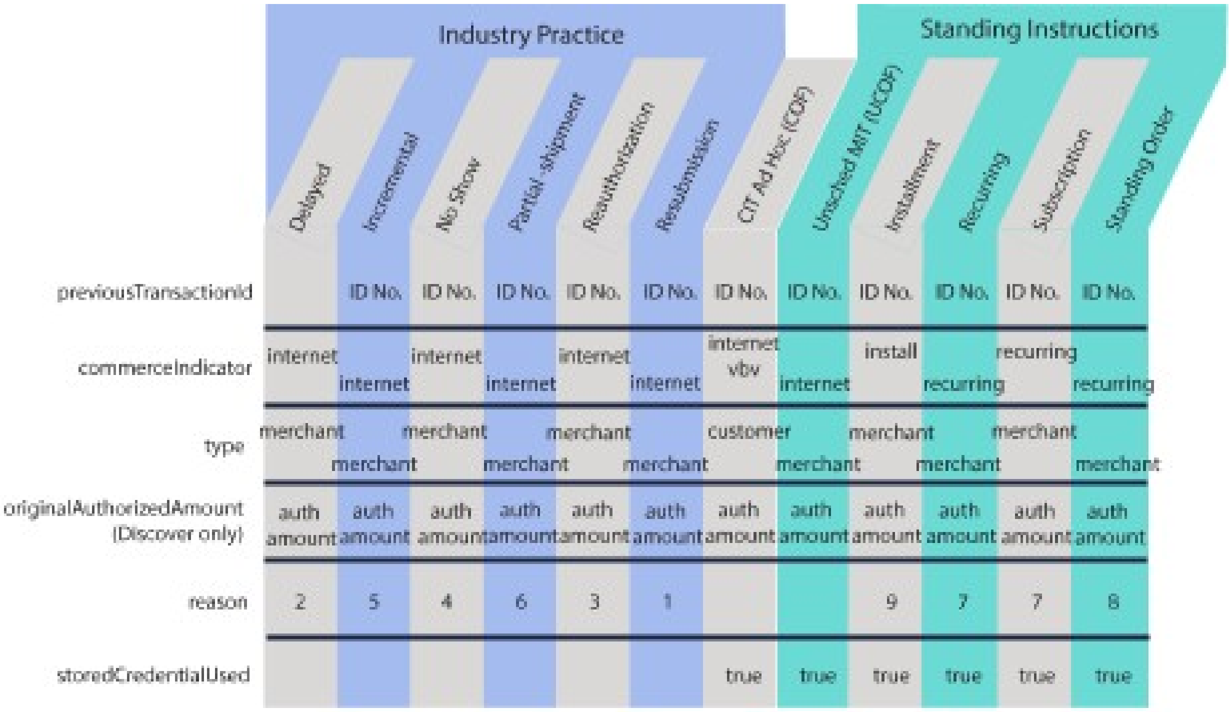


			Standing Instructions			
	CIT Transaction w/o Storing Credential	CIT Tansaction While Storing Credential	Initial Installment Transaction	Initial Recurring Transaction	Initial Subscription Transaction (Mastercard)	Initial Standing Order Transaction (Mastercard)
type		customer	customer	customer	customer	customer
commerceIndicator	any Indicator	internet payer auth	internet payer auth	internet payer auth	internet payer auth	internet payer auth
reason (Mastercard Only)	N-A	N-A	9	N-A	7	8

Subsequent Transactions

When you process subsequent transactions, all card types require a reason code. For a subsequent transaction, include these fields with a standard authorization request:

```
"processorInformation": {
  "transactionId": "01234567898765"
},
"processingInformation": {
  "authorizationOptions": {
    "initiator": {
      "type": "merchant",
      "merchantInitiatedTransaction": {
        "originalAuthorizedAmount": "100.00", // Discover only
        "previousTransactionID": "01234567890987",
        "reason": "7",
        "storedCredentialUsed": "true"
      }
    }
  },
  "commerceIndicator": "recurring"
}
```



	Industry Practice							Standing Instructions				
	Delayed	Incremental	No Show	Partial - shipment	Reauthorization	Resubmission	CIT Ad Hoc (COF)	Unsched MIT (UCOF)	Installment	Recurring	Subscription	Standing Order
previousTransactionId		ID No.	ID No.	ID No.	ID No.	ID No.	ID No.	ID No.	ID No.	ID No.	ID No.	ID No.
commerceIndicator	internet	internet	internet	internet	internet	internet	internet vbv	internet	install	recurring	recurring	recurring
type	merchant	merchant	merchant	merchant	merchant	merchant	customer	merchant	merchant	merchant	merchant	merchant
originalAuthorizedAmount (Discover only)	auth amount	auth amount	auth amount	auth amount	auth amount	auth amount	auth amount	auth amount	auth amount	auth amount	auth amount	auth amount
reason	2	5	4	6	3	1			9	7	7	8
storedCredentialUsed							true	true	true	true	true	true

Customer-Initiated Transactions with Credentials on File

A customer-initiated transaction (CIT) is a transaction initiated by the customer. There are two types of CIT transactions:

- Customer transactions where the credentials are stored for future merchant initiated transactions
- Customer transactions where the credentials are stored for future customer initiated transaction

Customers can initiate a CIT at a merchant payment terminal, through an online purchase transaction, or by making a purchase using a previously stored credential.

Storing Customer Credentials with a Customer-Initiated Transaction

Before you can perform a merchant-initiated transaction (MIT) or a customer-initiated transaction (CIT) with credentials-on-file (COF), you must store the customer's credentials for later use. Further, before you can store the user's credentials, you must get the customer's consent to store their private information. This is also known as establishing a relationship with the customer.

Fields Specific to this Use Case

Include this field with a standard authorization request when storing customer credentials:

- **processingInformation.authorizationOptions.initiator.credentialStoredOnFile**–Set this field to `true` to indicate that the customer credentials will be stored for future use.

Endpoint

POST https://api.cybersource.com/pts/v2/payments

Required Fields

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.phoneNumber

orderInformation.billTo.postalCode

paymentInformation.card.expirationMonth

paymentInformation.card.expirationYear

paymentInformation.card.number

processingInformation.authorizationOptions.initiator.credentialStoredOnFile

Set to `true`.

REST Example: Storing Customer Credentials During a Customer-Initiated Transaction

Request

Endpoint: POST https://api.cybersource.com/pts/v2/payments

```
{
  "processingInformation": {
    "authorizationOptions": {
      "initiator": {
        "credentialStoredOnFile": "true"
      }
    }
  },
  "orderInformation": {
    "billTo": {
      "firstName": "John",
      "lastName": "Deo",
      "address1": "201 S. Division St.",
      "postalCode": "48104-2201",
      "locality": "Ann Arbor",
      "administrativeArea": "MI",
      "country": "US",
      "email": "test@cybs.com",
      "phoneNumber": "6504327113"
    },
    "amountDetails": {
      "totalAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentInformation": {
    "card": {
      "expirationYear": "2031",
      "number": "4111xxxxxxxxxxxx",
      "expirationMonth": "12"
    }
  }
}
```

Response

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6528187198946076303004/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6528187198946076303004"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6528187198946076303004/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1652818719876"
  },
  "id": "6528187198946076303004",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "001"
    },
    "card": {
      "type": "001"
      }
   },
   "pointOfSaleInformation": {
      "terminalId": "111111"
   },
   "processorInformation": {
      "approvalCode": "888888",
      "networkTransactionId": "123456789619999",
      "transactionId": "123456789619999",
      "responseCode": "100",
      "avs": {
         "code": "X",
         "codeRaw": "I1"
      }
   },
   "reconciliationId": "63165088Z3AHV91G",
   "status": "AUTHORIZED",
   "submitTimeUtc": "2022-05-17T20:18:40Z"
}
```

Using Stored Customer Credentials During a Customer-Initiated Transaction

After customers store their credentials on file, they can recall these credentials to use with subsequent transactions.

Fields Specific to this Use Case

When a customer-initiated transaction (CIT) uses stored credentials, the transaction must include the following information in the authorization request:

- **processingInformation.authorizationOptions.initiator.storedCredentialUsed**—Set the field to `true` to indicate that the previously stored customer credentials were used.

Endpoint

`POST https://api.cybersource.com/pts/v2/payments`

Required Fields

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.phoneNumber

orderInformation.billTo.postalCode

paymentInformation.card.expirationMonth

paymentInformation.card.expirationYear

paymentInformation.card.number

processingInformation.authorizationOptions.initiator.storedCredentialUsed Set field to `true`.

Card-Specific Required Fields

Some card companies require additional information when making authorizations with stored credentials.

Discover

Discover requires the authorization amount from the original transaction when sending a request:

processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.
originalAuthorizedAmount

REST Example: Retrieving Customer Credentials During a Customer-Initiated Transaction

Request

Endpoint: `POST https://api.cybersource.com/pts/v2/payments`

```
{
  "processingInformation": {
    "authorizationOptions": {
      "initiator": {
        "storedCredentialUsed": "true"
      }
    }
  },
  "orderInformation": {
    "billTo": {
      "firstName": "John",
      "lastName": "Deo",
      "address1": "201 S. Division St.",
      "postalCode": "48104-2201",
      "locality": "Ann Arbor",
      "administrativeArea": "MI",
      "country": "US",
      "email": "test@cybs.com",
      "phoneNumber": "6504327113"
    },
    "amountDetails": {
      "totalAmount": "100.00",
      "currency": "USD",
      "originalAmount": "100"
        // Discover card Only
    }
  },
  "paymentInformation": {
    "card": {
      "expirationYear": "2031",
      "number": "4111xxxxxxxxxxxx",
      "expirationMonth": "12"
    }
  },
 "processorInformation": {
   "transactionId": "12345678961000"
 }
}
```

Response

```
},
  "paymentAccountInformation": {
    "card": {
      "type": "002"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "002"
    },
    "card": {
      "type": "002"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processorInformation": {
    "approvalCode": "888888",
    "authIndicator": "1",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "63740353A3AJ2NSH",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2022-05-20T19:13:06Z"
}
```

Merchant-Initiated Transactions

Merchants can initiate a payment on the behalf of a customer. This type of transaction is called a merchant-initiated transaction (MIT). When initiating a MIT, the customer is not present. However, customers must authorize the storage of their credentials and the use of these credentials for future payments.
There are two types of MITs:

- Industry Practice transactions: Follow-on transactions to a customer-initiated transaction.
- Standing Instruction transactions: Agreed upon standing instructions from the customer for the provision of goods or services. For example, a subscription to an internet music service may involve an agreed upon standing instruction allowing the merchant to bill a customer monthly for their subscription.

Industry Practice Transactions

Industry practice transactions are follow-on transactions to a previous customer-initiated transaction. There are six types of Industry Practice transactions:

- Delayed charges
- Incremental charges
- Partial authorizations or partial shipments
- Reauthorizations
- Resubmissions
- No-shows

Merchant-Initiated Delayed Transaction

Delayed charge transaction is performed to process a supplemental account charge after original services have been rendered and respective payment has been processed.

Fields Specific to this Use Case

The following information is required with a standard authorization request:

- **processingInformation.authorizationOptions.initiator.type**–Indicate that this transaction is a merchant-initiated follow-on transaction. Set the field to merchant.
- **processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.reason**–Include the reason for the transaction. Set the field to 2.

Endpoint

POST https://api.cybersource.com/pts/v2/payments

Required Fields

orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
orderInformation.billTo.address1	
orderInformation.billTo.administrativeArea	
orderInformation.billTo.country	
orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.phoneNumber	
orderInformation.billTo.postalCode	
paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	
processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.previousTransactionId	For Discover and American Express cards, use the transaction ID from the original transaction. For Visa, use the last successful transaction ID.
processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.reason	Set to 2.
processingInformation.authorizationOptions.initiator	Set to merchant.

Card-Specific Required Fields

Discover

The listed card requires additional field:

processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.originalAuthoriz

Provide the original transaction amount.

REST Example: Merchant-Initiated Delayed Authorization Transaction

Request

Endpoint: POST https://api.cybersource.com/pts/v2/payments

```
{
   "orderInformation": {
  "billTo" : {
     "country" : "US",
     "lastName" : "Kim",
     "address1" : "201 S. Division St.",
     "postalCode" : "48104-2201",
     "locality" : "Ann Arbor",
     "administrativeArea" : "MI",
     "firstName" : "Kyong-Jin",
         "phoneNumber": "6504327113",
     "email" : "test@cybs.com"
    },
      "amountDetails": {
         "totalAmount": "120.00",
         "currency": "USD"
      }
   },
   "paymentInformation": {
      "card": {
         "expirationYear": "2031",
         "number": "4111xxxxxxxxxxxx",
         "expirationMonth": "12"
      }
   },
   "processingInformation": {
      "authorizationOptions": {
         "initiator": {
       "type": "merchant",
         "merchantInitiatedTransaction": {
          "originalAuthorizedAmount": "100",
             // Discover only
          "previousTransactionId": "123456789619999",
          "reason": "2"
         }
        }
      }
   }
}
```

This example shows a sample response for a merchant-initiated delayed transaction.

Response

```
{
   "_links": {
      "authReversal": {
         "method": "POST",
         "href": "/pts/v2/payments/6534213653516599003001/reversals"
```

```
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6534213653516599003001"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6534213653516599003001/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1653421365327"
  },
  "id": "6534213653516599003001",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "120.00",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "002"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "002"
    },
    "card": {
      "type": "002"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processorInformation": {
    "approvalCode": "888888",
    "authIndicator": "1",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "64365475T3K10Q1D",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2022-05-24T19:42:45Z"
}
```

Merchant-Initiated Incremental Transaction

An incremental authorization is used to increase the total amount authorized for a payment if the initial authorization does not cover the total cost of goods and services. An incremental transaction is an additional amount to the original authorization. The final authorized total includes amounts for both the initial and the incremental authorizations.

Fields Specific to this Use Case

Include these fields with a standard authorization request:

- **processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.type**–Set the field to `merchant` to indicate this transaction is a merchant-initiated follow-on transaction.
- **processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.reason**–Set the reason field to `5`.
- **processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.previousTransactionID**– Provide the original transaction ID.

Endpoint

Send a `PATCH` request to this endpoint:
`pts/v2/payments/{id}` where `id` is the original authorization ID.

Required Fields

orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.reason	Set to `5`.
processingInformation.authorizationOptions.initiat	Set to `merchant`.

Card-Specific Required Fields

Discover

The listed card requires additional field:

processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.originalAuthoriz	Provide the original transaction amount.

REST Example: Merchant-Initiated Incremental Transaction

Send a `PATCH` request to this endpoint:
`pts/v2/payments/{id}` where `id` is the original authorization ID.

Request

```
{
   "orderInformation": {
  "billTo" : {
     "country" : "US",
     "lastName" : "Kim",
     "address1" : "201 S. Division St.",
     "postalCode" : "48104-2201",
     "locality" : "Ann Arbor",
     "administrativeArea" : "MI",
     "firstName" : "Kyong-Jin",
        "phoneNumber": "6504327113",
     "email" : "test@cybs.com"
    },
      "amountDetails": {
        "totalAmount": "120.00",
        "currency": "USD"
      }
   },
   "paymentInformation": {
      "card": {
        "expirationYear": "2031",
        "number": "4111xxxxxxxxxxxx",
        "expirationMonth": "12"
      }
   },
   "processingInformation": {
      "authorizationOptions": {
        "initiator": {
       "type": "merchant",
         "merchantInitiatedTransaction": {
         "originalAuthorizedAmount": "100",
                // Required for Discover
         "previousTransactionId": "123456789619999",
         "reason": "5"
         }
        }
      }
   }
}
```

This example shows a sample response for a merchant-initiated incremental transaction.

Example: Response

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6533225006556860003002/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6533225006556860003002"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6533225006556860003002/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1653322500637"
  },
  "id": "6533225006556860003002",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "001"
    },
    "card": {
      "type": "001"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processorInformation": {
    "approvalCode": "888888",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "64143477A3AJ4P2Z",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2022-05-23T16:15:00Z"
}
```

Merchant-Initiated Reauthorization Transactions

A reauthorization occurs when the completion or fulfillment of the original order or service extends beyond the authorized amount time limit. There are two common reauthorization scenarios:

- Split or delayed shipments by a retailer
- Extended car rentals, hotel stays, or cruise line bookings

Fields Specific to this Use Case

Include these fields with a standard authorization request:

- **processingInformation.authorizationOptions.initiator.type**—Set the field to merchant to indicate this transaction is a merchant-initiated follow-on transaction.

- **processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.reason**— Set the reason field to 3.

Endpoint

POST https://api.cybersource.com/pts/v2/payments

Required Fields

orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
orderInformation.billTo.address1	
orderInformation.billTo.administrativeArea	
orderInformation.billTo.country	
orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.phoneNumber	
orderInformation.billTo.postalCode	
paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	
processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.previousTransactionId	For Discover and American Express cards, use the transaction ID from the original transaction. For Visa, use the last successful transaction ID.
processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.reason	Set to 3.
processingInformation.authorizationOptions.initiator.type	Set to merchant.

Card-Specific Required Fields

Discover

The listed card requires additional field:

processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.originalAuthoriz	Provide the original transaction amount.

REST Example: Merchant-Initiated Reauthorization Transactions

Request

Endpoint: POST https://api.cybersource.com/pts/v2/payments

```
{
   "processingInformation": {
      "authorizationOptions": {
         "initiator": {
        "type": "merchant",
          "merchantInitiatedTransaction": {
           "originalAuthorizedAmount": "100", // Discover Only
           "previousTransactionId": "123456789619999",
           "reason": "3"
          }
         }
      }
   },
   "orderInformation": {
 "billTo" : {
    "country" : "US",
    "lastName" : "Kim",
    "address1" : "201 S. Division St.",
    "postalCode" : "48104-2201",
    "locality" : "Ann Arbor",
    "administrativeArea" : "MI",
    "firstName" : "Kyong-Jin",
         "phoneNumber": "6504327113",
    "email" : "test@cybs.com"
   },
      "amountDetails": {
         "totalAmount": "130.00",
         "currency": "USD"
      }
   },
   "paymentInformation": {
      "card": {
         "expirationYear": "2031",
         "number": "4111xxxxxxxxxxxx",
         "expirationMonth": "12"
      }
   }
}
```

Response

```
{
   "_links": {
      "authReversal": {
         "method": "POST",
         "href": "/pts/v2/payments/6541178668686490403003/reversals"
      },
      "self": {
         "method": "GET",
         "href": "/pts/v2/payments/6541178668686490403003"
      },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6541178668686490403003/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1654117866849"
  },
  "id": "6541178668686490403003",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "130.00",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "001"
    },
    "card": {
      "type": "001"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processorInformation": {
    "approvalCode": "888888",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "65313868D3TXXC05",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2022-06-01T21:11:06Z"
}
```

Merchant-Initiated Resubmission Transaction

A resubmission transaction is used when a merchant resubmits an authorization to recover an outstanding debt from the customer. A common scenario is when a card was initially declined due to insufficient funds, but the goods or services were already delivered to the customer.

Specific Fields for this Use Case

Include these fields with a standard authorization request:

- **processingInformation.authorizationOptions.initiator.type**–
- Set the field to merchant to indicate this transaction is a merchant-initiated follow-on transaction.
- **processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.reason**–
  Set the reason field to 1.

Required Fields

orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
orderInformation.billTo.address1	
orderInformation.billTo.administrativeArea	
orderInformation.billTo.country	
orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.phoneNumber	
orderInformation.billTo.postalCode	
paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	
processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.previousTransactionId	**For Discover and American Express cards, use the transaction ID from the original transaction. For Visa, use the last successful transaction ID.**
processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.reaso	**Set to** 1**.**
processingInformation.authorizationOptions.initiator.type	**Set to** merchant**.**

Card-Specific Required Fields

Discover

The listed card requires additional field:

processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.originalAuthoriz	**Provide the original transaction amount.**

REST Example: Merchant-Initiated Resubmitted Authorization Transactions

Request

Endpoint: POST https://api.cybersource.com/pts/v2/payments

```
{
  "processingInformation": {
    "authorizationOptions": {
      "initiator": {
     "type": "merchant",
      "merchantInitiatedTransaction": {
       "originalAuthorizedAmount": "100",  // Discover Only
       "previousTransactionId": "123456789619999",
       "reason": "1"
      }
      }
    }
  },
  "orderInformation": {
 "billTo" : {
   "country" : "US",
   "lastName" : "Kim",
   "address1" : "201 S. Division St.",
   "postalCode" : "48104-2201",
   "locality" : "Ann Arbor",
   "administrativeArea" : "MI",
   "firstName" : "Kyong-Jin",
      "phoneNumber": "6504327113",
   "email" : "test@cybs.com"
  },
    "amountDetails": {
      "totalAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentInformation": {
    "card": {
      "expirationYear": "2031",
      "number": "4111xxxxxxxxxxxx",
      "expirationMonth": "12"
    }
  }
}
```

This example shows a sample response for a resubmitted merchant-initiated authorization transaction.

Response

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6534232293716260503006/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6534232293716260503006"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6534232293716260503006/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1653423229353"
  },
  "id": "6534232293716260503006",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "004"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "004"
    },
    "card": {
      "type": "004"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processorInformation": {
    "approvalCode": "888888",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "64365912G3K7HFDJ",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2022-05-24T20:13:49Z"
}
```

Merchant-Initiated No-Show Transactions

A no-show authorization occurs when a merchant charges a customer after the customer makes a reservation, and does not show up to claim the reservation. In this situation, the customer is charged an agreed upon fee for not showing up as expected.

Fields Specific to this Use Case

Include these fields with a standard authorization request:

- **processingInformation.authorizationOptions.initiator.type**–
  Set the field to merchant to indicate this transaction is a merchant-initiated follow on transaction.
- **processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.reason**–
  Set the reason field to 4.

Required Fields

orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
orderInformation.billTo.address1	
orderInformation.billTo.administrativeArea	
orderInformation.billTo.country	
orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.phoneNumber	
orderInformation.billTo.postalCode	
paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	
processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.previousTransactionId	**For Discover and American Express cards, use the transaction ID from the original transaction. For Visa, use the last successful transaction ID.**
processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.reason	**Set to** 4.
processingInformation.authorizationOptions.initiator.type	**Set to** merchant.

Card-Specific Required Fields

Discover

The listed card requires additional field:

processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.originalAuthoriz	**Provide the original transaction amount.**

REST Example: Merchant-Initiated No-Show Transaction

Request

Endpoint: POST https://api.cybersource.com/pts/v2/payments

```
{
    "processingInformation": {
        "authorizationOptions": {
            "initiator": {
          "type": "merchant",
             "merchantInitiatedTransaction": {
              "originalAuthorizedAmount": "100", //Discover only
              "previousTransactionId": "123456789619999",
              "reason": "4"
             }
            }
        }
    },
    "orderInformation": {
  "billTo" : {
      "country" : "US",
      "lastName" : "Kim",
      "address1" : "201 S. Division St.",
      "postalCode" : "48104-2201",
      "locality" : "Ann Arbor",
      "administrativeArea" : "MI",
      "firstName" : "Kyong-Jin",
            "phoneNumber": "6504327113",
      "email" : "test@cybs.com"
     },
        "amountDetails": {
            "totalAmount": "150.00",
            "currency": "USD"
        }
    },
    "paymentInformation": {
        "card": {
            "expirationYear": "2031",
            "number": "4111xxxxxxxxxxxx",
            "expirationMonth": "12"
        }
    }
}
```

Response

```
{
    "_links": {
        "authReversal": {
            "method": "POST",
            "href": "/pts/v2/payments/6534214295466223903006/reversals"
        },
        "self": {
            "method": "GET",
            "href": "/pts/v2/payments/6534214295466223903006"
        },
        "capture": {
            "method": "POST",
            "href": "/pts/v2/payments/6534214295466223903006/captures"
        }
    },
    "clientReferenceInformation": {
        "code": "1653421429522"
    },
    "id": "6534214295466223903006",
    "orderInformation": {
        "amountDetails": {
            "authorizedAmount": "150.00",
            "currency": "USD"
        }
    },
    "paymentAccountInformation": {
        "card": {
            "type": "001"
        }
    },
    "paymentInformation": {
        "tokenizedCard": {
            "type": "001"
        },
        "card": {
            "type": "001"
        }
    },
    "pointOfSaleInformation": {
        "terminalId": "111111"
    },
    "processorInformation": {
        "approvalCode": "888888",
        "networkTransactionId": "123456789619999",
        "transactionId": "123456789619999",
        "responseCode": "100",
        "avs": {
            "code": "X",
            "codeRaw": "I1"
        }
    },
    "reconciliationId": "64365823G3K7HFAM",
    "status": "AUTHORIZED",
    "submitTimeUtc": "2022-05-24T19:43:49Z"
}
```

Standing Instruction Transactions

Standing instruction transactions are follow-on merchant-initiated transactions to a previous customer-initiated transaction. These are the types of standing instruction transactions:

- Installment payments

- Recurring payments (Visa, Discover, and American Express only)
- Subscription payments (Mastercard only)
- Standing order payments (Mastercard only)
- Unscheduled MIT COF payments

Installment Payments

An installment payment is a single purchase of goods or services billed to a customer in multiple transactions over a period of time agreed to by you and the customer. The agreement enables you to charge a specific amount at specified intervals.

Initial CIT Installment Payment

The first transaction in an installment payment is a customer-initiated transaction (CIT). Before you can perform a subsequent merchant-initiated transaction (MIT), you must store the customer's credentials for later use. Before you can store the user's credentials, you need to get the customer's consent to store their private information. This process is also known as establishing a relationship with the customer.

Endpoint

POST https://api.cybersource.com/pts/v2/payments

Fields Specific to this Use Case

Include the following information with a standard authorization request when storing customer credentials during the initial CIT installment payment:

- **processingInformation.authorizationOptions.initiator.credentialStoredOnFile**—Set the field to `true` to indicate the customer credentials will be stored for future use.
- **processingInformation.authorizationOptions.initiator.type**—Set the field to `customer` to indicate this transaction is a customer-initiated initial transaction.
- **processingInformation.commerceIndicator**—Set the transaction type to `internet` or a payer authentication value.
- In the CIT authorization response, capture the value in the **processorInformation.networkTransactionId** field. You will need to include the network transaction ID in subsequent MIT authorization requests.

Required Fields for Authorizing Initial Installment Payments Using REST APIs

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.phoneNumber	
orderInformation.billTo.postalCode	
paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	
processingInformation.authorizationOptions.initiator.credentialStoredOnFile	**Set to** `true`**.**
processingInformation.authorizationOptions.initiator.type	**Set to** `customer`**.**
processingInformation.commerceIndicator	**Set to** `internet` **or a payer authentication value.**

Card-Specific Required Fields

Some card companies require additional information when making authorizations with stored credentials.

Mastercard

Mastercard requires a reason code for an initial transaction:

processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.reason	**Set to** `9`**.**

REST Example: Initial Customer-Initiated Transaction Installment Payment

Request

Endpoint: `POST https://api.cybersource.com/pts/v2/payments`

```
{
  "processingInformation": {
    "commerceIndicator": "internet",
    "authorizationOptions": {
      "initiator": {
        "type": "customer",
        "credentialStoredOnFile": "true",
        "merchantInitiatedTransaction": {
          "reason": "9"   //Mastercard only
        }
      }
    }
  },
  "orderInformation": {
    "billTo": {
      "firstName": "John",
      "lastName": "Deo",
      "address1": "201 S. Division St.",
      "postalCode": "48104-2201",
      "locality": "Ann Arbor",
      "administrativeArea": "MI",
      "country": "US",
      "phoneNumber": "6504327113",
      "email": "test@cybs.com"
    },
    "amountDetails": {
      "totalAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentInformation": {
    "card": {
      "expirationYear": "2031",
      "number": "4111xxxxxxxxxxxx",
      "expirationMonth": "12"
    }
  }
}
```

Response

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6528187198946076303004/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6528187198946076303004"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6528187198946076303004/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1652818719876"
  },
  "id": "6528187198946076303004",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "001"
    },
    "card": {
      "type": "001"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processorInformation": {
    "approvalCode": "888888",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "63165088Z3AHV91G",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2022-05-17T20:18:40Z"
}
```

Subsequent MIT Installment Payments

After the initial CIT installment payment, subsequent installment payments are merchant-initiated transactions (MITs).

Endpoint

`POST https://api.cybersource.com/pts/v2/payments`

Fields Required for this Use Case

Include these information with a standard authorization request when sending a subsequent installment payment request:

- **processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.previousTransactionID**—Include the previous network transaction ID to indicate previously stored customer credentials were used.
- **processingInformation.authorizationOptions.initiator.storedCredentialUsed**—Set the field to `true`.
- **processingInformation.authorizationOptions.initiator.type**—Set the field to `merchant` to indicate a merchant-initiated transaction.
- **processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.reason**—Set the reason code field to `9`.
- **processingInformation.commerceIndicator**—Set the field to `install` to indicate this is an installment transaction.

Required Fields for Authorizing Subsequent Installment Payments Using REST APIs

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea	
orderInformation.billTo.country	
orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.phoneNumber	
orderInformation.billTo.postalCode	
paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	
processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction. previousTransactionID	**For Discover and American Express cards, use the transaction ID from the original transaction. For Visa, use the last successful transaction ID.**
processingInformation.authorizationOptions.initiator.storedCredentialUsed	**Set to `true`.**
processingInformation.authorizationOptions.initiator.type	**Set to `merchant`.**
processingInformation.commerceIndicator	**Set to `install`.**
processingInformation.initiator.merchantInitiatedTransaction.reason	**Set to `9`.**

Card-Specific Required Fields

Some card companies require additional information when making authorizations with stored credentials.

Discover

Discover requires the authorization amount from the original transaction when sending a request:

processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction. originalAuthorizedAmount

REST Example: Subsequent Installment Payment Authorization

Request

Endpoint: `POST https://api.cybersource.com/pts/v2/payments`

```
{
  "processingInformation": {
    "commerceIndicator": "install",
    "authorizationOptions": {
      "initiator": {
        "storedCredentialUsed": "true",
        "type": "merchant",
        "merchantInitiatedTransaction": {
          "reason": "9",
          "previousTransactionId": "123456789619999",
          "originalAuthorizedAmount": "100"    //Discover Only
        }
      }
    }
  },
  "orderInformation": {
    "billTo": {
      "firstName": "John",
      "lastName": "Deo",
      "address1": "201 S. Division St.",
      "postalCode": "48104-2201",
      "locality": "Ann Arbor",
      "administrativeArea": "MI",
      "country": "US",
      "phoneNumber": "6504327113",
      "email": "test@cybs.com"
    },
    "amountDetails": {
      "totalAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentInformation": {
    "card": {
      "expirationYear": "2031",
      "number": "4111xxxxxxxxxxxx",
      "expirationMonth": "12"
    }
  }
}
```

Response

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6530824710046809304002/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6530824710046809304002"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6530824710046809304002/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1653082470983"
  },
  "id": "6530824710046809304002",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "002"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "002"
    },
    "card": {
      "type": "002"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processorInformation": {
    "approvalCode": "888888",
    "authIndicator": "1",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "79710341A39WTT5W",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2022-05-20T21:34:31Z"
}
```

Recurring Payments

A recurring payment is a credentials-on-file (COF) transaction in a series of payments that you bill to a customer at a fixed amount at fixed, regular intervals not to exceed one year between transactions. The series of recurring payments is the result of an agreement between you and the customer for the purchase of goods or services that are provided at regular intervals.

Mastercard uses subscription and standing order payments instead of recurring payments. See *Mastercard Subscription Payments* on page 45 and *Mastercard Standing Order Payments* on page 42.

Recurring Billing for Recurring Payments

If you are using the Cybersource Recurring Billing service, do not use this document. Cybersource saves and stores payment credentials for recurring transactions, ensuring compliance with COF best practices.

For more information on Recurring Billing, see *Recurring Billing*.

Initial CIT Recurring Payment

The first transaction in a recurring payment is a customer-initiated transaction (CIT). Before you can perform a subsequent merchant-initiated transaction (MIT), you must store the customer's credentials for later use. Before you can store the user's credentials, you need to get the customer's consent to store their private information. This is also known as establishing a relationship with the customer.

Endpoint

POST https://api.cybersource.com/pts/v2/payments

Fields Specific to this Use Case

Include the following information with a standard authorization request when storing customer credentials during the initial CIT recurring payment:

- **processingInformation.authorizationOptions.initiator.credentialStoredOnFile**—Set the field to `true` to indicate the customer credentials will be stored for future use.
- **processingInformation.authorizationOptions.initiator.type**—Set the field to `customer` to indicate this transaction is a customer-initiated initial transaction.
- **processingInformation.commerceIndicator**—Set the field to `internet` or a payer authentication value.
- In the CIT authorization response, save the Transaction ID. You will need to include the network transaction ID in subsequent MIT recurring authorization requests.

Required Fields for Authorizing Initial Recurring Payments Using REST APIs

For an Initial CIT Recurring Payment, you must provide either field:

processingInformation.recurringOptions.firstRecurringPayment	Set to `true`.
processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.reason	Set to `7`.

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName

orderInformation.billTo.locality

orderInformation.billTo.postalCode

paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	
processingInformation.authorizationOptions.initiator.credentialStoredOnFile	**Set to `true`.**
processingInformation.authorizationOptions.initiator.type	**Set to `customer`.**
processingInformation.commerceIndicator	**Set to `internet`, a payer authentication value, or `MOTO`.**

REST Example: Initial CIT Recurring Payment
Use this REST API request for the initial CIT recurring payment. In the response, find the transaction ID value in the **processorInformation.transactionId** field and save it for subsequent MIT recurring authorization requests. If you are using a token, the transaction ID will be automatically stored.

Endpoint

`POST https://api.cybersource.com/pts/v2/payments`

Request

```
{
  "processingInformation": {
    "commerceIndicator": "internet",
    "authorizationOptions": {
      "initiator": {
        "credentialStoredOnFile": "true",
        "type": "customer"
      }
    }
  },
  "orderInformation": {
    "billTo": {
      "firstName": "John",
      "lastName": "Deo",
      "address1": "201 S. Division St.",
      "postalCode": "48104-2201",
      "locality": "Ann Arbor",
      "administrativeArea": "MI",
      "country": "US",
      "phoneNumber": "6504327113",
      "email": "test@cybs.com"
    },
    "amountDetails": {
      "totalAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentInformation": {
    "card": {
      "expirationYear": "2031",
      "number": "4111xxxxxxxxxxxx",
      "expirationMonth": "12"
    }
  }
}
```

Response

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6528187198946076303004/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6528187198946076303004"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6528187198946076303004/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1652818719876"
  },
  "id": "6528187198946076303004",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "001"
    },
    "card": {
      "type": "001"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processorInformation": {
    "approvalCode": "888888",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
   "reconciliationId": "63165088Z3AHV91G",
   "status": "AUTHORIZED",
   "submitTimeUtc": "2022-05-17T20:18:40Z"
}
```

Mastercard Standing Order Payments

A standing order payment is a recurring COF transaction that is a variable amount at a regular interval, such as a utility bill, not to exceed one year between transactions. The series of recurring payments is the result of an agreement between you and the customer for the purchase of goods or services that are provided at regular intervals.

Mastercard Initial CIT Standing Order Payment

The first transaction in a standing order payment is a customer-initiated transaction (CIT). Before you can perform a subsequent merchant-initiated transaction (MIT), you must store the customer's credentials for later use. Before you can store the user's credentials, you need to get the customer's consent to store their private information. This is also known as establishing a relationship with the customer.

Fields Specific to this Use Case

Include the following information with a standard authorization request:

- **processingInformation.authorizationOptions.initiator.credentialStoredOnFile**–Set the field to `true` to indicate the customer credentials will be stored for future use.
- **processingInformation.commerceIndicator**– Set the field to `internet` or a payer authentication value to indicate the transaction type.
- **processingInformation.initiator.merchantInitiatedTransaction.reason** –Set the field to reason code `8`.

- In the CIT authorization response, capture the value in the **processorInformation.networkTransactionId** field. You will need to include the network transaction ID in subsequent MIT authorization requests.

Endpoint

`POST https://api.cybersource.com/pts/v2/payments`

Required Fields for Authorizing Standing Order Payments Using REST APIs

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country

orderInformation.billTo.email

orderInformation.billTo.firstName

orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.phoneNumber	
orderInformation.billTo.postalCode	
paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	
processingInformation.authorizationOptions.initiator.credentialStoredOnFile	**Set to** `true`**.**
processingInformation.authorizationOptions.initiator.type	**Set to** `customer`**.**
processingInformation.commerceIndicator	**Set to** `internet` **or** `payer auth`**.**
processingInformation.initiator.merchantInitiatedTransaction.reason	**Set to** `8`**.**

REST Example: Initial CIT Standing Order Payments

Request

Endpoint: `POST https://api.cybersource.com/pts/v2/payments`

```
{
  "processingInformation": {
    "commerceIndicator": "internet",
    "authorizationOptions": {
      "initiator": {
        "credentialStoredOnFile": "true",
        "type": "customer",
        "merchantInitiatedTransaction": {
          "reason": "8"
        }
      }
    }
  },
  "orderInformation": {
    "billTo": {
      "firstName": "John",
      "lastName": "Deo",
      "address1": "201 S. Division St.",
      "postalCode": "48104-2201",
      "locality": "Ann Arbor",
      "administrativeArea": "MI",
      "country": "US",
      "phoneNumber": "6504327113",
      "email": "test@cybs.com"
    },
    "amountDetails": {
      "totalAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentInformation": {
    "card": {
      "expirationYear": "2031",
      "number": "5555xxxxxxxxxxxx",
      "expirationMonth": "12"
    }
  }
}
```

Response

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6530824710046809304002/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6530824710046809304002"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6530824710046809304002/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1653082470983"
  },
  "id": "6530824710046809304002",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "002"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "002"
    },
    "card": {
      "type": "002"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processorInformation": {
    "approvalCode": "888888",
    "authIndicator": "1",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
      "avs": {
        "code": "X",
        "codeRaw": "I1"
      }
    },
    "reconciliationId": "79710341A39WTT5W",
    "status": "AUTHORIZED",
    "submitTimeUtc": "2022-05-20T21:34:31Z"
}
```

Mastercard Subscription Payments

A subscription payment is a recurring COF transaction that is processed at a fixed amount at regular intervals not to exceed one year between transactions. The series of recurring payments is the result of an agreement between you and the customer for the purchase of goods or services that are provided at regular intervals.

Mastercard Initial CIT Subscription Payment

The first transaction in a subscription payment is a customer-initiated transaction (CIT). Before you can perform a subsequent merchant-initiated transaction (MIT), you must store the customer's credentials for later use. Before you can store the user's credentials, you need to get the customer's consent to store their private information. This is also known as establishing a relationship with the customer.

Fields Specific to this Use Case

Include the following information with a standard authorization request when storing customer credentials during the initial CIT subscription payment:

- **processingInformation.authorizationOptions.initiator.credentialStoredOnFile**–Set the field to `true` to indicate the customer credentials will be stored for future use.
- **processingInformation.commerceIndicator**–Set the field to `internet` or a payer authentication value.
- **processingInformation.initiator.merchantInitiatedTransaction.reason** –Set the reason code field to `7`.
- In the CIT authorization response, capture the value in the **processorInformation.networkTransactionId** field. You will need to include the network transaction ID in subsequent MIT authorization requests.

Endpoint

`POST https://api.cybersource.com/pts/v2/payments`

Required Fields for Authorizing Subscription Payments Using REST APIs

orderInformation.amountDetails.currency

orderInformation.amountDetails.totalAmount

orderInformation.billTo.address1

orderInformation.billTo.administrativeArea

orderInformation.billTo.country	
orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.phoneNumber	
orderInformation.billTo.postalCode	
paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	
processingInformation.authorizationOptions.initiator.credentialStoredOnFile	Set to `true`.
processingInformation.authorizationOptions.initiator.type	Set to `customer` or `payer auth`.
processingInformation.commerceIndicator	Set to `recurring`.
processingInformation.initiator.merchantInitiatedTransaction.reason	Set to `7`.

REST Example: Initial CIT Subscription Payments

Request

Endpoint: `POST https://api.cybersource.com/pts/v2/payments`

```
{
  "processingInformation": {
    "commerceIndicator": "internet",
    "authorizationOptions": {
      "initiator": {
        "type": "customer",
        "credentialStoredOnFile": "true",
        "merchantInitiatedTransaction": {
          "reason": "7"
        }
      }
    }
  },
  "orderInformation": {
    "billTo": {
      "firstName": "John",
      "lastName": "Deo",
      "address1": "201 S. Division St.",
      "postalCode": "48104-2201",
      "locality": "Ann Arbor",
      "administrativeArea": "MI",
      "country": "US",
      "phoneNumber": "6504327113",
      "email": "test@cybs.com"
    },
    "amountDetails": {
      "totalAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentInformation": {
    "card": {
      "expirationYear": "2031",
      "number": "4111xxxxxxxxxxxx",
      "expirationMonth": "12"
    }
  }
}
```

Response

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6530824710046809304002/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6530824710046809304002"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6530824710046809304002/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1653082470983"
  },
  "id": "6530824710046809304002",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "002"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "002"
    },
    "card": {
      "type": "002"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processorInformation": {
    "approvalCode": "888888",
    "authIndicator": "1",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "79710341A39WTT5W",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2022-05-20T21:34:31Z"
}
```

Unscheduled COF Payments

An unscheduled credentials-on-file (COF) transaction uses stored payment information for a fixed or variable amount that does not occur regularly. An account top-up is one kind of unscheduled COF.

Initial CIT Unscheduled COF Payment

The first transaction in an unscheduled COF payment is a customer-initiated transaction (CIT). Before you can perform a subsequent merchant-initiated transaction (MIT), you must store the customer's credentials for later use. Before you can store the user's credentials, you need to get the customer's consent to store their private information. This is also known as establishing a relationship with the customer.

Endpoint

`POST https://api.cybersource.com/pts/v2/payments`

Fields Specific to this Use Case

Include the following information with a standard authorization request when storing customer credentials during the initial CIT unscheduled COF payment:

- **processingInformation.authorizationOptions.initiator.credentialStoredOnFile**–Set the field to `true` to indicate the customer credentials will be stored for future use.
- **processingInformation.authorizationOptions.initiator.type**– Set the field to `customer` to indicate this transaction is a customer-initiated initial transaction.
- **processingInformation.commerceIndicator**–Set the field to `internet`.
- In the CIT authorization response, capture the value in the **processorInformation.networkTransactionId** field. You will need to include the network transaction ID in subsequent MIT authorization requests.

Required Fields for Authorizing Initial Unscheduled COF Payments Using REST APIs

These fields are required in a subsequent authorization request for an initial unscheduled COF payment:

orderInformation.amountDetails.currency	
orderInformation.amountDetails.totalAmount	
orderInformation.billTo.address1	
orderInformation.billTo.administrativeArea	
orderInformation.billTo.country	
orderInformation.billTo.email	
orderInformation.billTo.firstName	
orderInformation.billTo.lastName	
orderInformation.billTo.locality	
orderInformation.billTo.phoneNumber	
orderInformation.billTo.postalCode	
paymentInformation.card.expirationMonth	
paymentInformation.card.expirationYear	
paymentInformation.card.number	
processingInformation.authorizationOptions.initiator.credentialStoredOnFile	**Set to `true`.**
processingInformation.authorizationOptions.initiator.type	**Set to `customer`.**
processingInformation.commerceIndicator	**Set to `internet` or a payer authentication value.**

REST Example: Initial CIT Unscheduled COF Payment

Request

Endpoint: `POST https://api.cybersource.com/pts/v2/payments`

```
{
  "processingInformation": {
    "commerceIndicator": "internet",
    "authorizationOptions": {
      "initiator": {
        "credentialStoredOnFile": "true",
        "type": "customer"
      }
    }
  },
  "orderInformation": {
    "billTo": {
      "firstName": "John",
      "lastName": "Deo",
      "address1": "201 S. Division St.",
      "postalCode": "48104-2201",
      "locality": "Ann Arbor",
      "administrativeArea": "MI",
      "country": "US",
      "phoneNumber": "6504327113",
      "email": "test@cybs.com"
    },
    "amountDetails": {
      "totalAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentInformation": {
    "card": {
      "expirationYear": "2031",
      "number": "4111xxxxxxxxxxxx",
      "expirationMonth": "12"
    }
  }
}
```

Response

```
{
  "_links": {
    "authReversal": {
      "method": "POST",
      "href": "/pts/v2/payments/6528187198946076303004/reversals"
    },
    "self": {
      "method": "GET",
      "href": "/pts/v2/payments/6528187198946076303004"
    },
    "capture": {
      "method": "POST",
      "href": "/pts/v2/payments/6528187198946076303004/captures"
    }
  },
  "clientReferenceInformation": {
    "code": "1652818719876"
  },
  "id": "6528187198946076303004",
  "orderInformation": {
    "amountDetails": {
      "authorizedAmount": "100.00",
      "currency": "USD"
    }
  },
  "paymentAccountInformation": {
    "card": {
      "type": "001"
    }
  },
  "paymentInformation": {
    "tokenizedCard": {
      "type": "001"
    },
    "card": {
      "type": "001"
    }
  },
  "pointOfSaleInformation": {
    "terminalId": "111111"
  },
  "processorInformation": {
    "approvalCode": "888888",
    "networkTransactionId": "123456789619999",
    "transactionId": "123456789619999",
    "responseCode": "100",
    "avs": {
      "code": "X",
      "codeRaw": "I1"
    }
  },
  "reconciliationId": "63165088Z3AHV91G",
  "status": "AUTHORIZED",
  "submitTimeUtc": "2022-05-17T20:18:40Z"
}
```