vSafe API Messages
Common Use Cases

vSafe API Messages Common Use Cases

Contents:

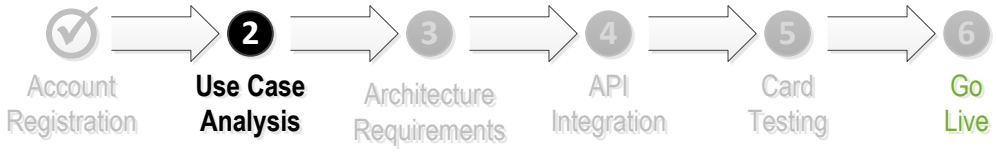
Change History

Date	Author	Notes
4/11/2014	L. Humbird	First publication.
9/05/2014	L. Humbird	Add use cases for Wallet and Reporting (draft) Organize all use cases into categories.
9/10/2014	L. Humbird	Prepare Wallet and Reporting use cases for review. Updated all diagrams: standardized appearance, minor corrections to process flow, parm naming. Add “How to read the diagrams” to the intro.
9/19/2014	L. Humbird	New use cases; reorganized to separate out PCI in-scope and out-of-scope.
9/23/2014	L. Humbird	Add sections on Tokenization and Session Tags. Consolidated separate use cases into “alternate flows” for clarity. Add Partner-Initiated Automated Payment Set Up. Initial-capped all section titles for clearer cross-referencing. Demoted 3rd order section titles - except Primary and Alternate Flows - to make them more prominent for skimming and scanning.
10/06/2014	L. Humbird	Add a glossary to the Introduction.
10/29/2014	L. Humbird	Split non-indemnified and PAN use cases to separate section. Correct non-indem flows, minus fraud engine references. Change use case “Get Transaction Detailed Information” to “Get Payment Status Information" (pg 31). Add new PaymentStatus codes, and a next-step table. Modify all diagrams that return PaymentStatus after back-end communication to branch on delayed response to the GetPaymentStatus flow. Add System Status use case. Change “Indemnified service” to “guaranteed payments service”. ”Non-indemnified” stays as-is. Add “Alternate Flow, Part 1, Take PAN” to use case “Process a Payment, Authorize and Confirm, Non-Indemnified”.
12/19/2014	L. Humbird	Removed Delayed Payment conditional branch from API calls that return a payment status.

Introduction

Purpose

These use cases are intended to help Vesta's partner businesses understand and integrate vSafe API messages into their checkout workflow.



***You Are Here**. Use Case Analysis is your first technical step to understand and perform API Integration.*

The following use cases provide a visual and procedural description for typical payment workflows, and specifically howthese API messages are used for payment processing and fraud indemnification service.

If you are using the guaranteed payments service, refer to the indemnified use cases. If you are not using the guaranteed payments service, refer to the Non-Indemnified and Take PAN Flows.

Glossary

Term	Definition
Acquirer	The bank or financial institution that receives, or "acquires" funds in a credit card transaction on behalf of the merchant.
Cardholder	Name of the individual associated with the payment card or device.
Indemnification	This is the guaranteed payments service that provides financial protection from chargeback fraud. The standardservice provides no financial protection, but is somewhat simpler to implement and test.
PCI DSS	Payment Card Industry Data Security Standard. PCI is the governing body for setting and maintaining industry-wide DSS requirements and procedures for handling major brands of credit card data.
Partner	A vSafe authorized merchant.
Payment card	A credit or signature debit card.
Payment device	Stored information about a payment card.
PCI JavaScript Library	A set of embedded vSafe functions at the customer's web browser. Its purpose is to minimize a partner's PCI scope. It does this by intercepting the customer's payment card PAN and exchanging it for a temporary token.

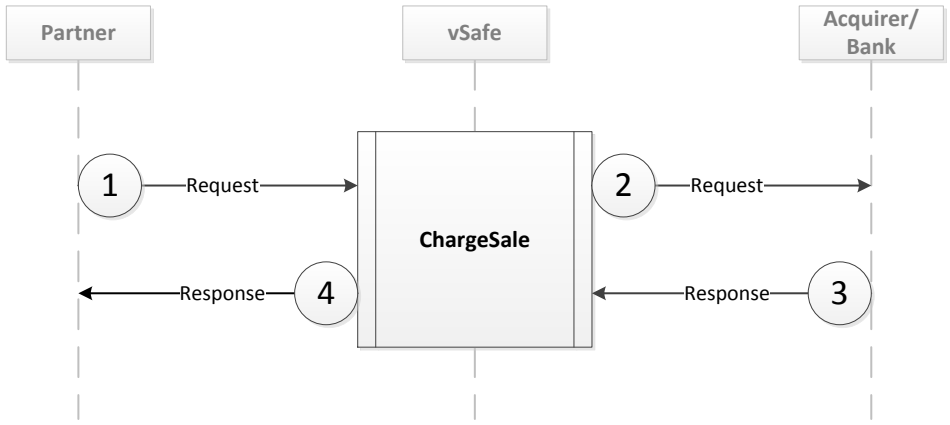
Tokens and IDs

Term	Definition
Token	A unique value that represents and is exchanged for a payment card PAN. This is a security measure that reduces a partner's PCI exposure, and reduces the risk of hackers accessing PANs stored or transmitted in partner systems. Tokens are created automatically and as needed by vSafe components, such as API calls and the embedded scripts in the customer's browser. Tokens can be temporary or permanent.
Temporary token	A type of token that can be used for one non-recurring transaction, and expires after a set period of time, usually a few days.
Permanent token	A type of token that never expires. It represents a payment device that is intended for future or recurring charges. A permanent token can be issued to a payment device following a successful payment or authorization with the acquirer. When a permanent token is assigned, any previously assigned temporary token becomes expired.
PAN	Primary Account Number. The 12-16-digit number associated with the bank and account number, and is commonly printed or embossed on the surface of the card. Capturing, storing or transmitting this information is considered in-scope for PCI.
Web Session ID	Also known as a device ID. This is metadata used to identify a customer for the duration of a web-based transaction, such as a user session or a checkout process. A Web Session ID is generated by the GetSessionTags API call. It is required for the guaranteed payments service, and optional for the non-indemnified service.
Transaction ID	A partner-generated uniqueidentifierfor partners to identify and track transactions.
Payment ID	A vSafe-generated unique identifier used to identify and track transactions. This is required for operations on an existing transaction, such as reversals or refunds, pended payments, and recurring payments.
Enrollment ID	A vSafe-generated unique identifier for tracking recurring payment enrollments.

How to Read the Diagrams

The flow diagrams in this document depict a number of lanes, each representing an actor: customer, partner, vSafe, and acquirer. Other lanes may appear as needed.

The diagrams in this document use the following shape to indicate a vSafe API call, as shown. The numbered circles represent the order of events. Generally flow order is top-to-bottom, following directional arrows.



This simplified diagram illustrates the top-down sequence of events:

1. Partner calls ChargeSale API with request parameters, such as payment info, charge amount.
2. ChargeSale sends a payment request to the acquirer.
3. Acquirer responds with payment status.
4. ChargeSale responds to the partner with the payment status.

In all use case diagrams, vSafe API calls are located in the vSafe lane, and are indicated by boxes with double-lines. Other boxes indicate related processes, but are not vSafe API calls.

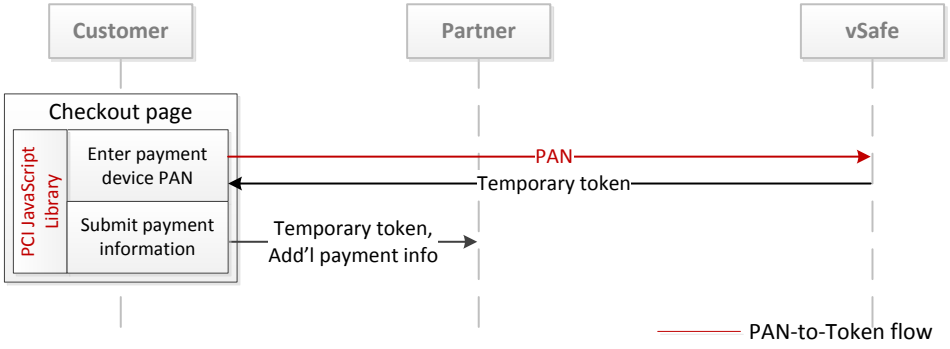
Tokenization

vSafe helps you minimize your PCI scope by exchanging the payment card PAN with a token. vSafe provides two methods to implement tokenization.

Primary Method: Use PCI JavaScript Library

vSafe'sPCI JavaScript Library isintended for use in web and mobile web channels. This libraryis injected to your customer's browser via your checkout web page. The scripts in the library are triggered by when the customer types in the payment device PAN.

This diagram shows how thePAN is intercepted and exchanged for a token.



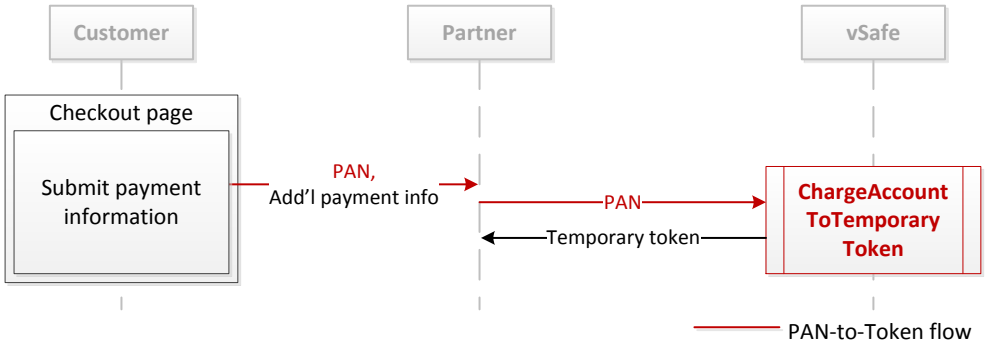
1. Customer enters payment cardPAN, then moves the cursor to the next input field.
2. PCI JavaScript Library encrypts the PAN and sendsit to vSafe (red flow line).
3. vSafe stores the PAN, and returns a temporary token to the customer's web browser.
4. PCI JavaScript Library receives the temporary token from the browser. This temporary token is returned to the partner when the payment information form is submitted (black flow line).

Note: In this diagram, the red flow linesare an automatic process, and require no interaction by the partner. Therefore, in the use cases you will only see the temporary token flow from the customer lane.

When the customer submits the payment form, you are collecting a temporary token along with the rest of the payment information. Since the PAN never arrives on your systems, your PCI scope is significantly reduced, and may be eliminated.

Alternate Method: Use API Call

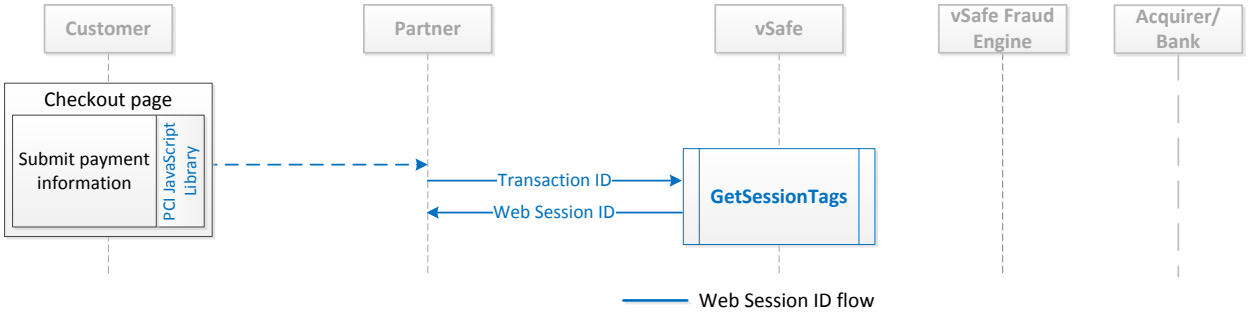
This vSafe API exchanges the payment card PAN with a temporary token. This method is intended for channels that cannot accommodate the PCI JavaScript Library method, such as IVR, call centers, or mobile applications. Although this method exposes your systems to the PAN, this gives you the option of not storing or transmitting after the initial capture.



1. Customer submits payment information. This includes the payment card PAN, expiration date, CVN, cardholder information, and payment amount.
2. Partner calls **ChargeAccountToTemporaryToken**with the PAN in the request parameters.
3. vSafe returns a temporary token for use in subsequent calls, such as ChargeSale.

Session Tags

The GetSessionTags API call retrieves Web Session ID, which is the customer's web fingerprint for use in fraud detection, and is used in subsequent API calls.Session tags are required for the guaranteed payments service, and are optional for the non-indemnified service.



1. Partner calls **GetSessionTags** with the Transaction ID in the request parameters.
2. vSafe returns the Web Session ID for the customer.

Are You Still There?

vSafe provides methods for interrogating system status and transaction payment status. This provides assurance that vSafe is operating normally, and that your transactions are in progress or completed.

Get System Status You can poll vSafe regularly and on an as-needed basis to see if it is up and running. The **HeartBeat**API is designed for quick response, and has no impact on any transaction processing. For more information, see Get System Status.

Get Payment Status If a transaction payment status did not arrive when you expected it, you can check the status at any time. The **GetPaymentStatus** API does this for you. For more information, see Get Payment Status.

Pre-Conditions for All Use Cases

In order to utilize automatic PAN interception and tokenizationduring checkout, vSafe's PCI JavaScript library must beembedded in the customer's browser at the checkout page.

Purchases and Reversals

Process a Payment, Single Call

Description

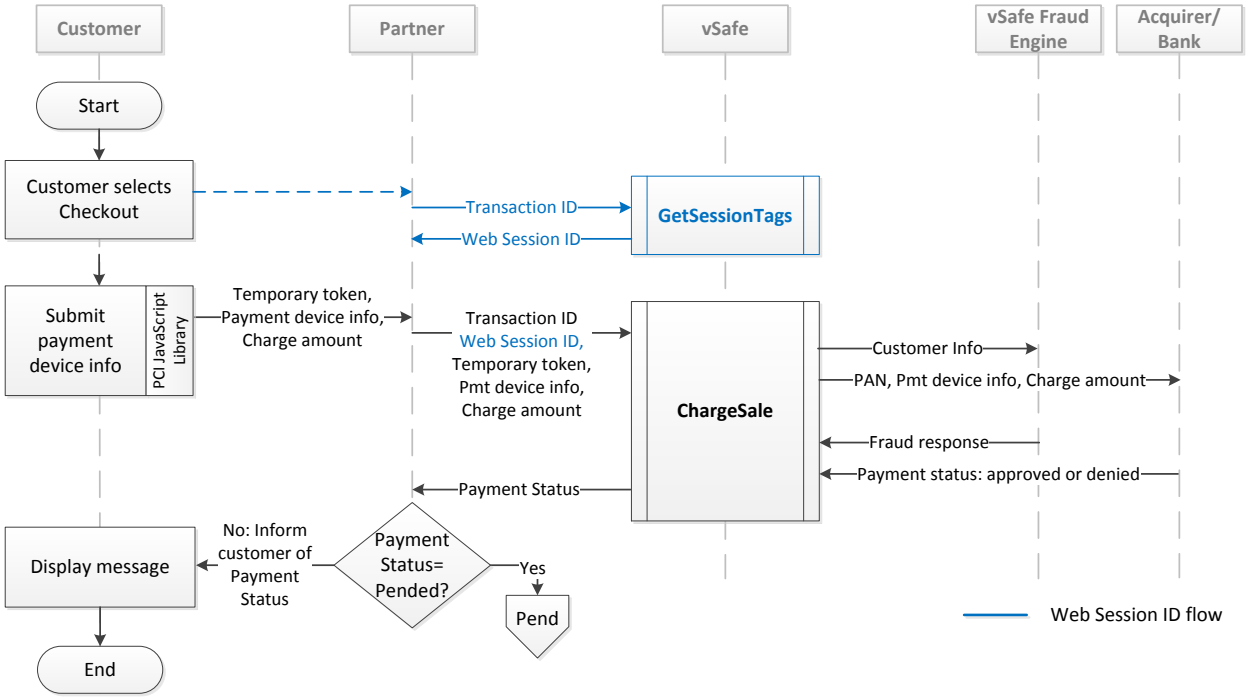
This is a simple one-step process to take a payment with immediate fulfillment.With theguaranteed payments service, each monetized transaction is checked with vSafe's fraud engine.

Preconditions

- Customer has at least one item with a total due of more than $0.00 in the shopping cart.
- Customer has a payment card.
- You are using vSafe's PCI JavaScript Library at the customer's browser to intercept the PAN and inject a temporary token.

Note:**GetSessionTags** is called to retrieve the web device fingerprint data, which is used by **ChargeSale**.

Primary Flow



1. Get the current Web Session ID for the customer (blue flow lines). This is required if the payment has been taken via the web, mobile web or mobile application. This is captured during the current customer session, before the card information is collected.
    a. Partner calls **GetSessionTags** with the Transaction ID.
    b. vSafe returns the customer's Web Session ID.(The call retrieves this information from the PCI Shield JavaScript on the customer's web browser.)
2. Process payment.
    a. Partner captures name, address, expiration date, CVN, etc. from the customer.
    b. Partner calls **ChargeSale** with the following parameters:
        - Web session ID (This is required if the payment source is via the web, mobile web or mobile application.)
        - Temporary token
        - Card information (cardholder's name and address, card expiration date, CVN)
        - Charge amount
        - Payment source (web or phone)
        - Risk information (This is an XML string that describes the details of the purchase.
    c. vSafe evaluates the fraud risk.
    d. vSafe authorizes the transaction with bank.
    e. vSafe responds with payment outcome to **ChargeSale** with payment status:

Payment Status	Meaning	Next Step
1	Bank Denied	Communicate the outcome to the customer.
2	vSafe Pended	Refer to: Resolve a Pended Payment.
3	vSafe Denied	Communicate the outcome to the customer.
5	Authorized	Communicate the outcome to the customer. If the StoreCard flag is set, vSafe returns a permanent token.

Key Business Rules

- Charge amount must be greater than $0.00.
- Credit card number, expiration date, and CVN must be valid.

Best Practices

- If the payment status is not returned in the expected time, refer to Get Payment Status.

Process a Payment, Authorize and Confirm

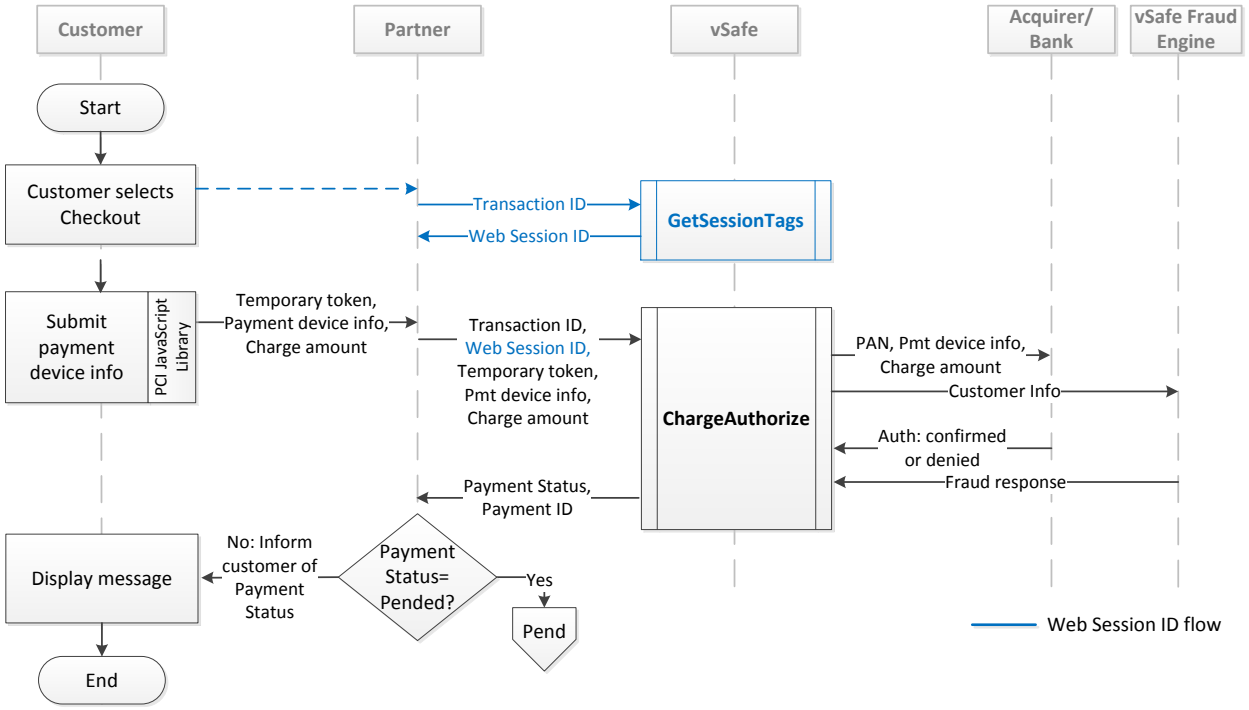
Description

This use case outlines the process to complete a payment for purchase in two steps: authorization and capture.

Preconditions

- Customer has at least one item with a total due of more than $0.00 in the shopping cart.
- Customer has a payment card.
- You are using vSafe's PCI JavaScript Library at the customer's browser to intercept the PAN and inject a temporary token.

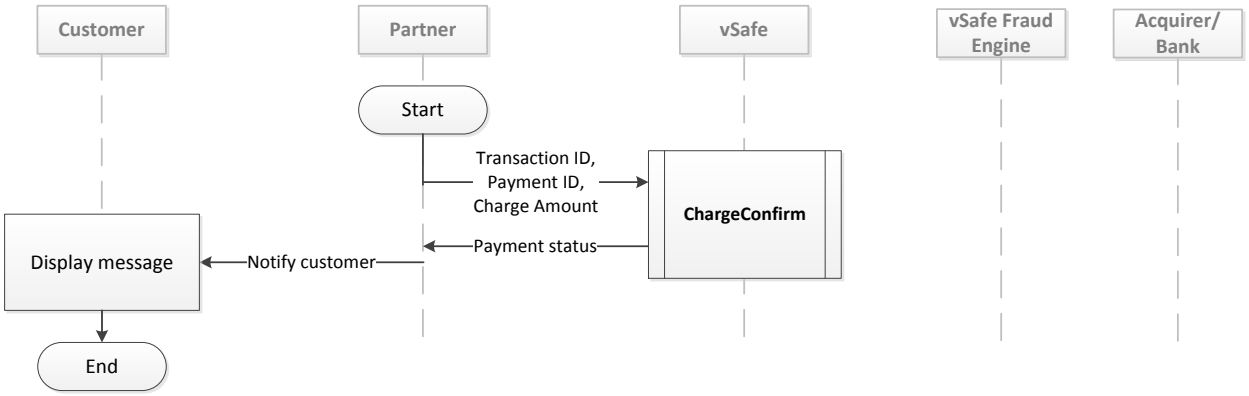
Primary Flow, Part 1: Authorize Payment



1. Customer selects Checkout.
2. Get the current Web Session ID for the customer (blue flow lines). This is required if the payment has been taken via the web, mobile web or mobile application. This is captured during the current customer session, before the card information is collected.
    a. Partner calls **GetSessionTags** with the Transaction ID.
    b. vSafe returns the customer's Web Session ID.(The call retrieves this information from the PCI JavaScript on the customer's web browser.)
3. Authorize the transaction.
    a. Partner captures name, address, expiration date, CVN, etc. from customer.
    b. Partner calls **ChargeAuthorize**. with the following parameters:
        - Web session ID (This is required if the payment source is via the web, mobile web or mobile application.)
        - Temporary token
        - Card information (cardholder's name and address, card expiration date, CVN)
        - Charge amount
        - Payment source (web, phone, or prearranged)
        - Risk information (This is an XML string that describes the details of the purchase.
    c. vSafe authorizes transaction with the bank.
    d. vSafe evaluates fraud risk.
    e. vSafe responds with payment outcome to **ChargeAuthorize** with payment status:

Payment Status	Meaning	Next Step
1	Bank Denied	Communicate the outcome to the customer.
2	vSafe Pended	Refer to: Resolve a Pended Payment.
3	vSafe Denied	Communicate the outcome to the customer.
5	Authorized	Communicate the outcome to the customer. If the StoreCard flag is set, vSafe returns a permanent token.

Primary Flow, Part 2: Confirm Payment



4. Confirm the transaction.
    a. Partner calls **ChargeConfirm** with the payment ID created by **ChargeAuthorize**, and the charge amount.
    b. vSafe performs the following:
        i. Check that the amount is greater than$0.00 and is less thanthe authorized amount.
        ii. Check that the authorization hasn't expired.
        iii. Contact the acquirer with the payment card information and the confirmed charge amount.
5. Partner notifies the customer of the outcome.

Key Business Rules

- Charge amount to be authorized must be > $0 and ≤ the authorized amount.
- A charge must be confirmed within a predefined time frame set by the bank that issued the customer's card.
- An authorized charge can be confirmed only once.

Best Practices

- If the payment status is not returned in the expected time, refer to Get Payment Status.

Resolve a Pended Payment

Description

This use case is to handlepaymentsthat are incomplete due to entering a pended status and you are licensed to use the online pending service (KBA). A transaction can result in a pended payment status from the following use cases:

- Process a Payment, Single Call

Best Practices

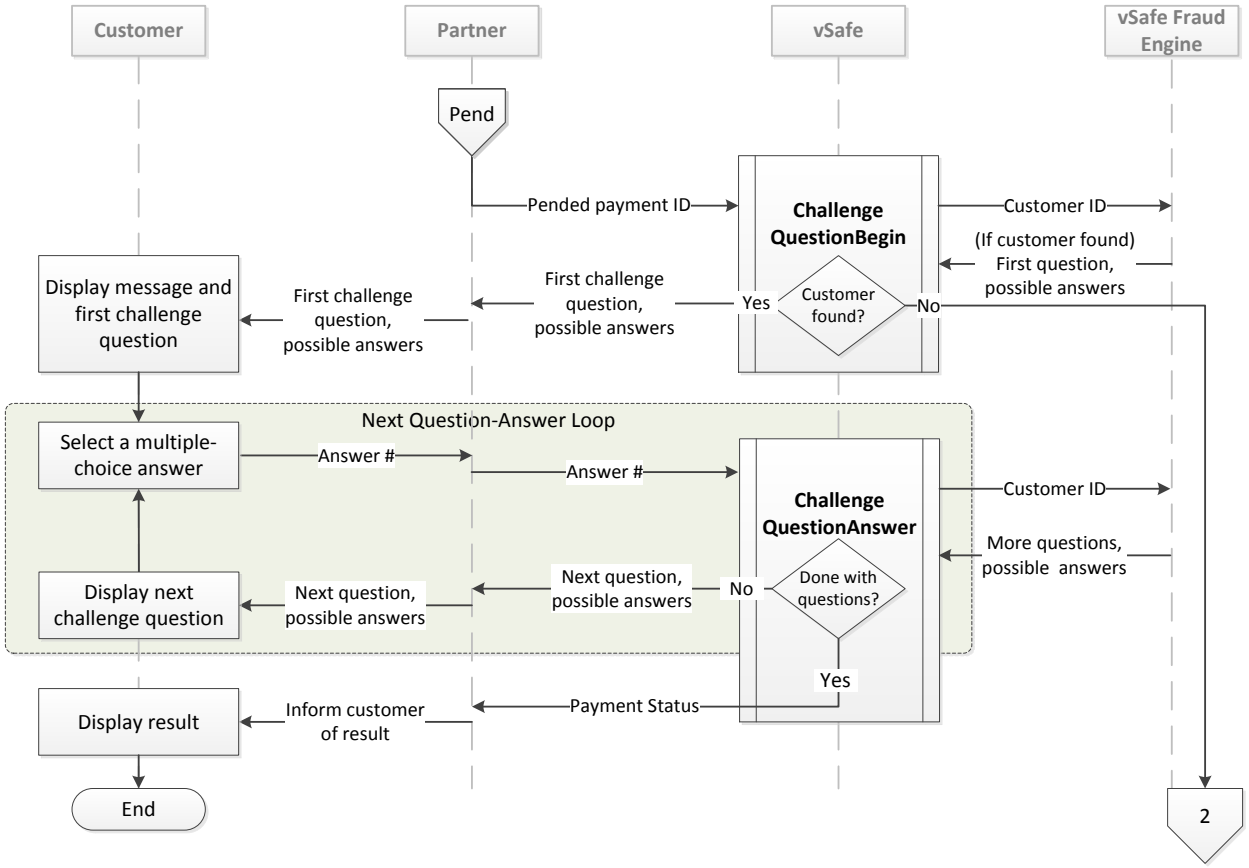
- If the payment status is not returned in the expected time, refer to Get Payment Status.
- Process a Payment, Authorize and

A pended status indicates that there is a moderate fraud risk. In order to move a pended payment to a completed state, the customer's identity needs to be verified.

Preconditions

- Partner is using the guaranteed payments service.
- Partner has licensed the online pend resolution service
- The transaction Payment Status = 2 (Pended).

Primary Flow: Customer Found



1. Partner calls **ChallengeQuestionBegin**with the payment ID.
    a. If re-running **ChallengeQuestionBegin**, include the requested additional customer information – date of birth or Social Security number.
2. vSafe looks up the customer information associated with the payment ID.
3. vSafe returns the following in **ChallengeQuestionBegin** response:
    - An action code that indicates that thecustomer was found,
    - The first challenge question and multiple-choice answers.
4. Customer responds to question by selecting an answer.
5. Partner calls **ChallengeQuestionAnswer** with the customer's selection.
6. vSafe responds to **ChallengeQuestionAnswer** with next question and multiple-choice answers.
    a. Repeat while additional questions remain.
7. vSafe evaluates the customer's answers, and provides payment status in final **ChallengeQuestionAnswer**response.

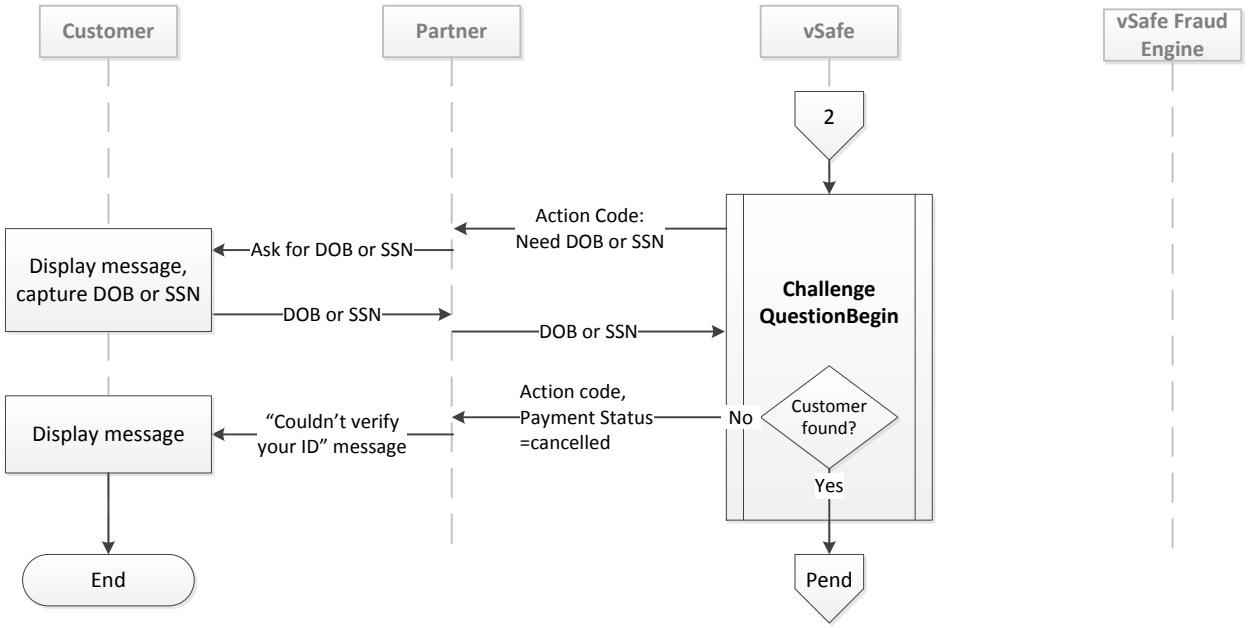
Payment Status	Meaning	Next Step
3	vSafe Denied	Communicate outcome to customer.
5	Payment is Authorized	Complete this transaction using ChargeConfirm.
10	Successful Payment	Communicate outcome to customer.

    a. If the payment status indicates that the transaction was denied, vSafe returns a Payment Status = 3 (vSafe Denied).
    b. If the payment status indicates that the transaction was approved and the call was issued from a **ChargeAuthorize**, the PaymentStatus value will be 5 (Authorized).
    c. If the payment status indicates that the transaction was approved and the call was issued from a **ChargeSale**, the PaymentStatus value will be 10 (Successful Payment).
8. Partner communicates outcome to customer.

Alternate Flow: Customer Not Found, Need More Information

Use this flow when the response contains Action Code=2(Customer not found, need more information).

This is similar to the main flow, but includes steps to aid in identifying the customer prior to beginning the challenge question session.



1. vSafe returns an Action Code that indicates if the customer was found:

Action Code	Meaning	Next Step
0	Customer has been found	Present customer with the first challenge question and multiple-choice answers (provided in response).
2	Exact customer match not found after final attempt	The payment is cancelled.
3	Exact customer match not found.	Go back to Step 1 and provide customer's DOB in the request message.
4	Exact customer match not found.	Go back to Step 1 and provide customer's last 4 SSN in the request message.

2. Continue from Step 4 of the Primary Flow.

Key Business Rules

- **ChallengeQuestionAnswer** can be performed only on a pended payment.
- If the session is unavailable or interrupted by an unrecoverable communication problem, the payment state changes from pended to cancelled.
- The number of question-answer sessions and the number of required correct answers is configured by vSafe.

Best Practices

- **ChallengeQuestionAnswer** is intended for repeated use. Each time **ChallengeQuestionAnswer** is used, vSafe collects the answer from the previous question, and provides another challenge question and multiple-choice answers.
- If the payment status is not returned in the expected time, refer to Get Payment Status.

Refundor Void aCompleted Payment

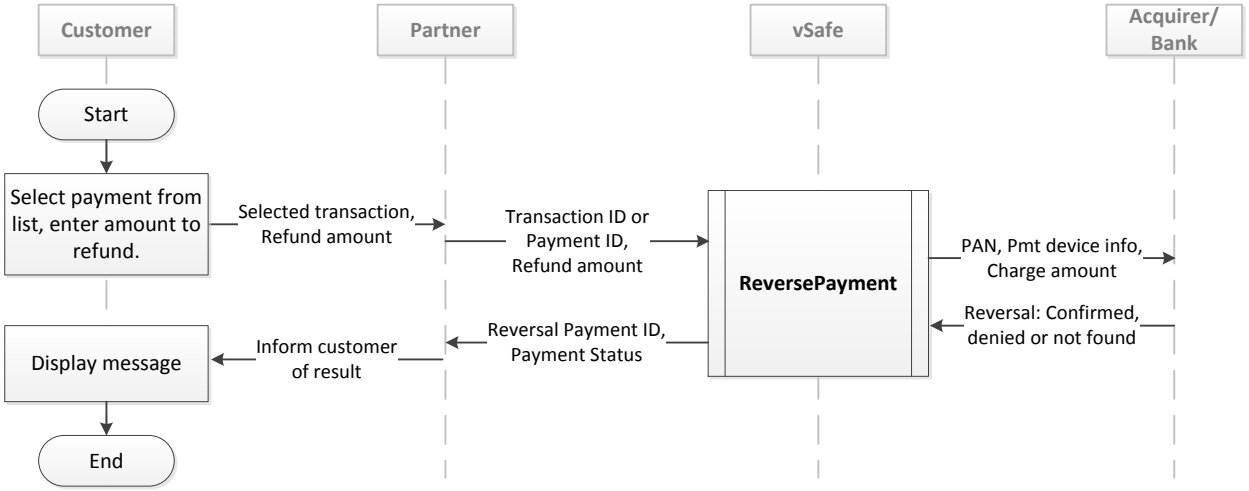
Description

This use case is to reverse a completed purchase. If the purchase hasn't settled with the acquirer, then this becomes a void rather than a refund.

Preconditions

- Payment ID is known.
- Payment amount to refund is known.
- Payment Status = 10 (Completed)

Primary Flow



1. Partner calls **ReversePayment** with the payment ID to be reversed, and a refund amount.
2. vSafesends the reversal information to the acquirer/bank.
3. vSaferesponds with a new Payment ID for the reversal, and a Payment Status=10 (successful refund or void).
4. Partner communicates refundstatus to the customer.

Key Business Rules

- The payment ID must be valid.
- A partner must be authorized by vSafe to perform a reversal.
- The reversal amount must be greater than $0.00 and less than or equal to the payment amount.
- A reversal must be performed on transactions that resulted in a payment amount.
- To reverse an authorized payment, see Cancelan Authorized Payment.

Best Practices

- If the payment status is not returned in the expected time, refer to Get Payment Status.

Cancelan Authorized Payment

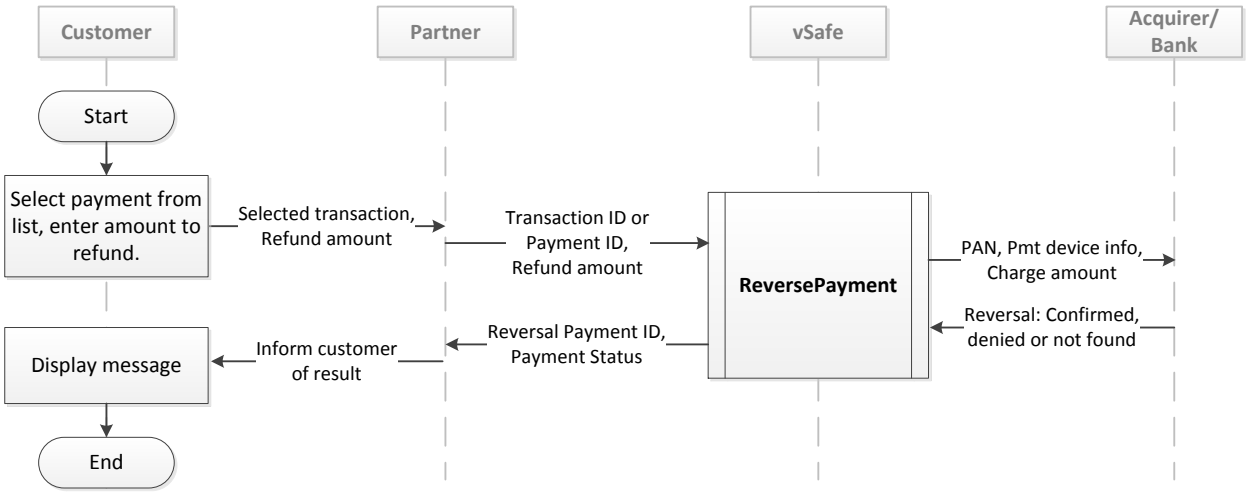
Description

This use case is to cancel a charge authorization. A charge authorization is normally finished with a charge confirmation.

Preconditions

- Payment ID is known and is valid.
- The Payment Status = 5 (Authorized).
- The reversal amount must be equal to the authorized amount.

Primary Flow



1. Partner calls **ReversePayment** with the payment ID to be reversed, and a refund amount.
2. vSafe sends the reversal information to the acquirer/bank.
3. vSafe responds by changing the Payment status from 5 (Authorized) to 4 (Cancelled).
4. Partner communicates refund status to the customer.

Key Business Rules

- The partner must be authorized by vSafe to perform reversals.
- An authorized transaction will expire after a set period of time (usually a few days) if there is no confirmation.

Best Practices

- If the payment status is not returned in the expected time, refer to Get Payment Status.

Supporting Functions

Get Payment Card Detailed Information

Use this when a customer or partner wants to identify the type of payment card being presented.

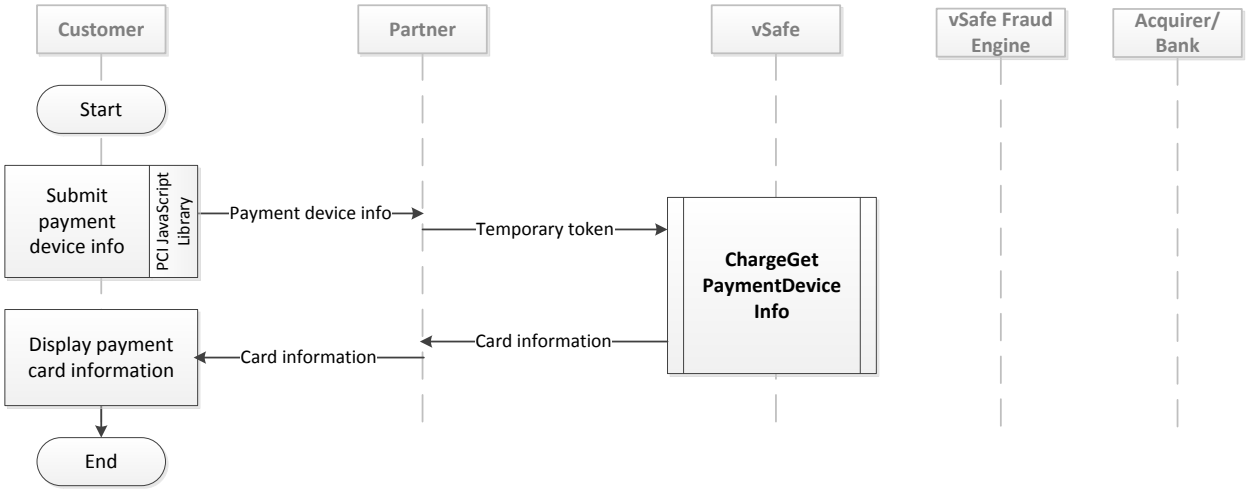
Description

This use case returns metadata on a payment card. This is mainly used during integration testing, but is also useful if an unknown card is being presented, or if a partner wants to process a subset of card transactions, such as credit only or debit only.

Preconditions

- Customer or partner provides a payment card to evaluate.
- You are using vSafe's PCI JavaScript Library at the customer's browser to intercept the PAN and inject a temporary token.

Primary Flow



1. Get card information.
    a. Partner calls **ChargeGetPaymentDeviceInfo** with the temporary token.
    b. vSafe returns the following information in **ChargeGetPaymentDeviceInfo** response parameters:
        - IsCreditRoutable (Yes/No)
        - IsPrepaidCard (Yes/No)
        - IsSignatureDebit (Yes/No)
        - IsNonSignatureDebit (Yes/No)
        - PaymentDeviceCVNLength (zero-length indicates a non-signature debit card)
        - PaymentDeviceTypeCD
        - PaymentDeviceTypeName

Business Rules

- Be sure that the IsTempToken Boolean is set correctly.Otherwise an error response code is returned.

Get Payment Status

Description

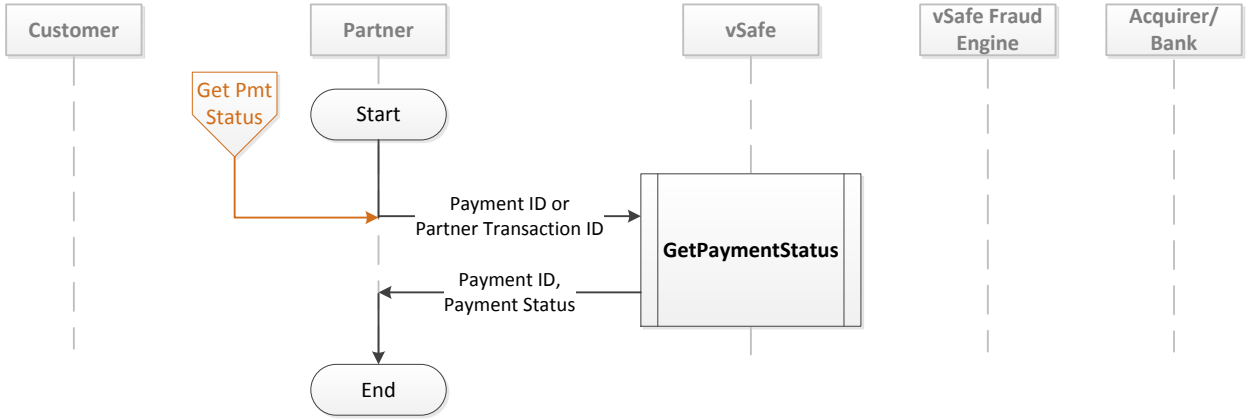
This use case supports obtaining status for payments that are:

- Recently submitted transactions awaiting final payment decision
- Previously processed but have not returned a status update

Preconditions

- Transaction ID or Payment ID is known.
- ChargeSale, ChargeAuthorize or ChargeConfirm has been called.

Primary Flow



1. Partner calls **GetPaymentStatus**with the Payment ID or the Partner Transaction ID.
2. vSafereturns the payment status.
3. Perform the appropriate next step based on the payment status:

Payment Status	Meaning	Next Step
-1	In-progress	Wait a few seconds and try **GetPaymentStatus** again. If the status continues to indicate in-progress retry **GetPaymentStatus** or take the following steps: 1) Resubmit the payment request with a new Payment ID 2) Create a background task to continually call**GetPaymentStatus** on the original transaction request every few seconds until a final status is received. a) If **GetPaymentStatus** returns a Payment Status 5 (Pended) or 10 (Successful) for the original transaction request, call **ReversePayment** using the Payment ID provided in the **GetPaymentStatus** response. This will avoid duplicate charges to the customer.
1	Bank Denied	Notify the customer of the denied status.

Payment Status	Meaning	Next Step
2	vSafe Pended	Go to use case: Resolve a Pended Payment.
3	vSafe Denied	Notify the customer of the denied status.
4	Transaction Cancelled	Notify the customer of the cancelled status.
5	Authorized	Complete the transaction using Primary Flow, Part 2: Confirm Payment.
10	Successful Payment	Notify the customer of the completed status.
16 105	Transaction Expired Transaction Failed, payment not taken	Do one of the following: • Re-authorize the payment with a new Transaction ID using **ChargeAuthorize**. • Submit the payment with a new Transaction ID using**ChargeSale**. • Notify the customer of the failed status.

Get System Status

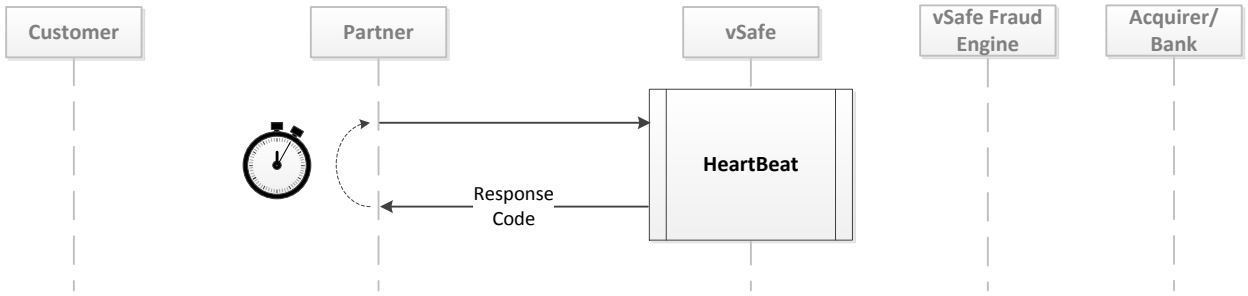
Description

Use this to check if vSafe is online.

Preconditions

- None

Primary Flow



1. Partner calls **HeartBeat**. No parameters needed.
2. vSafe returns the system status in the Response Code:
    a. 0 (Normal operation)
    b. 1 (System is not available)

Best Practices

- Use HeartBeat periodically and concurrently with your normal payment processing activity.
- Typical use is polling every 20-30 seconds or at your discretion.

Create a Permanent Token

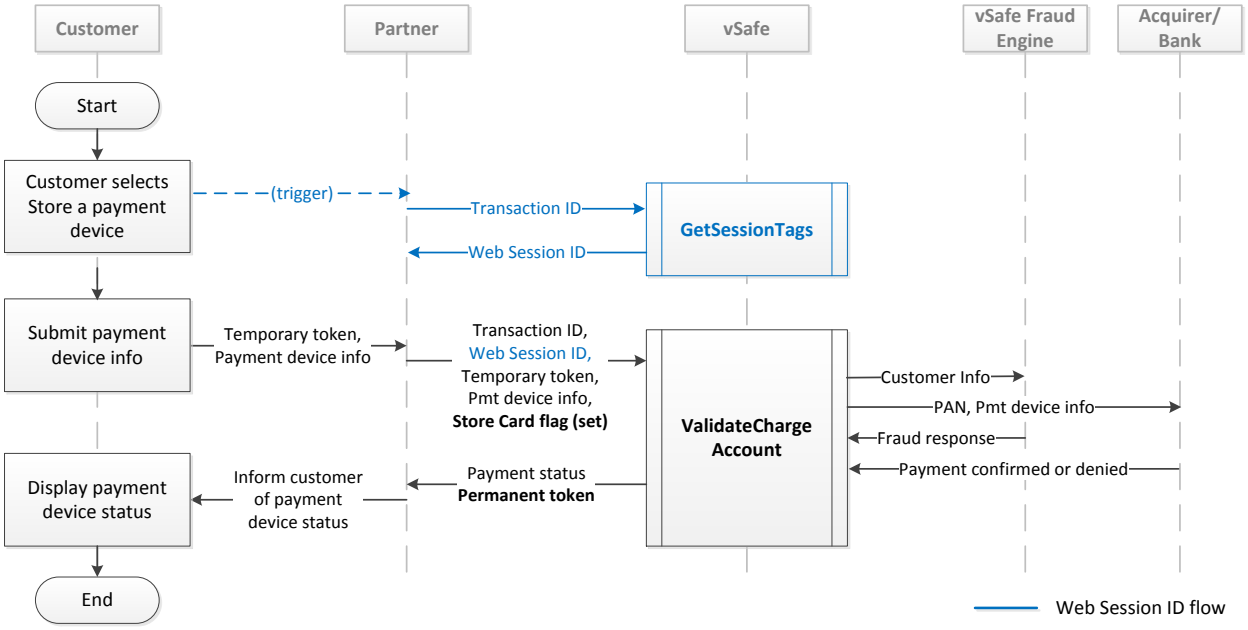
Description

Use this to create a token that doesn't expire, e.g., for customers who engage repeat business.

Preconditions

- Customer has a payment card to store for future use.
- You are using vSafe's PCI JavaScript Library at the customer's browser to intercept the PAN and inject a temporary token.

Primary Flow



1. Customer provides a payment device information.
2. Partner gets session tags for web device fingerprinting (blue flow lines).
    a. Partner calls **GetSessionTags** with the Transaction ID.
    b. vSafe returns the Web Session ID.
3. Get a permanent token.
    a. Partner calls **ValidateChargeAccount**with the following parameters:
        - Web session ID (required for web or mobile transactions)
        - Temporary token
        - Card information (cardholder's name and address, card expiration date, CVN)
        - Payment source (web, phone, or prearranged)
        - Risk information (an XML string that describes the details of the purchase)
        - Store Card flag is **set**, indicating that a permanent token is requested
    b. vSafe contacts the bank for authorization.
    c. vSafe evaluates the fraud risk.

d. vSafe returns the authorization request status:
   - 1 (Bank Denied)
   - 3 (vSafe Denied)
   - 10 (Success)

e. If the authorization request is successful, vSafe returns permanent token in **ValidateChargeAccount**response parameters.

Best Practices

- If the payment status is not returned in the expected time, refer to Get Payment Status.

Validate a Card

Description

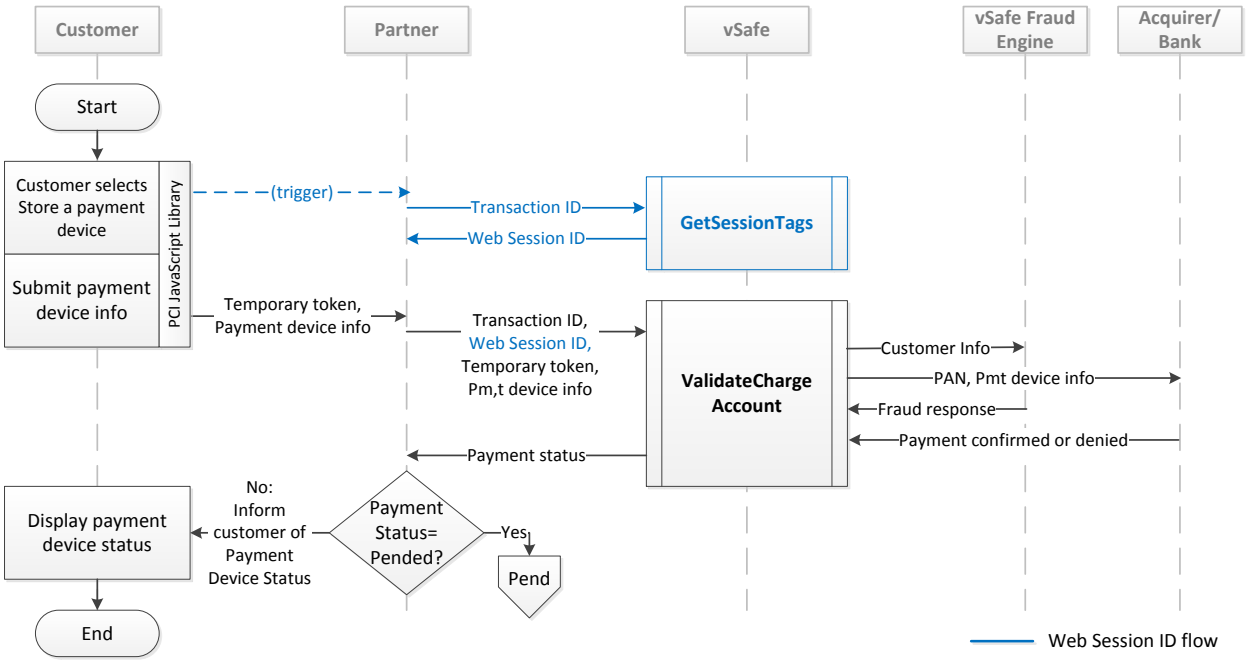
This use case determines if a payment card is valid and can be processed.It also is used to add a new payment card without charging anything against it.

Preconditions

- Customer has card information.

Primary Flow

The guaranteed payments service for web-based transactions require session tags for processing. This is done using the GetSessionTags API call (shown in blue).



1. Partner gets the current Web Session ID for the customer (blue flow lines). This is required if the payment has been taken via the web, mobile web or mobile application. This is captured during the current customer session, before the card information is collected. It is optional for non-indemnified partners.
    a. Partner calls **GetSessionTags** with the Transaction ID.
    b. vSafe returns the customer's Web Session ID.(The call retrieves this information from the PCI JavaScript on the customer's web browser.)
2. Validate the card information.
    a. Partner captures card information, customer's name and address from customer.
    b. Partner calls **ValidateChargeAccount**with the following in the request:
        - Card and customer information
        - Web Session ID
        - Temporary token
        - StoreCard flag is set
    c. vSafe contacts the bank for authorization.
    d. vSafe evaluates the fraud risk.
    e. vSafe providesthis information in the response:
    f. Payment status:
        - 1 (Bank Denied)
        - 2 (vSafe Pended)
        - 3 (vSafe Denied)
        - 10 (Successful Validation)
    g. If Payment status was successful, and if StoreCard is set, vSafe returns a permanent token.
    h. If the acquirer returns an NSF, the authorization result code will contain 1(Non-sufficient funds).
3. Partner informs customer of card status.

Key Business Rules

- The credit card number, expiration date, and CVN must be valid.
- GetSessionTags is optional for the non-indemnified service.

Best Practices

- If the payment status is not returned in the expected time, refer to Get Payment Status.

Recurring Payments

Validate Payment Device

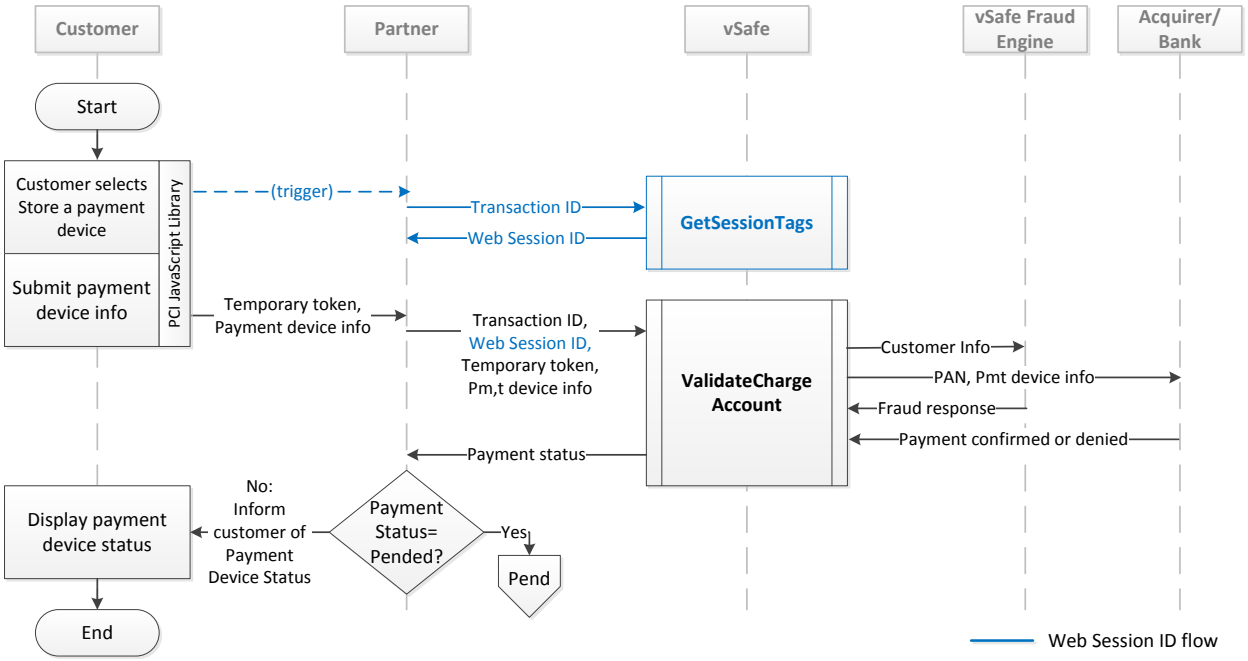
Description

Use this to determine if a customer's payment device is eligible for recurring paymentsas an initial step prior to setting up automatic payments.This is a preemptive measure to prevent enrolling payment devices that are not eligible for automatic payments, such as prepaid cards.

Preconditions

- Customer provides a payment device for use in automatic payments.

Primary Flow



1. Partner gets the current Web Session ID for the customer (blue flow lines). This is required if you are using the guaranteed payments service, and the payment has been taken via the web or mobile.
    a. Partner calls **GetSessionTags** with the Transaction ID.
    b. vSafe returns the customer's Web Session ID.(The call retrieves this information from the PCI JavaScript on the customer's web browser.)
2. Partner validates a payment device with the acquirer.
    a. Partner calls **ValidateChargeAccount**with the following parameters:
        - Web session ID (required for web or mobile transactions)
        - Temporary token
        - Card information (cardholder's name and address, card expiration date, CVN)
        - Payment source (set to "PPD", a prearranged transaction)
        - Store Card flag is **set**, indicating that a permanent token is requested

b. vSafe contacts the acquirer for authorization.
c. vSafe evaluates the fraud risk.
d. vSafe returns the authorization request status:
   - 10 (Success)
   - 1 (Bank Declined)
   - 3 (vSafe Denied)
e. If the authorization status = 10 (Success), vSafe returns permanent token.

3. Partner communicates outcome to the customer.

Key Business Rules

- If the validation fails, then the payment device should not be used for recurring payments.

Best Practices

- The permanent token can be used for immediate and recurring payments.
- The next step is to set up automatic payments.
- If the payment status is not returned in the expected time, refer to Get Payment Status.

Set Up Automatic Payments

Description

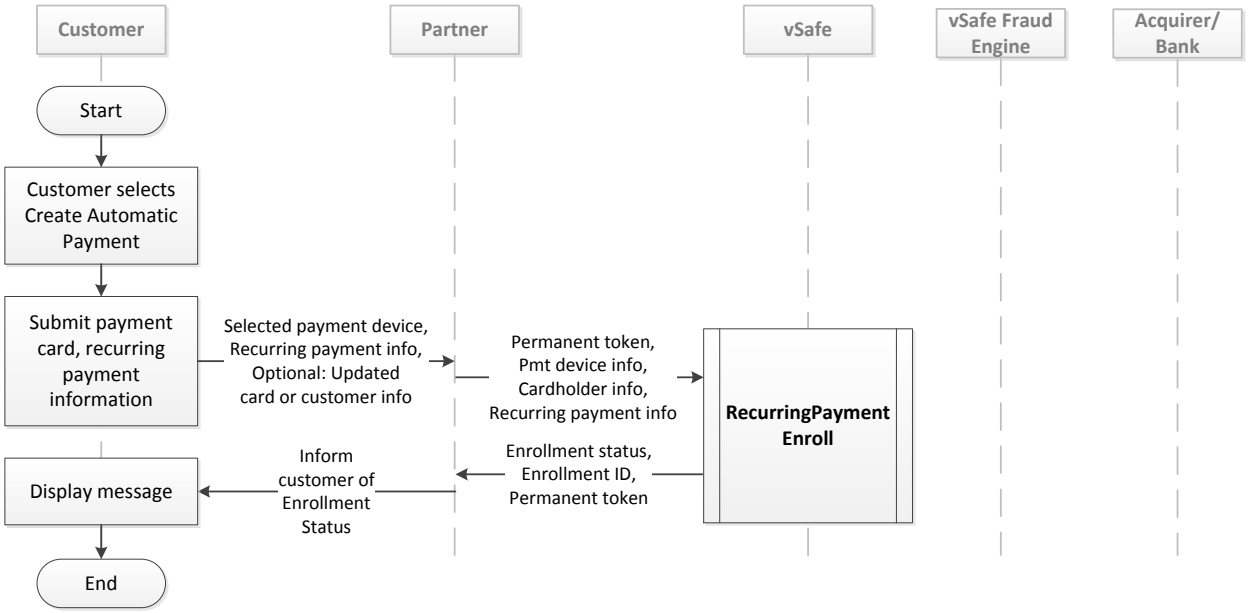
This sets up an automated payment process for a customer.

Preconditions

- Customer presents a payment device, or has an account and charge card set up with the partner (a permanent token will be used).

Primary Flow: Customer Selects a Stored Payment Device

Customer selects a stored payment device instead of typing in information for a new one.

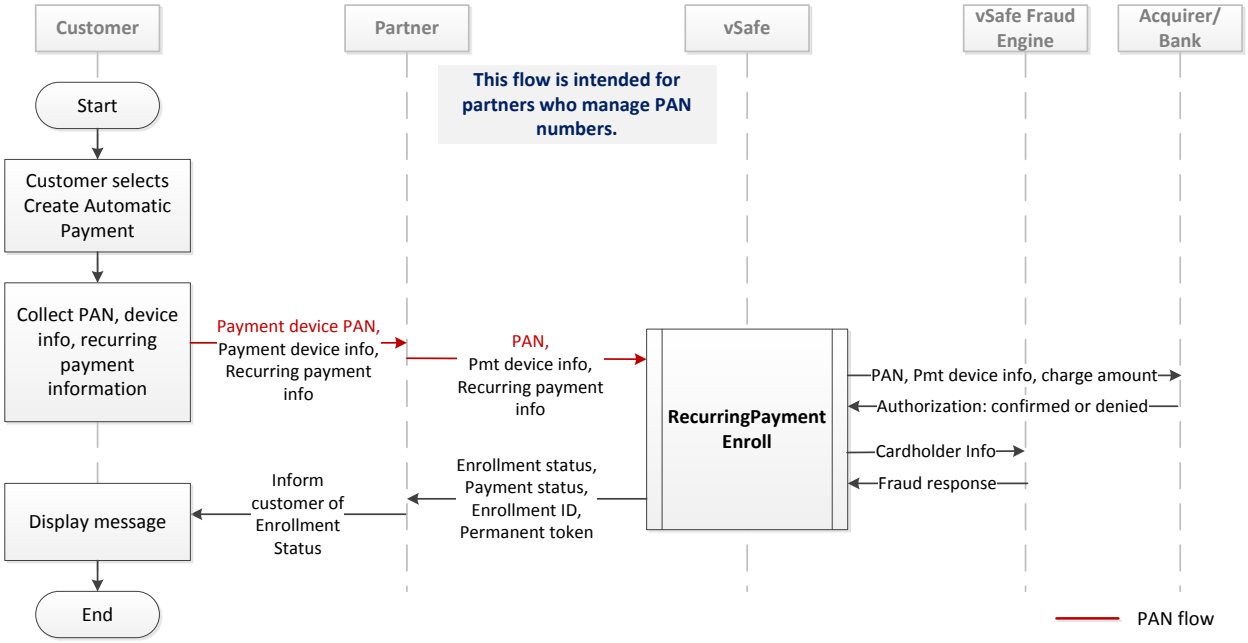


1. Partner captures the permanent token associated with the stored card, charge amount, recurring frequency and start date from the customer.
    a. The partner may optionally capture updated customer and/or updated card information. Otherwise the partner uses the stored customer and card information.
2. Partner calls **RecurringPaymentEnroll** with the following information:
    - Permanent token
    - Customer information (name, address)
    - Card information (permanent token, card expiration date, CVN)
    - Recurring payment information (start date, payment frequency, payment amount).
3. vSafe returns the enrollment information in the **RecurringPaymentEnroll** response:
    - Enrollment status (enrolled or not)
    - Enrollment ID (if enrolled)
    - Confirmation of start date
    - Confirmation of payment frequency (weekly or monthly)
    - Confirmation of the payment amount

- Permanent token (If the ChargeAccountNumberIndicatorhas a value of 1 or 2, this will return the permanent token.)
- Enrollment ID. This number uniquely identifies this enrollment for any subsequent automatic payment activity.

4. Partner communicates outcome to the customer.

Alternate Flow: Customer Provides New Payment Device, Take PAN



1. Partner captures customer information, card information, charge amount, recurring frequency and start date from the customer.
2. Partner calls **RecurringPaymentEnroll** with the following information:
    a. Card information (cardholder's name and address, card number, card expiration date, CVN)
    b. Recurring payment information (start date, payment frequency, payment amount).
3. vSafe evaluates the fraud risk (indemnified only).
4. vSafe contacts the bank for authorization.
5. vSafe returns the enrollment information in the **RecurringPaymentEnroll** response:
    a. Enrollment status (enrolled or not)
    b. Enrollment ID (if enrolled)
    c. Payment status:
        - 1 (Bank Denied)
        - 3 (vSafe Denied)
        - 5 (Authorized)
        - 6 (Not Authorized)
    d. Confirmation of start date
    e. Confirmation of payment frequency (weekly or monthly)
    f. Confirmation of the payment amount

g. Permanent token (If the ChargeAccountNumberIndicator has a value of 1 or 2, this will return the permanent token.)

6. Partner communicates outcome to the customer.

Key Business Rules

- Start date must be in the future.
- The day number must be in the range 1-28.
- The credit card number must be valid:
  - The number contains 13-19 digits.
  - It passes a Mod-10 test.
  - It must be chargeable on cc networks.
  - The card type used must match the value set in ChargeAccountNumberIndicator.
- The credit card expiration date must be valid, and set in the future.
- Payment amount must be greater than $0.00.
- When an automatic payment is made, vSafe sends a **GetPaymentStatus** response message through a URL endpoint, if configured.
- When vSafe processes a recurring payment, the response is automatically generated and sent to the partner. The response contains the status (success or denied) of a recurring payment. This is actively sent to the partner after each payment is processed.
- If a denial occurs, vSafe does not re-attempt the transaction; a single payment attempt is made for each scheduled payment.
- If enrolled with a permanent token, no payment status is returned.

Best Practices

- If a bank decline occurs for a scheduled payment, it is up to the partner to notify the customer and determine the proper course of action:
  - Cancel the service and unenroll the customer
  - Re-attempt a single payment using **ChargeSale**
  - Do nothing and wait for the next scheduled payment
- If the payment status is not returned in the expected time, refer to Get Payment Status.

Check Automated Payment Enrollment Status

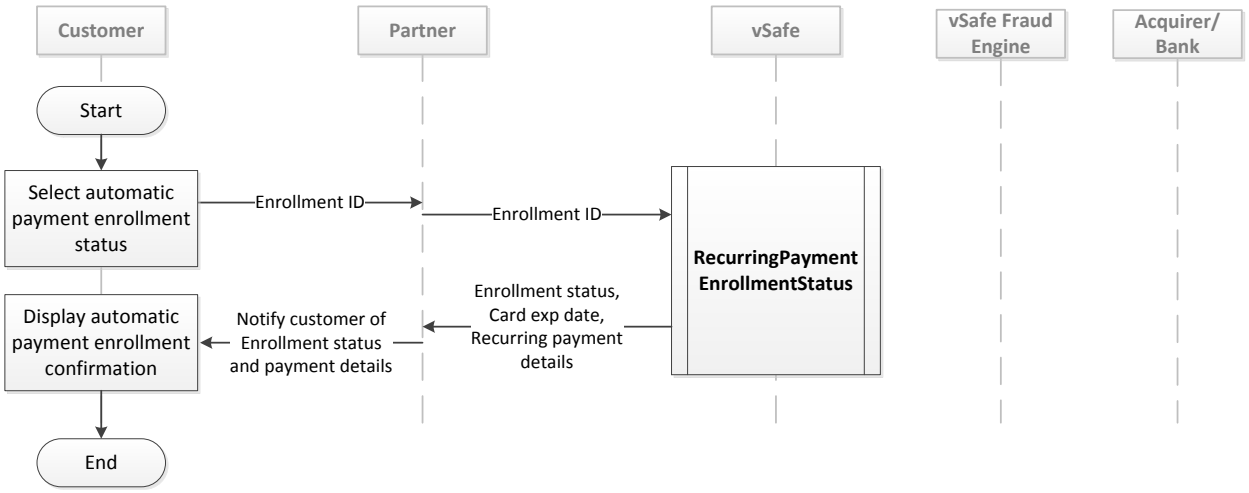
Description

The customer wants to verify that the automated payments are set up.

Preconditions

- The enrollment ID is known.

Primary Flow



1. Partner calls **RecurringPaymentEnrollmentStatus** with the enrollment ID.
2. vSafe finds the automatic payment enrollment information and puts it in the **RecurringPaymentEnrollmentStatus** response:
    a. Enrollment Status:
        - 0 (Enrolled)
        - 1 (Unenrolled)
    a. Charge Account last-four-digits
    b. Recurring Payment Start Date
    c. Next Payment Date
    d. Payment Amount
    b. Recurring Type:
        - 1 (Weekly)
        - 2 (Monthly)
3. Partner communicates outcome to the customer.

Key Business Rules

- The Enrollment ID must be valid.

Best Practices

- The customer UI can present a list of enrollments. This would prevent a user from typing an incorrect enrollment ID.

CheckAutomatic Payment Status

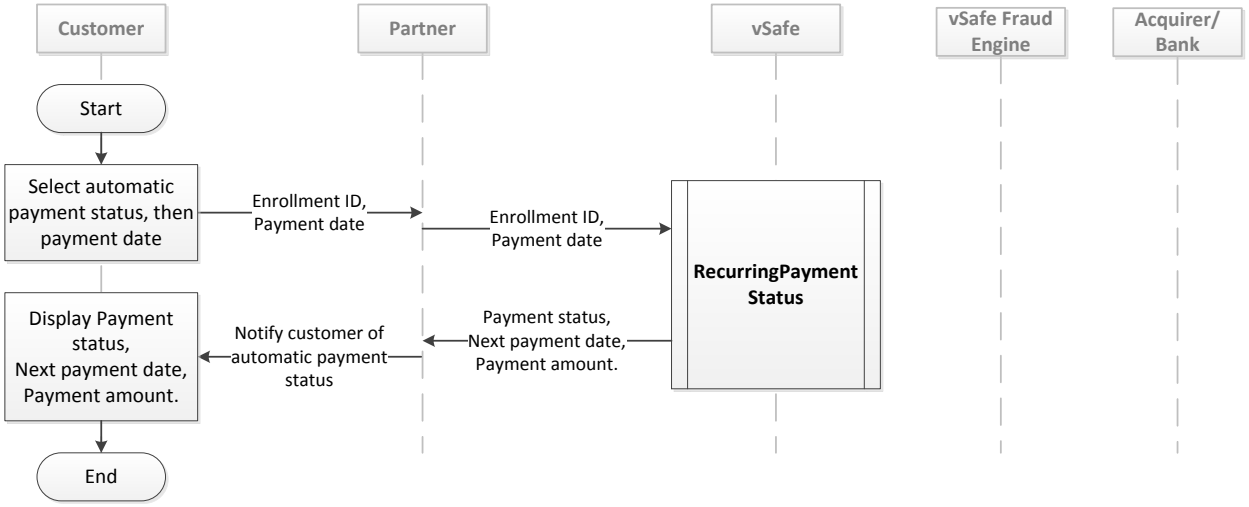
Description

The customer wants to check if an automatic payment has been made.

Preconditions

- The enrollment ID is known.
- Partner or customer knows when the payment should have occurred.

Primary Flow



1. Partner collects the date of the automatic payment from the customer.
2. Partner calls **RecurringPaymentStatus** with the enrollment ID and the payment date.
3. vSafe looks up the automatic payment enrollment and responds with the enrollment information in the **RecurringPaymentStatus** response message.
4. vSafe puts the payment information in the **RecurringPaymentStatus** response:
    a. Payment status:
        - 1 (Bank Denied)
        - 3 (vSafe Denied)
        - 10 (Successful Payment)
    b. Next scheduled payment date
    c. Payment amount
5. Partner communicates outcome to the customer

Key Business Rules

- Enrollment ID must be valid.
- Payment date must be in the past.

Update an Automated Payment

Description

The customer wants to make a change to an existing automated payment.

Preconditions

- Automated payment exists.
- The enrollment ID is known.

Primary Flow

1. Cancel the currently enrolled automatic payment. See Cancel Automatic Payments.
2. Schedule a new automatic payment. SeeGet Payment Status.

Cancel Automatic Payments

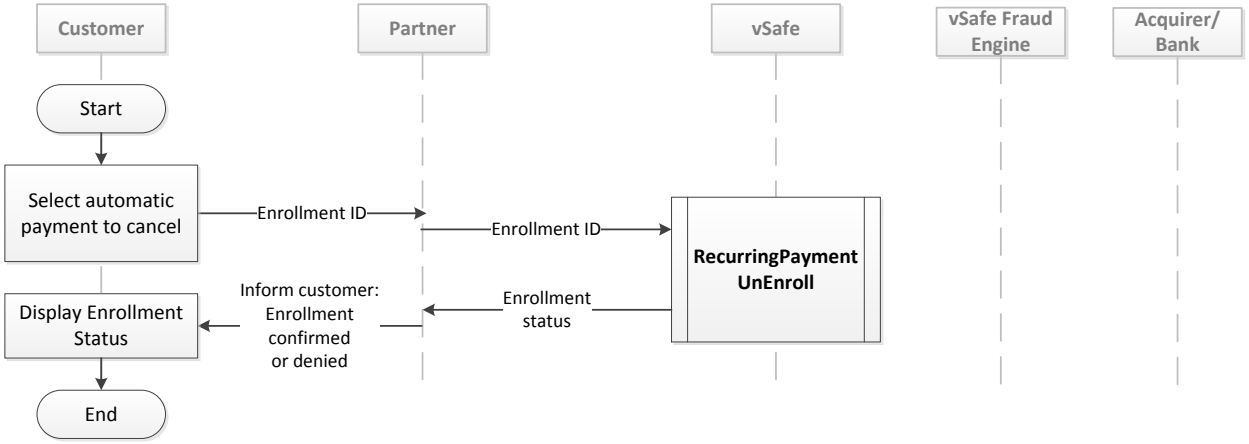
Description

Use this to cancel enrollment in recurring payments.

Preconditions

- Automatic payments have been set up.
- The enrollment ID is known.

Primary Flow



1. Customer selects the automatic payment from a list.
2. Partner calls **RecurringPaymentUnenroll** with the enrollment ID.
3. vSafe cancels the automatic payment enrollment and responds with the enrollment status in the **RecurringPaymentUnenroll** response message.
4. Partner communicates outcome to the customer.

Key Business Rules

- The Enrollment ID must be valid.

Best Practices

- Unenroll at least one business day ahead of the next scheduled payment. If an unenrollment is performed on a scheduled payment day, the payment may not be cancelled by the unenrollment.

Wallet

A wallet provides a method to consolidate all customer payment device information to a single reference ID (the wallet ID). When payment devices are associated with a wallet, a partner does not need to store and manage this information for each customer. Wallet APIs let you create a wallet, and add, modify and remove payment devices. Each wallet call also returns a detailed list of all current payment devices associated with a wallet ID.

Create a Customer Wallet

Description

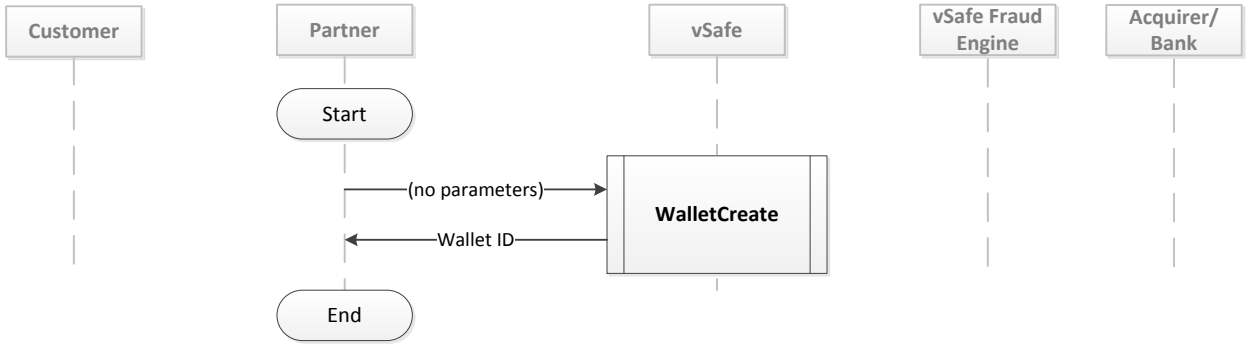
Use this to create a wallet to associate multiple payment devices, such as credit and debit cards, for a customer.Each wallet has a unique ID, which you can associate with a customer account. Once a wallet is created, you can add, modify or delete payment devices for the wallet.

Creating a wallet is the first step when a customer presentsa new payment device.

Preconditions

n/a

Primary Flow



1. Partner calls **WalletCreate** (only your login credentials are required).
2. vSafe returns with a confirmation status and a wallet ID.

Key Business Rules

- Once the wallet has been created, the next step is to associate the wallet ID with a customer.
- If a wallet is empty for 30 days, it is automatically deleted.

Best Practices

- **Note**: Be careful when associating a new wallet ID for an intended customer. If a new wallet ID is associated with an unintended customer, any new payment devices associated with that wallet ID would become visible to another customer. This could result in unauthorized charges, subsequent chargebacks, and unhappy customers.
- The wallet is a convenience device to associate multiple payment devices together.
- The partner is responsible for associating the walletID with a customer's account.
- A customer can be assigned more than one wallet for keeping various payment devices separate.

Add a New Payment Method to aWallet

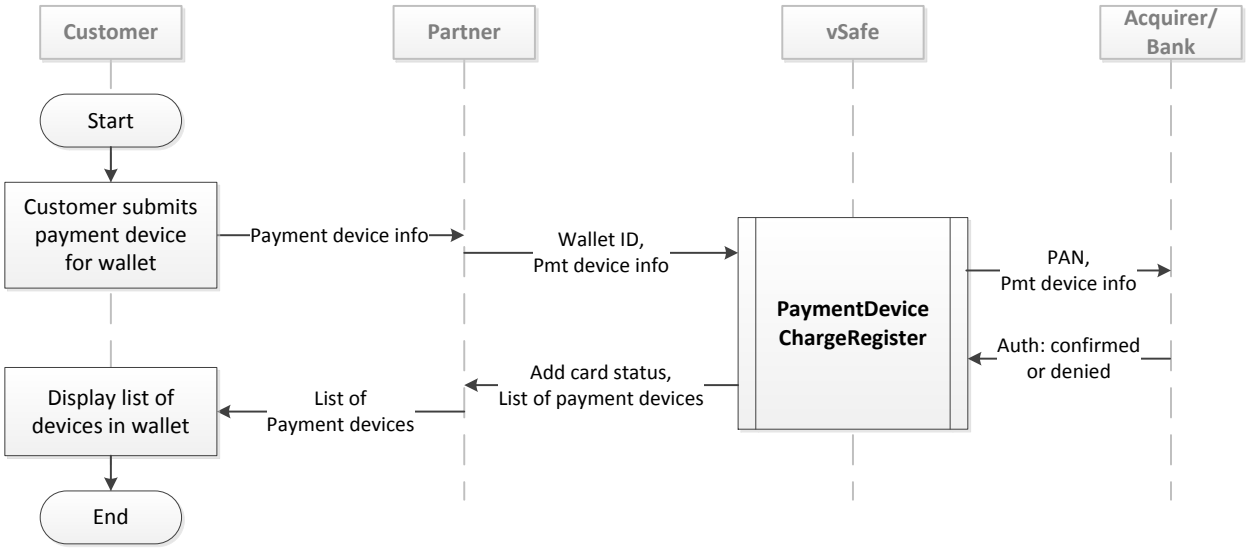
Description

Use this to add a new payment device to a wallet, and get a detailed list of all payment devices in the wallet.

Preconditions

- Customer has a wallet
- Customer provides a new payment device to store in a wallet

Primary Flow



1. Customer provides payment device information to add to their wallet.
2. Partner calls **PaymentDeviceChargeRegister** with the following parameters:
   - Wallet ID
   - Payment device PAN or temporary token
   - PAN/tokenidentifier (1=credit or debit card number, 2=temporary token, 3=permanent token)
   - Payment device expiration date
   - Payment device description (a short, familiar name for the card)
   - Cardholder information (name, address, city, region, postal code, country code)
3. vSafecontacts the acquirer for authorization.

4. The authorization status is returned to vSafe.
    - If the authorization is approved (ResponseCode=0), the payment device is added to the wallet.
    - If the authorization is denied (ResponseCode=1006), the payment device is not added to the wallet.
5. vSafereturns the confirmation status to the partner, and an updated list of payment devices in the wallet. Each list item contains:
    - Temporary token for the payment device
    - Payment device last-four PAN digits
    - Payment device expiration date
    - Payment device type (e.g., Visa, MasterCard, Discover)
    - Cardholder's info (name, address, city, region, postal code, country code)
    - Payment device description
6. Partner presents a formatted list of payment devices for the customer, and the payment device confirmation status.

Key Business Rules

- This use case may be run any number of times to add new payment devices to the wallet.
- Each payment device in the list has a new temporary token, which can be used for purchases in the current session.

Best Practices

- **Note**: Be careful when associating a new payment device to a wallet ID for an intended customer. If a new payment device is mistakenly associated with the wallet ID of an unintended customer, it could allow unauthorized charges by other customers, subsequent chargebacks, and unhappy customers.
- Provide an exception condition for customer attempting to add a duplicate payment device. If a duplicate payment device PAN is entered, it the device information is treated as an update,not an addition.

List Payment Devices in a Wallet

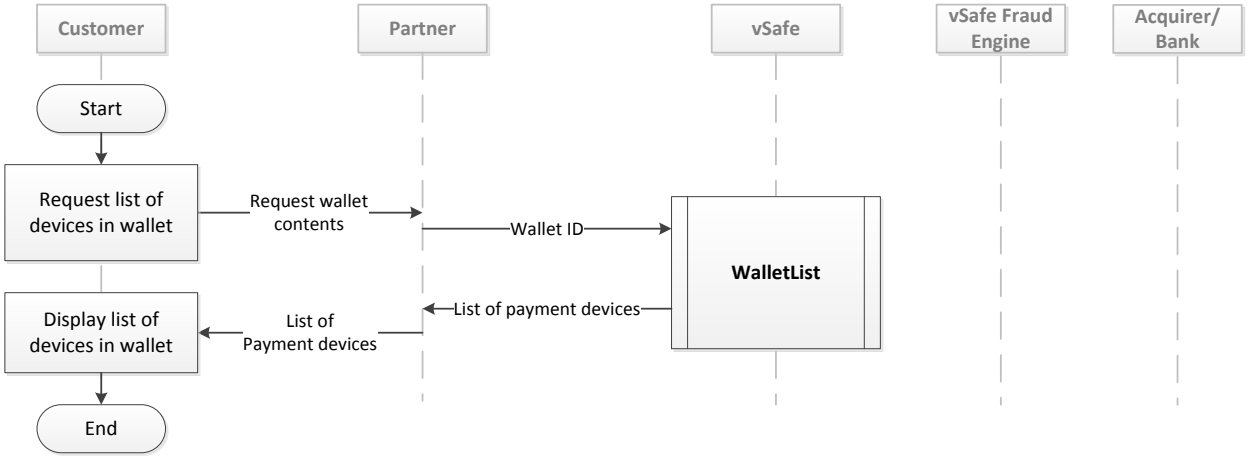
Description

Use this to get a detailed list of all payment devices in the wallet.

Preconditions

- Wallet exists
- Partner provides a method for customer to retrieve list of payment devices in their wallet.

Primary Flow



1. Customer requests a list of stored payment devices in the wallet.
2. Partner calls **WalletList** with the customer's wallet ID.
3. vSafe responds with a list of payment devices. Each list item contains:
   - Temporary token for the payment device
   - Payment device last-four PAN digits
   - Payment device expiration date
   - Payment device type (e.g., Visa, MasterCard, Discover)
   - Cardholder's info (name, address, city, region, postal code, country code)
   - Payment device description (a short, familiar name for the card)
4. Partner presents a formatted list of payment devices for the customer.

Key Business Rules

- Each payment device in the list has a new temporary token. This can be used for purchases during the current session.
- If the wallet has no payment devices, vSafe returns an empty list.

Best Practices

- Create an exception condition where no payment devices exist in the wallet, and provide a friendly message to the customer.

Update Information in aWallet

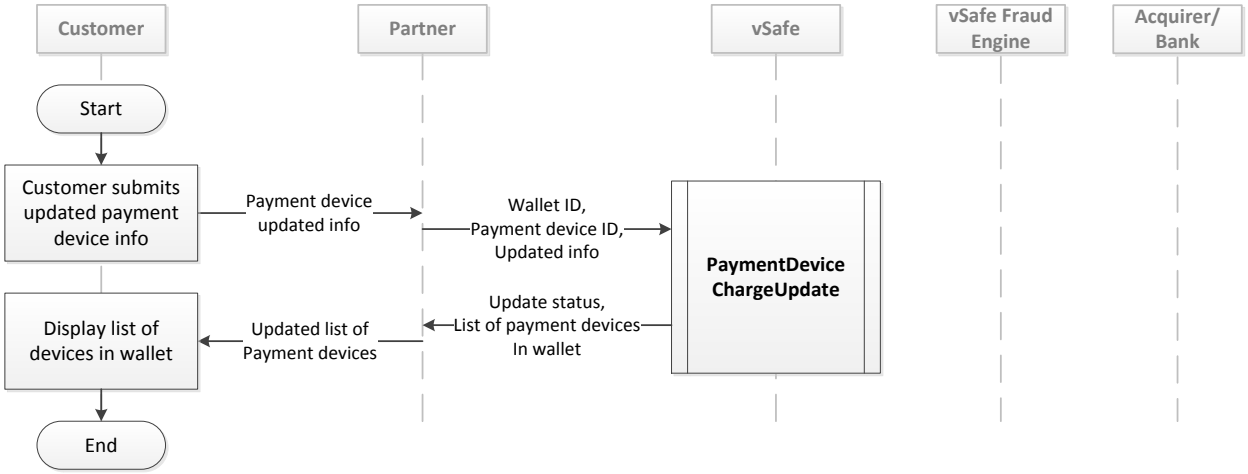
Description

Use this to update information such as name, address or card expiration date, on an existing payment device, and get a list of all payment devices in the wallet.

Preconditions

- Customer's wallet exists
- Customer's wallet has at least one payment device
- Partner provides a method for customer to select an existing payment device from their wallet
- Partner provides a method to collect new card or cardholder information from the customer

Primary Flow



1. Customer selects an existing payment card from their wallet.
2. Customer provides new card or cardholder information.
3. Partner calls **PaymentDeviceChargeUpdate** with the following information:
   - Customer's wallet ID
   - Payment device ID to be updated
   - Payment device last-four PAN digits
   - Payment device expiration date
   - Cardholder's info (name, address, city, region, postal code, country code)
   - Payment device description (a short, familiar name for the card)
4. vSafe responds with a confirmation status, and an updated list of payment devices in the wallet. Each list item contains:
   - Temporary token for the payment device
   - Payment device last-four PAN digits
   - Payment device expiration date
   - Payment device type (e.g., Visa, MasterCard, Discover)
   - Cardholder's info (name, address, city, region, postal code, country code)
   - Payment device description

5. Partner presents a list of payment devices for the customer, along with a confirmation status of the updated payment device.

Key Business Rules

- The payment device ID can be retrieved using any of these wallet API calls:
  - WalletList
  - PaymentDeviceChargeRegister
  - PaymentDeviceChargeUpdate
  - PaymentDeviceOnFileRemove
- Each payment device in the list has a new temporary token, which can be used for purchases in the current session.

A temporary token can be converted to a permanent token. See the Get System Status

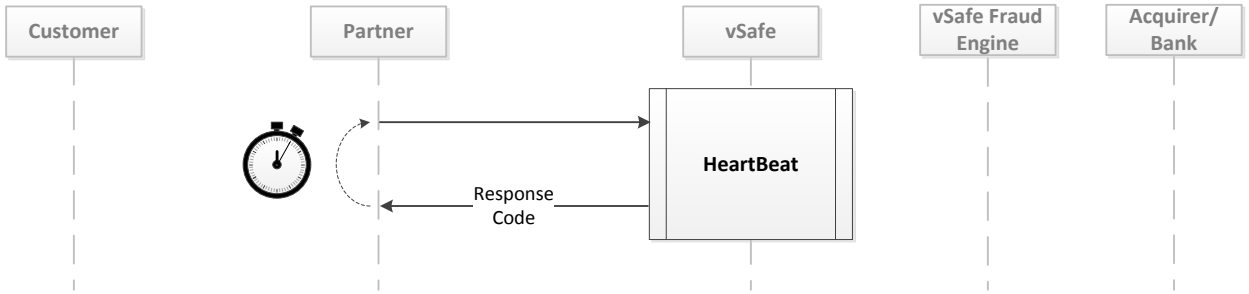
Description

Use this to check if vSafe is online.

Preconditions

- None

Primary Flow



3. Partner calls **HeartBeat**. No parameters needed.
4. vSafe returns the system status in the Response Code:
    a. 0 (Normal operation)
    b. 1 (System is not available)

Best Practices

- Use HeartBeat periodically and concurrently with your normal payment processing activity.
- Typical use is polling every 20-30 seconds or at your discretion.
- Create a Permanent Token use case.
- If the PAN needs to be updated, this must be treated as a new payment device.

Best Practices

- Create an exception condition where no wallet exists for a customer.
- Create an exception condition where no payment devices are in a customer's wallet.

Remove a Stored Payment Method from aWallet

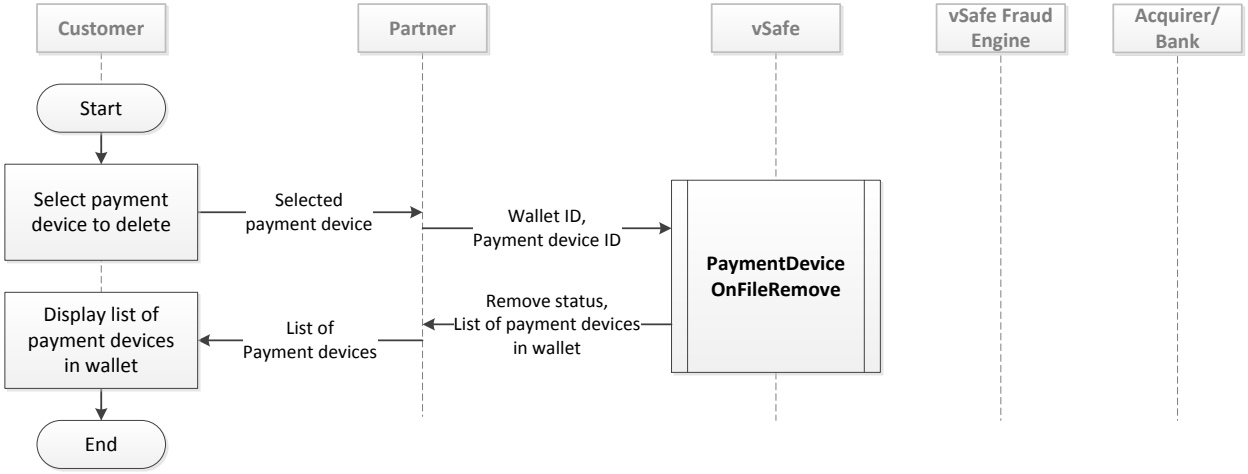
Description

Use this to remove a payment device from a customer's wallet, and get a list of all payment devices in the wallet.

Preconditions

- Wallet exists
- At least one payment method exists in the wallet
- Partner provides a method for customer to select a payment device from their wallet to remove.

Primary Flow



1. Customer selects a payment device from their wallet.
2. Partner calls **PaymentDeviceOnFileRemove** with the following information
   - Customer's wallet ID
   - Payment device ID to be removed
3. vSafe responds with a confirmation status, and an updated list of payment devices in the wallet. Each list item contains:
   - Temporary token for the payment device
   - Payment device last-four PAN digits
   - Payment device expiration date
   - Payment device type (e.g., Visa, MasterCard, Discover)
   - Cardholder's info (name, address, city, region, postal code, country code)
   - Payment device description
4. Partner presents a formatted list of payment devices for the customer, along with a confirmation status on the deleted payment device.

Key Business Rules

- This use case may be run any number of times to remove existing payment devices to the wallet.
- Each payment device in the list has a new temporary token, which can be used for purchases in the current session.
- If the wallet has no payment devices after the operation, vSafe returns an empty list.

Best Practices

- Create an exception condition where no payment devices exist in the wallet after processing, and provide a friendly message to the customer.

Reports

You can get a detailed daily report of all transaction activity. This is provided in a downloadable CSV file.

Generate a Detailed Transaction Report File

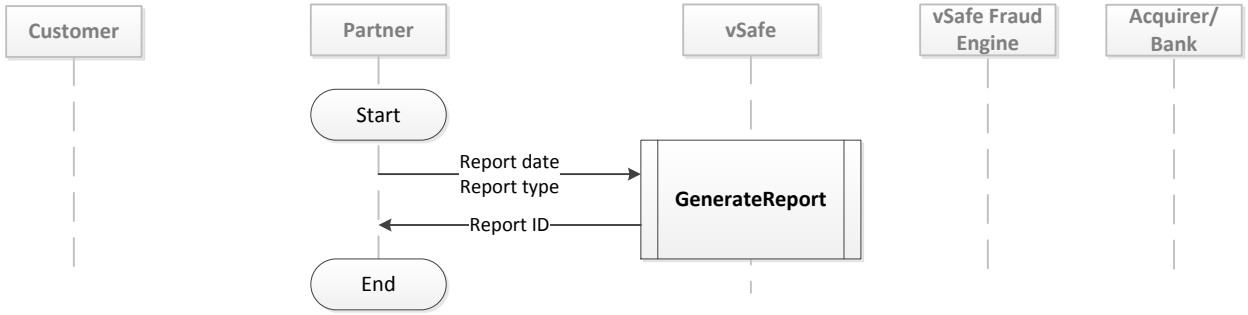
Description

This generates a detailed report for one days' worth of transactions. When generation is completed, a CSV-formatted report file is available for download.

Preconditions

- n/a

Primary Flow



1. Partner calls **GenerateReport** to generate a report.
   a. Provides date to report, type of report (daily)
2. vSafe responds with a ReportID.

Key Business Rules

- The requested report date must be retrievable; that is, in the past, but not before you started using vSafe.
- The report ID is used in subsequent requests for report generation status and downloading the report.

Best Practices

Report generation is not performed in real time, and may take an hour or more depending on server load.

Get the Status of a Report File

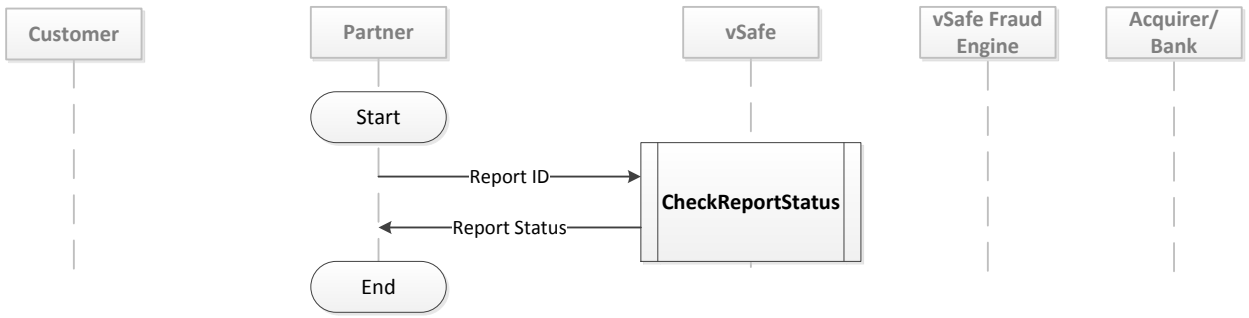
Description

This returns the status of a requested report file.

Preconditions

- Report generation has been initiated.

Primary Flow



1. Partner calls **CheckReportStatus** with the report ID created by GenerateReport.
2. vSafe responds with a ReportStatusCDvalue:
   - 0 (Queued) Processing has not started.
   - 1 (In Progress) Report has begun processing.
   - 2 (Complete) Report file completed processing; ready for download.
   - 3 (Failed) Report file could not be created.
   - 4 (Deleted) Report file has been deleted; automatically deleted after 48 hours.

Key Business Rules

What to do when ReportStatusCD indicates:

- Queued: Wait at least 15 minutes and then call **CheckReportStatus** again.
- In Progress: Wait at least 15 minutes and then call **CheckReportStatus** again.
- Complete: Use **DownloadReportFile** to retrieve the report CSV file.
- Failed: Use **CheckReportStatus** again. If you still get this status, contact vSafe support.
- Deleted: Use **GenerateReport** again, or contact vSafe support.

Best Practices

- If you do not receive a ReportStatusCD=0 (Queued) within two hours, call vSafe support.

Download a Report File

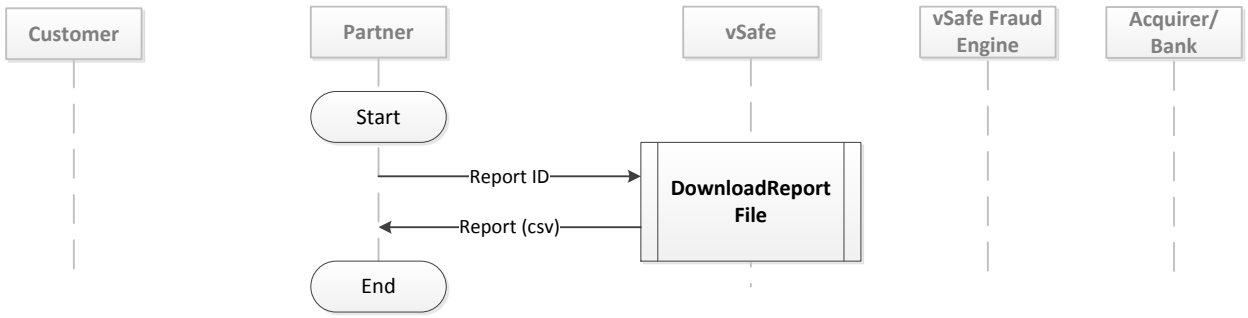
Description

This provides a link to download the report file. The file is in CSV format.

Preconditions

- A call to CheckReportStatus for your requested report returned a “Complete” code.

Primary Flow



1. Partner calls **DownloadReportFile** with the ReportID created by GenerateReport.
2. vSafe responds with a link to download the report file.

Key Business Rules

- Use CheckReportStatus before using this API. CheckStatusReport response should indicate “Complete” (the report is ready for download).

Best Practices

- Use this API to download the report within 48 hours of generation. It is automatically deleted after this period of time.

Non-Indemnified and Take PAN Flows

Process a Payment, Single Call, Non-Indemnified

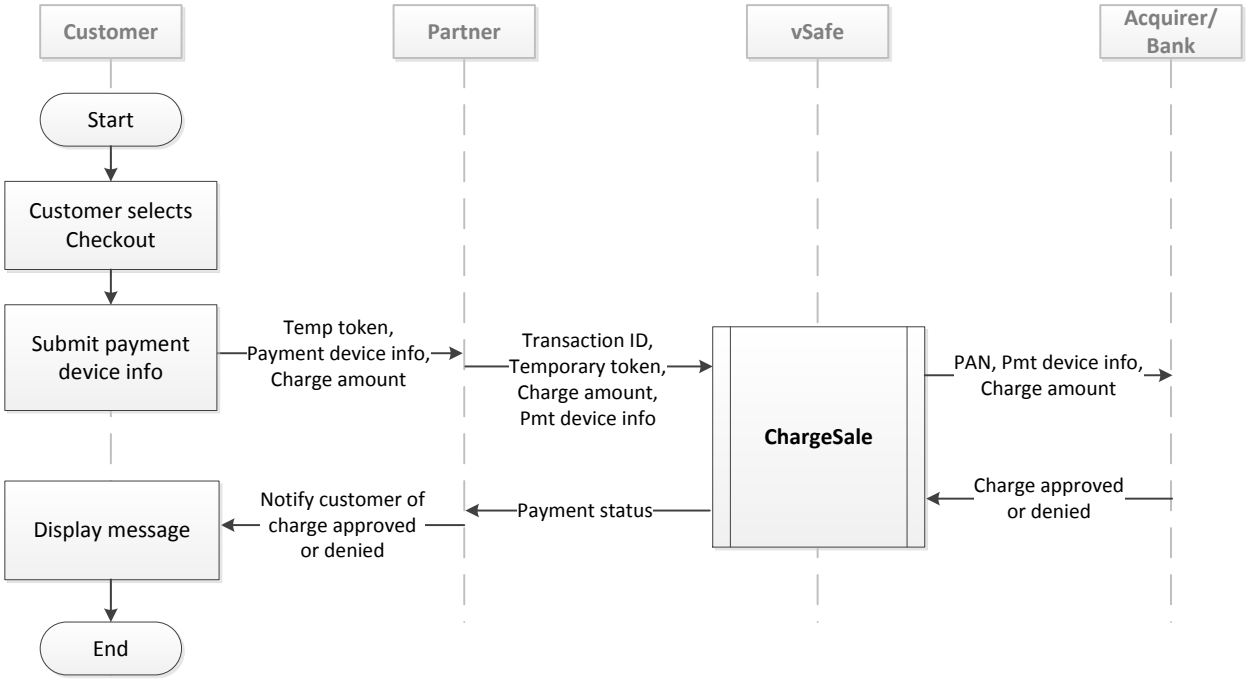
Description

This is a simple one-step process to take a payment.

Preconditions

- Customer has at least one item with a total due of more than $0.00 in the shopping cart.
- Customer has a payment card.

Primary Flow: Pass Token



1. Partner captures name, address, expiration date, CVN, etc. from the customer.
2. Create a temporary token.
    a. Partner captures card number from customer.
    b. Partner calls **ChargeAccountToTemporaryToken** with card number in the request parameters.
    c. vSafe returns temporary token in **ChargeAccountToTemporaryToken** response parameters.
3. Process the payment.
    a. Partner calls **ChargeSale**. With the following in the request parameters:
        - Temporary token
        - Card info (cardholder's name and address, card expiration date, CVN)
        - Charge amount

- Payment source (web, phone, or prearranged)

4. vSafe authorizes the transaction with bank.
5. vSafe responds with the following in the **ChargeSale** result message:
    a. Payment status:
        i. 1 (Bank Denied)
        ii. 3 (vSafe Denied)
        iii. 10 (Successful Payment)
    b. If the payment was successful, and if the StoreCard flag is set, vSafe returns a permanent token.
6. Partner communicates outcome to the customer.

Key Business Rules

- Charge amount must be greater than $0.00.
- Credit card number and expiration datemust be valid.

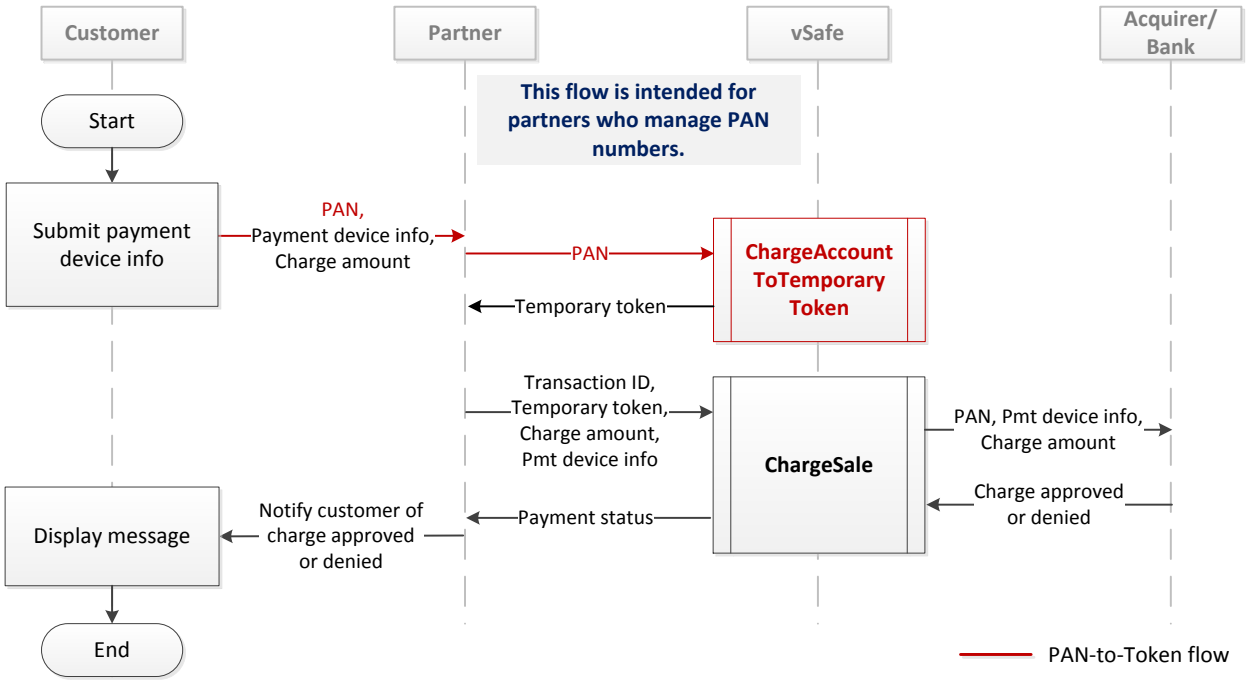
Best Practices

- If the payment status is not returned in the expected time, refer to Get Payment Status.
- Depending on the acquirer, a transaction may be approved even though the Card Validation Number (CVN) or the Address Verification Service (AVS) returned a non-match. If you decide to decline the transaction, refer to Refundor Void aCompleted Payment.

Alternate Flow: Take PAN

This alternate flow is the same as the primary flow except that a step is added to convert the incoming PAN into a temporary token. This flow assumes that the PAN is not being intercepted by other means, such as vSafe's PCI JavaScript Library at the customer's browser.

Note: Handling credit card numbers affects your PCI scope.



1. Partner captures cardholder information, payment card PAN, expiration date, CVN, etc. from the customer.
2. Create a temporary token (red flow lines).
    a. Partner calls **ChargeAccountToTemporaryToken** with payment card PAN in the request parameter.
    b. vSafe returns temporary token in **ChargeAccountToTemporaryToken** response parameters.
3. Process payment.
    a. Partner calls **ChargeSale**. With the following in the request parameters:
        - Temporary token
        - Card info (cardholder's name and address, card expiration date, CVN)
        - Charge amount
        - Payment source (web, phone, or prearranged)
    b. vSafe authorizes the transaction with bank.
    c. vSafe responds with the following in the **ChargeSale** result message:
        i. Payment status:
            - 1 (Bank Denied)
            - 3 (vSafe Denied)
            - 10 (Successful Payment)
    d. Partner communicates outcome to the customer.

Key Business Rules

- Charge amount must be greater than $0.00.
- Credit card number and expiration datemust be valid.

Best Practices

- If the payment status is not returned in the expected time, refer to Get Payment Status.
- Depending on the acquirer, a transaction may be approved even though the Card Validation Number (CVN) or the Address Verification Service (AVS) returned a non-match. If you decide to decline the transaction, refer to Refundor Void aCompleted Payment.

Process a Payment, Authorize and Confirm, Non-Indemnified

Description

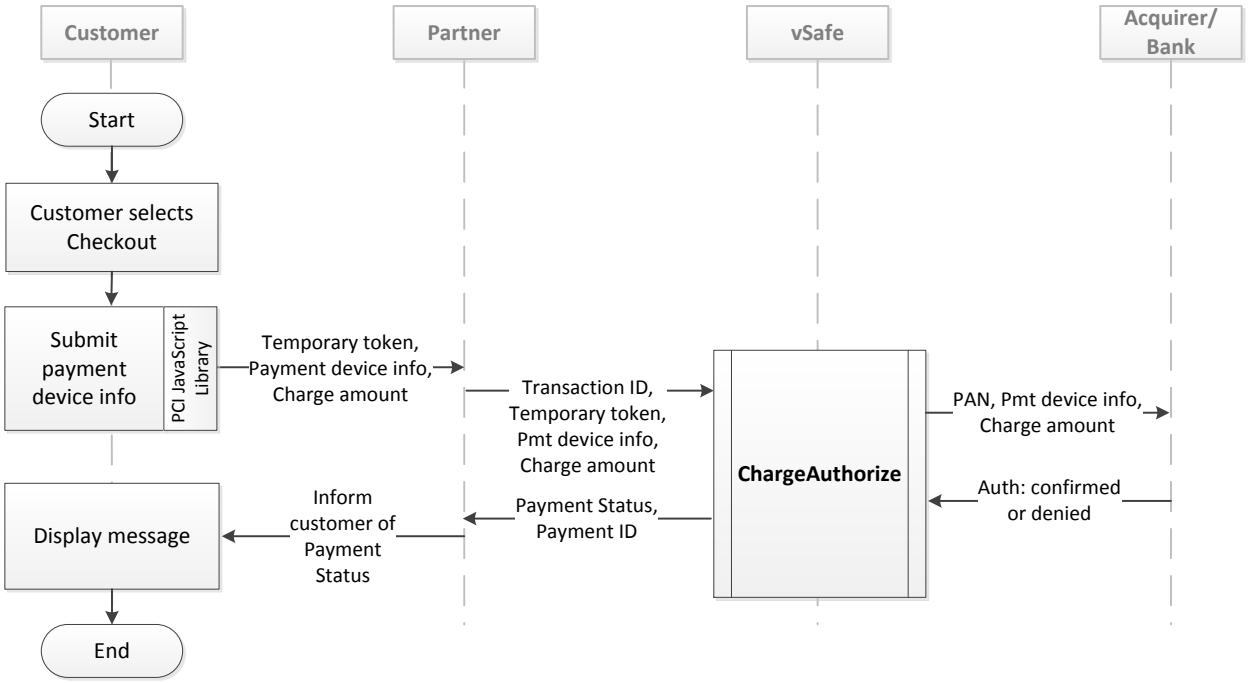
This use case is for transactions that require a business-related time delay between authorization and confirmation, such as between order processing and shipment.This process has two parts:

- Authorizing the payment.
- Confirming the payment, usually at fulfillment time. This is the step where the payment device is charged.

Preconditions

- Customer has at least one item with a total due of more than $0.00 in the shopping cart.
- Customer has a payment device.
- You are using vSafe's PCI JavaScript Library at the customer's browser to intercept the PAN.

Primary Flow, Part 1: Authorize Payment



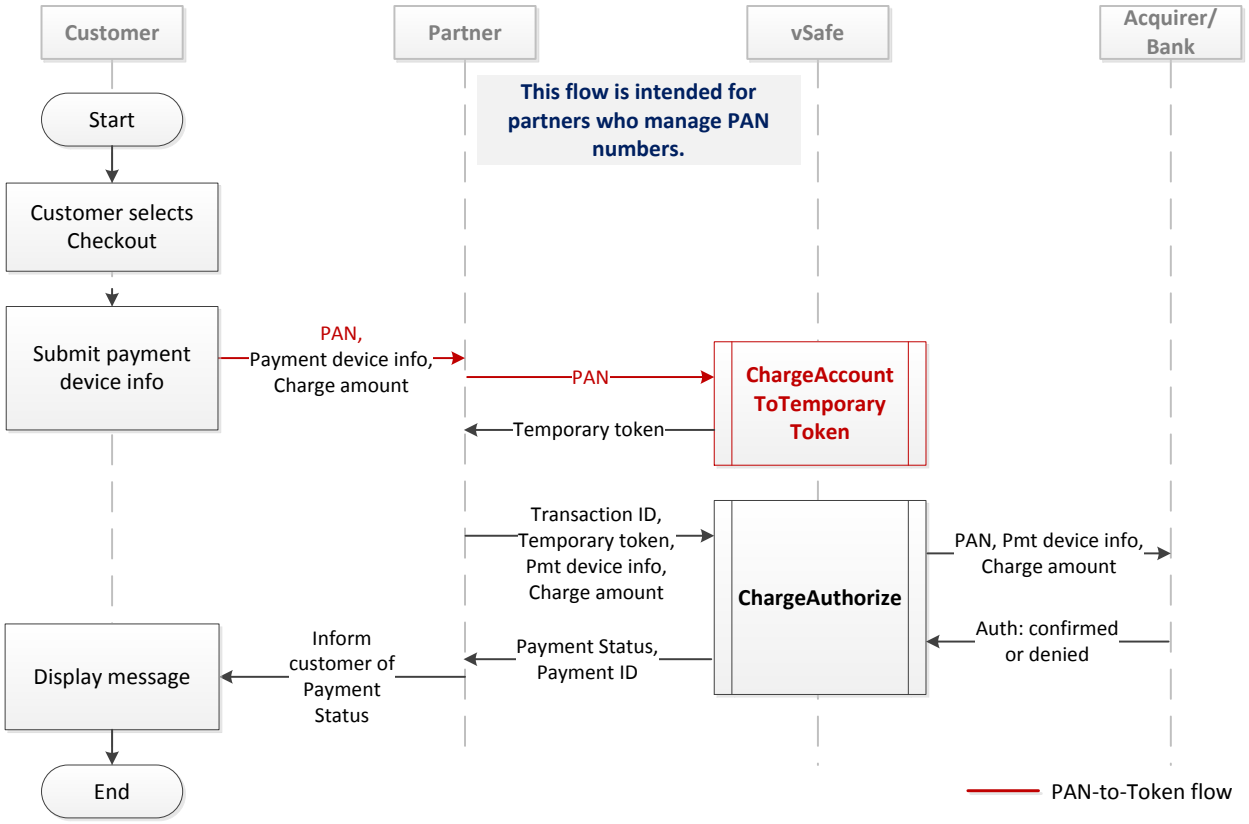
1. Customer selects Checkout.
2. Authorize the transaction.
    a. Partner captures name, address, expiration date, CVN, etc. from customer.
    b. Partner calls **ChargeAuthorize**. with the following parameters:
        - Temporary token
        - Card information (cardholder's name and address, card expiration date, CVN)
        - Charge amount
        - Payment source (web, phone, or prearranged)
    c. vSafe authorizes transaction with the bank.

d. vSafe evaluates fraud risk.
e. vSafe responds with payment outcome to **ChargeAuthorize** with payment status:
   - 1 (Bank Denied)
   - 3 (vSafe Denied)
   - 5 (Authorized)
f. If the payment is authorized and the StoreCard flag is set, vSafe returns a permanent token.
g. If the payment status value indicates that the purchase was denied (the payment status is 1), partner communicates outcome to the customer.

Alternate Flow, Part 1, Take PAN

This flow assumes that you are not using vSafe's PCI JavaScript Library at the customer's browser to intercept the PAN. It is identical to the Primary Flow, Part 1, except that it adds a call to **ChargeAccountToTemporaryToken** to exchange a PAN for a temporary token (shown in red).

Note: Handling credit card numbers affects your PCI scope.



1. Customer selects Checkout.
2. Authorize the transaction.
   a. Partner captures name, address, expiration date, CVN, etc. from customer.
   b. Partner calls **ChargeAuthorize**. with the following parameters:
      - Payment device PAN
      - Card information (cardholder's name and address, card expiration date, CVN)

- Charge amount
- Payment source (web, phone, or prearranged)

c. vSafe authorizes transaction with the bank.

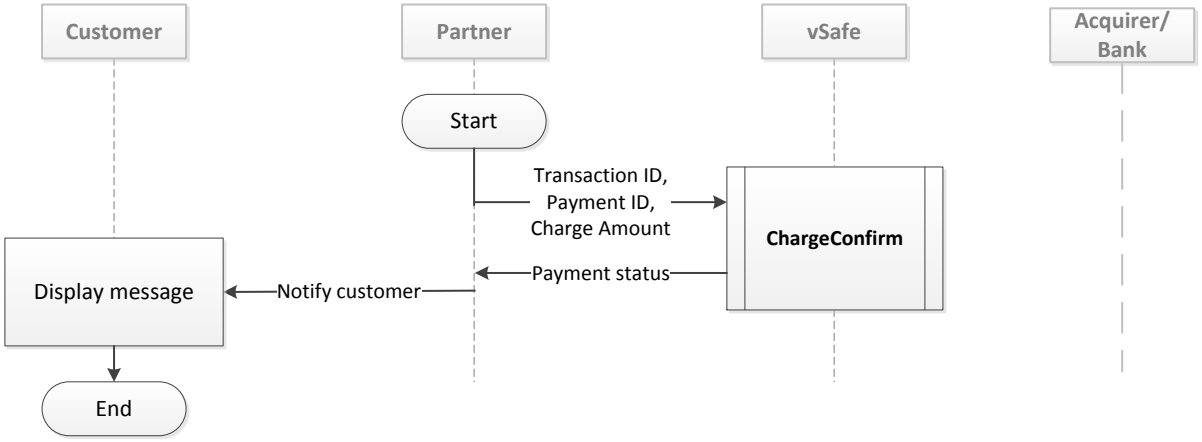
d. vSafe responds with payment outcome to **ChargeAuthorize** with payment status:

- 1 (Bank Denied)
- 3 (vSafe Denied)
- 5 (Authorized)

e. If the payment is authorized and the StoreCard flag is set, vSafe returns a permanent token.

f. If the payment status value indicates that the purchase was denied (the payment status is 1), partner communicates outcome to the customer.

Primary Flow, Part 2: Confirm Payment



3. Confirm the transaction (after the time delay).
   a. Partner calls **ChargeConfirm** with the payment ID created by **ChargeAuthorize**, and the charge amount.
   b. vSafe performs the following:
      i. Check that the amount is greater than $0.00 and is less than the authorized amount.
      ii. Check that the authorization hasn't expired.
      iii. Contact the acquirer with the payment card information and the confirmed charge amount.
      iv. If the authorization hasn't expired, vSafe returns a Payment status of 10 (Completed payment) in the ChargeConfirm response message.
4. Partner notifies the customer of the outcome.

Key Business Rules

- Charge amount to be authorized must be > $0 and ≤ the authorized amount.
- A charge must be confirmed within a predefined time frame set by the bank that issued the customer's card.
- An authorized charge can be confirmed only once.

Best Practices

- Use the **ChargeAuthorize** when you want to authorize the funds, but not actually charge the credit card until a future date (e.g., when an item ships or fulfillment occurs for a non-shipped good or service).
- Use **ChargeConfirm** on a transaction that was authorized using **ChargeAuthorize**.
- If a time delay is not needed for payment processing, use **ChargeSale** instead of the **ChargeAuthorize**/**ChargeConfirm** method.
- If the payment status is not returned in the expected time, refer to Get Payment Status.
- Depending on the acquirer, a transaction may be approved even though the Card Validation Number (CVN) or the Address Verification Service (AVS) returned a non-match. If you decide to decline the transaction, refer to Refundor Void aCompleted Payment.

Get Payment Card Detailed Information, Non-Indemnified, Take PAN

Description

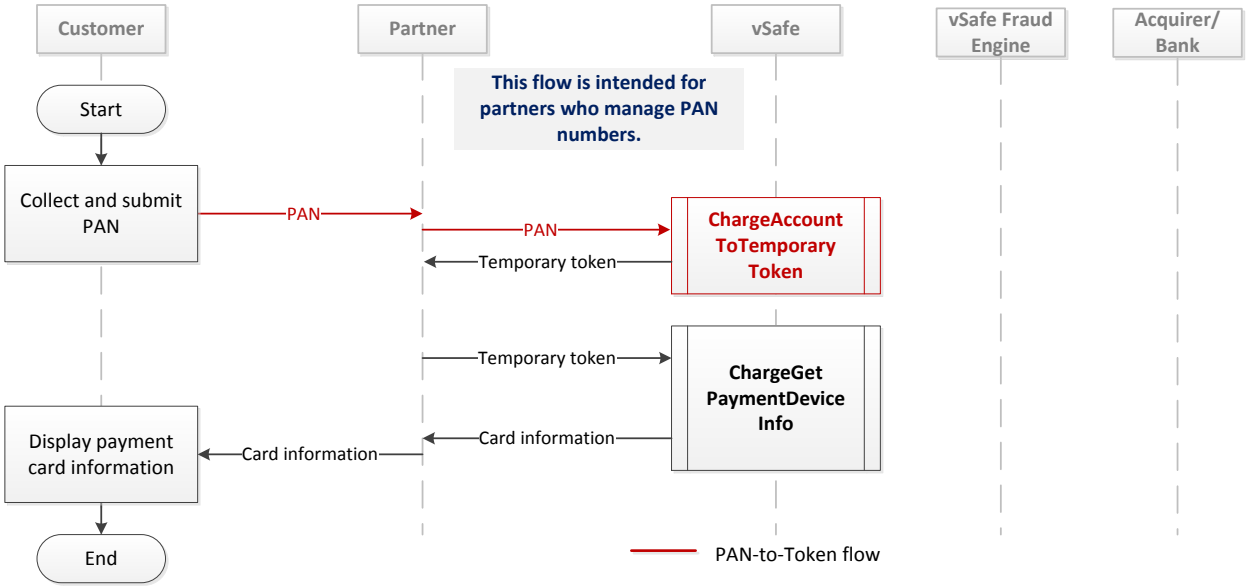
This flow starts by making an API call to exchange the incoming PAN for a temporary token (shown in red). This flow assumes that you are not using vSafe's PCI JavaScript Library at the customer's browser to intercept the PAN. The temporary token is used by the second API call that returns the card detailed information.

Note: Handling credit card numbers affects your PCI scope.

Preconditions

- Customer presents payment device

Primary Flow



1. Create a temporary token (red flow lines).
    a. Partner captures card number from customer.
    b. Partner calls **ChargeAccountToTemporaryToken** with a PAN in the request parameters.
    c. vSafe returns:
        i. a temporary token
        ii. a payment device type code, which indicates the charge account card type:
            - 3 (American Express)
            - 4 (Visa)
            - 5 (MasterCard)
            - 6 (Discover)
            - 7 (Diners Club)
            - 10 (Optima)

2. Get card information.
    a. Partner calls **ChargeGetPaymentDeviceInfo** with the temporary token.
    b. vSafe returns the following information in **ChargeGetPaymentDeviceInfo** response parameters:
        - IsCreditRoutable (Yes/No)
        - IsPrepaidCard (Yes/No)
        - IsSignatureDebit (Yes/No)
        - IsNonSignatureDebit (Yes/No)
        - PaymentDeviceCVNLength (zero-length indicates a non-signature debit card)
        - PaymentDeviceTypeCD (this is identical to what ChargeAccountToTemporaryToken returns)
        - PaymentDeviceTypeName

Business Rules

- Be sure that the IsTempToken Boolean is correctly set, otherwise an error response code is returned.

Best Practices

Debit cards are either signature debit or non-signature debit.

- Signature debit cards can be run as a debit card or as a credit card.
- Non-signature debit cards can only be run as a debit card.

vSaferunsall debit cards as credit cards, therefore non-signature debit cards cannot be used to process payments. This use case can indicate the type of debit card being presented.

Create a Permanent Token, Non-Indemnified

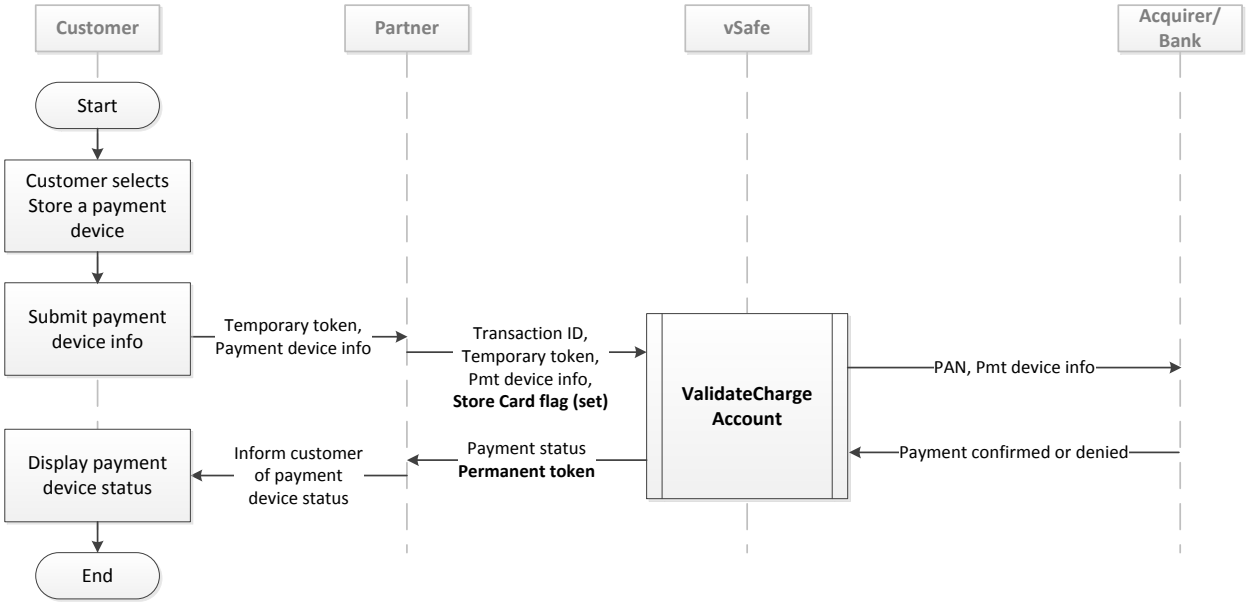
Description

Use this to create a token that doesn't expire, e.g., for customers who engage repeat business.

Preconditions

Customer has a payment card to store.

Primary Flow: Pass Token



1. Partner gets a temporary token. This can originate from the following sources:
   a. A customer provides a paymentcard information, and the PCI JavaScript Library automatically exchanges it for a temporary token.
   b. A customer provides payment card information, and the PAN is converted to a temporary token using **ChargeAccountToTemporaryToken**.
   c. A stored payment device is retrieved from a customer wallet. Payment card PANs are represented by temporary tokens.
2. Partner calls ValidateChargeAccount with the following parameters:
   - Temporary token
   - Payment device information (cardholder's name and address, card expiration date, CVN)
   - Payment source (web, phone, or prearranged)
   - Store Card flag is set, indicating that a permanent token is requested
3. vSafe contacts the bank for authorization.
4. vSafe returns the authorization request status:
   a. 1 (Bank Denied)
   b. 3 (vSafe Denied)
   c. 10 (Success)

5. If the authorization request is successful, vSafe returns permanent token in ValidateChargeAccount response parameters.
6. Partner informs customer of payment device status.

Best Practices

- If the payment status is not returned in the expected time, refer to Get Payment Status.

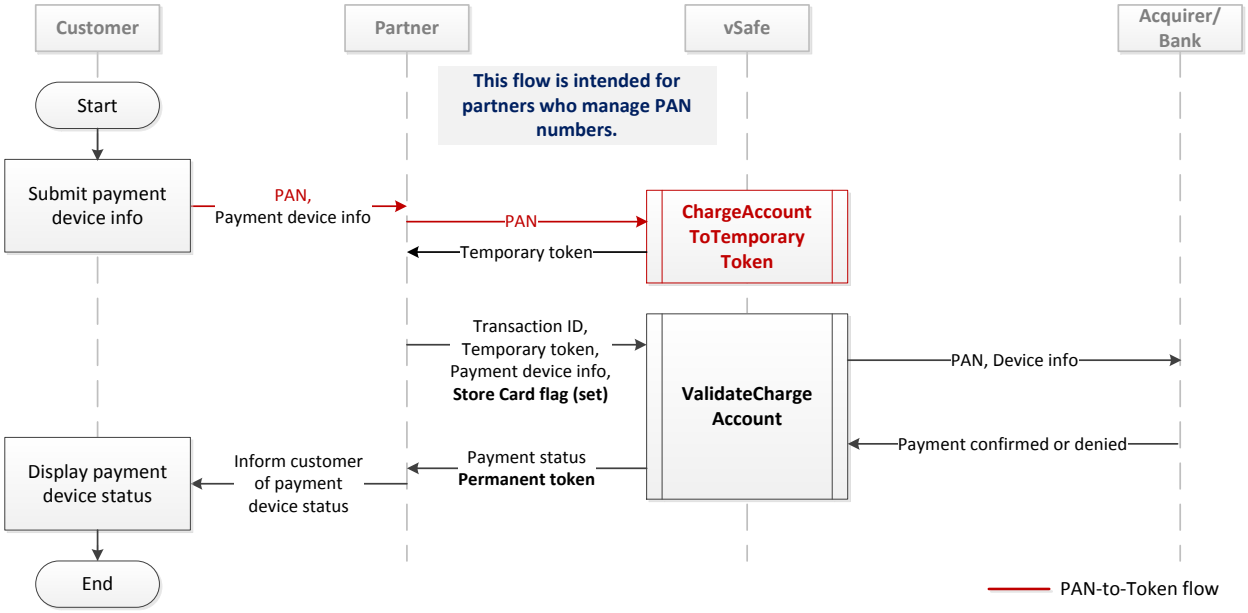
Alternate Flow: Take PAN

This flow is identical to the primary flow except that a step is added to convert the PAN into a temporary token. This assumes that you are **not** using vSafe's PCI JavaScript Library at the customer's browser to intercept the PAN.

Note: Handling credit card numbers affects your PCI scope.

Preconditions

- Customer presents payment device



1. Partner captures card number from customer.
2. Partner creates a temporary token (red flow lines).
   a. Partner calls **ChargeAccountToTemporaryToken** with card number in the request parameters (shown in red).
   b. vSafe returns temporary token in the response parameters.
3. Partner calls **ValidateChargeAccount**with the following parameters:
   - Temporary token
   - Payment device information (cardholder's name and address, card expiration date, CVN)
   - Payment source (web, phone, or prearranged)
   - Store Card flag is **set**, indicating that a permanent token is requested

a. vSafe contacts the bank for authorization.
b. vSafe returns the authorization request status:
    - 1 (Bank Denied)
    - 3 (vSafe Denied)
    - 10 (Success)
c. If the authorization request is successful, vSafe returns permanent token in **ValidateChargeAccount**response parameters.
d. If the authorization request is either bank denied (value of 1) or vSafe denied (value of 3), the partner informs customer.

Key Business Rules

- Any of the following API calls can return a permanent token as an option:
    - **ValidateChargeAccount** (this use case example)
    - **ChargeSale**
    - **ChargeAuthorize**
- When issuing the API call, be sure the StoreCard flag is **set**. This flag determines if a permanent token is returned.
- **RecurringPaymentEnroll** always returns a permanent token.
- The credit card number, expiration date and CVN must be valid.
- When the API returns a permanent token, the temporary token becomes invalid.

Best Practices

- If the payment status is not returned in the expected time, refer to Get Payment Status.

Validate a Card, Non-Indemnified, Take PAN

Description

This use case is for partners using the non-indemnified service who are handling PAN data.

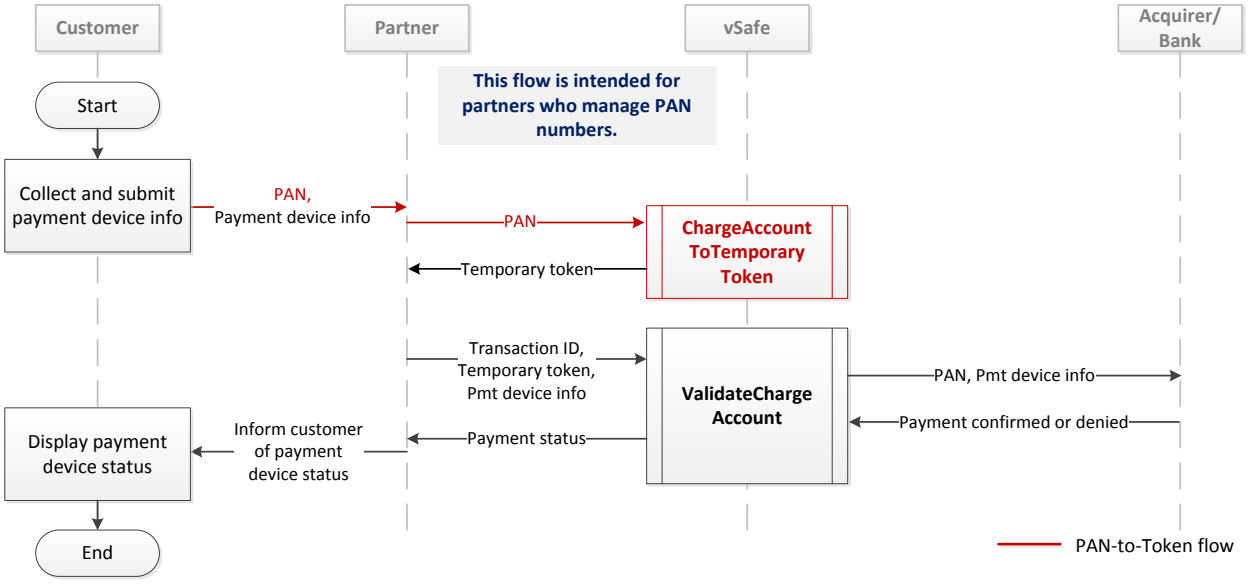
- This flow calls **ChargeAccountToTemporaryToken** to exchange a PAN for a temporary token (shown in red).
- This flow assumes that you are not using vSafe's PCI JavaScript Library at the customer's browser to intercept the PAN and inject a temporary token.

Note: Handling credit card numbers affects your PCI scope.

Preconditions

- Customer presents payment device

Primary Flow



1. Partner captures card number from customer.
2. Partner creates a temporary token (red flow lines).
   a. Partner calls **ChargeAccountToTemporaryToken** with card number in the request parameters (shown in red).
   b. vSafe returns temporary token in the response parameters.
3. Partner calls **ValidateChargeAccount**with the following parameters:
   - Temporary token
   - Card information (cardholder's name and address, card expiration date, CVN)
   - Payment source (web, phone, or prearranged)

   a. vSafe contacts the bank for authorization.
   b. vSafe evaluates the fraud risk.

c. vSafe returns the authorization request status:
   - 1 (Bank Denied)
   - 3 (vSafe Denied)
   - 10 (Success)

d. If the authorization request is successful, vSafe returns permanent token in **ValidateChargeAccount**response parameters.

e. If the authorization request results in bank denied (Payments Status = 1), the partner informs customer.

Key Business Rules

- **ValidateChargeAccount** is one of several API calls can optionally return a permanent token:
  - **ValidateChargeAccount** (this example)
  - **ChargeSale**
  - **ChargeAuthorize**
- The credit card number, expiration date and CVN must be valid.