Payer Authentication

Simple Order API

cybersource
A Visa Solution

Developer Guide

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

Trademarks

Confidentiality Notice

Revision

Version: 24.02

Contents

Recent Revisions to This Document

24.02

Added a *short description* of the other products in the risk management portfoio that work with payer authentication.

24.01

Updated the date that a *Visa Secure 3-D Secure mandate* that changes some conditionally optional fields to required fields occurs. The effective date was pushed back by six months to August 12, 2024.

23.10

Removed References to 3-D Secure 1.0	Removed the 3-D Secure 1.0 test cases section as 3-D Secure 1.0 is no longer supported. Other references to 3-D Secure 1.0 were removed from the guide.
Test Card Numbers Updated	Mastercard test card numbers for the EMV 3-D Secure *test case 2.4* were corrected and a Visa test card number for *test case 2.6* was corrected.
ECI Value Tables combined	For each *EMV 3-D Secure 2.0 test case*, the Economic Indicator (ECI) raw numeric value table and its respective character string value table were combined into one table to better indicate the relationship between the two ECI values.

23.09

This revision contains only editorial changes and no technical updates.

23.08

Editorial Content updated	While no technical content was added, the entire guide was edited and updated to remove outdated content.
Consistency issues addressed	Inconsistent content between the REST and Simple Order versions of the guide was updated to ensure that both API versions have the same information. Content alignment in table cells was addressed to enable easier reading of the table information.

23.07

Added Acquirer Information section	Added a section about the information needed from the acquirer. See *Acquirer Information* on page 21.
Moved Combining Services section	Moved the section about combining authentication and authorization services. This updated information now comes after the authentication steps for implementing the Direct API section. See *Combining the Authentication and the Authorization Services* on page 47.

23.06

Updated Implementing Direct API Section	The Implementing Direct API section was updated to include recommended practices and process flow diagrams. See *Implementing Direct API for Payer Authentication* on page 24. The Cardinal Cruise Direct Connection API was renamed to Direct API.
Method URL Test Case Updated	The Method URL test case was updated. See *Test Case 2.14: Require Method URL* on page 94.
Fixed Tables Spacing	A spacing issue within table cells that caused random indents within the cell content making it difficult to read was addressed.

23.05

New Use Cases Chapter Added	Added new section listing use cases for payer authentication. Combined the lists of required and optional fields from the API chapter and the examples at the end of the Implementing Direct API section to create a use cases section. Removed the section that listed the API fields. See *Payer Authentication Use Cases*.

23.04

Introduction Chapter Rewritten	• The Introduction chapter was rewritten. See *Introduction to Payer Authentication*. • The Hybrid integration chapter was removed as this integration is no longer installed. • The test cases for 3-D Secure 1.0 were moved from the Testing section to the back of the guide in preparation for removal when support for the 1.0 version is discontinued. (The 3-D Secure 1.0 test cases section was removed in November 2023.) • The Interpreting the Response sections for checking enrollment and validating authentication were updated with more detailed response statuses. See *Interpreting the Check Enrollment Response* on page 37 and *Interpreting the Validation Response* on page 45.
Test Card Number for Exemption Test Case Corrected	There was a typo in the JCB test card number for testing unsuccessful step-up authentication that was corrected. See *Test Case 2.11a: Unsuccessful Step-Up Authentication* on page 90.

23.03

JCB Added to Required Card Type	The JCB card was added to the cards listed that require the **card_cardType** field.
Test Results for Exemption Test Case Updated	The reason code result for the exemption test case 2c: Cardholder Not Enrolled in

Service was updated. See *Test Case 2c: Cardholder Not Enrolled in Service* on page 97.

About This Guide

Audience and Purpose

This guide is written for application developers who want to use the Simple Order API to integrate payer authentication services into their system. It describes the tasks you must perform in order to complete this integration.

Implementing payer authentication services requires software development skills.

You must write code that uses the API request and response fields to integrate payer authentication services into your existing order management system.

Scope

This guide describes how to use the Simple Order API to integrate payer authentication services with your order management system. It does not describe how to get started using the Simple Order API nor does it explain how to use services other than payer authentication. For that information, see the Related Documents section.

Conventions

These special statements are used in this document:

Important

An Important statement contains information essential to successfully completing a task or learning a concept.

Warning

A Warning contains information or instructions, which, if not followed, can result in a security risk, irreversible loss of data, or significant cost in time or revenue.

Related Documentation

Visit the *Technical Documentation Hub* on the Cybersource Developer Center for links to further documentation resources.

Customer Support

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

VISA Platform Connect: Specifications and Conditions for Resellers/ Partners

The following are specifications and conditions that apply to a Reseller/Partner enabling its merchants through Cybersource for Visa Platform Connect ("VPC") processing. Failure to meet any of the specifications and conditions below is subject to the liability provisions and indemnification obligations under Reseller/Partner's contract with Visa/Cybersource.

1. Before boarding merchants for payment processing on a VPC acquirer's connection, Reseller/Partner and the VPC acquirer must have a contract or other legal agreement that permits Reseller/Partner to enable its merchants to process payments with the acquirer through the dedicated VPC connection and/or traditional connection with such VPC acquirer.
2. Reseller/Partner is responsible for boarding and enabling its merchants in accordance with the terms of the contract or other legal agreement with the relevant VPC acquirer.
3. Reseller/Partner acknowledges and agrees that all considerations and fees associated with chargebacks, interchange downgrades, settlement issues, funding delays, and other processing related activities are strictly between Reseller and the relevant VPC acquirer.
4. Reseller/Partner acknowledges and agrees that the relevant VPC acquirer is responsible for payment processing issues, including but not limited to, transaction declines by network/ issuer, decline rates, and interchange qualification, as may be agreed to or outlined in the contract or other legal agreement between Reseller/ Partner and such VPC acquirer.

DISCLAIMER: NEITHER VISA NOR CYBERSOURCE WILL BE RESPONSIBLE OR LIABLE FOR ANY ERRORS OR OMISSIONS BY THE VISA PLATFORM CONNECT ACQUIRER IN PROCESSING TRANSACTIONS. NEITHER VISA NOR CYBERSOURCE WILL BE RESPONSIBLE OR LIABLE FOR RESELLER/PARTNER BOARDING MERCHANTS OR ENABLING MERCHANT PROCESSING IN VIOLATION OF THE TERMS AND CONDITIONS IMPOSED BY THE RELEVANT VISA PLATFORM CONNECT ACQUIRER.

Introduction to Payer Authentication

Cybersource has a variety of products to manage and minimize the risk of fraud that merchants face in their daily transactions. While these risk management products can operate independently to address specific areas of risk, the best results are achieved when the entire suite of products works in concert to detect patterns of fraud in a business's online activity.

- Decision Manager: Decision Manager uses AI to help large enterprises analyze the vast amount of data from their online transactions to detect known patterns of fraudulent behavior. Each potential transaction can be compared to past patterns and automatically assigned a risk score before authorizing a transaction. Behavior analysis of past transaction data enables you to recommend rules that identify risky transactions and to suggest how to handle them. Machine learning capabilities in Decision Manager enables you to create hypothetical environments to test strategies for dealing with risky scenarios so that you can either reject them or require payer authentication.
- Fraud Management Essentials: Fraud Management Essentials helps small-to-medium businesses monitor their online transactions using AI and preconfigured rules to spot and avoid fraudulent transactions. You can adjust the fraud detection settings to match your risk tolerance and manually review transactions flagged for risk review.
- Account Takeover Protection: Account Takeover Protection monitors customer account activity to detect compromised accounts. You create account events and define rules to determine the types and levels of activity in a customer account that trigger a manual review for potential fraud. The activity data that happens within a customer account can be easily integrated into Decision Manager and used to assess risky payment behavior.
- Payer Authentication: Payer authentication uses the 3-D Secure protocol in online transactions to verify that payment is coming from the actual cardholder. Most transactions can be authenticated without the customer being aware of the process, but higher risk transactions might require an exchange of one-time passwords (OTP) during authentication. This authentication of the payer before the transaction is authorized benefits the merchant by shifting chargeback liability from the merchant

to the card issuer. You can use Decision Manager with payer authentication services so that the risk level of an order determines when to invoke payer authentication. For example, low-risk orders can be set to skip payer authentication and proceed directly to authorization.

This guide documents the payer authentication aspect of fraud management and how payer authentication can be used to satisfy the Strong Customer Authentication (SCA) requirement of the Payment Services Directive (PSD2) that applies to the European Economic Area (EEA) and the United Kingdom. SCA requires banks to perform additional verification when consumers make payments to confirm their identity. Access to the documentation for other aspects of the risk management portfolio requires a Cybersource support license for that product.

Transactions where the card is not present have a high risk of fraud, so authenticating a payer before processing a transaction greatly reduces the merchant risk for chargebacks. Payer authentication is a way of verifying that a customer making an e-commerce purchase is the owner of the payment card being used. The protocol that is followed to authenticate customers during online transactions is called *EMV 3-D Secure*.

This EMV 3-D Secure protocol is used by all major payment cards to implement payer authentication, but payment companies usually brand it under a different name:

- Visa: Visa Secure
- Mastercard: Mastercard Identity Check
- American Express: American Express SafeKey
- JCB: J/Secure
- Discover/Diners: ProtectBuy

Why Payer Authentication Is Needed

As e-commerce developed, fraudulent transactions also grew, taking advantage of the difficulty authenticating a cardholder during a transaction when the card is not present. To create a standard for secure payment card processing, Europay, Mastercard, and Visa collaborated as EMV. Other card providers wanted input on creating new payment standards, so a consortium called EMVCo was formed to enable equal input from Visa, Mastercard, JCB, American Express, China UnionPay, and Discover.

EMVCo developed 3-D Secure as the protocol to provide customer authentication during an online transaction. EMV 3-D Secure reduced chargebacks to merchants, and when the buyer was authenticated, the issuing bank assumed any liability when a chargeback occurred.

The same need to reduce fraud prompted Europe to develop a standard called Strong Customer Authentication (SCA) to regulate authentication during electronic payment. The use of SCA is mandated by the European Banking Authority in the Payment Services Directive (PSD2) that took effect in 2018 to promote and regulate the technical aspects of financial transactions between merchants and their customers in Europe. SCA requires two-factor authentication. A customer must be able to authenticate by providing two of these three factors:

- Something the customer knows (such as a password, PIN, or challenge questions)
- Something the customer has (such as a phone or hardware token)
- Something the customer is (biometric data, such as fingerprint or face recognition)

Although SCA is required for almost all online transactions, some exceptions are allowed. If a payment is considered low risk, the merchant can request an exemption from SCA to bypass authentication of the customer. The issuing bank must approve of the exemption before the transaction can be exempted from SCA. Although an exemption from SCA results in a frictionless transaction, liability is not shifted to the issuing bank, and the merchant assumes responsibility for any chargeback that occurs. An exemption from SCA might apply to these types of transactions:

- Payer authentication is unavailable because of a system outage.
- Payment cards used specifically for business-to-business transactions are exempt.
- Payer authentication is performed outside of the authorization workflow.
- Follow-on installment payments of a fixed amount are exempt after the first transaction.
- Follow-on recurring payments of a fixed amount are exempt after the first transaction.
- Fraud levels associated with this type of transaction are considered a low risk.
- Low transaction value does not warrant SCA.
- Merchant-initiated transactions (MITs) are follow-on transactions that are also exempt.
- Stored credentials were authenticated before storing, so stored credential transactions are exempt.
- Trusted merchants, registered as trusted beneficiaries, are exempt.

For additional information about transactions that are exempt from SCA, see the *Payments Guide*.

EMV 3-D Secure meets the SCA mandate for authenticating the customer during e-commerce transactions. The first version was called 3-D Secure 1.0 and was designed to authenticate by having the customer enter a static password that they had created to prove that they were the actual cardholder. Although this authentication process was an improvement in reducing fraud, the process had drawbacks:

- The authentication process was slow and intrusive.
- The cardholder had to remember a password and answer security questions.
- Transaction data shared between the merchant and issuing bank was not extensive enough for good risk analysis by the bank.
- Authentication for phones and tablets was not available.

Merchants lost sales when impatient customers grew frustrated over the length of time required for transaction approval. They did not trust being redirected to a different webpage to authenticate, and many had trouble remembering their passwords. Shopping cart abandonment caused merchants to lose sales. EMV 3-D Secure 2.0 was developed to address those problems.

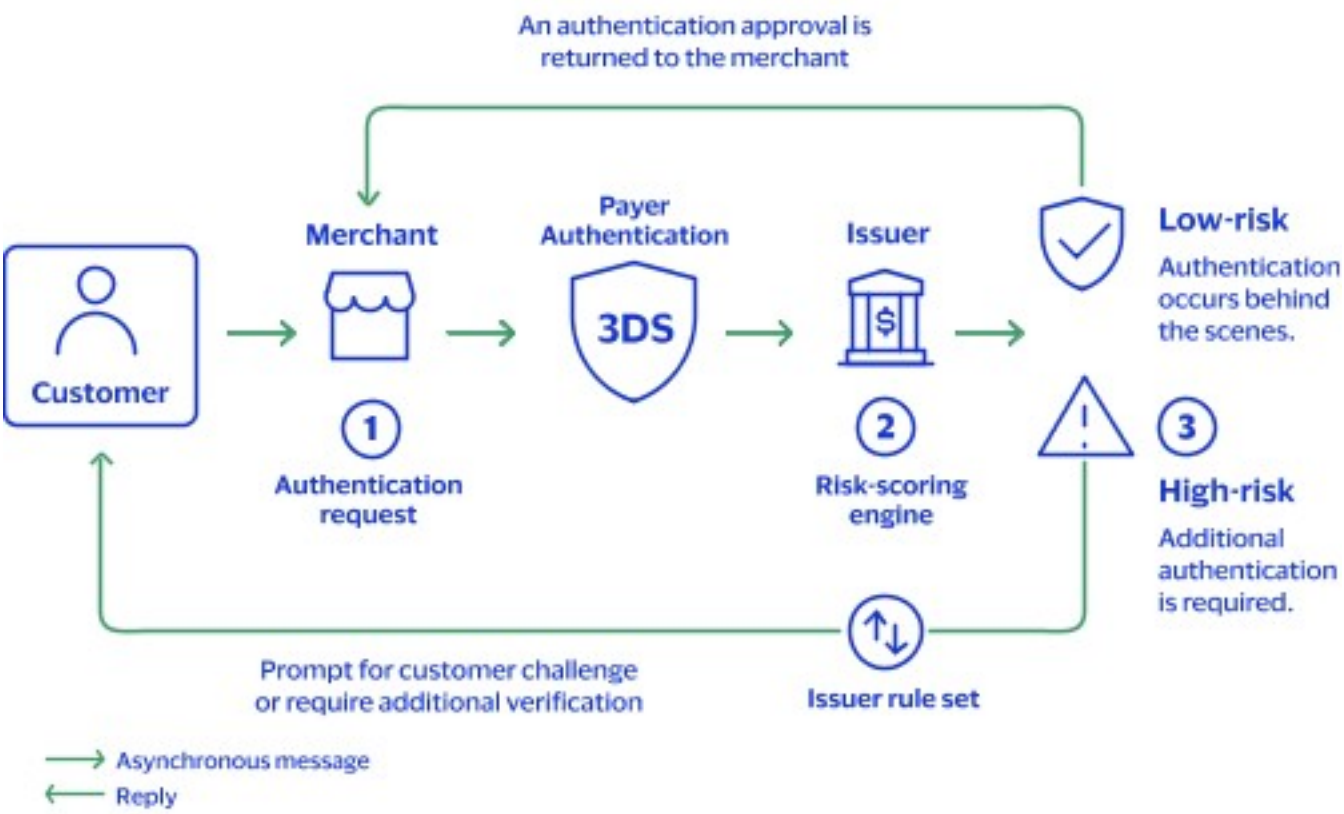
EMV 3-D Secure 2.0

To improve the experience of authenticating payers, a new version of 3-D Secure was developed. EMV 3-D Secure 2.0 uses a less intrusive process to authenticate a buyer, which provides a better customer experience. Enhancements include:

- More robust device data collection to improve risk analysis by the issuing bank.
- Small screens that pop up on your webpage to avoid full page redirects.
- SDK version for mobile devices like phones and tablets.
- Streamlined shopping experience.

Payer Authentication Customer Workflow

During authentication for each transaction, data is collected about the device that the customer is using and the shipping and billing information is compared with transaction history. The workflow in payer authentication can go one of two possible ways: frictionless and challenge (also known as step-up).
Payer Authentication Workflow



Frictionless Workflow

With frictionless workflow, the authentication process is not visible to the customer. The data collected during the transaction matches the data collected from past transactions with the customer. The risk for fraud is calculated to be low enough that further authentication is unnecessary. The transaction can continue to authorization.

1. The customer enters card information at checkout. Information about the device being used by the customer and shopping behavior is collected and relayed from the merchant to the issuing bank. A delay of about 10 seconds is built into the process to ensure that the device data can be transmitted and assessed before the **Buy** option is enabled.
2. The customer selects **Buy**.
3. The issuing bank verifies the information it receives against previous transactions. If the device data correlates with the information, the transaction is approved without the buyer having to provide any additional information.

Challenge Workflow

A challenge workflow occurs when the data collected during the transaction does not match the information on file from previous transactions with the customer. This process occurs for multiple reasons and does not necessarily mean that the customer has fraudulent intent. It could occur because the customer got a new device that has not been registered yet or because they bought something while traveling on vacation. The issuer of the card decides whether further authentication of the customer is required and requests that the customer prove that they are the cardholder. The customer is asked to return a passcode to the issuer. Below is a general description of this workflow from the customer viewpoint.

1. The customer enters card information at checkout. Information about the customer's device is collected and sent from the merchant to the issuing bank.
2. The customer selects **Buy**.
3. The issuing bank assesses risk by comparing the information it receives to information on file from previous transactions with the customer. If the device data does not match the information collected previously, the issuer requests further authentication.
4. A small window opens on the checkout page where a message from the bank asks if the customer wants to use email or text to receive a one-time password (OTP) from the bank.
5. The customer chooses how the password is sent.
6. The issuer sends an OTP to the account on file for the buyer.
7. A window opens on the checkout page on the customer device prompting the customer to enter the OTP sent by the issuer.
8. The customer enters the password that was received and sends it back to the issuer.
9. If the password entered by the customer matches the password sent by the issuer, the customer is authenticated, and the transaction can proceed to authorization. If the password does not match the password that the bank sent, the customer sees a

message that the transaction is declined and that another form of payment should be attempted.

Payer Authentication Merchant Workflow

Transaction circumstances might result in differences to the more detailed payer authentication process described below.

1. Before the **Buy** button is selected at checkout, the Setup service is called. The full card number identifies how to contact the issuing bank. The issuing bank sends an access token and a URL (called the DCC URL) to use for the data collected about the device where the transaction is occurring.
2. The merchant collects data about the device and includes billing and shipping information. The merchant posts this data to a hidden 10 pixel x 10 pixel iframe to send to the DDC URL provided by the card issuer for comparision with past transactions. After the data points are collected and sent, the issuing bank confirms that data collection ended and the **Buy** button is enabled. An 8-10 second delay ensures enough time for data collection.
3. Clicking **Buy** triggers the Check Enrollment service sending the order data (and session ID) to the issuer. If the bank is not part of an EMV 3-D Secure program, the payer authentication process stops. If the issuing bank is part of an EMV 3-D Secure program, the device data is compared to information on file collected at the bank during previous transactions with the cardholder.
   - The issuer's risk analysis software determines whether enough data points collected by the merchant match the data in the bank's files. If the data matches well, no further interaction is needed. This is called frictionless flow because no challenge to the buyer is necessary. The response returned to the merchant includes a payload with values like the ECI, CAVV, DS Transaction Id, and the PARes Status. These values must be passed on during the request for authorization. It is important to note that while frictionless flow can occur because the payer is authenticated, it can also occur for other reasons. For example, the issuing bank does not participate in payer authentication. Therefore, response values must be verified to determine why no step-up is needed.
   - If a significant discrepancy occurs between the transaction data and the data on file with the bank, the bank requests that the payer authenticate. This is a friction workflow and is called a step up or challenge. The response from the bank contains the same values returned for a frictionless workflow but also includes additional values like the *Access Control Server (ACS) URL*, the *PAReq* payload, a Pares Status = C, a Step Up URL, a new JWT, and a Transaction Id.
4. The JWT and the step up URL received in the check enrollment response are returned to the customer. Using the step up URL with the JWT as a POST parameter, a challenge screen opens in a viewable iframe on the buyer's device so that the cardholder can view and respond to the bank challenge. The challenge consists of the bank sending a pass code that the customer returns to the bank. The challenge asks how the customer

wants to receive a pass code, by text or email. After the customer chooses, they receive a pass code that they must enter into the challenge screen.

5. After the cardholder enters and sends the passcode, the response is sent to the merchant's return URL contained in the JWT. This response causes the merchant to make a validation call to the bank to obtain the final authentication outcome. The response to this validation request contains the final authentication results including these values: *ECI*, *CAVV* (if successful), DSTransactionId, ThreeDSVersion, and *PARes Status* (`Y` or `A` = successful or `N`, `U`, `R` = failed, unavailable, or rejected).
6. The next action depends on the outcome:
   - Successful: proceed to authorization, and append the EMV 3-D Secure data points to the authorization message.
   - Failed, unavailable, or rejected: display a message to prompt the customer to try payment with a different card.

Acquirer Information

To properly configure payer authentication, Cybersource needs three items of information that your *acquiring bank* uses to manage payments to your account. If you do not know this information, contact your acquiring bank.

- Acquiring *Merchant ID* (MID): This unique identifier for your business account is assigned by your acquiring bank or payment processor. A MID consists of 8-24 alpha-numeric characters. The MID can be different than the business deposit identifier used in settlements.
- *Acquiring Bank Identification Number* (BIN): This unique number is assigned to the acquiring bank by a payment card network to identify that bank when settling transactions. Each payment card assigns its own BIN for an acquiring bank, and the BINs have their own unique characteristics. For example, all Visa BINs start with a 4, Mastercard BINs start with a 2 or 5, and Discover BINs start with a 3 or 6.
- Merchant Category Code (MCC): This four-digit numeric value is assigned by the acquirer to the merchant to classify the merchandise or services provided by the business. The MCC indicates the kind of business transaction that the merchant processes.

Enable Merchant Account for EMV 3-D Secure

Partners and merchants use the Business Center to go online and view transaction activity and to generate reports about their transactions. For each partner, an account is created, and a portfolio merchant ID (MID) is assigned. For each of the merchants within the partner's portfolio, an account is also created and assigned a merchant ID (MID).

Access to the various functions in the Business Center is managed by the partner through the MID.
When the MID account is created, the various services that the merchant needs must be enabled. Payer authentication is a service that might need to be turned on by support. To set up an account for payer authentication, you need this information:

- MID
- Merchant website URL.
- Two-character ISO code for your country.
- Merchant category code.
- EMV 3-D Secure requestor ID (optional).
- EMV 3-D Secure requestor name (optional).
- Name of merchant's bank.
- Name, address, and email address of bank contact.

For each payment card that you accept, your acquirer must provide you with this information:

- Eight-digit BIN number.
- Merchant ID assigned by your acquirer.
- List of all of the currencies that you can process.

Payer Authentication Configuration Testing

After the payer authentication functionality is enabled for your account and you have installed and configured the software, you must run tests to ensure that payer authentication is working properly. You must ensure that the proper data is being collected and sent to the issuer and that the proper status for a particular circumstance is returned. To ensure that the proper statuses are returned under all possible circumstances, extensive testing is required before you go live with payer authentication.
A sandbox testing environment is provided to resolve any bugs in your system. In this testing environment, you can simulate various transaction scenarios with the types of payment cards that you accept. Test card numbers for the various types of payment cards are provided so that you can run transaction simulations. You can verify that the values generated during the simulations are the correct values that should occur during that transaction scenario.
When your test results are correct, contact customer support and request to go live.
When you go live, you will use the production host name to process transactions instead of the test host name that you used when processing transactions in the test environment.
The host name for the testing environment:
POST https://apitest.cybersource.com
The host name for the production environment:
POST https://api.cybersource.com
Details about testing your payer authentication configuration are available in the *Testing Payer Authentication Services* section.

Request Endpoints

When posting a request for payer authentication, you must add an endpoint to each hostname, whether you are using the test environment or the production environment. These endpoints are used with payer authentication.
`/risk/v1/authentications`: use when verifying that a card is enrolled in a card authentication program or requesting authentication from the issuer.
`/risk/v1/authentication-results`: use when retrieving and validating authentication results from the issuer so that the merchant can process the payment.
`/pts/v2/payments`: use when bundling multiple payments together.
For example, a test request might look like this:
`POST https://apitest.cybersource.com/risk/v1/authentications`

Payer Authentication Integrations

Payer authentication was designed to authenticate buyers during online transactions. During the early growth of internet transactions, e-commerce was conducted only on computers. Mobile phones had limited capabilities. When mobile phones (and tablets) could access the internet, online transactions quickly grew, and now they comprise almost half of all e-commerce transactions. A key part of updating the EMV 3-D Secure protocol from 1.0 to 2.0 was to ensure that payer authentication became available for e-commerce done on mobile devices.
Two types of payer authentication integration are available for merchants:

- API for browser authentication from a computer.
- SDK for authentication from mobile devices (available for Android and iOS). Contact support to obtain the SDK.

Merchants should integrate payer authentication for online shopping on both types of devices. The next sections in this guide describe how to integrate payer authentication into those shopping experiences.

Implementing Direct API for Payer Authentication

The Direct API integrates EMV 3-D Secure 2.x into your business's website. This integration uses an iframe to complete the device profiling and EMV 3-D Secure authentication requirements without including third-party JavaScript directly on your site. This implementation requires the use of JavaScript to leverage the authentication. The JavaScript is hosted and contained inside the iframe and does not directly access your web page.

Important

Payer Authentication uses Cardinal (a Visa owned company) Centinel as the technology platform to manage all EMV 3-D Secure authentication processes. Any references to Cardinal in this document refer to the underlying services that are provided by Cardinal technology.

A website is available at *https://developer.cybersource.com/demo/index.html* that provides a demo tool to help users understand how payer authentication works. Users can complete the sequence of steps required to implement payer authentication on their website and examine the code underlying the process. Use test card numbers to walk through the process and enter 123 as the security code.

Prerequisites

Notify your account representative that you want to implement payer authentication (3-D Secure) using the Direct API. Provide the merchant ID that you will use for testing. For more information, see *Required Merchant Information*.

Before you can implement payer authentication services, your business team must contact your acquirer and Cybersource to establish the service. Your software development team should become familiar with the API fields and technical details of this service.

After Implementation and Before Go Live

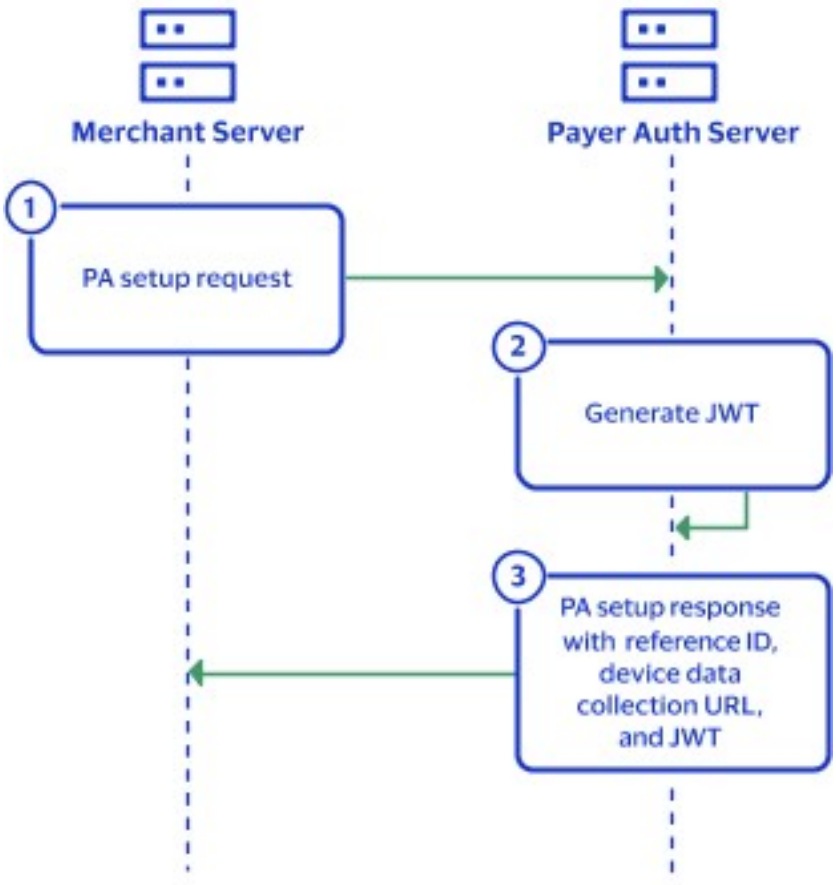
Use the test cases to test your preliminary code and make appropriate changes. See *Testing Payer Authentication* on page 69. Testing ensures that your account is configured for production and that your transactions are processed quickly and correctly.

Step 1: Setup Service

Request the Setup service before selecting the button to submit payment. Request the Setup service separately without including other services. The Setup service response will include a JSON Web Token (JWT) that contains credentials to create a secure channel with the merchant. The Setup response also includes a reference ID to use during the authentication and a URL to use when transferring the device data that is collected in the next step.

Run the Setup service as soon as the customer enters their card number, to avoid any delay in the customer experience. The next step in the process, device data collection, cannot start until the Setup response is received since the response has the URL where the device data will be sent.

Process Flow for Setup for Payer Authentication



Best Practices

This practice should be followed for this step to achieve optimal performance and to minimize potential operating issues.

- After the customer credit card is entered, immediately begin device data collection.

Request Fields

When requesting the Setup service, you must send the customer's card number, encrypted payment data, transient token, or a TMS token or some other equivalent of card data used by your integration. Besides the required fields, the request might also include any of these fields:

- **card_accountNumber**
- **recurringSubscriptionInfo_subscriptionID**
- **tokenSource_transientToken**

The **card_cardType** field is required when the card type is Cartes Bancaires, JCB, or UPI.

Important Response Fields

The response from the issuing bank might include these API fields.

- **payerAuthSetupReply_accessToken** is used in *Step 2: Device Data Collection* on page 29.
- **payerAuthSetupReply_deviceDataCollectionURL** is used in *Step 2: Device Data Collection* on page 29.
- **payerAuthSetupReply_referenceID** is used in *Step 3: Payer Authentication Check Enrollment Service* on page 34.

For further details on examples, see *Use Case: Setting Up Payer Authentication*.

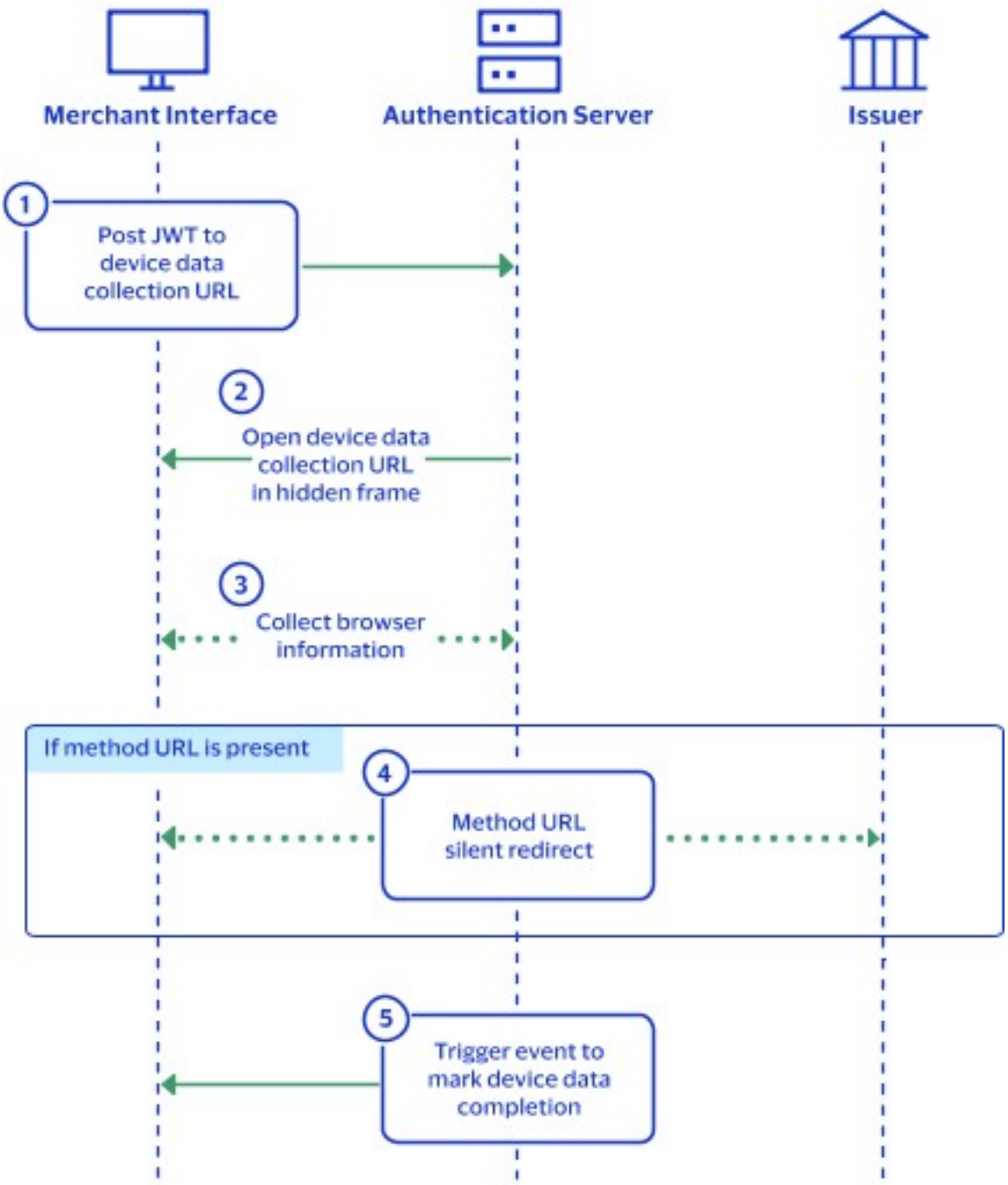
Step 2: Device Data Collection

Device data collection starts on the merchant end after you receive the server-side Setup service response as described in *Step 1: Setup Service* on page 26 and pass the access token and the device data collection URL to the front end. When device data gets to the data collection URL, a Method URL stipulated in the 3-D Secure protocal captures the entire card number to identify the issuing bank.

A hidden 10 x 10 pixel iframe is rendered in the browser, and using the access token, the merchant sends the customer device data to the device data collection URL. The device data collection can take up to 10 seconds. If you proceed with the check enrollment service as described in *Step 3: Payer Authentication Check Enrollment Service* on page 34 before a response is received, the collection process is short-circuited and an error occurs. Despite the error, as long as you include the data from the eleven browser fields as explained in *Step 3:Payer Authentication Check Enrollment Service*, you can still proceed with the EMV authentication.

It is recommended that the device data collection start immediately after the merchant receives the customer card number to ensure that the data collection runs in the background while the customer continues with the checkout process, ensuring a minimum of waiting. When a customer changes to a different card number, begin the Setup and device data collection process again as soon as the new card number is entered.

Process Flow for Device Data Collection



Best Practices

These practices should be followed for this step to achieve optimal performance and to minimize potential operating issues.

- After the customer credit card number is entered, immediately begin the device data collection.
- Device data collection must complete before beginning the enrollment check.

- While not required, it is highly recommended to pass the values from the 11 browser-based fields in the request. The information from these fields serve as a backup, for when the device data collection does not complete correctly.
- As much billing data as possible (unless restricted by regional mandates) should be supplied to the issuing bank to ensure that the issuer's risk assessment software has the most comprehensive data.
- The billing data such as state and country must be formatted according to ISO 3166-2 format specifications to ensure that the network can properly validate the data.

Which Device Data is Collected

One of the key components to authenticating a cardholder during an online transaction is to compare information about the device that the buyer is currently using to information in the bank database about devices the buyer used in past transactions. This information is maintained in the acess control server (ACS) at the issuing bank. This device information focuses on the web browser and includes these types of data:

- IP address
- Browser language
- Browser type
- Browser version
- Computer operating system
- System time zone
- Screen dimensions
- Color depth

A successful device data collection process that includes the 11 browser fields listed in the check enrollment step increases the chances of a frictionless authentication. Business rules evaluate whether a transaction is risky enough to require the buyer to authenticate their identity. These business rules are configured in the issuer's risk analysis software that evaluates each transaction.

Building the Iframe

The iframe has these parameters.

- Form POST Action: The POST goes to the URL that is opened within an iframe. This URL is obtained from the **payerAuthSetupReply_deviceDataCollectionURL** response field discussed in *Step 1: Setup Service* on page 26.
- JWT POST Parameter: Use the value from the **payerAuthSetupReply_accessToken** response field discussed in *Step 1: Setup Service* on page 26.

Initiating the Device Data Collection Iframe

Initiate a form POST in a hidden 10 x 10 pixel iframe and send it to the device data collection URL (**payerAuthSetupReply_deviceDataCollectionURL**).
Place the HTML anywhere inside the `<body>` element of the checkout page. Dynamically replace the value of the form action attribute and JWT POST parameter with the response values discussed in *Step 1: Setup Service* on page 26. See this example.

Initiate the Device Data Collection Iframe

```
<iframe id="cardinal_collection_iframe" name="collectionIframe" height="10" width="10" style="display:
 none;"></iframe>
<form id="cardinal_collection_form" method="POST" target="collectionIframe" action=https://
centinelapistag.cardinalcommerce.com/V1/Cruise/Collect>
  <input id="cardinal_collection_form_input" type="hidden" name="JWT"
 value="eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJSZWZlcmVuY2VJZCI6ImE0NjVlYzU1LTMwNTEtN
 GYwZC05MGE0LWZjMDNlMGE2MWQxOSIsIlJldHVyblVybCI6Imh0dHA6XC9cL2xvY2FsaG9zdDo4MDgyXC9yZXF1ZX
 N0LWNhdGNoZXJcL2NhdGNoLXJlcXVlc3QucGhwIiwianRpIjoianRpXzVmMDVkM2VkY2U0MjYzLjc5MjQwNzMzIi
 wiaWF0IjoxNTk0MjE3NDUzLCJpc3MiOiI1YjIzZjhjMGJmOWUyZjBkMzQ3ZGQ1YmEiLCJPcmdVbml0SWQiOiI1
 NWVmM2YwY2Y3MjNhYTQzMWM5OWI0MzgifQ.Yw9cB9Hdrg71GPL40oAC0g3CVKYElNGe0uvN9JAaw2E">
</form>
```

Submitting the Device Data Collection Iframe

Add JavaScript to invoke the iframe form POST. Place the JavaScript after the closing `</body>` element as shown in this example. The JavaScript invokes the iframe form POST automatically when the window loads. You must submit it before requesting the Check Enrollment service.

JavaScript to Invoke the Iframe Form POST

```
 <script>
window.onload = function() {
var cardinalCollectionForm = document.querySelector('#cardinal_collection_form');
if(cardinalCollectionForm) // form exists
cardinalCollectionForm.submit();
}
</script>
```

Receiving the Device Data Collection URL Response

Receiving the response indicates that the device data collection URL completed its processing. The response is an event callback that contains a message with the status of the device data collection process.
Which `event.origin` URL that you use depends on whether you are in a test or production environment:

- Test: `https://centinelapistag.cardinalcommerce.com`
- Production: `https://centinelapi.cardinalcommerce.com`

Study the example below to understand how to subscribe to the event. Add JavaScript to receive the response from the device data collection iframe. Place the JavaScript after the closing `</body>` element.

Listen for Device Data Collection Response

```
window.addEventListener("message", function(event) {
  if (event.origin === https://centinelapistag.cardinalcommerce.com) {
    console.log(event.data);
  }
}, false);
```

This example shows a response payload from the event. None of the returned data needs to be stored for future use.
Event Listener Callback Payload

```
{
  "MessageType": "profile.completed",
  "Session Id": "f54ea591-51ac-48de-b908-eecf4ff6beff",
  "Status": true
}
```

Step 3: Payer Authentication Check Enrollment Service

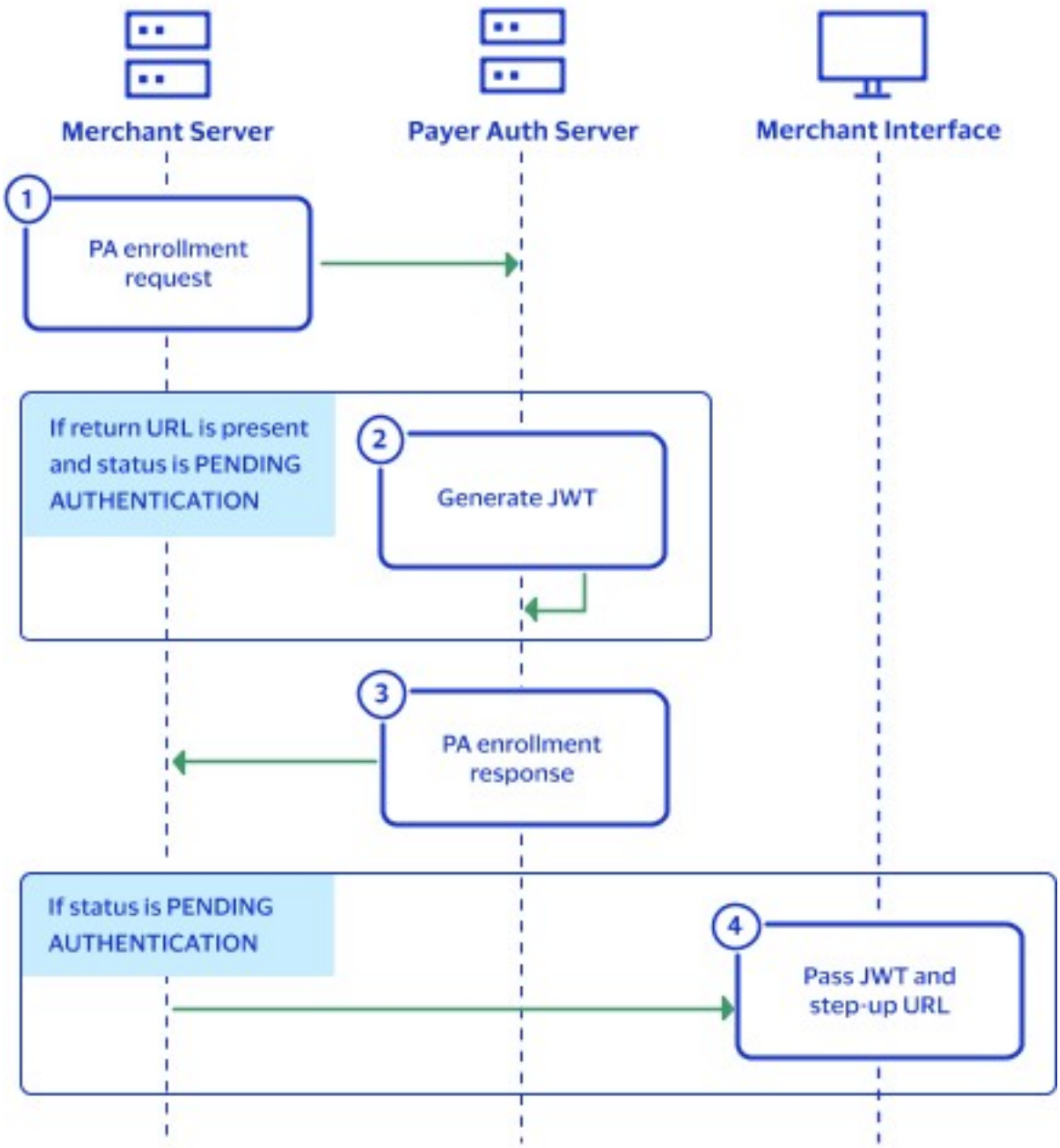
Request the Check Enrollment service only after you receive the device data collection response. Checking enrollment before receiving the data device collection response stops the data collection process. Data collection can take up to 10 seconds. The merchant should set a timer that expires after 10 seconds of waiting for a response to the data collection so that the check enrollment service starts even when the device data collection response was not received.
As a backup, in cases when the device data collection fails to complete, you can still qualify for EMV 3-D Secure 2.x by including all 11 browser field values in the check enrollment request.

- **payerAuthEnrollService_httpAccept**
- **payerAuthEnrollService_httpUserAccept**
- **billTo_httpBrowserColorDepth**
- **billTo_httpBrowserJavaEnabled**
- **billTo_httpBrowserJavaScriptEnabled**
- **billTo_httpBrowserLanguage**
- **billTo_httpBrowserScreenHeight**
- **billTo_httpBrowserScreenWidth**
- **billTo_httpBrowserTimeDifference**
- **billTo_ipAddress**
- **payerAuthEnrollService_httpUserAgent**

With the device data collected, including the 11 browser fields listed above, the issuer runs a risk assesment that results in one of these outcomes:

- Frictionless success (low risk)
- Challenge required (moderate risk)
- Frictionless failure or decline (high risk)

Process Flow for Checking Enrollment in Payer Authentication



Best Practices

Follow these practices for this step to achieve optimal performance and to minimize potential operating issues.

- Do not start checking enrollment until the device data collection has completed.
- Notifiy cardholders to contact their bank for instructions if a problem occurs. Information about additional action required of the cardholder should be displayed on the checkout page. Providing instructions to the customer avoids multiple attempts to resubmit the same card.

Request Fields

The **payerAuthEnrollService_referenceID** field is mapped from the **payerAuthSetupReply_referenceID** field as discussed in *Step 1: Setup Service* on page 26. **payerAuthEnrollService_returnURL** is set to the URL to which the issuing bank redirects the customer as discussed in *Step 4: Step-Up Iframe* on page 39.
To request the Check Enrollment service, you must send either the customer's card number, encrypted payment data, transient token, or a TMS token or transient token or some other equivalent of card data used by your integration. The request fields can include any of these:

- **card_accountNumber**
- **encryptedPayment_data**
- **encryptedPayment_descriptor**
- **recurringSubscriptionInfo_subscriptionID**
- **tokenSource_transientToken**

These fields are required (merchant ID is in the header):

- **billTo_country**
- **billTo_email**
- **billTo_firstName**
- **billTo_lastName**
- **billTo_postalCode**
- **billTo_state**
- **billTo_street1**
- **card_cardType**
- **card_expirationMonth**
- **card_expirationYear**
- **merchantID**
- **merchantReference Code**
- **payerAuthEnrollService_referenceID**
- **payerAuthEnrollService_returnURL**
- **purchaseTotals_currency**
- **purchaseTotals_grandTotalAmount**

You can send additional request data to reduce your issuer step-up authentication rates. Send all available fields. As a backup, if device data collection fails, include the 11 device information fields listed among the optional fields for the Check Enrollment service in your request. If a failure does occur, adding these device information fields ensures a transaction is not downgraded. If you do not have data for a field, do not send dummy data.
The size of the step-up iframe discussed in *Step 4: Step-Up Iframe* on page 39 can vary depending on the EMV 3-D Secure version of the transaction. You can request the size of the challenge window in the **payerAuthEnrollService_acsWindowSize** request field.

Requesting a specific window size does not guarantee this size. Parsing the PAReq as described in *Step 4: Step-Up Iframe* on page 39 determines the actual size.
For further details on individual API fields, refer to the *API Field Reference Guide*. The field values should use the ISO 3166-2 format.

Interpreting the Check Enrollment Response

It is important to check the status values in the response. These possible statuses are the same for all card types.

Reason Code 475

- VERes enrolled = `Y`
- PARes status = `C`

The account number is enrolled in payer authentication. The cardholder is challenged to authenticate. Authenticate the cardholder before authorizing the transaction.

Reason Code 100

Frictionless authentication was successful/Stepup authentication is not required

- VERes enrolled = `Y`
- PARes status = `Y`

The account is enrolled in payer authentication, and the cardholder was successfully authenticated. If enrollment and authorization are made in separate calls, the payer authentication data must be included in the authorization request to receive liability shift protection.
Attempts Stand-in Frictionless Authentication

- VERes enrolled = `Y`
- PARes status = `A`

This status indicates that the account is enrolled in paper authentication, but the issuer does not support the program. This is called stand-in authentication. If check enrollment and authorization are made in separate calls, the payer authentication data must be included in the authorization request to receive liability shift protection.
Card not enrolled

- VERes enrolled = `B` or `U`

This status indicates that the account is not eligible for a payer authentication program, authentication was bypassed, or an error or timeout occurred. If enrollment and authorization are made in separate calls, you can request authorization, but there is no liability shift protection.
Unavailable Frictionless Authentication

- VERes enrolled = `Y`

- PARes status = `U`

This status indicates that the account is enrolled in payer authentication, but authentication is currently unavailable. The merchant can attempt to retry authentication or proceed with authorization. If enrollment and authorization are made in separate calls, you can continue and request authorization, but there is no liability shift protection. Without authentication of the customer, the merchant remains liable for any chargeback.

Reason Code 476

Failed Frictionless Authentication

- VERes enrolled = `Y`
- PARes status = `N`

Indicates that the account is enrolled in payer authentication but frictionless authentication failed. Merchants cannot submit this transaction for authorization. Instead ask for another form of payment.
Rejected Frictionless Authentication

- VERes enrolled = `Y`
- PARes status = `R`

Indicates that the account is enrolled in payer authentication but frictionless authentication was rejected by the issuing bank without requiring a challenge. Merchants cannot submit this transaction for authorization. Instead ask for another form of payment. When a 476 status occurs, the merchant should display a message from the card issuer to the cardholder using the **payerAuthEnrollReply_cardholderMessage** field. The text of the message is provided by the ACS/issuer during a frictionless or decoupled transaction to convey information to the cardholder. An example message might be, “Additional authentication is needed for this transaction, contact (issuer name) at xxx-xxx-xxxx.” An example of the entry that would appear in the log for such an occurrence is: "cardholderInfo":"You cannot complete this purchase right now. For help, call CommBank at (111) 555-2222"

Important Response Fields

When you receive a reason code 475 response, you also receive these fields:

- **payerAuthEnrollReply_stepUpUrl** discussed in *Step 4: Step-Up Iframe* on page 39.
- **payerAuthEnrollReply_accessToken** discussed in *Step 4: Step-Up Iframe* on page 39.

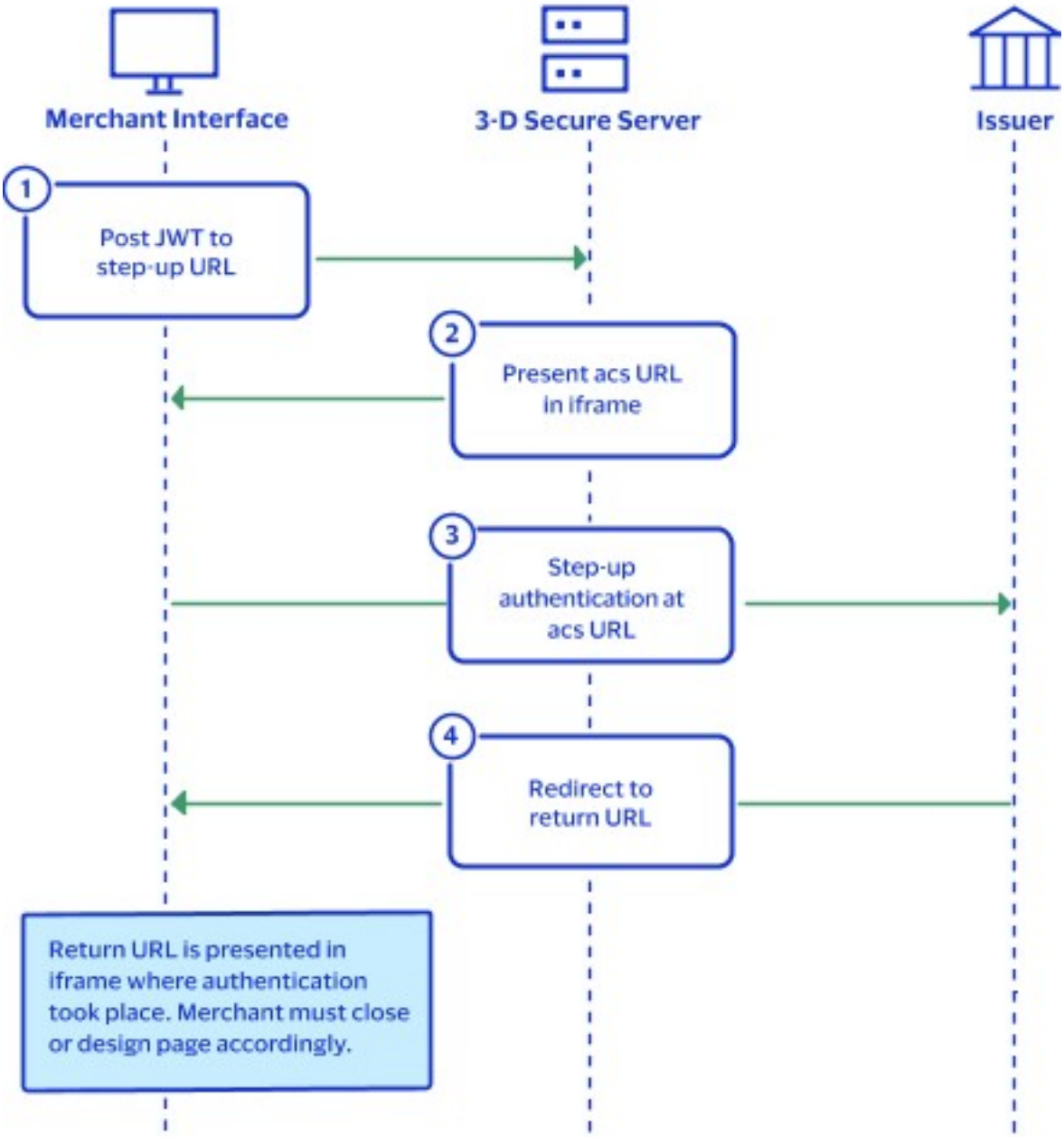
These fields contain values that are used in the Step-Up service, which runs when a customer is challenged to authenticate.

Step 4: Step-Up Iframe

Initiate step-up authentication on the front end after you receive the response as discussed in *Step 3: Payer Authentication Check Enrollment Service* on page 34. Note that frictionless authentication does not require this step-up iframe step. This step is only for step-up authentication when the issuing bank wants to challenge the cardholder.

When a challenge is needed to prove a customer's identity, a JSON Web Token is returned to the merchant that contains a step-up URL. The merchant opens an iframe where the access token to the step-up URL (also known as the endpoint) is posted. The iframe must be sized appropriately to enable the cardholder to complete the challenge. The iframe manages customer interaction with the card-issuing bank's access control server. The bank asks the customer to provide identifying information. Once the customer completes the challenge, the process moves to validating the information that the customer sent.

Process Flow for Step-Up Authentication



Best Practices

These practices should be followed for this step to achieve optimal performance and to minimize potential operating issues.

- When a transaction requires a challenge, according to EMVCo protocol, the challenge must be issued within 30 seconds of the Enrollment Check response. When more than 30 seconds elapses the ACS times out.

Building the Iframe Parameters

The iframe that the merchant displays should be sized to enable the customer bank to exchange authentication information between itself and the customer. Because a bank can use various methods to authenticate, the iframe has four size options. The bank will request that the merchant ensure that the iframe size provides room to display the bank logo and the card network being used, the amount of the transaction, and a brief explanation of what the customer needs to do. The size of the challenge window is managed by the merchant to ensure that the challenge window matches with the presentation screen provided by the merchant. The merchant chooses the iframe parameters and passes the window size to the issuer.

- Use the JWT POST Parameter value from the **payerAuthEnrollReply_accessToken**response field and do a form POST within the iframe to the StepUpUrl value that is passed by the **payerAuthEnrollReply_stepUpUrl** response field
- MD POST Parameter: Merchant-defined data returned in the response. This field is optional.
- Iframe height and width: EMV 3-D Secure 2.x offers multiple size options:
  - Use the **payerAuthEnrollService_acsWindowSize** request field to request a specific window size.
  - Use the **payerAuthEnrollReply_paReq** response field to determine iframe dimensions by Base64 decoding the string and cross-referencing a Challenge Window Size value with its corresponding size.

This table lists the possible values for iframe size and the sizes associated with the value.

Challenge Window Size Value and Corresponding Size

Challenge Window Size Value	Step-Up Iframe Dimensions (Width x Height in pixels)
01	250 x 400
02	390 x 400
03	500 x 600
04	600 x 400
05	Full screen

This is an example for the decoded value.

Challenge Window Size Decoded Value


```
{
  "messageType":"CReq","messageVersion":"2.2.0",
  "threeDSServerTransID":"c4b911d6-1f5c-40a4-bc2b-51986a98f991",
  "acsTransID":"47956453-b477-4f02-a9ef-0ec3f9f779b3",
```

```
  "challengeWindowSize":"02"
}
```


Creating the Iframe

Create an iframe that is the same size as the Challenge Window Size to send a POST request to the step-up URL. Study this example.

Send a POST Request to the Step-Up URL


```
<iframe name=”step-up-iframe" height="400" width="400"></iframe>
<form id="step-up-form" target=”step-up-iframe" method="post" action=" https://centinelapistag.
 cardinalcommerce.com/V2/Cruise/StepUp"> <input type="hidden" name="JWT" value="eyJhbGciOiJIUz
 I1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiJmNmFmMTRmOS04YWRjLTRiNzktOGVkYS04YWVlMTI2NTkzZTEiLCJp
 YXQiOjE1OTYwNTEyNzYsImlzcyI6IjVkZDgzYmYwMGU0MjNkMTQ5OGRjYmFjYSIsImV4cCI6MTU5NjA1NDg3NiwiT3Jn
 VW5pdElkIjoiNTVlZjNmNTZmNzIzYWE0MzFjOTlkNTRiIiwiUGF5bG9hZCI6eyJBQ1NVcmwiOiJodHRwczovLzBt
 ZXJjaGFudGFjc3N0YWcuY2FyZGluYWxjb21tZXJjZS5jb20vTWVyY2hhbnRBQ1NXZWIvY3JlcS5qc3AiLCJQYXlsb2
 FkIjoiZXlKdFpYTnpZV2RsVkhsd1pTSTZJa05TWlhFaUxDSnRaWE56WVdkbFZtVnljMmx2YmlJNklqSXVNaTR3SWl3
 aWRHaHlaV1ZFVTFObGNuWmxjbFJ5WVc1elNVUWlPaUpsTkdKaVpqazNNeTFqTW1FeUxUUTNOREF0T1RWak5
 DMWpNVGhoTVRNeE16TmlPRFFpTENKaFkzTlVjbUZ1YzBsRUlqb2lNVGMzT0RFM016SXROREk1TVMwME1HUmlMVG
 xoTkRndE1ESm1OREpoTlRZd1lqYzVJaXdpWTJoaGJHeGxibWRsVjJsdVpHOTNVMmw2WlNJNklqQXlJbjAiLCJUcm
 Fuc2FjdGlvbklkIjoiQnh5a0hYVEp4M1JuNHBGWnF1bjAifSwiT2JqZWN0aWZ5UGF5bG9hZCI6dHJ1ZSwiUmV0
 dXJuVXJsIjoiaHR0cHM6Ly9taWNoYWVsdGF5bG9yLmlvL2N5YnMvc3RvcmVEZW1vL3B1YmxpYy9saXN0ZW5lci5
 weSJ9.H8j-VYCJK_7ZEHxGz82_IwZGKBODzPaceJNNC99xZRo" /> <input type="hidden" name="MD"
 value="optionally_include_custom_data_that_will_be_returned_as_is”/> </form>
```


Invoking the Iframe

Add JavaScript to invoke the iframe form POST. Place the JavaScript after the closing </ body> tag as shown in the example below. The JavaScript invokes the iframe form POST automatically when the window loads. While you can submit the form at a different time, you must submit the form before requesting the validation service.

```
<script>
window.onload = function() {
   var stepUpForm = document.querySelector('#step-up-form');
   if(stepUpForm) // Step-Up form exists
   stepUpForm.submit();
}
</script>
```

Receiving the Step-Up Results

After the customer interacts with the issuing bank, the customer is returned to the **payerAuthEnrollService_returnURL** within the iframe as specified in *Step 3: Payer Authentication Check Enrollment Service* on page 34. The payload sent to the return URL is URL-encoded and Base64-encoded (see the example below). The merchant hosting the return URL can then close the iframe after redirection.
The response sent back to the return URL contains these values:

- Transaction ID: (**payerAuthEnrollReply_authenticationTransactionID** response field). This value is used in *Step 5: Payer Authentication Validation Service* on page 44.
- MD: merchant data returned if present in the POST to step-up URL; otherwise, null.

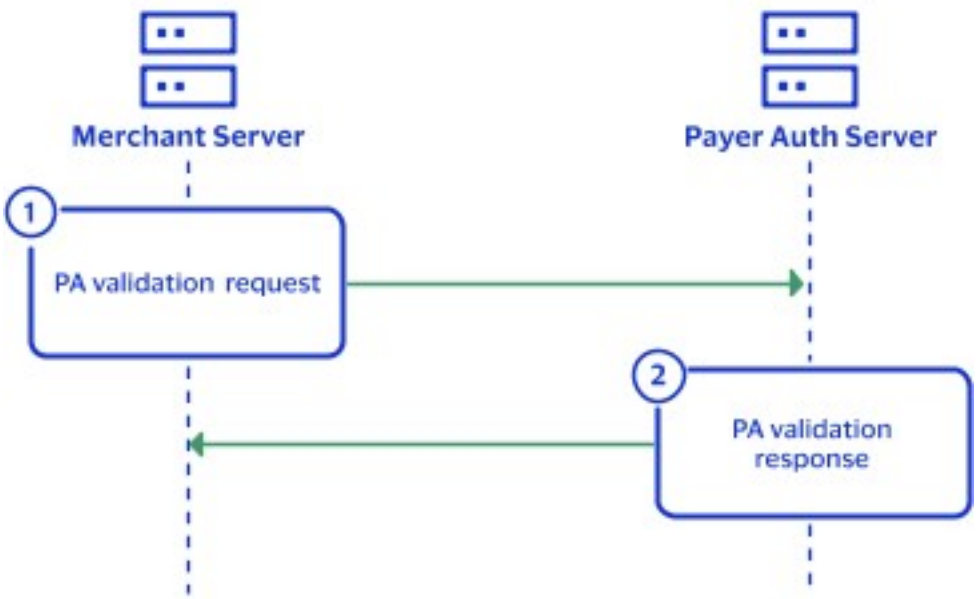
POST to Return URL

```
TransactionId=BwNsDeDPsQV4q8uy1Kq1&MD=null
```

Step 5: Payer Authentication Validation Service

When you receive the step-up response as discussed in *Step 4: Step-Up Iframe* on page 39, make a validation call to verify that the customer successfully authenticated. Note that frictionless authentication does not require this validation step. Validation is required only for step-up authentication.

Process Flow for Validation of the Payer



Request Fields

The `payerAuthValidateService_authenticationTransactionID` field in this step is mapped from the `payerAuthEnrollReply_authenticationTransactionID` field in *Step 4: Step-Up Iframe* on page 39.

These fields are required:

- **card_cardType**
- **card_expirationMonth**
- **card_expirationYear**
- **card_accountNumber**
- **merchantReferenceCode**
- **payerAuthValidateService_authenticationTransactionID**
- **purchaseTotals_currency**
- **purchaseTotals_grandTotalAmount** or **item_#_unitPrice**

For examples, see *Use Case: Validating Payer Authentication* on page 131.
For further details on individual API fields, refer to the *API Field Reference Guide*.

Interpreting the Validation Response

If the authentication is rejected (TransStatus R), Visa, American Express, JCB, Diners Club, Discover, China UnionPay, and Elo recommend not proceeding to authorization. Instead, ask the customer to use another payment method.
Proceed with the order according to the validation response that you receive. The possible validation response statuses are the same for all of the card types.

Reason Code 100

Successful Step-up Authentication

- PARes status = `Y`

Step-up authentication of the customer was successful. If you request the Validate Authentication and Authorization services separately, you must add the required payer validate payload values to your authorization request before you can receive chargeback protection that shifts the liability to the issuer.
Unavailable Step-up Authentication

- PARes status = `U`

Step-up authentication was unavailable and the customer could not be authenticated. This status does not necessarily indicate any fraudulent intent from the customer. Merchants can either attempt to retry authentication or continue to authorization. If you are making separate validatation and authorization calls, you can still proceed with the authorization request but there is no liability shift. Without authentication, the merchant remains liable for any chargeback if it should occur with the transaction.

Reason Code 476

Unavailable Step-up Authentication

- PARes status = `N`

The customer could not be authenticated. Do not submit this transaction for authorization. Instead ask the customer for another form of payment.

Error

If you receive an error from the payment card company, process the order according to your business rules. If the error occurs frequently, report it to customer supportcustomer supportcustomer support. If you receive a system error, determine the cause of the error and proceed with card authorization only when appropriate.

Redirecting Customers to Pass or Fail Message Page

After authentication is complete, redirect the customer to a page containing a success or failure message. You must ensure that the messages that display to customers are accurate and complete, and that the message addresses all possible scenarios for enrolled and non-enrolled cards. For example, if the authentication fails, display a message such as this to the customer:

```
Authentication Failed
Your card issuer cannot authenticate this card. Please select another card or form of payment to
 complete your purchase.
```

Combining the Authentication and the Authorization Services

After the customer is successfully authenticated, you must get authorization from the issuing bank to proceed with the transaction. While these are separate processes, it is recommended that you link these services by immediately passing the returned values into a request to authorize the transaction. The two services can be linked when:

- Checking enrollment determines that no challenge is required. Pass the values returned from checking enrollment to the authorization request.
- Validating a challenge authenticates the cardholder. Pass the values returned from validating the challenge to the authorization request.

With the same request transactions, a different endpoint must be referenced for the authorization, and an additional element must be added to the JSON. When step-up authentication is required, transaction processing stops to allow completion of authentication, and authorization is not called until after the challenge response is validated. This integration method is recommended.
Depending on your card type, you might not receive the XID value. If you receive this field under a frictionless scenario, it is required for authorization.

Combining Check Enrollment and the Authorization Services

Receiving certain responses from checking enrollment allows the authorization to be requested immediately afterwards. The possible checking enrollment responses are:

- Successful frictionless authentication
- Attempted stand-in frictionless authentication
- Issuer does not support the payer authentication program
- Account is not eligible for a payer authentication program

- Unavailable frictionless authentication
- Failed frictionless authentication
- Rejected frictionless authentication

In all checking enrollment scenarios, it is recommended that you integrate these services by combining the checking enrollment and authorization services into a single transaction. When the services are combined, one of these conditions occurs:

- No additional integration work is required to manually map the appropriate check enrollment results to the corresponding authorization request fields.
- If further authentication is needed, the authorization cannot happen until after authentication completes and you can proceed to the next steps for challenging.

With same request transactions, a different endpoint must be referenced for the authorization, and an additional element must be added to the JSON. Depending on your card type, you might not receive the XID value. If you receive this field under a frictionless scenario, it is required for authorization.

Check Enrollment Response Fields and Their Equivalent Authorization Request Fields

When a customer is authenticated without a challenge, the transaction can be authorized either in the same request or in a separate authorization request. Whether authorization occurs in the same request or a separate request, the values from the check enrollment response must be passed to the authorization request to qualify for a liability shift. This table matches the check enrollment fields with their equivalent authorization fields. Sometimes a check enrollment response field is the same field used in the authorization request.
Be sure to include the following card-specific information in your authorization request:

- For Visa, American Express, JCB, Diners Club, Discover, China UnionPay, and Elo, include the *CAVV* (Cardholder Authentication Verification Value).
- For Mastercard only, include the collection indicator and the *AAV* (also known as *UCAF*).

Enrollment Check and Response Fields

Identifier	Enrollment Check Response Field	Card Authorization Request Field
E-commerce indicator	payerAuthEnrollReply_commerceIndicator	ccAuthService_commerceIndicator
Collection indicator	payerAuthEnrollReply_ucafCollectionIndicator	ucaf_collectionIndicator
CAVV	payerAuthValidateReply_cavv	ccAuthService_cavv
AAV	payerAuthValidateReply_ucafAuthenticationData	ucaf_authenticationData

Identifier	Enrollment Check Response Field	Card Authorization Request Field
XID	payerAuthEnrollReply_xid	ccAuthService_xid
Result of the enrollment che ck for Asia, Middle East, and Africa Gateway	payerAuthEnrollReply_veres Enrolled	
3-D Secure version	payerAuthEnrollReply_specif icationVersion	ccAuthService_paSpecificat ionVersion
Directory server transaction ID(Not required for 3-D Sec ure 1.0.)	payerAuthEnrollReply_direc toryServerTransactionID	ccAuthService_directorySe r verTransactionID

Combining the Validation and the Authorization Services

After the customer is successfully authenticated, you must get authorization from the issuing bank to proceed with the transaction. While these are separate processes, you should integrate these two services into a single process whenever possible. When you do so, no additional integration work is required on your part to manually map the appropriate validation results to corresponding authorization request fields.
With the same request transactions, a different endpoint must be referenced for the authorization, and an additional element must be added to the JSON. When step-up authentication is required, transaction processing stops to allow authentication to complete, and authorization is not called until after the challenge response is validated. This integration method is highly recommended. Depending on your card type, you might not receive the XID value. If you receive this field under a frictionless scenario, it is required for authorization.

Validation Fields and their Equivalent Authorization Fields

When a customer is authenticated after a challenge, the transaction can be authorized in the same request or in a separate authorization request. Whether authorization is combined with validation or occurs in a separate request, the values from the validation response must be passed to the authorization request to qualify for a liability shift to the issuing bank. This table pairs the Validation field with its equivalent Authorization API field. Be sure to include the following card-specific information in your authorization request:

- For Visa, American Express, JCB, Diners Club, Discover, China UnionPay, and Elo, include the CAVV.
- For Mastercard only, include the collection indicator and the AAV (also known as UCAF).

Validation Check and Response Fields

Identifier	Validation Check Response Field	Card Authorization Request Field
E-commerce indicator	payerAuthValidateReply_commerceIndicator	e_commerce_indicator
Collection indicator	payerAuthValidateReply_ucafCollectionIndicator	ucaf_collection_indicator
CAVV	payerAuthValidateReply_cavv	ccAuthService_cavv
AAV	payerAuthValidateReply_ucafAuthenticationData	ucaf_authenticationData
XID	payerAuthValidateReply_xid	ccAuthService_xid
3-D Secure version	payerAuthValidateReply_specificationVersion	ccAuthService_paSpecificationVersion
Directory server transaction ID	payerAuthValidateReply_directory ServerTransactionID	ccAuthService_directoryServerTransactionID

Implementing SDK Payer Authentication

This chapter summarizes the process of integrating SDK Payer Authentication services into your mobile application. Payer authentication services use the Mobile SDK for iOS or Android to facilitate the authentication. New SDK versions are frequently released and you should ensure that you stay current with the latest release. One way to stay informed on about new releases is to subscribe to a distribution list to be informed of updates and other product announcements. You can subscribe by going to this link: *https://win.cardinalcommerce.com/CardinalMobileSDKNotifications*
Implementing the SDK in your mobile application requires either Android or iOS platform application programming skills. Android API 21 or iOS 9 and XCode 8 are required.
The SDK is only designed to handle EMV 3-D Secure 2.x transactions.

Implementation Overview

Notify your account representative that you want to implement payer authentication (EMV 3-D Secure). Give the representative the merchant ID that you will use for testing. For more information, see *Payer Authentication Merchant Workflow* on page 20.
Implementation tasks include:

- Download, import, and configure the Mobile SDK for either iOS or Android.
- For each purchase request:
  - Build the authentication request.
  - Invoke the authentication.
  - Handle declines.
  - Make another back-end, server-to-server call to request these services:
    `payerAuthValidateService`: Payer Authentication Validation
    `ccAuthService`: Card Authorization service (optional)
- Use the test cases to test your preliminary code and make appropriate changes. See *Testing Payer Authentication* on page 69.
- Ensure that your account is configured for production.

Note that calling the Payer Authentication Setup Service is not required with the SDK mobile version.

Process Flow for SDK Integration

The steps required to integrate payer authentication into an SDK mobile application are described below.

1. Contact customer support to register for an API key.
2. Download and import the Mobile SDK for either iOS or Android.
3. Set up your build environment.
4. Configure your SDK.
5. Setup the initial call to Cardinal.
6. Create an API call to your merchant server to request the Enrollment Check service, passing in transaction details and the `payerAuthEnrollService_referenceID` request field.
7. If the issuing bank does not require authentication, you receive this information in the Enrollment Check response:
   - E-commerce indicator (**payerAuthEnrollReply_commerceIndicator**)
   - CAVV (all card types except Mastercard) (**payerAuthEnrollReply_cavv**)
   - AAV (Mastercard only) (**payerAuthEnrollReply_ucafCollectionIndicator**)
   - Transaction ID (**payerAuthEnrollReply_xid**)
   - 3-D Secure version (**payerAuthEnrollReply_specificationVersion**)
   - Directory server transaction ID (**payerAuthEnrollReply_directoryServerTransactionID**)
8. If the issuing bank requires authentication, you receive a response with the payload and the transaction ID that you include in the Cardinal.continue call from your SDK.
9. The Mobile SDK displays an authentication window, and the customer enters the authentication information into that window.
10. The bank validates the customer credentials and a Java Web Token (JWT) is returned by the SDK in the onValidated callback that the merchant is required to validate server-side for security reasons.
11. Create an API call to your merchant server to request the Validate Authentication service, extracting the processor transaction ID value from the JWT and sending it in the `payerAuthValidateService_authenticationTransactionID` request field. You receive the e-commerce indicator, CAVV or AAV, transaction ID, 3-D Secure version, and directory server transaction ID.

Verify that the authentication was successful and continue processing your order. You must pass all pertinent data for the card type and processor in your authorization request. For more information, see *Requesting the Validation Service* on page 66.

Prerequisites for SDK Implementation

Before you can implement payer authentication services, your business team must contact your acquirer and Cybersource to establish the service. Your software development team should become familiar with the API fields and technical details of this service.
Creating a mobile application with the SDK implementation, requires that you perform some preliminary procedures before the starting the actual payer authentication implementation process. These processes involving JWTs are described in this section.

Credentials/API Keys

API keys are required to create the JSON Web Token (JWT). For further information, contact *customer support* .
You will receive an email with your username and a temporary password. Your username will be in this format:
`cybersource_merchant name_contact name`
For example:
`cybersource_petairways_peter`
Once you receive your credentials, log in to your JFrog account and update your temporary password. Follow the process below to generate your API key.

Generating your API Key:

1. Log in to your JFrog account.
2. In the top-right of the JFrog Platform, select the Welcome drop-down menu and click **Edit Profile**.
3. Enter your password and click **Unlock**.
4. Under Authentication Settings, click **Generate API Key**.

What Mobile Device Data is Collected

One of the key components to authenticating a cardholder during an online transaction is to compare information about the mobile device that the buyer is using to the information about mobile devices that the buyer used in past transactions. This information is maintained in the acess control server (ACS) at the issuing bank.
In mobile device transactions, information collected about the buyer device can include:

- Device ID
- Device model
- Operating system version
- System language
- Country
- Time zone

- Screen dimensions

A successful device data collection process that includes the eleven browser fields listed in the check enrollment step, increases the chances of a frictionless authentication. The decision to escalate a transaction to a level of risk high enough to require challenging the buyer to authenticate their identity is managed by business rules that are configured in the issuer's risk analysis software that evaluates each transaction.

Using the Android SDK

A mobile SDK is available for integrating payer authentication services into mobile applications running on the Android platform.

Updating the Gradle Build Properties

In Android Studio, open the app directory (which can also be labeled Module: app) and open the build.gradle file. Edit the Gradle file located in the app directory. Add the contents shown in the example below to the Gradle file.

```
repositories {
  ...
  maven {
      url "https://cardinalcommerceprod.jfrog.io/artifactory/android"
      credentials {
          username Artifactory username
          password Artifactory user API Key
      }
  }
}
dependencies {
  ...
  //Cardinal Mobile SDK
  implementation 2.5-1
}
```

If your project uses Progurad, add the lines shown below to the proguard-rules.pro file.

```
-keep class com.cardinalcommerce.dependencies.internal.bouncycastle.**
-keep class com.cardinalcommerce.dependencies.internal.nimbusds.**
```

Configuring the Android SDK

Get the instance of the Cardinal object by Cardinal.getInstance(). Use the default configuration options. See the example below to complete Cardinal.configure().

For more details on configuration, refer to the configuration options table after the example.

```
private Cardinal cardinal = Cardinal.getInstance();
@Override
protected void onCreate(Bundle savedInstanceState) {

CardinalConfigurationParameters cardinalConfigurationParameters = new
 CardinalConfigurationParameters();
      cardinalConfigurationParameters.setEnvironment(CardinalEnvironment.STAGING);
      cardinalConfigurationParameters.setTimeout(8000);
      JSONArray rType = new JSONArray();
      rType.put(CardinalRenderType.OTP);
      rType.put(CardinalRenderType.SINGLE_SELECT);
      rType.put(CardinalRenderType.MULTI_SELECT);
      rType.put(CardinalRenderType.OOB);
      rType.put(CardinalRenderType.HTML);
      cardinalConfigurationParameters.setRenderType(rType);

      cardinalConfigurationParameters.setUiType(CardinalUiType.BOTH);

      UiCustomization yourUICustomizationObject = new UiCustomization();
cardinalConfigurationParameters.setUICustomization(yourUICustomizationObject);

      cardinal.configure(this,cardinalConfigurationParameters);
}
```

Android Configuration Options

Method	Description	Default Values
setEnableDFSync (boolean enableDFSync)	On setting true, onSetupCompleted is called after the collected device data is sent to the server.	False
setEnableQuickAuth (boolean enableQuickAuth)	Sets enable quick auth false.	False
setEnvironment(Setting up mobile SDK - Android- V 2.1# CardinalEnvironment environment)	Sets the environment to which the SDK must connect.	CardinalEnvironment. PRODUCTION
setProxyAddress(java.lang. String proxyAddress)	Sets the proxy to which the SDK must connect.	“ “

Method	Description	Default Values
setRenderType(org.json. JSONArray renderType)	Sets renderLists all user interface types that the device supports for displaying specific challenge user interfaces within the SDK.	JSONArray rType = new JSONArray(); rType.put(Cardinal RenderType.OTP); rType.put(Cardinal RenderType.SINGLE_SELECT); rType.put(Cardinal RenderType.MULTI_SELECT); rType.put(Cardinal RenderType.OOB); rType.put(Cardinal RenderType.HTML);
setTimeout(int timeout)	Sets the maximum amount of time (in milliseconds) for all exchanges.	8000
setUICustomization (UiCustomization UI Customization)	Sets UICustomization	Device Default Values
setUiType(CardinalUiType uiType)	Sets all user interface types that the device supports for displaying specific challenge user interfaces within the SDK.	CardinalUiType.BOTH

Setting Up the Initial Call

Calling Cardinal.init():

- begins the communication process with Cardinal
- authenticates your credentials (server JWT)
- completes the data collection process

By the time the customer is ready to check out, all necessary preprocessing is complete. Each time a user begins a mobile transaction, Cardinal assigns a unique identifier to the session called a **consumerSessionId**. This **consumerSessionId** ensures that Cardinal matches the correct device data collection results to a request. Cybersource calls this session identifier, **payerAuthEnrollService_referenceID**. You must assign the value of the **consumerSessionId** field to the **payerAuthEnrollService_referenceID** field so that Cybersource can also track the calls for each user session.
Study the code example shown below for completing the cardinal.init().

Cardinal.init() (Android SDK)

```
cardinal = Cardinal.getInstance();
String serverJwt = "INSERT_YOUR_JWT_HERE";
cardinal.init(serverJwt ,
new CardinalInitService() {
   /**
   * You may have your Submit button disabled on page load. Once you are
```

```
    * set up for CCA, you may then enable it. This will prevent users
    * from submitting their order before CCA is ready.
    */
    @Override
    public void onSetupCompleted(String consumerSessionId) {

    }
    /**
    * If there was an error with set up, Cardinal will call this function
    * with validate response and empty serverJWT
    * @param validateResponse
    * @param serverJwt will be an empty
    */
    @Override
    public void onValidated(ValidateResponse validateResponse, String serverJwt) {

    }
});
```

See *Running Payer Authentication with SDK* on page 61 for the next steps.

Using the iOS SDK

A mobile SDK is available for integrating payer authentication services into mobile applications running on the iOS platform.

Downloading and Importing the SDK

Download the CardinalMobile.framework file using cURL in this example.

Download CardinalMobile.framework

```
curl -L -u <USER_NAME>
      :<API_KEY> https://cardinalcommerceprod.jfrog.io/artifactory/ios/<VERSION>-<BUILD_NUMBER>/
cardinalmobilesdk.zip
      -o <LOCAL_FILE_NAME.EXT>

#Example:
curl -L -u UserName:ApiKey "https://cardinalcommerceprod.jfrog.io/artifactory/ios/2.2.5-1/
cardinalmobilesdk.zip" -o cardinalmobile2.2.5-1.zip
```

Download the CardinalMobile.xcframework file using the cURL in this example.

Download CardinalMobile.xcframework

```
curl -L -u <USER_NAME>
      :<API_KEY> https://cardinalcommerceprod.jfrog.io/artifactory/ios/<VERSION>-<BUILD_NUMBER>/
CardinalMobileiOSXC.zip
      -o <LOCAL_FILE_NAME.EXT>

#Example:
curl -L -u UserName:ApiKey "https://cardinalcommerceprod.jfrog.io/artifactory/ios/2.2.5-1/
CardinalMobileiOSXC.zip" -o cardinalmobile2.2.5-1.zip
```

In your XCode project, drag the CardinalMobile.framework file into the Frameworks group in your Xcode Project. (Create the group if it doesn't already exist.) In the import dialog box, check the box to Copy items into the destinations group folder (or Destination: Copy items if needed). The iOS SDK files are now available for linking in your project.

Configuring Your Build Environment

1. Open Xcode and in the source list to the left of the main editor area, choose your project.
2. Under the Targets section, select your application and open the General tab.
3. Expand the Embedded Binaries section and click the small plus (+) at the bottom of the list.
4. Add CardinalMobile.framework from the list.

Configuring the iOS SDK

Create a new instance of the cardinal object by CardinalSession new. Use the default configuration options. Study these examples to complete the iOS SDK configuration. For more details on configuration options, refer to the table after the examples.

CardinalSession new (iOS SDK - Objective-C)

```
#import <CardinalMobile/CardinalMobile.h>

CardinalSession *session;

//Setup can be called in viewDidLoad
- (void)setupCardinalSession {
   session = [CardinalSession new];
   CardinalSessionConfiguration *config = [CardinalSessionConfiguration new];
   config.deploymentEnvironment = CardinalSessionEnvironmentProduction;
   config.timeout = CardinalSessionTimeoutStandard;
   config.uiType = CardinalSessionUITypeBoth;

   UiCustomization *yourCustomUi = [[UiCustomization alloc] init];
   //Set various customizations here. See "iOS UI Customization" documentation for detail.
   config.uiCustomization = yourCustomUi;

   CardinalSessionRenderTypeArray *renderType = [[CardinalSessionRenderTypeArray alloc]
 initWithObjects:
                    CardinalSessionRenderTypeOTP,
                    CardinalSessionRenderTypeHTML,
                    nil];
   config.renderType = renderType;

   config.enableQuickAuth = false;
   [session configure:config];
}
```

CardinalSession new (iOS SDK - Swift)

```
import CardinalMobile
```

```
var session : CardinalSession!

//Setup can be called in viewDidLoad
func setupCardinalSession{
   session = CardinalSession()
   var config = CardinalSessionConfiguration()
   config.deploymentEnvironment = .production
   config.timeout = 8000
   config.uiType = .both

   let yourCustomUi = UiCustomization()
   //Set various customizations here. See "iOS UI Customization" documentation for detail.
   config.uiCustomization = yourCustomUi

   config.renderType = [CardinalSessionRenderTypeOTP, CardinalSessionRenderTypeHTML]
   config.enableQuickAuth = true
   session.configure(config)
}
```

iOS Configuration Options

Method	Description	Default Values	Possible Values
deploymentEnvironment	The environment to which the SDK connects.	CardinalSessionEnvironmentProduction	CardinalSessionEnvironment Staging CardinalSessionEnvironment Production
timeoutInMilliseconds	Maximum amount of time (in milliseconds) for all exchanges.	8000	
uiType	Interface types that the device supports for displaying specific challenge user interfaces within the SDK.	CardinalSessionUIType Both	CardinalSessionUITypeBoth CardinalSessionUITypeNative CardinalSessionUITypeHTML
renderType	List of all the render types that the device supports for displaying specific challenge user interfaces within the SDK.	[CardinalSessionRenderTypeOTP, CardinalSessionRenderTypeHTML, CardinalSessionRenderTypeOOB, CardinalSessionRenderTypeSingleSelect, CardinalSessionRenderTypeMultiSelect]	CardinalSessionRenderType OTP CardinalSessionRenderType HTML CardinalSessionRenderType OOB CardinalSessionRenderType SingleSelect CardinalSessionRenderType MultiSelect

Method	Description	Default Values	Possible Values
proxyServerURL	Proxy server through which the Cardinal SDK Session operates.	nil	
enableQuickAuth	Enable Quick Authentication	false	
uiCustomization	Set Custom UICustomization for SDK-Controlled Challenge UI.	nil	
enableDFSync	Enable DF Sync to get onSetupCompleted called after collected device data is sent to the server.	false	

Setting Up the Initial Call

Calling cardinal session setup begins the communication process, authenticates your credentials (server JWT), and completes the data collection process. By the time the customer is ready to check out, all necessary preprocessing is complete.
Each time a user begins a mobile transaction, a unique value is assigned to the **consumerSessionId** API field to identify the session. This **consumerSessionId** value ensures that the correct device data collection results is matched to each user request. Cybersource uses its **payerAuthEnrollService_referenceID** field to contain Cardinal's **consumerSessionId** value. You must assign the value of the **consumerSessionId** field to the **payerAuthEnrollService_referenceID** field so that Cybersource can also track the calls for each user session.
Study these code examples to understand how to complete the cardinal session setup. The function call must be placed in your Checkout ViewController.

Cardinal session setup (iOS SDK - Objective-C)

```
NSString *accountNumberString = @"1234567890123456";
NSString *jwtString = @"INSERT_YOUR_JWT_HERE";

[session setupWithJWT:jwtString
        didComplete:^(NSString * _Nonnull consumerSessionId){
//
    // You may have your Submit button disabled on page load. Once you are
    // setup for CCA, you may then enable it. This will prevent users
    // from submitting their order before CCA is ready.
    //
} didValidate:^(CardinalResponse * _Nonnull validateResponse) {
    // Handle failed setup
    // If there was an error with setup, cardinal will call this
    // function with validate response and empty serverJWT
```

```
}];
```

Cardinal session setup (iOS SDK - Swift)

```
let accountNumberString = "1234567890123456"
let jwtString = "INSERT_YOUR_JWT_HERE"

session.setup(jwtString: jwtString, completed: { (consumerSessionId: String) in
    //
    // You may have your Submit button disabled on page load. Once you
    // are setup for CCA, you may then enable it. This will prevent
    // users from submitting their order before CCA is ready.
    //
}) { (validateResponse: CardinalResponse) in
    // Handle failed setup
    // If there was an error with setup, cardinal will call this
    // function with validate response and empty serverJWT
}
```

Running Payer Authentication with SDK

The payer authentication process in SDK requires checking whether a customer is participating in a card authentication program. If the customer is enrolled in payer authentication, you validate their current status in the program and authorize the transaction. The following procedures describe how to ensure the correct data values are passed during the payer authentication process.

Requesting the Check Enrollment Service (SDK)

After the SDK completes the device collection from your mobile application, and after the customer clicks the Buy button, you must make a back-end, server-to-server call to request the Enrollment Check service.
The Check Enrollment service verifies that the card is enrolled in a card authentication program. The merchant ID is included as part of the header, but these fields are required in the request:

- **billTo_city**
- **billTo_country**
- **billTo_email**
- **billTo_firstName**
- **billTo_lastName**
- **billTo_postalCode**
- **billTo_state**
- **billTo_street1**
- **card_accountNumber**
- **card_cardType**
- **card_expirationMonth**
- **card_expirationYear**

- **merchantID**
- **merchantReference Code**
- **payerAuthEnrollService_referenceID**
- **payerAuthEnrollService_run**
- **purchaseTotals_currency**
- **purchaseTotals_grandTotalAmount**

> **Important**
>
> To reduce your issuer step-up authentication rates, you can send additional request data in order. It is best to send all available fields.

Use the enrollment check and card authorization services in the same request or in separate requests:

- Same request: Cybersource attempts to authorize the card if your customer is not enrolled in a payer authentication program. In this case, the field values that are required to prove that you attempted to check enrollment are passed automatically to the authorization service. If authentication is required, processing automatically stops.
- Separate requests: Manually include the enrollment check result values (Enrollment Check response fields) in the authorization service request (Card Authorization request fields).

Be sure to include the following card-specific information in your authorization request:

- For Visa, American Express, JCB, Diners Club, Discover, China UnionPay, and Elo, include the CAVV.
- For Mastercard only, include the collection indicator and the AAV (also known as UCAF).

These fields are listed in this table.

Enrollment Check and Response Fields

Identifier	Enrollment Check Response Field	Card Authorization Request Field
E-commerce indicator	payerAuthEnrollReply_ commerceIndicator	ccAuthService_commerceIndicator
Collection indicator	payerAuthEnrollReply_ucafCollectionIndicator	ucaf_collectionIndicator
CAVV	payerAuthValidateReply_cavv	ccAuthService_cavv
AAV	payerAuthValidateReply_ucafAuthenticationData	ucaf_authenticationData
XID	payerAuthEnrollReply_xid and payerAuthValidateReply_xid	ccAuthService_xid

Identifier	Enrollment Check Response Field	Card Authorization Request Field
Result of the enrollment check for Asia, Middle East, and Africa Gateway	payerAuthEnrollReply_veresEnrolled	
3-D Secure version	payerAuthEnrollReply_specificationVersion	ccAuthService_paSpecificationVersion
Directory server transaction ID(Not required for 3-D Secure 1.0.)	payerAuthEnrollReply_directoryServerTransactionID	ccAuthService_directoryServerTransactionID

Interpreting the Response

In EMV 3-D Secure, there are two possible responses:

- Frictionless: No challenge or stepup to the cardholder. While frictionless authentication can indicate a successfully authenticated outcome because the customer's card is enrolled in a payer authentication program, it can also result from the bank failing or rejecting authentication without challenging the cardholder. In the frictionless authentication flow, you receive a PAResStatus of either `Y`, `A`, `N`, `I`, `R`, or `U` with an associated ECI value. With successful frictionless authentication, the PAResStatus = `Y` or `A` and you receive a CAVV. You may also receive a PAResStatus = `I` indicating successful authentication but it might not include a CAVV.
- Challenge: The response contains PAResStatus = `C`. A challenge response has a payload and contains an ACS URL and a StepUpUrl. Challenge the cardholder to provide additional authentication information and display an authentication challenge window to the cardholder so the cardholder can respond to a validation request and receive a validation response.

Authenticating Enrolled Cards

In the response from the enrollment check service, confirm that you receive these fields and values:

- 3-D Secure version = 2.x
- VERes enrolled = Y
- PARes status = C

These values identify whether it is a EMV 3-D Secure 2.x transaction and that a challenge is required. If the 3-D Secure version is 1.0, then the SDK is no longer applicable and you must open up a WebView.
Once you validate these fields, you call Cardinal.cca_continue (Android SDK) or Cardinal session continue (iOS SDK) for the SDK to perform the challenge between the customer and the issuing bank.

Calling Cardinal.cca_continue (Android SDK)

When you have verified that a customer's card is enrolled in a card authentication program, you must take the payload, and the `payerAuthEnrollReply_authenticationTransactionID` response field and include them in the Cardinal.cca_continue function before proceeding with the authentication session as shown in this example.

```
/**
 * Cca continue.
 *
 * @param transactionId     the transaction id
 * @param payload           the payload
 * @param currentActivity   the current activity
 * @throws InvalidInputException       the invalid input exception
 * @throws JSONException               the json exception
 * @throws UnsupportedEncodingException the unsupported encoding exception
 */
try {
   cardinal.cca_continue("[TRANSACTION ID ]", "[PAYLOAD]", this, new CardinalValidateReceiver() {
        /**
         * This method is triggered when the transaction
         * has been terminated. This is how SDK hands back
         * control to the merchant's application. This method will
         * include data on how the transaction attempt ended and
         * you should have your logic for reviewing the results of
         * the transaction and making decisions regarding next steps.
         * JWT will be empty if validate was not successful.
         *
         * @param validateResponse
         * @param serverJWT
         */
        @Override
        public void onValidated(Context currentContext, ValidateResponse validateResponse, String
 serverJWT) {
        }
     });
 }
catch (Exception e) {
  // Handle exception
 }
```

Calling Cardinal session continue (iOS SDK)

When you have verified that a customer's card is enrolled in a card authentication program, take the payload, and the `payerAuthEnrollReply_authenticationTransactionID` response field and include them in the Cardinal session continue function before proceeding with the authentication session as shown in *Example 22* .
In Continue, you should pass a class conforming to a protocol CardinalValidationDelegate (and implement a method stepUpDidValidate) as a parameter. These examples show a class conforming to CardinalValidationDelegate protocol.

Objective-C Examples

Cardinal session continue (iOS SDK - Objective-C)

```
@interface YourViewController()<CardinalValidationDelegate>{ //Conform your ViewController or any
 other class to CardinalValidationDelegate protocol

}
@end

@implementation YourViewController

   /**
    * This method is triggered when the transaction has
    * been terminated.This is how SDK hands back
    * control to the merchant's application. This method will
    * include data on how the transaction attempt ended and
    * you should have your logic for reviewing the results of
    * the transaction and making decisions regarding next steps.
    * JWT will be empty if validate was not successful
    *
    * @param session
    * @param validateResponse
    * @param serverJWT
    */
   -(void)cardinalSession:(CardinalSession *)session stepUpDidValidateWithResponse:(CardinalResponse
 *)validateResponse serverJWT:(NSString *)serverJWT{

   }

@end
```

If Continue is called in the same class, call the method shown in the following example to start StepUpFlow.

Cardinal.continue Call in the Same Class (Objective-C)

```
[session continueWithTransactionId: @"[TRANSACTION_ID]"
                 payload: @"[PAYLOAD]"
        didValidateDelegate: self];
```

Swift Examples

Cardinal session continue (iOS SDK - Swift)

```
class YourViewController:CardinalValidationDelegate {

   /**
    * This method is triggered when the transaction has been
    * terminated.This is how SDK hands back
    * control to the merchant's application. This method will
    * include data on how the transaction attempt ended and
    * you should have your logic for reviewing the results of
    * the transaction and making decisions regarding next steps.
    * JWT will be empty if validate was not successful
    *
```

```
    * @param session
    * @param validateResponse
    * @param serverJWT
    */
   func cardinalSession(cardinalSession session: CardinalSession!, stepUpValidated validateResponse:
 CardinalResponse!, serverJWT: String!) {

   }

}
```

If Continue is called in the same class, call the method shown in the example below to start StepUpFlow.

Cardinal.continue Call in the Same Class (Swift)

```
session.continueWith(transactionId: "[TRANSACTION_ID]", payload: "[PAYLOAD]", validationDelegate: self)
```

When necessary, the SDK displays the authentication window and the customer enters their authentication information.

Receiving the Authentication Results

Next onValidated() (Android SDK) or stepUpDidValidate (iOS SDK) launches and returns the authentication results and response JWT along with the processor transaction ID as shown in this example.

Decoded Response JWT


```
{
  "iss": "5a4504be6fe3d1127cdfd94e",
  "iat": 1555075930,
  "exp": 1555083130,
  "jti": "cc532159-636d-4fa8-931d-d4b0f4c83b99",
  "ConsumerSessionId": "0_9a16b7f5-8b94-480d-bf92-09cd302c9230",
  "aud": "d0cf3392-62c5-4107-bf6a-8fc3bb49922b",
  "Payload": {
    "Payment": {
      "Type": "CCA",
      "ProcessorTransactionId": "YGSaOBivyG0dzCFs2Zv0"
    },
    "ErrorNumber": 0,
    "ErrorDescription": "Success"
  }
}
```


Requesting the Validation Service

For enrolled cards, the next step is to make a back-end, server-to-server call to request the validation service.

When you make the validation request, you must:

- Send the `payerAuthValidateService_authenticationTransactionID` request field.

- Send the credit card information including the PAN, currency, and expiration date (month and year).

The response that you receive contains the validation result.
It is recommended that you request the payer authentication and card authorization services at the same time. Doing this automatically sends the correct information to your payment processor and converts the values of these fields to the proper format required by your payment processor:

- `payerAuthEnrollReply_commerceIndicator`
- `payerAuthValidateReply_cavv`
- `payerAuthValidateReply_ucafAuthenticationData`
- `payerAuthEnrollReply_xid` and `payerAuthValidateReply_xid`

If you request the services separately, manually include the validation result values (Validation Check response fields) in the authorization service request (Card Authorization request fields). To receive liability shift protection, you must ensure that you pass all pertinent data for the card type and processor in your request. Failure to do so might invalidate your liability shift for that transaction. Include the electronic commerce indicator (ECI), the transaction ID (XID), the 3-D Secure version, the directory server transaction ID, and this card-specific information in your authorization request.

- For Visa, American Express, JCB, Diners Club, Discover, China UnionPay, and Elo, include the CAVV.
- For Mastercard only, include the collection indicator and the AAV (also known as UCAF).

Validation Check and Response Fields

Identifier	Validation Check Response Field	Card Authorization Request Field
E-commerce indicator	payerAuthValidateReply_commerceIndicator	e_commerce_indicator
Collection indicator	payerAuthValidateReply_ucafCollectionIndicator	ucaf_collection_indicator
CAVV	payerAuthValidateReply_cavv	ccAuthService_cavv
AAV	payerAuthValidateReply_ucafAuthenticationData	ucaf_authenticationData
XID	payerAuthValidateReply_xid	ccAuthService_xid
3-D Secure version	payerAuthValidateReply_specificationVersion	ccAuthService_paSpecificationVersion
Directory server transaction ID	payerAuthValidateReply_directory ServerTransactionID	ccAuthService_directoryServerTransactionID

Interpreting the Response

Important

If the authentication fails, Visa, American Express, JCB, Diners Club, Discover, China UnionPay, and Elo require that you not accept the card. Instead, you must ask the customer to use another payment method.

Proceed with the order according to the validation response received. The responses are similar for all card types:

- Success: You receive reason code 100, and other service requests, including authorization, are processed normally.
- Failure: You receive reason code 476 indicating that the authentication failed, so the other services in your request are not processed.
- Error: If you receive an error from the payment card company, process the order according to your business rules. If the error occurs frequently, report it to *customer support*. If you receive a system error, determine the cause, and proceed with card authorization only if appropriate.

To verify that the enrollment and validation checks are for the same transaction, ensure that the XID in the enrollment check and validation responses are identical.

Redirecting Customers to the Message Page

After authentication is complete, redirect the customer to a page containing a success or failure message. Ensure that all messages that display to customers are accurate, complete, and address all possible scenarios for enrolled and non-enrolled cards. For example, if the authentication fails, display a message such as this to the customer:

```
Authentication Failed
Your card issuer cannot authenticate this card. Please select another card or form of payment to
 complete your purchase.
```

Testing Payer Authentication

After you complete the necessary changes to your Web and API integration, verify that all components are working correctly by performing all the tests for the cards that you support. Each test contains the specific input data and the most important results fields that you receive in the API response.

Important

In October 2022, support for 3-D Secure 1.0 was largely discontinued as EMV 3-D Secure 2.0 superseded the earlier version. A few countries were granted an extension of time to continue to use the 1.0 version. This compliance extension expired in November 2023. Due to the 3-D Secure 1.0 no longer being supported, the section documenting the test cases for 1.0 was removed from the manual.

Testing Process

Use the card number specified in the test with the card's expiration date set to the month of December and the current year plus three. For example, for 2023, use 2026. You also need the minimum required fields for an order.

Important

In most countries, card network support for 3-D Secure 1.0 was discontinued in October 2022 as EMV 3-D Secure 2.0 superseded the older 1.0 version. The compliance extension to continue to use the 3-D Secure 1.0 protocol that was granted to merchants in some South Asian countries expired in November 2023. All countries should now use the EMV 3-D Secure 2.0 test cases when configuring responses to various transaction scenarios. The 3-D Secure 1.0 test cases were removed from this guide. Contact *customer support* for further information.

Enrollment Check Response Fields

Name Used in Test Cases	API Field
ACS URL	payerAuthEnrollReply_acsURL
E-commerce indicator	payerAuthEnrollReply_commerceIndicator
ECI	payerAuthEnrollReply_eci
PAReq	payerAuthEnrollReply_paReq
proofXML	payerAuthEnrollReply_proofXML
Reason code	payerAuthEnrollReply_reasonCode
VERes enrolled	payerAuthEnrollReply_veresEnrolled
XID	payerAuthEnrollReply_xid

Authentication Validation Response Fields

The following table lists only the response fields used in the test cases.

Response Fields Used in the Authentication Validation Test Cases

Name Used in Test Cases	API Field
Authentication result	payerAuthValidateReply_authenticationR esult
E-commerce indicator	payerAuthValidateReply_commerceIndicator
AAV (Mastercard only)	payerAuthValidateReply_ucafAuthenticationData
CAVV (all card types except Mastercard)	payerAuthValidateReply_cavv
Collection indicator	payerAuthValidateReply_ucafCollectionIndi cator
ECI	payerAuthValidateReply_eci
PARes status	payerAuthValidateReply_authenticationSt atusMessage
Reason code	payerAuthValidateReply_reasonCode
XID	payerAuthValidateReply_xid

Test Cases for 3-D Secure 2.x

Use the card number specified in the test with the card's expiration date set to the month of January and the current year plus three. For example, for 2023, use 2026. You also need the minimum required fields for an order.
Be sure to remove spaces in card numbers when testing.
The transaction ID (XID) values are included in 3-D Secure 2.x test cases for legacy reasons. Only Mastercard transactions do not return XID.

While the 3-D Secure version and directory server transaction ID fields are returned for the Check Enrollment and Validate Authentication services, this data is not included in the 3-D Secure 2.x test cases.

Important

Mastercard requires that the 3-D Secure version and directory server transaction ID be included along with all pertinent data in your authorization request.

Test Case 2.1: Successful Frictionless Authentication

Successful frictionless authentication of the cardholder by the card issuer.

Card Numbers

Card Type	Test Card	Number
	3-D Secure 2.1.0	3-D Secure 2.2.0
American Express Card Type = 003	34000 00 0000 1007	34000 00 0000 2708
Cartes Bancaires Mastercard Card Type = 036	520000 00 0000 3001	520000 00 0000 4801
Cartes Bancaires Visa Card Type = 036	400000 00 0000 3006	400000 00 0000 4970
China UnionPay Card Type = 062	620001 00 0020 0000	—
Diners Club Card Type = 005	601100 00 0000 1002	—
Discover Card Type = 004	601100 00 0000 1002	—
Elo Card Type = 054	650529 00 0000 1002	—
JCB J/Secure Card Type = 007	333700 00 0000 0008	333800 00 0000 0296
Mastercard Card Type = 002	520000 00 0000 1005	520000 00 0000 2235
Visa Card Type = 001	400000 00 0000 1000	400000 00 0000 2701

Results for the Check Enrollment Service

Reason code = `100`

`ics_pa_enroll` service was successful.

VERes enrolled = `Y`
PARes status = `Y`
CAVV = <CAVV value>
AVV = <AVV value> (Mastercard only)
XID = <XID value> (American Express only)

E-Commerce Indicator (ECI) Values

This table lists the expected ECI raw value and its respective string value. These values indicate whether the payer was authenticated by the card network. These values should be passed under this test condition when a transaction is submitted for payment authorization.

Network	ECI Raw Value	ECI String Value
American Express	05	aesk
Cartes Bancaires Mastercard	02	spa
Cartes Bancaires Visa	05	vbv
China UnionPay	05	up3ds
Diners Club	05	pb
Discover	05	dipb
Elo	05	cs
ITMX	05	lss
JCB J/Secure	05	js
Mastercard	02	spa
Visa	05	vbv

Results for the Validation Authentication Service

Validation does not apply to this test as no validation is needed when no challenge is issued during the transaction.

Action

If you request Check Enrollment and authorization services separately, add the required payer authentication values to your authorization request. If you request the Check Enrollment and authorization services together, the process described above occurs automatically.

Test Case 2.2: Unsuccessful Frictionless Authentication

Cardholder authentication without a challenge by the card issuer failed.

Card Numbers

Card Type	Test Card	Number
	3-D Secure 2.1.0	3-D Secure 2.2.0
American Express Card Type = 003	34000 00 0000 1015	34000 00 0000 2096
Cartes Bancaires Mastercard Card Type = 036	520000 00 0000 3019	520000 00 0000 4538
Cartes Bancaires Visa Card Type = 036	400000 00 0000 3014	400000 00 0000 4574
China UnionPay Card Type = 062	620001 00 0010 0010	—
Diners Club Card Type = 005	601100 00 0000 1010	—
Discover Card Type = 004	601100 00 0000 1010	—
Elo Card Type = 054	650529 00 0000 1010	—
JCB J/Secure Card Type = 007	333700 00 0000 0990	333800 00 0000 0361
Mastercard Card Type = 002	520000 00 0000 1013	520000 00 0000 2276
Visa Card Type = 001	400000 00 0000 1018	400000 00 0000 2925

Results for the Check Enrollment Service

Reason code = `476`

- User failed authentication.
- Payer cannot be authenticated.

VERes enrolled = `Y`

PARes status = `N`

E-Commerce Indicator (ECI) Values

This table lists the expected ECI raw value and its respective string value. These values indicate whether the payer was authenticated by the card network. These values should be passed under this test condition when a transaction is submitted for payment authorization.

Network	ECI Raw Value	ECI String Value
American Express	07	internet
Cartes Bancaires Mastercard	00	internet
Cartes Bancaires Visa	07	internet or vbv_failure
China UnionPay	07	up3ds_failure
Diners Club	07	internet
Discover	07	internet
Elo	07	internet
ITMX	07	lss_failure
JCB J/Secure	07	internet
Mastercard	00	internet
Visa	07	internet or vbv_failure

Results for the Validation Authentication Service

No results are returned.

Action

While the merchant can still authorize a failed 3-D Secure transaction as a non-authenticated transaction, it is not recommended to submit this transaction for authorization. Instead ask the customer for another form of payment.

Test Case 2.3: Attempts Processing Frictionless Authentication

While the cardholder is enrolled in 3-D Secure, the card issuer does not support 3-D Secure, requiring a stand-in authentication experience.

Card Numbers

Card Type	Test Card	Number
	3-D Secure 2.1.0	3-D Secure 2.2.0
American Express Card Type = 003	34000 00 0000 1023	34000 00 0000 2872
Cartes Bancaires Mastercard Card Type = 036	520000 00 0000 3027	520000 00 0000 4587
Cartes Bancaires Visa Card Type = 036	400000 00 0000 3022	400000 00 0000 4111
China UnionPay Card Type = 062	620001 00 0000 0020	—
Diners Club Card Type = 005	601100 00 0000 1028	—
Discover Card Type = 004	601100 00 0000 1028	—
Elo Card Type = 054	650529 00 0000 1069	—
JCB J/Secure Card Type = 007	333700 00 0000 7045	333800 00 0000 0585
Mastercard Card Type = 002	520000 00 0000 1021	520000 00 0000 2482
Visa Card Type = 001	400000 00 0000 1026	400000 00 0000 2719

Results for the Check Enrollment Service

Reason code = `100`

`ics_pa_enroll service was successful.`

VERes enrolled = `Y`

PARes status = `A`

CAVV = <CAVV value>

AVV = <AVV value> (Mastercard only)

XID = <XID value> (American Express only)

E-Commerce Indicator (ECI) Values

This table lists the expected ECI raw value and its respective string value. These values indicate whether the payer was authenticated by the card network. These values should be passed under this test condition when a transaction is submitted for payment authorization.

Network	ECI Raw Value	ECI String Value
American Express	06	aesk_attempted
Cartes Bancaires Mastercard	01	spa
Cartes Bancaires Visa	06	vbv_attempted
China UnionPay	06	up3ds_attempted
Diners Club	06	pb_attempted
Discover	06	dipb_attempted
Elo	06	cs_attempted
ITMX	06	lss_attempted
JCB J/Secure	06	js_attempted
Mastercard	01	spa
Visa	06	vbv_attempted

Results for the Validation Authentication Service

No results are returned.

Action

If you request Check Enrollment and authorization services separately, add the required payer authentication values (CAVV and ECI) to your authorization request. If you request the Check Enrollment and authorization services together, the process described above occurs automatically.

Test Case 2.4: Unavailable Frictionless Authentication

Authentication is unavailable at the time of transaction.

Card Numbers

Card Type	Test Card	Number
	3-D Secure 2.1.0	3-D Secure 2.2.0
American Express Card Type = 003	34000 00 0000 1031	34000 00 0000 2922
Cartes Bancaires Mastercard Card Type = 036	520000 00 0000 3035	520000 00 0000 4306
Cartes Bancaires Visa Card Type = 036	400000 00 0000 3030	400000 00 0000 4160
China UnionPay Card Type = 062	620001 00 0040 0030	—
Diners Club Card Type = 005	601100 00 0000 1036	—
Discover Card Type = 004	601100 00 0000 1036	—
Elo Card Type = 054	650529 00 0000 1085	—
JCB J/Secure Card Type = 007	333700 00 0000 0735	333800 00 0000 0221
Mastercard Card Type = 002	520000 00 0000 1039	520000 00 0000 2268
Visa Card Type = 001	400000 00 0000 1034	400000 00 0000 2313

Results for the Check Enrollment Service

Reason code = `100`
`ics_pa_enroll service was successful.`
VERes enrolled = `Y`
PARes status = `U`
AVV = <AVV value> (Mastercard only)

CAAV = <No value provided>

E-Commerce Indicator (ECI) Values

This table lists the expected ECI raw value and its respective string value. These values indicate whether the payer was authenticated by the card network. These values should be passed under this test condition when a transaction is submitted for payment authorization.

Network	ECI Raw Value	ECI String Value
American Express	07	internet
Cartes Bancaires Mastercard	00	internet
Cartes Bancaires Visa	07	internet or vbv_failure
China UnionPay	07	up3ds_failure
Diners Club	07	internet
Discover	07	internet
Elo	07	internet
ITMX	07	lss_failure
JCB J/Secure	07	internet
Mastercard	00	internet
Visa	07	internet or vbv_failure

Results for the Validation Authentication Service

No results are returned.

Action

Submit your authorization request. No liability shift.

Test Case 2.5: Rejected Frictionless Authentication

Cardholder authentication was rejected without a challenge by the issuer.

Card Numbers

Card Type	Test Card	Number
	3-D Secure 2.1.0	3-D Secure 2.2.0
American Express Card Type = 003	34000 00 0000 1049	34000 00 0000 2062

Card Type	Test Card	Number
	3-D Secure 2.1.0	3-D Secure 2.2.0
Cartes Bancaires Mastercard Card Type = 036	520000 00 0000 3043	520000 00 0000 4405
Cartes Bancaires Visa Card Type = 036	400000 00 0000 3048	400000 00 0000 4517
China UnionPay Card Type = 062	620001 00 0030 0040	—
Diners Club Card Type = 005	601100 00 0000 1044	—
Discover Card Type = 004	601100 00 0000 1044	—
Elo Card Type = 054	650529 00 0000 1143	—
JCB J/Secure Card Type = 007	333700 00 0000 0321	333800 00 0000 0734
Mastercard Card Type = 002	520000 00 0000 1047	520000 00 0000 2185
Visa Card Type = 001	400000 00 0000 1042	400000 00 0000 2537

Results for the Check Enrollment Service

Reason code = `476`

- User failed authentication.
- Payer cannot be authenticated.

VERes enrolled = `Y`
PARes status = `R`
AVV = <AVV value> (American Express only)
CAAV = <No value provided>

E-Commerce Indicator (ECI) Values

This table lists the expected ECI raw value and its respective string value. These values indicate whether the payer was authenticated by the card network. These values should be passed under this test condition when a transaction is submitted for payment authorization.

Network	ECI Raw Value	ECI String Value
American Express	07	internet
Cartes Bancaires Mastercard	00	internet
Cartes Bancaires Visa	07	internet or vbv_failure
China UnionPay	07	up3ds_failure
Diners Club	07	internet
Discover	07	internet
Elo	07	internet
ITMX	07	lss_failure
JCB J/Secure	07	internet
Mastercard	00	internet
Visa	07	internet or vbv_failure

Results for the Validation Authentication Service

No results are returned.

Action

You are not permitted to submit this transaction for authorization. Instead ask the customer for another form of payment.

Test Case 2.6: Authentication not Available on Lookup

A system error prevented authentication on Lookup.

Card Numbers

Card Type	Test Card	Number
	3-D Secure 2.1.0	3-D Secure 2.2.0
American Express Card Type = 003	34000 00 0000 1056	34000 00 0000 2468

Card Type	Test Card	Number
	3-D Secure 2.1.0	3-D Secure 2.2.0
Cartes Bancaires Mastercard Card Type = 036	520000 00 0000 3050	520000 00 0000 4090
Cartes Bancaires Visa Card Type = 036	400000 00 0000 3055	400000 00 0000 4285
China UnionPay Card Type = 062	620001 00 0060 0050	—
Diners Club Card Type = 005	601100 00 0000 1051	—
Discover Card Type = 004	601100 00 0000 1051	—
Elo Card Type = 054	650529 00 0000 1150	—
JCB J/Secure Card Type = 007	333700 00 0000 6765	333800 00 0000 0940
Mastercard Card Type = 002	520000 00 0000 1054	520000 00 0000 2409
Visa Card Type = 001	400000 00 0000 1059	400000 00 0000 2990

Results for the Check Enrollment Service

Reason code = `100`
`ics_pa_enroll service was successful.`
VERes enrolled = `U`

E-Commerce Indicator (ECI) Values

This table lists the expected ECI raw value and its respective string value. These values indicate whether the payer was authenticated by the card network. These values should be passed under this test condition when a transaction is submitted for payment authorization.

Network	ECI Raw Value	ECI String Value
American Express	07	internet
Cartes Bancaires Mastercard	00	internet
Cartes Bancaires Visa	07	internet or vbv_failure
China UnionPay	07	up3ds_failure
Diners Club	07	internet
Discover	07	internet
Elo	07	internet
ITMX	07	lss_failure
JCB J/Secure	07	internet
Mastercard	00	internet
Visa	07	internet or vbv_failure

Results for the Validation Authentication Service

No results are returned.

Action

Submit your authorization request. No liability shift.

Test Case 2.7: Enrollment Check Error

An error occurred while attempting to authenticate the cardholder.

Card Numbers

Card Type	Test Card	Number
	3-D Secure 2.1.0	3-D Secure 2.2.0
American Express Card Type = 003	34000 00 0000 1064	34000 00 0000 2732
Cartes Bancaires Mastercard Card Type = 036	520000 00 0000 3068	520000 00 0000 4058
Cartes Bancaires Visa Card Type = 036	400000 00 0000 3063	400000 00 0000 4194

Card Type	Test Card	Number
	3-D Secure 2.1.0	3-D Secure 2.2.0
China UnionPay Card Type = 062	620001 00 0050 0060	—
Diners Club Card Type = 005	601100 00 0000 1069	—
Discover Card Type = 004	601100 00 0000 1069	—
Elo Card Type = 054	650529 00 0000 1176	—
JCB J/Secure Card Type = 007	333700 00 0000 0016	333800 00 0000 0650
Mastercard Card Type = 002	520000 00 0000 1062	520000 00 0000 2037
Visa Card Type = 001	400000 00 0000 1067	400000 00 0000 2446

Results for the Check Enrollment Service

Reason code = `100`
`ics_pa_enroll` service was successful.
VERes enrolled = `U`

E-Commerce Indicator (ECI) Values

This table lists the expected ECI raw value and its respective string value. These values indicate whether the payer was authenticated by the card network. These values should be passed under this test condition when a transaction is submitted for payment authorization.

Network	ECI Raw Value	ECI String Value
American Express	07	internet
Cartes Bancaires Mastercard	00	internet
Cartes Bancaires Visa	07	internet or vbv_failure
China UnionPay	07	up3ds_failure

Network	ECI Raw Value	ECI String Value
Diners Club	07	internet
Discover	07	internet
Elo	07	internet
ITMX	07	lss_failure
JCB J/Secure	07	internet
Mastercard	00	internet
Visa	07	internet or vbv_failure

Results for the Validation Authentication Service

No results are returned.
While Mastercard would normally return the directory server transaction ID, in this test case it is not returned.

Action

Proceed with the authorization request, and contact your support representative to resolve the issue. No liability shift. If you requested payer authentication and authorization together, the authorization is processed automatically.

Test Case 2.8: Time-Out

Timeout occurred while checking enrollment, causing an error on the transaction.

Card Numbers

Card Type	Test Card	Number
	3-D Secure 2.1.0	3-D Secure 2.2.0
American Express Card Type = 003	34000 00 0000 1072	34000 00 0000 2047
Cartes Bancaires Mastercard Card Type = 036	520000 00 0000 3076	520000 00 0000 4694
Cartes Bancaires Visa Card Type = 036	400000 00 0000 3071	400000 00 0000 4277
China UnionPay Card Type = 062	620001 00 0090 0070	—

Card Type	Test Card	Number
	3-D Secure 2.1.0	3-D Secure 2.2.0
Diners Club Card Type = 005	601100 00 0000 1077	—
Discover Card Type = 004	601100 00 0000 1077	—
Elo Card Type = 054	650529 00 0000 1192	—
JCB J/Secure Card Type = 007	333700 00 0000 0081	333800 00 0000 0577
Mastercard Card Type = 002	520000 00 0000 1070	520000 00 0000 2326
Visa Card Type = 001	400000 00 0000 1075	400000 00 0000 2354

Results for the Check Enrollment Service

Reason code = `100`
VERes enrolled = `U`

E-Commerce Indicator (ECI) Values

This table lists the expected ECI raw value and its respective string value from this transaction. These values indicate whether the payer was authenticated by the card network. These values should be passed under this test condition when a transaction is submitted for payment authorization.

Network	ECI Raw Value	ECI String Value
American Express	07	internet
Cartes Bancaires Mastercard	00	internet
Cartes Bancaires Visa	07	internet or vbv_failure
China UnionPay	07	up3ds_failure
Diners Club	07	internet
Discover	07	internet

Network	ECI Raw Value	ECI String Value
Elo	07	internet
ITMX	07	lss_failure
JCB J/Secure	07	internet
Mastercard	00	internet
Visa	07	internet or vbv_failure

Results for the Validation Authentication Service

No results are returned.

Action

After 10-12 seconds, proceed with the authorization request. No liability shift.

Test Case 2.9: Bypassed Authentication

The challenge requested by the issuer was bypassed for this transaction.

Card Numbers

Card Type	Test Card	Number
	3-D Secure 2.1.0	3-D Secure 2.2.0
American Express Card Type = 003	34000 00 0000 1080	34000 00 0000 2948
Cartes Bancaires Mastercard Card Type = 036	520000 00 0000 3084	520000 00 0000 4991
Cartes Bancaires Visa Card Type = 036	400000 00 0000 3089	400000 00 0000 4400
China UnionPay Card Type = 062	620001 00 0080 0080	—
Diners Club Card Type = 005	601100 00 0000 1085	—
Discover Card Type = 004	601100 00 0000 1085	—

Card Type	Test Card	Number
	3-D Secure 2.1.0	3-D Secure 2.2.0
Elo Card Type = 054	650529 00 0000 1226	—
JCB J/Secure Card Type = 007	333700 00 0000 0537	333800 00 0000 0122
Mastercard Card Type = 002	520000 00 0000 1088	520000 00 0000 2508
Visa Card Type = 001	400000 00 0000 1083	400000 00 0000 2560

Results for the Check Enrollment Service

Reason code = `100`
`ics_pa_enroll service was successful.`
VERes enrolled = `B`
XID = <XID value>

E-Commerce Indicator (ECI) Values

This table lists the expected ECI raw value and its respective string value from this transaction. These values indicate whether the payer was authenticated by the card network. These values should be passed under this test condition when a transaction is submitted for payment authorization.

Network	ECI Raw Value	ECI String Value
American Express	07	internet
Cartes Bancaires Mastercard	00	internet
Cartes Bancaires Visa	07	internet
China UnionPay	07	up3ds_failure
Diners Club	07	internet
Discover	07	internet
Elo	07	internet
ITMX	07	lss_failure
JCB J/Secure	07	internet
Mastercard	00	internet

Network	ECI Raw Value	ECI String Value
Visa	07	internet

Results for the Validation Authentication Service

No results are returned.

Action

Submit your authorization request. No liability shift.

Test Case 2.10a: Successful Step-Up Authentication

Successful step up (or challenge) authentication transaction.

Card Numbers

Card Type	Test Card	Number
	3-D Secure 2.1.0	3-D Secure 2.2.0
American Express Card Type = 003	34000 00 0000 1098	34000 00 0000 2534
Cartes Bancaires Mastercard Card Type = 036	520000 00 0000 3092	520000 00 0000 4074
Cartes Bancaires Visa Card Type = 036	400000 00 0000 3139	400000 00 0000 4855
China UnionPay Card Type = 062	620001 99 9980 0019	—
Diners Club Card Type = 005	601100 00 0000 1093	—
Discover Card Type = 004	601100 00 0000 1093	—
Elo Card Type = 054	650529 00 0000 1234	—
JCB J/Secure Card Type = 007	333700 00 0020 0004	333800 00 0000 0569

Card Type	Test Card	Number
	3-D Secure 2.1.0	3-D Secure 2.2.0
Mastercard Card Type = 002	520000 00 0000 1096	520000 00 0000 2151
Visa Card Type = 001	400000 00 0000 1091	400000 00 0000 2503

Results for the Check Enrollment Service

Reason code = 475
The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.
VERes enrolled = `Y`
PARes status = `C`
XID = <XID value>

Results for the Validation Authentication Service

Reason code = `100`
`ics_pa_validate service was successful.`
PARes status = `Y`
XID = <XID value>
CAVV = <CAVV value>

E-Commerce Indicator (ECI) Values

This table lists the expected ECI raw value and its respective string value from this transaction. These values indicate whether the payer was authenticated by the card network. These values should be passed under this test condition when a transaction is submitted for payment authorization.

Network	ECI Raw Value	ECI String Value
American Express	05	aesk
Cartes Bancaires Mastercard	02	spa
Cartes Bancaires Visa	05	vbv
China UnionPay	05	up3ds
Diners Club	05	pb
Discover	05	dipb
Elo	05	cs
ITMX	05	lss

Network	ECI Raw Value	ECI String Value
JCB J/Secure	05	js
Mastercard	02	spa
Visa	05	vbv

Action

If you request Validate Authentication and authorization services separately, add the required payer authentication values to your authorization request. If you request the Validate Authentication and authorization services together, the process described above occurs automatically. Merchant should include the CAVV and ECI vlues in the authorization message.

Test Case 2.11a: Unsuccessful Step-Up Authentication

Step up (challenge) authentication transaction where the cardholder challenge failed.

Card Numbers

Card Type	Test Card	Number
	3-D Secure 2.1.0	3-D Secure 2.2.0
American Express Card Type = 003	34000 00 0000 1106	34000 00 0000 2237
Cartes Bancaires Mastercard Card Type = 036	520000 00 0000 3100	520000 00 0000 4041
Cartes Bancaires Visa Card Type = 036	400000 00 0000 3097	400000 00 0000 4293
China UnionPay Card Type = 062	620001 99 9970 0029	—
Diners Club Card Type = 005	601100 00 0000 1101	—
Discover Card Type = 004	601100 00 0000 1101	—
Elo Card Type = 054	650529 00 0000 1275	—

Card Type	Test Card	Number
	3-D Secure 2.1.0	3-D Secure 2.2.0
JCB J/Secure Card Type = 007	333700 00 0020 0087	333800 00 0000 0874
Mastercard Card Type = 002	520000 00 0000 1104	520000 00 0000 2490
Visa Card Type = 001	400000 00 0000 1109	400000 00 0000 2370

Results for the Check Enrollment Service

Reason code = 475The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.
VERes enrolled = Y
PARes status = C
PAReq = <PAReq value>
ACS URL = <URL value>

Results for the Validation Authentication Service

Reason code = 476

- User failed authentication.
- Payer cannot be authenticated.

PARes status = N
XID = <XID value> (American Express only)

E-Commerce Indicator (ECI) Values

This table lists the expected ECI raw value and its respective string value from this transaction. These values indicate whether the payer was authenticated by the card network. These values should be passed under this test condition when a transaction is submitted for payment authorization.

Network	ECI Raw Value	ECI String Value
American Express	07	internet
Cartes Bancaires Mastercard	00	internet
Cartes Bancaires Visa	07	internet or vbv_failure
China UnionPay	07	up3ds_failure

Network	ECI Raw Value	ECI String Value
Diners Club	07	internet
Discover	07	internet
Elo	07	internet
ITMX	07	Iss_failure
JCB J/Secure	07	internet
Mastercard	00	internet
Visa	07	internet or vbv_failure

Action

You are not permitted to submit this transaction for authorization. Instead ask the customer for another form of payment.

Test Case 2.12a: Unavailable Step-Up Authentication

Step up authentication is unavailable.

Card Numbers

Card Type	Test Card	Number
	3-D Secure 2.1.0	3-D Secure 2.2.0
American Express Card Type = 003	34000 00 0000 1114	34000 00 0000 2484
Cartes Bancaires Mastercard Card Type = 036	520000 00 0000 3118	520000 00 0000 4124
Cartes Bancaires Visa Card Type = 036	400000 00 0000 3105	400000 00 0000 4640
China UnionPay Card Type = 062	620001 99 9960 0039	—
Diners Club Card Type = 005	601100 00 0000 1119	—
Discover Card Type = 004	601100 00 0000 1119	—

Card Type	Test Card	Number
	3-D Secure 2.1.0	3-D Secure 2.2.0
Elo Card Type = 054	650529 00 0000 1283	—
JCB J/Secure Card Type = 007	333700 00 0020 0079	333800 00 0000 0981
Mastercard Card Type = 002	520000 00 0000 1112	520000 00 0000 2664
Visa Card Type = 001	400000 00 0000 1117	400000 00 0000 2420

Results for the Check Enrollment Service

Reason code = `475`
The cardholder is enrolled in payer authentication. Authenticate before proceeding with authorization.
VERes enrolled = `Y`
PARes Status = `C`

Results for the Validation Authentication Service

Reason code = `100 ics_pa_validate service was successful.`
PARes status = `U`
XID = <XID value>

E-Commerce Indicator (ECI) Values

This table lists the expected ECI raw value and its respective string value from this transaction. These values indicate whether the payer was authenticated by the card network. These values should be passed under this test condition when a transaction is submitted for payment authorization.

Network	ECI Raw Value	ECI String Value
American Express	07	internet
Cartes Bancaires Mastercard	00	internet
Cartes Bancaires Visa	07	internet or vbv_failure
China UnionPay	07	up3ds_failure
Diners Club	07	internet

Network	ECI Raw Value	ECI String Value
Discover	07	internet
Elo	07	internet
ITMX	07	lss_failure
JCB J/Secure	07	internet
Mastercard	00	internet
Visa	07	internet or vbv_failure

Action

Merchant can retry authentication or process without the liability shift.

Test Case 2.14: Require Method URL

Ensures that the merchant has allowed sufficient time (10 seconds) for the issuer to complete their device data collection.

Card Numbers

The Method URL test runs before the authentication request to check how well your system implements device data collection. The enrollment check of the card account should not start until after the device data collection response is received. This test helps to ensure that there is enough time to collect the device data and to transmit it. This test attempts to collect the nine-digit BIN of the card number and checks that the delay between the DDC request and the response is at least seven seconds long. Test failure occurs when less than nine digits of the BIN is collected or the delay between the DDC request and response is too short in duration.
Do not run this test when your system does not collect device data. When device data is not collected, an older version of the EMV 3-D Secure protocol is automatically used and the transaction is automatically assessed as a higher risk.

Card Type	Test Card Number
Visa Card Type = 001	400010 00 0000 0000

Results for the Check Enrollment Service

VERes enrolled = `Y`
PARes status = `Y`
CAVV = <CAVV value>
ECI value = `07`

ECI/Collection Indicator Values

The following table lists the expected ECI or Collection Indicator values for each network.

Network	E-Commerce Indicator (ECI)
Visa	07

Action

If your device data collection method implements correctly and EMV 3-D Secure Method processing occurs, the test transaction produces a Frictionless Success result. A failure is indicated when PARes status = `C`. With the failure, a warning message displays to explain the cause of the test failure.

Payer Authentication Exemption Test Cases

These test cases cover payer authentication scenarios that can occur outside of typical testing. These special use cases might require including additional API fields to accommodate different data that is necessary for that test.

Test Case 1a: Initial/First Recurring Transaction: Fixed Amount

Merchant initiates a *Requester Initiated Payments* (3RI) recurring transaction of a fixed amount for a specified number of transactions or with no set number of transactions such as occurs with subscription purchases.

Card Type	Test Card Number
Mastercard Card Type = 002	520000 00 0000 2805

Required Fields for Check Enrollment

Message category = `01`
Device channel = `APP` (01), `BROWSER` (02)
Three RI Indicator = `01`
Challenge code = `03`
Authentication code = `02`
Purchase date = <yyyyMMDDHHMMSS>
Recurring frequency = <1 to 31>
Recurring end = <yyyyMMDD>

Results for the Check Enrollment Service

Reason code = `100`
VERes enrolled = `Y`
PARes status = `C`
CAVV = (No value provided)
ECI = `00`

Results for the Validation Authentication Service

Reason code = `100` **ics_pa_validate** service was successful.
PARes status = `Y`
CAVV = <CAVV>
ECI = `07`

Card Network and Version Specifications

Visa Secure 2.1 does not support this use case. Visa Secure 2.2 test cards are in development.
For Mastercard Identity Check 2.1, 3RI is not supported for Payment Authentication (PA). This means only the initial transaction is supported for Recurring Payments.
If you attempt to run a Device Channel of 3RI within Mastercard Identity Check 2.1, you receive a transStatusReason=21 (3RI Transaction not Supported) and a transaction status of “U” rather than “Y.”
In EMV 3-D Secure 2.2, Mastercard has allocated a new ECI value, ECI 07, for 3RI transactions. This is present on a Mastercard response message for this particular 3RI scenario. For EMV 3-D Secure 2.1, Mastercard will continue to use ECI 02.

Test Case 2a: Card Authentication Failed

The following test case scenarios test various Trans Status Reasons (failed, suspected fraud, and similar instances). When **PAResStatus** = N, the **CardholderInfo** field can be returned by the card issuer. When this cardholder information is returned, you must display this information within your checkout experience.

Card Type	Test Card Number
Visa Card Type = 002	400000 00 0000 2040

Results for the Check Enrollment Service

Reason code = `100`
VERes enrolled = `Y`
PARes status = `N`
CAVV = (No value provided)
Cardholder Info = <cardholder information>
ECI = `07`
Reason code = `01`

Test Case 2b: Suspected Fraud

This test case scenario checks for suspected fraud.

Card Type	Test Card Number
Visa Card Type = 001	400000 00 0000 2149

Results for the Check Enrollment Service

Reason code = `100`
VERes enrolled = `Y`
PARes status = `U`
CAVV = (No value provided)
ECI = `07`
Reason code = `11`

Test Case 2c: Cardholder Not Enrolled in Service

This test case scenario verifies whether the cardholder is enrolled in the service.

Card Type	Test Card Number
Visa Card Type = 001	400000 00 0000 2164

Results for the Check Enrollment Service

Reason code = `476`
VERes enrolled = `Y`
PARes status = `R`
CAVV = (No value provided)
ECI = `07`
Reason code = `13`

Test Case 2d: Transaction Timed Out at the ACS

This test case scenario verifies whether a transaction will time out at the Access Control Server (ACS). This test case is valid for both payer authentication and non-payer authentication transactions.

Card Type	Test Card Number
Visa Card Type = 001	400000 00 0000 2172

Results for the Check Enrollment Service

Reason code = `100`
VERes enrolled = `Y`
PARes status = `U`
CAVV = (No value provided)
ECI = `07`
Reason code = `14`

Test Case 2e: Non-Payment Transaction Not Supported

This test case scenario checks whether a non-payment transaction can occur. This test case is valid for both payer authentication and non-payer authentication transactions.

Card Type	Test Card Number
Visa Card Type = 001	400000 00 0000 2230

Results for the Check Enrollment Service

Reason code = `100`
VERes enrolled = `Y`
PARes status = `U`
CAVV = (No value provided)
ECI = `07`
Reason code = `20`

Test Case 2f: 3RI Transaction Not Supported

This test case scenario verifies whether the merchant can initiate a recurring 3RI transaction, such as with subscriptions.

Card Type	Test Card Number
Visa Card Type = 001	400000 00 0000 2248

Required Fields for Check Enrollment

Message category = `02`
Device channel = `3RI (03)`
Three RI Indicator = `01`

Results for the Check Enrollment Service

Reason code = 100
VERes enrolled = `Y`
PARes status = `U`
CAVV = (No value provided)
ECI = `07`
Reason code = `21`

Test Case 3a: Transaction Risk Analysis Exemption: Low Value: Mastercard EMV 3-D Secure 2.1 and 2.2

You have performed a proprietary risk assessment and are requesting a transaction risk analysis, low risk, or low value exemption based on fraud thresholds established with the

network. Be sure to use the correct test card number for your version of EMV 3-D Secure. The PARes Status will differ between the EMV 3-D Secure versions.

Card Type	Test Card Number
Mastercard Card Type = 002	(version 2.1.0) 5200 00 00 0000 1161 (version 2.2.0) 5200 00 00 0000 2052

Required Fields for Check Enrollment

Challenge code = `05`

Results for the Check Enrollment Service

Reason code = `100`
Version 2.1.0
VERes enrolled = `Y`
PARes status = `N`
CAVV = <CAVV value>
ECI = `06`
Reason code = `81`
For Mastercard Identity Check, the ChallengeIndicator should be passed as `05`.
Version 2.2.0
VERes enrolled = `Y`
PARes status = `I`
CAVV = <CAVV value>
ECI = `06`

Action

Proceed to Authorization.
You can also request the transaction risk analysis (TRA) exemption directly during authorization depending on the region and your agreements with your acquirer and the networks.

Test Case 3b: Transaction Risk Analysis: Low Value: Visa

The merchant has performed a proprietary risk assessment and is requesting a transaction risk analysis, low risk, or low value exemption based on fraud thresholds established with the network.

Card Type	Test Card Number
Visa Card Type = 001	400000 00 0000 2024

Required Fields for Check Enrollment

Challenge code = `05` (no challenge requested)

Results for the Check Enrollment Service

Reason code = `100`
VERes enrolled = `Y`
PARes status = `I`
CAVV = <CAVV value>
ECI = `07`

Action

Proceed to Authorization.
You can also request the TRA exemption directly during authorization depending on the region and your agreements with your acquirer and the networks.

Test Case 3c: Transaction Risk Analysis: Low Value: Discover

The merchant has performed a proprietary risk assessment and is requesting a transaction risk analysis, low risk, or low value exemption based on fraud thresholds established with the network.

Card Type	Test Card Number
Discover Card Type = 004	601100 00 0000 1002

Required Fields for Check Enrollment

Challenge code = `04` (challenge requested)

Results for the Check Enrollment Service

Reason code = 100
VERes enrolled = `Y`
PARes status = `Y`
CAVV = <CAVV value>

Action

Proceed to Authorization.
You can also request the TRA exemption directly during authorization depending on the region and your agreements with your acquirer and the networks.

Test Case 3d: Acquirer Transaction Risk Analysis: Cartes Bancaires

Merchant has performed a proprietary risk assessment and requests a transaction risk analysis, low risk, or low value exemption based on fraud thresholds established with the network.

Card Type	Test Card Number
Cartes Bancaires Visa Card Type = 036	400000 00 0000 3006
Cartes Bancaires Mastercard Card Type =036	520000 00 0000 3001

Required Fields for Check Enrollment

Challenge code = `05` (no challenge requested)

Results for the Check Enrollment Service

Reason code = `100`
VERes enrolled = `Y`
PARes status = `Y`
CAVV = <CAVV value> (The CAVV value is not returned during testing but can be returned in production based on issuer rules surrounding co-branding with Visa or Mastercard BINs.)
ECI = (no value provided)

Action

Proceed to Authorization.
You can also request the TRA exemption directly during authorization depending on the region and your agreements with your acquirer and the networks.

Test Case 4a: Trusted Beneficiary Prompt for Trustlist

You have a successful traditional step-up (challenge) authentication transaction with a prompt for the Trustlist and an accepted exemption result.

Card Type	Test Card Number
Visa Card Type = 001	400000 00 0000 2008
Mastercard Card Type = 002	520000 00 0000 2003

Required Fields for Check Enrollment

Challenge code = 09 (challenge requested)

Results for the Check Enrollment Service

With the Cardinal Cruise API, the response will also include a StepUpUrl.
VERes enrolled = `Y`
PARes status = `C`
CAVV = (No value provided)

ECI =

- Visa = `07`
- Mastercard = `00`

Results for the Authenticate Response

PARes status = `Y`
CAVV = <CAVV value>
ECI =

- Visa = `05`
- Mastercard = `02`

WhiteListStatus = <WhiteListStatus value>
WhiteListStatusSource = <WhiteListStatusSource value>

Action

You should append the CAVV and ECI values to the authorization message.

Test Case 4b: Utilize Trusted Beneficiary Exemption

There is a successful frictionless authentication transaction with a pre-whitelisted indication and an accepted exemption result.

Card Type	Test Card Number
Visa Card Type = 001	400000 00 0000 2016
Mastercard Card Type = 002	520000 00 0000 2011

Required Fields for Check Enrollment

Challenge code = `08` (No challenge requested)

Results for the Check Enrollment Service

Reason code = 100
PARes status = `Y`
CAVV = <CAVV value>
ECI =

- Visa = `05`
- Mastercard = `02`

WhiteListStatus = <WhiteListStatus value>
WhiteListStatusSource = <WhiteListStatusSource value>
ThreeDSVersion = <ThreeDSVersion value>

Action

Append the CAVV and ECI values to the authorization message.

Test Case 5a-1: Identity Check Insights (ScoreRequest = N)

This is a Mastercard Data Only authentication request.

Card Type	Test Card Number
Mastercard Card Type = 002	520000 00 0000 1005

Required Fields for Check Enrollment

MessageCategory = `80`

Results for the Check Enrollment Service

Reason code = 100
PAResStatus = U
CAVV = <CAVV value>
ECI = `04`
StatusReason = `80`
ThreeDSVersion = <ThreeDSVersion value>

Reason code = `100`

Action

Append the ECI and DS Transaction ID values to the authorization message.

Test Case 5a-2: Identity Check Insights (ScoreRequest = Y)

This is a Mastercard Data Only authentication request.

Card Type	Test Card Number
Mastercard Card Type = 002	520000 00 0000 1005

Required Fields for Check Enrollment

Message Category = `80`
Optional Fields for Check Enrollment
Score Request = Y
Merchant Reason Code = `A`

Results for the Check Enrollment Service

Reason code = 100
PAResStatus = `U`

CAVV = <CAVV value>
ECI = `04`
StatusReason = `80`
ThreeDSVersion = <ThreeDSVersion value>
Optional Results for the Check Enrollment Service (if ScoreRequest = `Y`)
IDCI_Score = `9`
IDCI_Decisions = `not low risk`
IDCI_ReasonCode1 = `A`
IDCI_ReasonCode2 = `GG`

Results for the Authentication Result

Reason code = 100

Action

Append the ECI and DS Transaction ID values to the authorization message.

Payer Authentication Use Cases

These use case examples show a request and response that occur with the Setup, Check Enrollment, and Validate Authentication services for Payer Authentication. In certain circumstances, some payment card companies and some countries require that additional information be collected when authenticating the customer. These circumstances are noted. Each use case includes an example and lists of the required and optional API fields for the use case.

Use Case: Setting Up Payer Authentication

Running the Setup service identifies the customer's bank and prepares for collecting data about the device that the customer is using to place the order.

Card-Specific Requirements

Some payment cards require specific information to be collected during a transaction.

card_cardType	Required when the card type is Cartes Bancaires, JCB, or UPI.

Country-Specific Requirements

These fields are required for transactions in specific countries.

billTo_state	Required for transactions in the US, Canada, and Mainland China.
billTo_postalCode	Required when the **billTo_country** field value is `US` or `CA`.

Endpoint

Set the `payerAuthSetupService_run` field to `true`.

Send the request to:
Production: `https://ics2ws.ic3.com/commerce/1.x/transactionProcessor`
Test: `https://ics2wstest.ic3.com/commerce/1.x/transactionProcessor`

Required Fields for Setting Up Payer Authentication

These fields are the minimum fields required when you request the Payer Authentication Setup service. Other fields that can be used to collect additional information during a transaction are listed in the optional fields section. Under certain circumstances, a field that normally is optional might be required. The circumstance that makes an optional field required is noted.

billTo_postalCode	Required when the **billTo_country** field value is U.S. or CA.
billTo_state	Required for U.S., Canada, and Mainland China.
card_accountNumber	
card_cardType	Required when the card type is Cartes Bancaires, JCB, or UPI.
card_expirationMonth	Required when **card_accountNumber** is included.
card_expirationYear	Required when **card_accountNumber** is included.
payerAuthSetupService_run	

Related information
- API Field Reference for the Simple Order API

Optional Fields for Setting Up Payer Authentication

These fields are optional for setting up a Payer Authentication transaction. Under certain circumstances, a field might appear as both an optional field and a required field.

billTo_city

billTo_country

billTo_email

billTo_firstName

billTo_lastName

billTo_postalCode

billTo_street1

card_cardType

encryptedPayment_data

encryptedPayment_descriptor

merchantID

merchantReferenceCode

recurringSubscriptionInfo_subscriptionID

tokenSource_transientToken

Simple Order Example: Setting Up with Payer Authentication

Request

```
card_accountNumber=XXXXXXXXXXXXXXXX
card_expirationMonth=12
card_expirationYear=2030
merchantID=patest
merchantReferenceCode=0001
payerAuthSetupService_run=true
```

Response to Successful Request

```
decision=ACCEPT
merchantReferenceCode=0001
payerAuththSetupReply_deviceDataCollectionURL=https://centinelapistag.cardinalcommerce.com/V1 /
Cruise/Collect
payerAuthSetupReply_reasonCode=100
payerAuthSetupReply_referenceID=f13fe5e0-9b47-4ea1-a03a-ec360f4d0f9f
payerAuthSetupReply_accessToken=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiI1MDc4OTI0Mi0z
 YmEzLTRhZTItYWQwOS1kZjZkODk2NWQ5MjciLCJpYXQiOjE1OTgyOTk1MjQsImlzcyI6IjVkZDgzYmYwMGU0MjNkMTQ
 5OGRjYmFjYSIsImV4cCI6MTU5ODMwMzEyNCwiT3JnVW5pdElkIjoiNTVlZjNmMTBmNzIzYWE0MzFjOTliNWViIiwiUGF
 5bG9hZCI6eyJBQ1NVcmwiOiJodHRwczovLzBtZXJjaGFudGFjc3N0YWcuY2FyZGluYWxjb21tZXJjZS5jb20vTWVyY2
 hhbnRBQ1NXZWIvY3JlcS5qc3AiLCJQYXlsb2FkIjoiZXlKdFpYTnpZV2RsVkhsd1pTSTZJa05TWlhFaUxDSnRaWE56W
 VdkbFZtVnljMmx2YmlJNklqSXVNaTR3SWl3aWRHaHlaV1ZFVTFObGNuWmxjbFJ5WVc1elNVUWlPaUkzTkRNeVlUWXdN
 QzA0TXpNMkxUUm1PRGN0WVdKbE9TMDJObVkzTkRFM01EaGhNV1FpTENKaFkzTlVjbUZ1YzBsRUlqb2lPR0U1TkRkaU9E
 TXRNRFJpTkMwMFltVmlMV0V5WWpNdFpHTmpNV0UxWmprMFlURXlJaXdpWTJoaGJHeGxibWRsVjJsdVpHOTNVMmw
 2WlNJNklqQXlJbjAiLCJUcmFuc2FjdGlvbklkIjoiVEQ1b1MwbzFGQzY1cWF2MHhzeDAifSwiT2JqZWN0aWZ5UGF5bG9h
 ZCI6dHJ1ZSwiUmV0dXJuVXJsIjoiaHR0cHM6Ly9leGFtcGxlLmNvbS9zdGVwLXVwLXJldHVybi11cmwuanNwIn0.8w
 Z8XhLgOIIRvgEUugvYrRAi-efavZTNM0tWInYLTfE
payerAuthSetupReply_reasonCode=100
requestID=5982993692286989203011
requestToken=AxjzbwSTRFa3h+A4wXZDABEBURwlqraRpAy7gDthk0kyro9JLIYA8AAA2wK2
```

Use Case: Setting Up Payer Authentication with Google Pay

Running the Setup service identifies the customer's bank and prepares for collecting data about the device that the customer is using to place the order. This use case demonstrates how the service works using a digital payment method like Google Pay.

Card-Specific Requirements

Some payment cards require specific information to be collected during a transaction.

card_cardType	Required when the card type is Cartes Bancaires, JCB, or UPI.

Country-Specific Requirements

These fields are required for transactions in specific countries.

billTo_state	Required for transactions in the US, Canada, and Mainland China.
billTo_postalCode	Required when the **billTo_country** field value is `US` or `CA`.

Endpoint

Set the `payerAuthSetupService_run` field to `true`.
Send the request to:
Production: `https://ics2ws.ic3.com/commerce/1.x/transactionProcessor`
Test: `https://ics2wstest.ic3.com/commerce/1.x/transactionProcessor`

Required Fields for Setting Up Payer Authentication

These fields are the minimum fields required when you request the Payer Authentication Setup service. Other fields that can be used to collect additional information during a transaction are listed in the optional fields section. Under certain circumstances, a field that normally is optional might be required. The circumstance that makes an optional field required is noted.

billTo_postalCode	Required when the **billTo_country** field value is U.S. or CA.
billTo_state	Required for U.S., Canada, and Mainland China.
card_accountNumber	
card_cardType	Required when the card type is Cartes Bancaires, JCB, or UPI.
card_expirationMonth	Required when **card_accountNumber** is included.
card_expirationYear	Required when **card_accountNumber** is included.
payerAuthSetupService_run	

Related information
- API Field Reference for the Simple Order API

Optional Fields for Setting Up Payer Authentication

These fields are optional for setting up a Payer Authentication transaction. Under certain circumstances, a field might appear as both an optional field and a required field.

billTo_city

billTo_country

billTo_email

billTo_firstName

billTo_lastName

billTo_postalCode

billTo_street1

card_cardType

encryptedPayment_data

encryptedPayment_descriptor

merchantID

merchantReferenceCode

recurringSubscriptionInfo_subscriptionID

tokenSource_transientToken

Simple Order Example: Setting Up Payer Authentication Using Google Pay

This is an example of an Payer Authentication Setup request and response using Google Pay as the digital payment option. The data in this example is for illustrative purposes.

Request


```
merchantID=patest
merchantReferenceCode=0001
payerAuthSetupService_run=true
payment_solution=012
encryptedPayment_data=eyJzaWduYXR1cmUiOiJNRVlDSVFDMnNQcmduTmQ1cUY5N0hIMU1uWXRGV1Q
    xSFlWbnFrek93NU4ySXNldUZBZ0loQU90dWF1anc5L3lXWm5BYUlxUlVySktLWFF3M1I0Y3Mr
    aWFlWHBJYlNPT2wiLCJwcm90b2NvbFZlcnNpb24iOiJFQ3YxIiwic2lnbmVkTWVzc2FnZSI6I
    ntcImVuY3J5cHRlZE1lc3NhZ2VcIjpcIld2blBMbXVnR3NNZ3N5TlBRU2toWjEwWkJKV09lV2
    Z1SncxZzR2ZVRjOENoTWpTaE1SWkdBemJ2TEhpM2RvTGtaYkNxcXJQeVIwVU1SNEFDQUFlSks
    rSWNvTzM4UlFDbE5XNTgyM1ZrNEFEMmlkSGxKQU90YjhjYXVrWlFOTkdQUmVJL3lwZ3c3Szhj
    M0lRRlBHQStMZ2ZaZ1dtWjNtWm4yMFFmYU9JRHZvcGt2V1hFTHNSVXdvaC9lNXFUeTlpb2RoO
    WlTQmUvN09HSlUvK2h6MTMrRnc2ZFk3d2F3Y3FVY0hXUkRSYkl3Tzk5dXU5L2NEbzQxZjZyT3
    JoaGNVTTBlY25Eak1lYzhMNy9RUWozMmZsMGNJMVQwdHg2UFpuMU1iby9iMG5VOTAwTzN1VXB
    nNWtheHBpRzg5a0NhemR0V2FlMC9MaitsWENMcm1UYjV5VGxmVXE3L25OTmwvTEwwT1BaUit3
    MENDdnBKZDB4b3QwRkd1OXRTdHYwOG9CTkJ2Y25kNDMzUmYraGlyaGVoOW1JNEJET01rODIxa
    nlxWUcxNWdGVGFlMFFYTDUzS2lFalpYZHV4VDdmc0F2YXc3OXkzemNhMEVnXFx1MDAzZFxcdT
    AwM2RcIixcImVwaGVtZXJhbFB1YmxpY0tleVwiOlwiQkRkR0xtQVg2MUoxZnRNTStLSU95dkZ
```

```
    1a1BPZWovVURld0krK0lqd2hyVXZwdVFpbDBRY29tK0JCRkh3QnN4U2VhZDYrK0tYanBCWUQ4
    VUVkWDR6ZFJnXFx1MDAzZFwiLFwidGFnXCI6XCJVallCemN6dEl6Q3pkMTJMVHRmTGF2OWRtV
    jc4aHM2NlVIT3c5WFRoQ2hzXFx1MDAzZFwifSJ9
```


Response to Successful Request


```
payerAuthSetupReply_accessToken=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiI2N
    jFlZjkwMS04MTFlLTQzZWQtkwMS04MTFlLTQzZWQtYWU0MS00MDM4ZGIyNDI5MjUiLCJpYXQi
    OjE2Nzk1ODIyNDAsImlzcyI6IjVkZDgzYmYwMGU0MjNkMTQ5OGRjYmFjYSIsImV4cCI6MTY3O
    TU4NTg0MCwiT3JnVW5pdElkIjoiNTVlZjNmMDNmNzIzYWE0MzFjOTliMDA2IiwiUmVmZXJlbm
    NlSWQiOiJjMjUxYmE4OC1hMjY2LTQ1YmItODE3OC02MDc4NzFjMWFhNzQifQ.picPcWjbtOLG
    ZmNyLEhlM0NV3GYNVu7nRXIt7diaO1w
decision=ACCEPT
payerAuthSetupReply_deviceDataCollectionURL=https://centinelapistag.cardinal
    commerce.com/V1/Cruise/Collect
payerAuthSetupReply_referenceID=c251ba88-a266-45bb-8178-607871c1aa74
merchantReferenceCode=0001
reasonCode=100
requestID=6795822405176891104008
payerAuthSetupReply_reasonCode=100
requestToken=AxizbwSTcGb7N7K5VVsI/7gBURy2bEzgAMCoZNJMq6PSYifATAAAFwni
```


Use Case: Checking Enrollment in Payer Authentication

Running the Check Enrollment service identifies the customer's bank and collects data about the device that the customer is using to place the order.

Card-Specific Requirements

Some payment cards require information to be collected during a transaction.

payerAuthEnrollService_defaultCard	Recommended for Discover ProtectBuy.
payerAuthEnrollService_MCC	Required when the card type is Cartes Bancaires.
payerAuthEnrollService_productCode	Required for American Express SafeKey (U.S.) when the product code is Airlinepurchase (AIR).
payerAuthEnrollService_merchantName	Required for Visa Secure travel.
shipTo_street1	Required only for American Express SafeKey (US).
shipTo_street2	Required only for American Express SafeKey (US.)

shipTo_city	Required only for American Express SafeKey (US).
shipTo_country	Required only for American Express SafeKey (US).
shipTo_postalCode>	Required for American Express SafeKey (US).

Country-Specific Requirements

These fields are required for transactions in specific countries.

payerAuthEnrollService_merchantScore	Required for transactions processed in France.
billTo_state	Required for transactions in US, Canada, and Mainland China.
billTo_city	Required for transactions in US, Canada, and Mainland China.
billTo_postalCode	Required when the **billTo_country** field value is `US` or `CA`.
shipTo_county	Required when the **shipTo_country** field value is `CA`, `US`, or `China`.
shipTo_postalCode	Required when the **shipTo_country** field value is `US` or `CA`.

Processor-Specific Requirements

These fields are required by specific processors for transactions.

transactionMode	Required only for merchants in Saudi Arabia.

Endpoint

Set the `ccAuthService_run` field to `true`.
Send the request to:
Production: `https://ics2ws.ic3.com/commerce/1.x/transactionProcessor`
Test: `https://ics2wstest.ic3.com/commerce/1.x/transactionProcessor`

Required Fields for Checking Enrollment in Payer Authentication

These fields are the minimum fields required for verifying that a customer is enrolled in a payer authentation program. Under certain circumstances, a field that normally is optional might be required. The circumstance that makes an optional field required is noted.

billTo_city

billTo_country	Required for US, Canada, and Mainland China. For Mainland China, use the ISO 3166-2 format.
billTo_email	
billTo_firstName	
billTo_lastName	
billTo_postalCode	Required for US, Canada, and Mainland China.
billTo_state	Required for US., Canada, and Mainland China.
billTo_street1	
card_accountNumber	
card_cardType	
card_expirationMonth	Required when **card_accountNumber** is included.
card_expirationYear	Required when **card_accountNumber** is included.
merchantReferenceCode	
payerAuthEnrollService_referenceID	
payerAuthEnrollService_returnURL	
payerAuthEnrollService_run	Required (when available) unless market or regional mandate restricts sending this information.
purchaseTotals_currency	
purchaseTotals_grandTotalAmount	Optional when you use the **item_#_unitPrice** field.

Related information
- API Field Reference for the Simple Order API

Optional Fields for Checking Enrollment in Payer Authentication

These fields are usually optional when verifying enrollment for a Payer Authentication transaction. In certain circumstances, the information provided by an optional field might be required before a transaction can proceed. Those optional fields that are sometimes required are also listed as required fields with the circumstance described.

Important

The fields that are marked with a single asterisk (*) in front are browser-related fields. The information collected by these browser-related fields, while not required, is highly recommended for all EMV 3-D Secure transactions.

Important

The fields that are marked with a double asterisk (**) at the end will be required as part of the EMV 3-D Secure 2.x minimum data requirements for Visa Secure. These conditionally optional fields will be removed from the optional fields list and moved to the required list on August 12, 2024.

airlineData_leg_#_carrierCode	Required for each leg.
airlineData_leg_#_departureDate	The numbered element name should contain 0 instead of #. Payer Authentication services only use the first leg of the trip.
airlineData_leg_#_destination	Required for each leg.
airlineData_leg_#_originatingAirportCode	
airlineData_numberOfPassengers	
airlineData_passenger_#_firstName	
airlineData_passenger_#_lastName	
billTo_country**	
billTo_email**	
billTo_firstName**	
billTo_lastName**	
billTo_postalCode**	Required when the **billTo_country** field is `US` or `CA`.
billTo_customerAccountChangeDate	
billTo_customerAccountCreateDate	
billTo_customerAccountPasswordChangeDate	
***billTo_httpBrowserColorDepth**	
***billTo_httpBrowserJavaEnabled**	
***billTo_httpBrowserJavaScriptEnabled**	
***billTo_httpBrowserLanguage**	
billTo_httpBrowserScreenHeight*	
billTo_httpBrowserScreenWidth*	
***billTo_httpBrowserTimeDifference**	

*** billTo_ipAddress****	
billTo_passportCountry	
billTo_passportNumber	
billTo_phoneNumber**	Required if **payerAuthEnrollService_mobilePhone** or **billTo_workNumber** is not used.
billTo_street1**	
billTo_street2	
billTo_street3	
card_cardType	
card_cvNumber	
ccAuthService_paChallengeCode	
encryptedPayment_data	
item_#_passengerFirstName	
item_#_passengerLastName	
item_#_productDescription	
item_#_productName	
item_#_productSKU	
item_#_quantity	
item_#_shippingAddress1	
item_#_shippingAddress2	
item_#_shippingCity	
item_#_shippingCountryCode	
item_#_shippingDestinationTypes	
item_#_shippingFirstName	
item_#_shippingLastName	
item_#_shippingMiddleName	
item_#_shippingPhone	
item_#_shippingPostalCode	
item_#_shippingState	
item_#_totalAmount	
item_#_unitPrice	

merchantDefinedData_mddField_1 to merchantDefinedData_mddField_5

Important

These fields override the old merchant-defined data fields. For example, when you use the obsolete field **merchantDefinedData_field5** and the new field **merchantDefinedData_mddField_5** in the same request, the new field value overwrites the value specified in the obsolete field.

Warning

Merchant-defined data fields are not intended to and must not be used to capture personally identifying information. Accordingly, merchants are prohibited from capturing, obtaining, and/or transmitting any personally identifying information in or via the merchant defined data fields. Personally identifying information includes, but is not limited to, address, credit card number, Social Security number, driver's license number, state-issued identification number, passport number, and card verification numbers (CVV, CVC2, CVV2, CID, CVN). When a merchant is discovered capturing and/or transmitting personally identifying information via the merchant-defined data fields, whether intentionally or accidentally, the merchant's account is immediately suspended, resulting in a rejection of any and all transaction requests submitted by the merchant after the point of suspension.

merchantReferenceCode

pa_otpToken

payerAuthEnrollService_accountPurchases	Recommended for Discover ProtectBuy.
payerAuthEnrollService_acquirerCountry	
payerAuthEnrollService_acsWindowSize	
payerAuthEnrollService_addCardAttempts	Recommended for Discover ProtectBuy.
payerAuthEnrollService_alternateAuthenticationData	
payerAuthEnrollService_alternateAuthenticationDate	
payerAuthEnrollService_alternateAuthenticationMethod	
payerAuthEnrollService_authenticationIndicator	
payerAuthEnrollService_authenticationTransactionID	Required for Standard integration.
payerAuthEnrollService_challengeCode	This field defaults to `01` on your account but is overridden by the merchant when you include this field. EMV 3-D Secure version 2.1.0 supports values `01`-`04`. Version 2.2.0 supports values `01`-`09`. **Warning** Modifying this field could affect liability shifts down the payment chain. Unless you are very familiar with the various types of authentication, do not change the default settings before consulting with customer support.
payerAuthEnrollService_customerCCAlias	Required when tokenization is enabled in the merchant profile settings.
payerAuthEnrollService_decoupled_AuthenticationMaxTime	
payerAuthEnrollService_decoupledAuthenticationIndicator	
payerAuthEnrollService_defaultCard	Recommended for Discover ProtectBuy.
*payerAuthEnrollService_deviceChannel	Required for SDK integration. When you use the SDK integration, this field is dynamically set to `SDK`. When you use the

	JavaScript code, this field is dynamically set to `Browser`. For merchant-initiated or 3RI transactions, you must set the field to `3RI`. When you use this field in addition to JavaScript code, you must set the field to `Browser`.
payerAuthEnrollService_fraudActivity	Recommended for Discover ProtectBuy.
payerAuthEnrollService_giftCardAmount	
payerAuthEnrollService_giftCardCount	
payerAuthEnrollService_giftCardCurrency	
***payerAuthEnrollService_httpAccept**	When the customer's browser provides a value, include that value in your request.
***payerAuthEnrollService_httpUserAccept**	
***payerAuthEnrollService_httpUserAgent**	
payerAuthEnrollService_installmentTotalCount	Required when the merchant and cardholder have agreed to installment payments.
payerAuthEnrollService_marketingOptIn	Recommended for Discover ProtectBuy.
payerAuthEnrollService_marketingSource	Recommended for Discover ProtectBuy.
payerAuthEnrollService_MCC	Required when the card type is Cartes Bancaires.
payerAuthEnrollService_merchantFraudRate	
payerAuthEnrollService_merchantID	Merchant bank identifier, such as Paymentech's division, FDC's Terminal ID, or Vital V number. Use this field for evaluation, testing, and production. This number is not your merchant ID.
payerAuthEnrollService_merchantName	Required for Visa Secure travel.
payerAuthEnrollService_merchantNewCustomer	
payerAuthEnrollService_merchantScore	Required for transactions processed in France.
payerAuthEnrollService_merchantURL	
payerAuthEnrollService_messageCategory	
payerAuthEnrollService_mobilePhone**	Required if **payerAuthEnrollService_workPhone** or **billTo_phoneNumber** is not used.
payerAuthEnrollService_overridePaymentMethod	

payerAuthEnrollService_paymentAccountDate	Recommended for Discover ProtectBuy.
payerAuthEnrollService_preorder	
payerAuthEnrollService_preorderDate	
payerAuthEnrollService_priorAuthenticationData	
payerAuthEnrollService_priorAuthenticationMethod	
payerAuthEnrollService_priorAuthenticationReferenceID	
payerAuthEnrollService_priorAuthenticationTime	
payerAuthEnrollService_productCode	Required for American Express SafeKey (U.S.) when the product code is AIR (Airline purchase).
payerAuthEnrollService_recurringEndDate	Required for recurring transactions.
payerAuthEnrollService_recurringFrequency	Required for recurring transactions.
payerAuthEnrollService_recurringOriginalPurchaseDate	Required for recurring transactions.
payerAuthEnrollService_referenceID	Required for Hybrid or Cardinal Cruise Direct Connection API integration.
payerAuthEnrollService_reorder	
payerAuthEnrollService_requestorInitiatedAuthenticationIndicator	EMV 3-D Secure version 2.1.0 supports values `01`-`05`. Version 2.2.0 supports values `01`-`11`.
payerAuthEnrollService_requestorName	
payerAuthEnrollService_returnURL	
payerAuthEnrollService_scoreRequest	
payerAuthEnrollService_sdkMaxTimeout	Required for 3-D Secure 2.x.
payerAuthEnrollService_secureCorporatePaymentIndicator	
payerAuthEnrollService_shipAddressUsageDate	Recommended for Discover ProtectBuy.
payerAuthEnrollService_totalOffersCount	
payerAuthEnrollService_transactionCountDay	Recommended for Discover ProtectBuy.
payerAuthEnrollService_transactionCountYear	Recommended for Discover ProtectBuy.

payerAuthEnrollService_transactionMode	
payerAuthEnrollService_whiteListStatus	
payerAuthEnrollService_workPhone**	**Required if payerAuthEnrollService_mobilePhone or billTo_phoneNumber is not used.**
paymentNetworkToken_transactionType	
requestID	
shipTo_city	Required when any shipping address information is included. Required for American Express SafeKey (US).
shipTo_country	Required only for American Express SafeKey (US).
shipTo_destinationCode	
shipTo_destinationTypes	Required when the **bill_country** field value is `US` or `CA`.
shipTo_firstName	
shipTo_lastName	
shipTo_middleName	
shipTo_phoneNumber	
shipTo_postalCode	Required when the **shipTo_country** field value is `US` or `CA`. Required for American Express SafeKey (U.S.).
shipTo_shippingMethod	Required only for American Express SafeKey (US).
shipTo_state	Required when the **shipTo_country** field value is `CA`, `US`, or `Mainland China`. Required for American Express SafeKey (U.S.).
shipTo_street1	Required when any shipping address information is included. Required for American Express SafeKey (US).
shipTo_street2	Required only for American Express SafeKey (US).
shipTo_street3	Required for American Express SafeKey (US).

Simple Order Example: Check Enrollment

Request

```
billTo_city=Mountain View
```

```
billTo_country=US
billTo_email=test@yahoo.com
billTo_firstName=Tanya
billTo_lastName=Lee
billTo_postalCode=94043
billTo_state=CA
billTo_street1=1234 Gold Ave
card_accountNumber=XXXXXXXXXXXXXXXX
card_cardType=001
card_cvNumber=111
card_expirationMonth=12
card_expirationYear=2030
ccAuthService_run=true
merchantID=patest
merchantReferenceCode=0001
payerAuthEnrollService_referenceID=f13fe5e0-9b47-4ea1-a03a-ec360f4d0f9f
payerAuthEnrollService_returnURL=https://example.com/step-up-return-url.jsp
payerAuthEnrollService_run=true
purchaseTotals_currency=USD
purchaseTotals_grandTotalAmount=30.00
```

Response to Successful Request

```
decision=REJECT
merchantReferenceCode=0001
payerAuthEnrollReply_accessToken=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiI1MDc4OTI0Mi0z
 YmEzLTRhZTItYWQwOS1kZjZkODk2NWQ5MjciLCJpYXQiOjE1OTgyOTk1MjQsImlzcyI6IjVkZDgzYmYwMGU0MjNk
 MTQ5OGRjYmFjYSIsImV4cCI6MTU5ODMwMzEyNCwiT3JnVW5pdElkIjoiNTVlZjNmMTBmNzIzYWE0MzFjOTliNWViIi
 wiUGF5bG9hZCI6eyJBQ1NVcmwiOiJodHRwczovLzBtZXJjaGFudGFjc3N0YWcuY2FyZGluYWxjb21tZXJjZS5jb20
 vTWVyY2hhbnRBQ1NXZWIvY3JlcS5qc3AiLCJQYXlsb2FkIjoiZXlKdFpYTnpZV2RsVkhsd1pTSTZJa05TWlhFaUxDS
 nRaWE56WVdkbFZtVnljMmx2YmlJNklqSXVNaTR3SWl3aWRHaHlaV1ZFVTFObGNuWmxjbFJ5WVc1elNVUWlPaUkzT
 kRNeVlUWXdNQzA0TXpNMkxUUm1PRGN0WVdKbE9TMDJObVkzTkRFM01EaGhNV1FpTENKaFkzTlVjbUZ1YzBsRUlqb
 2lPR0U1TkRkaU9ETXRNRFJpTkMwMFltVmlMV0V5WWpNdFpHTmpNV0UxWmprMFlURXlJaXdpWTJoaGJHeGxibWRsVj
 JsdVpHOTNVMmw2WlNJNklqQXlJbjAiLCJUcmFuc2FjdGlvbklkIjoiVEQ1b1MwbzFGQzY1cWF2MHhzeDAifS
 wiT2JqZWN0aWZ5UGF5bG9hZCI6dHJ1ZSwiUmV0dXJuVXJsIjoiaHR0cHM6Ly9leGFtcGxlLmNvbS9zdGVwLXV
 wLXJldHVybi11cmwuanNwIn0.8wZ8XhLgOIIRvgEUugvYrRAi-efavZTNM0tWInYLTfE
payerAuthEnrollReply_acsTransactionID=8a947b83-04b4-4beb-a2b3-dcc1a5f94a12
payerAuthEnrollReply_acsURL=https://0merchantacsstag.cardinalcommerce.com/MerchantACSWeb/
creq.jsp
payerAuthEnrollReply_authenticationTransactionID=TD5oS0o1FC65qav0xsx0
payerAuthEnrollReply_cardBin=40000000
payerAuthEnrollReply_cardTypeName=VISA
payerAuthEnrollReply_challengeRequired=false
payerAuthEnrollReply_directoryServerTransactionID=395fb036-cfc6-462b-b28d-d6ed7c970cdd
payerAuthEnrollReply_paReq=eyJtZXNzYWdlVHlwZSI6IkNSZXEiLCJtZXNzYWdlVmVyc2lvbiI6IjIuMi4wIiwidGhy
 ZWVEU1NlcnZlclRyYW5zSUQiOiI3NDMyYTYwMC04MzM2LTRmODctYWJlOS02NmY3NDE3MDhhMWQiLCJhY3NU
 cmFuc0lEIjoiOGE5NDdiODMtMDRiNC00YmViLWEyYjMtZGNjMWE1Zjk0YTEyIiwiY2hhbGxlbmdlV2luZG93U2l
 6ZSI6IjAyIn0
payerAuthEnrollReply_reasonCode=475
payerAuthEnrollReply_specificationVersion=2.2.0
payerAuthEnrollReply_stepUpUrl=https://centinelapistag.cardinalcommerce.com/V2/Cruise/StepUp
payerAuthEnrollReply_threeDSServerTransactionID=7432a600-8336-4f87-abe9-66f741708a1d
payerAuthEnrollReply_veresEnrolled=Y
reasonCode=475
requestID=5982995245816268803007
requestToken=AxjzbwSTRFa9DM1xnUu/ABEBURwlqsQ5pAy7gDtXb0kyro9JLIYA8AAA2wK2
```

Use Case: Checking Enrollment in Payer Authentication Using Google Pay

Running the Check Enrollment service collects data about the device that the customer is using to place the order and verifies that the customer is enrolled in a payer authentication program. This use case demonstrates how the service works with a digital payment method like Google Pay.

Card-Specific Requirements

Some payment cards require information to be collected during a transaction.

payerAuthEnrollService_defaultCard	Recommended for Discover ProtectBuy.
payerAuthEnrollService_MCC	Required when the card type is Cartes Bancaires.
payerAuthEnrollService_productCode	Required for American Express SafeKey (U.S.) when the product code is Airlinepurchase (AIR).
payerAuthEnrollService_merchantName	Required for Visa Secure travel.
shipTo_street1	Required only for American Express SafeKey (US).
shipTo_street2	Required only for American Express SafeKey (US.)
shipTo_city	Required only for American Express SafeKey (US).
shipTo_country	Required only for American Express SafeKey (US).
shipTo_postalCode>	Required for American Express SafeKey (US).

Country-Specific Requirements

These fields are required for transactions in specific countries.

payerAuthEnrollService_merchantScore	Required for transactions processed in France.
billTo_state	Required for transactions in US, Canada, and Mainland China.
billTo_city	Required for transactions in US, Canada, and Mainland China.

billTo_postalCode	Required when the **billTo_country** field value is `US` or `CA`.
shipTo_county	Required when the **shipTo_country** field value is `CA`, `US`, or `China`.
shipTo_postalCode	Required when the **shipTo_country** field value is `US` or `CA`.

Processor-Specific Requirements

These fields are required by specific processors for transactions.

transactionMode	Required only for merchants in Saudi Arabia.

Endpoint

Set the `ccAuthService_run` field to `true`.
Send the request to:
Production: `https://ics2ws.ic3.com/commerce/1.x/transactionProcessor`
Test: `https://ics2wstest.ic3.com/commerce/1.x/transactionProcessor`

Required Fields for Checking Enrollment in Payer Authentication

These fields are the minimum fields required for verifying that a customer is enrolled in a payer authentation program. Under certain circumstances, a field that normally is optional might be required. The circumstance that makes an optional field required is noted.

billTo_city	
billTo_country	Required for US, Canada, and Mainland China. For Mainland China, use the ISO 3166-2 format.
billTo_email	
billTo_firstName	
billTo_lastName	
billTo_postalCode	Required for US, Canada, and Mainland China.
billTo_state	Required for US., Canada, and Mainland China.
billTo_street1	
card_accountNumber	
card_cardType	
card_expirationMonth	Required when **card_accountNumber** is included.

card_expirationYear	Required when **card_accountNumber** is included.
merchantReferenceCode	
payerAuthEnrollService_referenceID	
payerAuthEnrollService_returnURL	
payerAuthEnrollService_run	Required (when available) unless market or regional mandate restricts sending this information.
purchaseTotals_currency	
purchaseTotals_grandTotalAmount	Optional when you use the **item_#_unitPrice** field.

Related information
- API Field Reference for the Simple Order API

Optional Fields for Checking Enrollment in Payer Authentication

These fields are usually optional when verifying enrollment for a Payer Authentication transaction. In certain circumstances, the information provided by an optional field might be required before a transaction can proceed. Those optional fields that are sometimes required are also listed as required fields with the circumstance described.

Important

The fields that are marked with a single asterisk (*) in front are browser-related fields. The information collected by these browser-related fields, while not required, is highly recommended for all EMV 3-D Secure transactions.

Important

The fields that are marked with a double asterisk (**) at the end will be required as part of the EMV 3-D Secure 2.x minimum data requirements for Visa Secure. These conditionally optional fields will be removed from the optional fields list and moved to the required list on August 12, 2024.

airlineData_leg_#_carrierCode	Required for each leg.
airlineData_leg_#_departureDate	The numbered element name should contain 0 instead of #. Payer Authentication services only use the first leg of the trip.
airlineData_leg_#_destination	Required for each leg.
airlineData_leg_#_originatingAirportCode	

airlineData_numberOfPassengers	
airlineData_passenger_#_firstName	
airlineData_passenger_#_lastName	
billTo_country**	
billTo_email**	
billTo_firstName**	
billTo_lastName**	
billTo_postalCode**	Required when the **billTo_country** field is US or CA.
billTo_customerAccountChangeDate	
billTo_customerAccountCreateDate	
billTo_customerAccountPasswordChangeDate	
***billTo_httpBrowserColorDepth**	
***billTo_httpBrowserJavaEnabled**	
***billTo_httpBrowserJavaScriptEnabled**	
***billTo_httpBrowserLanguage**	
billTo_httpBrowserScreenHeight*	
billTo_httpBrowserScreenWidth*	
***billTo_httpBrowserTimeDifference**	
*** billTo_ipAddress****	
billTo_passportCountry	
billTo_passportNumber	
billTo_phoneNumber**	Required if **payerAuthEnrollService_mobilePhone** or **billTo_workNumber** is not used.
billTo_street1**	
billTo_street2	
billTo_street3	
card_cardType	
card_cvNumber	
ccAuthService_paChallengeCode	
encryptedPayment_data	
item_#_passengerFirstName	

item_#_passengerLastName

item_#_productDescription

item_#_productName

item_#_productSKU

item_#_quantity

item_#_shippingAddress1

item_#_shippingAddress2

item_#_shippingCity

item_#_shippingCountryCode

item_#_shippingDestinationTypes

item_#_shippingFirstName

item_#_shippingLastName

item_#_shippingMiddleName

item_#_shippingPhone

item_#_shippingPostalCode

item_#_shippingState

item_#_totalAmount

item_#_unitPrice

merchantDefinedData_mddField_1 to merchantDefinedData_mddField_5

Important

These fields override the old merchant-defined data fields. For example, when you use the obsolete field **merchantDefinedData_field5** and the new field **merchantDefinedData_mddField_5** in the same request, the new field value overwrites the value specified in the obsolete field.

Warning

Merchant-defined data fields are not intended to and must not be used to capture personally identifying information. Accordingly, merchants are prohibited from capturing, obtaining, and/or

	transmitting any personally identifying information in or via the merchant defined data fields. Personally identifying information includes, but is not limited to, address, credit card number, Social Security number, driver's license number, state-issued identification number, passport number, and card verification numbers (CVV, CVC2, CVV2, CID, CVN). When a merchant is discovered capturing and/or transmitting personally identifying information via the merchant-defined data fields, whether intentionally or accidentally, the merchant's account is immediately suspended, resulting in a rejection of any and all transaction requests submitted by the merchant after the point of suspension.
merchantReferenceCode	
pa_otpToken	
payerAuthEnrollService_accountPurchases	Recommended for Discover ProtectBuy.
payerAuthEnrollService_acquirerCountry	
payerAuthEnrollService_acsWindowSize	
payerAuthEnrollService_addCardAttempts	Recommended for Discover ProtectBuy.
payerAuthEnrollService_alternateAuthenticationData	
payerAuthEnrollService_alternateAuthenticationDate	
payerAuthEnrollService_alternateAuthenticationMethod	
payerAuthEnrollService_authenticationIndicator	
payerAuthEnrollService_authenticationTransactionID	Required for Standard integration.
payerAuthEnrollService_challengeCode	This field defaults to `01` on your account but is overridden by the merchant when you include this field. EMV 3-D Secure version

	2.1.0 supports values `01`-`04`. Version 2.2.0 supports values `01`-`09`. **Warning** Modifying this field could affect liability shifts down the payment chain. Unless you are very familiar with the various types of authentication, do not change the default settings before consulting with customer support.
payerAuthEnrollService_customerCCAlias	Required when tokenization is enabled in the merchant profile settings.
payerAuthEnrollService_decoupled_AuthenticationMaxTime	
payerAuthEnrollService_decoupledAuthenticationIndicator	
payerAuthEnrollService_defaultCard	Recommended for Discover ProtectBuy.
***payerAuthEnrollService_deviceChannel**	Required for SDK integration. When you use the SDK integration, this field is dynamically set to `SDK`. When you use the JavaScript code, this field is dynamically set to `Browser`. For merchant-initiated or 3RI transactions, you must set the field to `3RI`. When you use this field in addition to JavaScript code, you must set the field to `Browser`.
payerAuthEnrollService_fraudActivity	Recommended for Discover ProtectBuy.
payerAuthEnrollService_giftCardAmount	
payerAuthEnrollService_giftCardCount	
payerAuthEnrollService_giftCardCurrency	
***payerAuthEnrollService_httpAccept**	When the customer's browser provides a value, include that value in your request.
***payerAuthEnrollService_httpUserAccept**	
***payerAuthEnrollService_httpUserAgent**	
payerAuthEnrollService_installmentTotalCount	Required when the merchant and cardholder have agreed to installment payments.
payerAuthEnrollService_marketingOptIn	Recommended for Discover ProtectBuy.

payerAuthEnrollService_marketingSource	Recommended for Discover ProtectBuy.
payerAuthEnrollService_MCC	Required when the card type is Cartes Bancaires.
payerAuthEnrollService_merchantFraudRate	
payerAuthEnrollService_merchantID	Merchant bank identifier, such as Paymentech's division, FDC's Terminal ID, or Vital V number. Use this field for evaluation, testing, and production. This number is not your merchant ID.
payerAuthEnrollService_merchantName	Required for Visa Secure travel.
payerAuthEnrollService_merchantNewCustomer	
payerAuthEnrollService_merchantScore	Required for transactions processed in France.
payerAuthEnrollService_merchantURL	
payerAuthEnrollService_messageCategory	
payerAuthEnrollService_mobilePhone**	Required if **payerAuthEnrollService_workPhone** or **billTo_phoneNumber** is not used.
payerAuthEnrollService_overridePaymentMethod	
payerAuthEnrollService_paymentAccountDate	Recommended for Discover ProtectBuy.
payerAuthEnrollService_preorder	
payerAuthEnrollService_preorderDate	
payerAuthEnrollService_priorAuthenticationData	
payerAuthEnrollService_priorAuthenticationMethod	
payerAuthEnrollService_priorAuthenticationReferenceID	
payerAuthEnrollService_priorAuthenticationTime	
payerAuthEnrollService_productCode	Required for American Express SafeKey (U.S.) when the product code is AIR (Airline purchase).
payerAuthEnrollService_recurringEndDate	Required for recurring transactions.
payerAuthEnrollService_recurringFrequency	Required for recurring transactions.

payerAuthEnrollService_recurringOriginalPurchaseDate	Required for recurring transactions.
payerAuthEnrollService_referenceID	Required for Hybrid or Cardinal Cruise Direct Connection API integration.
payerAuthEnrollService_reorder	
payerAuthEnrollService_requestorInitiatedAuthenticationIndicator	EMV 3-D Secure version 2.1.0 supports values `01`-`05`. Version 2.2.0 supports values `01`-`11`.
payerAuthEnrollService_requestorName	
payerAuthEnrollService_returnURL	
payerAuthEnrollService_scoreRequest	
payerAuthEnrollService_sdkMaxTimeout	Required for 3-D Secure 2.x.
payerAuthEnrollService_secureCorporatePaymentIndicator	
payerAuthEnrollService_shipAddressUsageDate	Recommended for Discover ProtectBuy.
payerAuthEnrollService_totalOffersCount	
payerAuthEnrollService_transactionCountDay	Recommended for Discover ProtectBuy.
payerAuthEnrollService_transactionCountYear	Recommended for Discover ProtectBuy.
payerAuthEnrollService_transactionMode	
payerAuthEnrollService_whiteListStatus	
payerAuthEnrollService_workPhone**	**Required if payerAuthEnrollService_mobilePhone or billTo_phoneNumber is not used.**
paymentNetworkToken_transactionType	
requestID	
shipTo_city	Required when any shipping address information is included. Required for American Express SafeKey (US).
shipTo_country	Required only for American Express SafeKey (US).
shipTo_destinationCode	
shipTo_destinationTypes	Required when the **bill_country** field value is `US` or `CA`.
shipTo_firstName	

shipTo_lastName	
shipTo_middleName	
shipTo_phoneNumber	
shipTo_postalCode	Required when the **shipTo_country** field value is `US` or `CA`. Required for American Express SafeKey (U.S.).
shipTo_shippingMethod	Required only for American Express SafeKey (US).
shipTo_state	Required when the **shipTo_country** field value is `CA`, `US`, or `Mainland China`. Required for American Express SafeKey (U.S.).
shipTo_street1	Required when any shipping address information is included. Required for American Express SafeKey (US).
shipTo_street2	Required only for American Express SafeKey (US).
shipTo_street3	Required for American Express SafeKey (US).

Simple Order Example: Checking Enrollment in Payer Authentication Using Google Pay

This is an example of an Payer Authentication Check Enrollment request and its corresponding response using Google Pay as the digital payment option.

Request


```
{
 "paymentInformation": {
"fluidData" : {
    "value" : "eyJzaWduYXR1cmUiOiJNRVFDSUVhVVBmc1RIMERyMnZmeG8wVkFlZ3N2bHlSRzdENEFsYm
Rwa1pPdlNzZGtBaUFVODE2aHpmMG5BMzJzQmx6anlUSURyZXBHNUY1eEtlRmNnSE9aK3RML2ZRXHUwMDNkXHUw
MDNkIiwiaW50ZXJtZWRpYXRlU2lnbmluZ0tleSI6eyJzaWduZWRLZXkiOiJ7XCJrZXlWYWx1ZVwiOlwiTUZrd0V
3WUhLb1pJemowQ0FRWUlLb1pJemowREFRY0RRZ0FFOFdKSHVMOFVuWW9WWDNHV3dGVkJpcnh6L3lJdGl0aW9ne
WhDeGpCRm5tS3pCcWs2K3lnVU5SUGF4THdaaWtILzBxV0s1QXhlc3BDNVhwN1NHNUN1T1FcXHUwMDNkXFx1MDAzZF
wiLFwia2V5RXhwaXJhdGlvblwiOlwiMTYzMDUxNjYyODA0OVwifSIsInNpZ25hdHVyZXMiOlsiTUVVQ0lRRHlTQTV
1T2t5UXQ5cFoyQlEzaXBmcGNWT0F5ZmIzM2ozUEZPQUw3K1o5S3dJZ2FjWWp2YWJpTEUyWHFkNU1xNGphNSt
EVldoREttVHpoMmk1RGlnbllFQndcdTAwM2QiXX0sInByb3RvY29sVmVyc2lvbiI6IkVDdjIiLCJzaWduZWR
NZXNzYWdlIjoie1wiZW5jcnlwdGVkTWVzc2FnZVwiOlwiWG5qOGxSSWhGMDVEWWdRK3hwNEE5YUhsVGE1U2l
jdUJac2w1L2NNRkJlclBBY1RzaE4zRFlOb1MvdEVkRkRYRzZJRXBpVlcxVnV6OUprejNWWGdpMzJrT21EVk9aakJNW
TFvVHdTQnA5WG53ejlLYUtOekYvRFBSTy9jbStobW9iZ2dSdmxGSStOekN5U1VNWW1hbTJjZFUyZGRZWmZHck9nZWN
Sc3FrdW1tNmlMa0xGQTFJcDFrNWFRV21EUElEdTh1SnNmbWs4bzMyMlpteVdMMVVWenE0WHFkNTZScXZoL1VFeEp3R
C9HZXU5SW00M0pmblZqckVkeDE0Ykx1OUpmMHJrcU5ONG5sM0NVZEFoMVNhZnBzdkduTVRML1Nmenk1ZGdDZlRD
cHJDdW85UVZPaXVValBJNUdXRlBKSWVVVVU1cUZhcis3NXFBT2dvZ0tNRUZ3OFVxL1A0UjBDcXczcFllNnc2enlaV
zdDV1YxRzRMc3BITTNRaE83bFZNNmRjSWZQWW00ZitubWI3UzgwY29KTXR1QjkxVEhjZzJmVXhwM2FrWEhSdzNyN
3BRZk9KWWFieUlURmtieDh0Yi9ieWl6VUZEVVU4S3EwTmVCVTVrQng2L21qUDg4bWxoWkE2ZERrNWJVc2o4SDBDSk
9nWUtCbVgyR09vamRtTDd5YlBnTU5vNnhsYjRtUzVkaTJjZUpFakFybEZFa3NWNTlsS2lodk5pckRZc1BTU2lTRF
```

```
VZNjBMUXVuTEErYjFMSnpCMkpYelFcIixcImVwaGVtZXJhbFB1YmxpY0tleVwiOlwiQk84bmtEbE0ycVlCQmpQ
d00wbDdUTFY2UytUbzZDTFl0eXArWGM2cXpQYklLTEgxVytySGh3NUlwU2lqb1lTb3VaclNuWU9LV21yRVAyYmt
LMk4rTWFZXFx1MDAzZFwiLFwidGFnXCI6XCJuVU5xUVlxcy9YRVlDMmg0WFlibnVpajFLb1NzUFpacEpqVGI4TVVZ
cUZNXFx1MDAzZFwifSJ9"
  }
 },
 "processingInformation": {
  "paymentSolution": "012"
 }
}
```


Response to Successful Request


```
{
 "consumerAuthenticationInformation": {
  "accessToken":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiI5MjlhZDAwYy0zZmEyLTQ5ODgtODRjNC1hYTcxMGFl
Y2I1OGEiLCJpYXQiOjE2Mjk4MjY5NjAsImlzcyI6IjVkZDgzYmYwMGU0MjNkMTQ5OGRjYmFjYSIsImV4cCI6MTYyOTgz
MDU2MCwiT3JnVW5pdElkIjoiNWI5YzRiYjNmZjYyNmIxMzQ0ODEwYTAxIiwiUmVmZXJlbmNlSWQiOiI5NDkzZjJiZi0
4NmIwLTQ0ZmYtYmJjZS0wZjU1MjNlMWIzNGEifQ.FgVbwbW9_lwnlr4ovYR5VVPuVl6CklAVHHXS_5ODskA",
  "deviceDataCollectionUrl": "https://centinelapistag.cardinalcommerce.com/V1/Cruise/Collect",
  "referenceId": "9493f2bf-86b0-44ff-bbce-0f5523e1b34a",
  "token": "AxizbwSTVW4Mj1fvsU27ABEBURxPZebOAE1IZNJMvRiuZhTA9AAA+QBf"
 },
 "id": "6298269599786696003003",
 "status": "COMPLETED",
 "submitTimeUtc": "2021-08-24T17:42:40Z"
}
```


Use Case: Validating Payer Authentication

Running the Validation service compares the customer's response to the challenge from the issuing bank to validate the customer identity.

Card-Specific Requirements

Some payment cards require information to be collected during a transaction.

payerAuthEnrollService_defaultCard	Recommended for Discover ProtectBuy.
payerAuthEnrollService_MCC	Required when the card type is Cartes Bancaires.
payerAuthEnrollService_productCode	Required for American Express SafeKey (US) when the product code is `AIR` for an airlinepurchase).
payerAuthEnrollService_merchantName	Required for Visa Secure travel.
shipTo_street1	Required only for American Express SafeKey (US).

shipTo_street2	Required only for American Express SafeKey (US)

Country-Specific Requirements

These fields are required for transactions in specific countries.

payerAuthEnrollService_merchantScore	Required for transactions processed in France.
billTo_city	Required for transactions in US., Canada, and Mainland China.
billTo_postalCode	Required when the **billTo_country** field value is `US` or `CA`.
billTo_state	Required for transactions in US, Canada, and Mainland China.

Endpoint

Set the `payerAuthValidateService_run` fields to `true`.
Send the request to:
Production: `https://ics2ws.ic3.com/commerce/1.x/transactionProcessor`
Test: `https://ics2wstest.ic3.com/commerce/1.x/transactionProcessor`

Required Fields for Validating Payer Authentication

card_accountNumber	
card_cardType	
card_expirationMonth	Required when **card_accountNumber** is included.
card_expirationYear	Required when **card_accountNumber** is included.
item_#_unitPrice	Required when the **purchaseTotals_grandTotalAmount** field is not used.
merchantReferenceCode	
payerAuthValidateService_authenticationTransactionID	
payerAuthValidateService_run	
purchaseTotals_currency	
purchaseTotals_grandTotalAmount	Required when the **item_#_unitPrice** field is not used.

Related information
- API Field Reference for the Simple Order API

Optional Fields for Validating Payer Authentication

These fields are optional when validating a Payer Authentication transaction. In certain circumstances, the information provided by an optional field might be required before a transaction can proceed. Those optional fields that are sometimes required are listed in the required fields with the circumstance described.

payerAuthValidateService_credentialEncrypted

payerAuthValidateService_responseAccessToken

payerAuthValidateService_signedPARes

REST Example: Validating the Challenge

This example shows a request to validate the challenge from the issuer and its corresponding response.

Validation Request

```
merchantID=patest
merchantReferenceCode=0001
payerAuthValidateService_authenticationTransactionID=hejNPA0YQlL5gVwZ6OX0
payerAuthValidateService_run=true
purchaseTotals_currency=USD
purchaseTotals_grandTotalAmount=30.00
```

Validation Response

```
decision=ACCEPT
merchantReferenceCode=0001
payerAuthValidateReply_acsTransactionID=ff412c09-4ea8-4f37-923e-4c405fb3951c
payerAuthValidateReply_authenticationResult=0
payerAuthValidateReply_authenticationStatusMessage=Success
payerAuthValidateReply_cardBin=40000000
payerAuthValidateReply_cardTypeName=VISA
payerAuthValidateReply_cavv=MTIzNDU2Nzg5MDEyMzQ1Njc4OTA=
payerAuthValidateReply_commerceIndicator=vbv
payerAuthValidateReply_directoryServerTransactionID=6c29615b-1a1e-4c13-9739-0394917163a3
payerAuthValidateReply_eci=05
payerAuthValidateReply_eciRaw=05
payerAuthValidateReply_paresStatus=Y
payerAuthValidateReply_reasonCode=100
payerAuthValidateReply_specificationVersion=2.1.0
payerAuthValidateReply_threeDSServerTransactionID=2c2294e9-6b70-4b19-bedb-7b43065f20ce
payerAuthValidateReply_xid=MTIzNDU2Nzg5MDEyMzQ1Njc4OTA=
reasonCode=100
requestID=6001869286506329603009
```

Simple Order Example: Validating the Challenge

Request

```
merchantID=patest
merchantReferenceCode=0001
payerAuthValidateService_authenticationTransactionID=hejNPA0YQlL5gVwZ6OX0
payerAuthValidateService_run=true
purchaseTotals_currency=USD
purchaseTotals_grandTotalAmount=30.00
```

Response to Successful Request

```
decision=ACCEPT
merchantReferenceCode=0001
payerAuthValidateReply_acsTransactionID=ff412c09-4ea8-4f37-923e-4c405fb3951c
payerAuthValidateReply_authenticationResult=0
payerAuthValidateReply_authenticationStatusMessage=Success
payerAuthValidateReply_cardBin=40000000
payerAuthValidateReply_cardTypeName=VISA
payerAuthValidateReply_cavv=MTIzNDU2Nzg5MDEyMzQ1Njc4OTA=
payerAuthValidateReply_commerceIndicator=vbv
payerAuthValidateReply_directoryServerTransactionID=6c29615b-1a1e-4c13-9739-0394917163a3
 payerAuthValidateReply_eci=05
payerAuthValidateReply_eciRaw=05
payerAuthValidateReply_paresStatus=Y
payerAuthValidateReply_reasonCode=100
payerAuthValidateReply_specificationVersion=2.1.0
 payerAuthValidateReply_threeDSServerTransactionID=2c2294e9-6b70-4b19-bedb-7b43065f20ce
 payerAuthValidateReply_xid=MTIzNDU2Nzg5MDEyMzQ1Njc4OTA=
reasonCode=100
requestID=6000186928650632960300 9
```

Use Case: Validating Payer Authentication Using Google Pay

Validating the customer identity compares the customer's response to the challenge that was sent. This use case demonstrates how the service works using a digital payment method like Google Pay.

Card-Specific Requirements

Some payment cards require information to be collected during a transaction.

payerAuthEnrollService_defaultCard	Recommended for Discover ProtectBuy.
payerAuthEnrollService_MCC	Required when the card type is Cartes Bancaires.

payerAuthEnrollService_productCode	Required for American Express SafeKey (US) when the product code is `AIR` for an airlinepurchase).
payerAuthEnrollService_merchantName	Required for Visa Secure travel.
shipTo_street1	Required only for American Express SafeKey (US).
shipTo_street2	Required only for American Express SafeKey (US)

Country-Specific Requirements

These fields are required for transactions in specific countries.

payerAuthEnrollService_merchantScore	Required for transactions processed in France.
billTo_city	Required for transactions in US., Canada, and Mainland China.
billTo_postalCode	Required when the **billTo_country** field value is `US` or `CA`.
billTo_state	Required for transactions in US, Canada, and Mainland China.

Endpoint

Set the `payerAuthValidateService_run` fields to `true`.
Send the request to:
Production: `https://ics2ws.ic3.com/commerce/1.x/transactionProcessor`
Test: `https://ics2wstest.ic3.com/commerce/1.x/transactionProcessor`

Required Fields for Validating Payer Authentication

card_accountNumber	
card_cardType	
card_expirationMonth	Required when **card_accountNumber** is included.
card_expirationYear	Required when **card_accountNumber** is included.
item_#_unitPrice	Required when the **purchaseTotals_grandTotalAmount** field is not used.
merchantReferenceCode	

payerAuthValidateService_authenticationTransactionID	
payerAuthValidateService_run	
purchaseTotals_currency	
purchaseTotals_grandTotalAmount	Required when the **item_#_unitPrice** field is not used.

Related information
- API Field Reference for the Simple Order API

Optional Fields for Validating Payer Authentication

These fields are optional when validating a Payer Authentication transaction. In certain circumstances, the information provided by an optional field might be required before a transaction can proceed. Those optional fields that are sometimes required are listed in the required fields with the circumstance described.

payerAuthValidateService_credentialEncrypted

payerAuthValidateService_responseAccessToken

payerAuthValidateService_signedPARes

Simple Order Example: Validating the Challenge When Using Google Pay

This is an example of an Payer Authentication Validate request and response when using Google Pay as the digital payment option. The data in this example is for illustrative purposes only.

Request

```
billTo_city=Mountain View
billTo_country=US
billTo_email=null@email.com
billTo_firstName=John
billTo_lastName=Doe
billTo_postalCode=94043
billTo_state=CA
billTo_street1=1295 Charleston Road
card_accountNumber=XXXXXXXXXXXXXXXX
card_cardType=001
card_cvNumber=111
card_expirationMonth=12
card_expirationYear=2030
ccAuthService_run=true
merchantID=patest
merchantReferenceCode=0001
payerAuthValidateService_authenticationTransactionID=TD5oS0o1FC65qav0xsx0
payerAuthValidateService_run=true
purchaseTotals_currency=USD
```

```
purchaseTotals_grandTotalAmount=30.00
```

Use Case: Validating and Authorizing a Transaction

The Validation service can be combined with the Authorization service so that when a customer's authentication is validated, the transaction is automatically submitted for authorization.

Fields Specific to the Visa Secure Use Case

These API fields are required specifically for this use case.

ccAuthService_commerceIndicator	Set this field to `vbv` for a successful authentication (EMV 3-D Secure value of `05`), `vbv_attempted` if authentication was attempted but did not succeed (EMV 3-D Secure value of `06`), or `vbv_failure` if authentication failed (EMV 3-D Secure value of `07`).
ccAuthService_cavv	Required when payer authentication is successful.

Card-Specific Requirements

Some payment cards require information to be collected during a transaction.

payerAuthEnrollService_defaultCard	Recommended for Discover ProtectBuy.
payerAuthEnrollService_MCC	Required when the card type is Cartes Bancaires.
payerAuthEnrollService_productCode	Required for American Express SafeKey (US) when the product code is `AIR` for an airlinepurchase).
payerAuthEnrollService_merchantName	Required for Visa Secure travel.
shipTo_street1	Required only for American Express SafeKey (US).
shipTo_street2	Required only for American Express SafeKey (US)

Country-Specific Requirements

These fields are required for transactions in specific countries.

payerAuthEnrollService_merchantScore	Required for transactions processed in France.
billTo_city	Required for transactions in US., Canada, and Mainland China.
billTo_postalCode	Required when the **billTo_country** field value is `US` or `CA`.
billTo_state	Required for transactions in US, Canada, and Mainland China.

Endpoint

Set the `payerAuthValidateService_run` and `ccAuthService_run` fields to `true`.
Send the request to `https://ics2ws.ic3.com/commerce/1.x/transactionProcessor`.

Required Fields for Processing an Authorization Using Visa Secure

Use these required fields to process an authorization using Visa Secure.

> **Important**
>
> When using relaxed requirements for address data and the expiration date, not all fields in this list are required. It is your responsibility to determine whether your account is enabled to use this feature and which fields are required. For details about relaxed requirements, see the support article *Relaxed Requirements for Address Data and Expiration Date in Credit Card Transactions*.

billTo_city	
billTo_country	
billTo_email	
billTo_firstName	
billTo_lastName	
billTo_postalCode	
billTo_state	
billTo_street1	
card_accountNumber	
card_expirationMonth	
card_expirationYear	
ccAuthService_cavv	Required when payer authentication is successful. Otherwise, this field is optional.
ccAuthService_commerceIndicatorv	Set this field to one of these values:

	• `vbv`: Successful authentication (EMV 3-D Secure value of `05`). • `vbv_attempted`: Authentication was attempted (EMV 3-D Secure value of `06`). • `vbv_failure`: or `internet`: Authentication failed or was not attempted (EMV 3-D Secure value of `07`)
ccAuthService_run	Set this field to `true`.
ccAuthService_xid	
merchant_referenceCode	
purchaseTotals_currency	
purchaseTotals_grandTotalAmount	

Related Information

- *API Field Reference for the Simple Order API*

Related information
- API Field Reference for the Simple Order API

Optional Fields for Validating Payer Authentication

These fields are optional when validating a Payer Authentication transaction. In certain circumstances, the information provided by an optional field might be required before a transaction can proceed. Those optional fields that are sometimes required are listed in the required fields with the circumstance described.

payerAuthValidateService_credentialEncrypted

payerAuthValidateService_responseAccessToken

payerAuthValidateService_signedPARes

Simple Order Example: Processing an Authorization Using Visa Secure

Request

```
billTo_city=Sao Paulo
billTo_country=BR
billTo_email=julia@example.com
billTo_firstname=Julia
billTo_lastname=Fernandez
billTo_postalCode=01310-000
billTo_state=SP
billTo_street1=R. Augusta
```

```
card_accountNumber=41111111XXXXXXXX
card_expirationMonth=12
card_expirationYear=2023
ccAuthService_run=true
ccAuthService_cavv=ABCDEFabcdefABCDEFabcdef0987654321234567
ccAuthService_commerceIndicator=vbv
ccAuthService_xid=MID23
merchant_referenceCode=Merchant_REF
purchaseTotals_currency=mxn
purchaseTotals_grandTotalAmount=100
```

Response to Successful Request

```
merchantReferenceCode=Merchant_REF
request_id=6461515866500167772420
decision=ACCEPT
reasonCode=100
requestToken=Ahj/7wSR5C/4Icd2fdAKakGLadfg5535r/ghx3Z90AoBj3u
purchaseTotals_currency=mxn
ccAuthReply_cardCategory=F
ccAuthService_reconciliationID=ZUDCXJO8KZRFXQJJ
ccAuthReply_reasonCode=100
ccAuthReply_amount=100.00
ccAuthReply_avsCode=5
ccAuthReply_authorizationCode=570110
ccAuthReply_processorResponse=1
ccAuthReply_authorizedDateTime=2022-03-01T161947Z
ccAuthReply_paymentNetworkTransactionID=111222
```

Website Modification Reference

This section describes how to modify your website to integrate Payer Authentication services into your checkout process. It also provides links to payment card company websites where you can download the appropriate logos.

Website Modification Checklist

Modify web page buttons:

- Order submission button: Disable the final “buy” button until the customer completes all payment and authentication requirements.
- Browser back button: Plan for unexpected customer behavior. Check throughout the authentication process so you do not authenticate transactions twice. Avoid confusing messages, such as warnings about expired pages.

Add appropriate logos:

- Download the appropriate logos of the cards that you support. Place these logos next to the card information entry fields on your checkout pages. For more information about obtaining logos and using them, see *EMV 3-D Secure Services Logos* on page 142.

Add informational message:

- Add a message next to the final “buy” button and the card logo to inform your customers that they might be prompted to provide their authentication password. For examples of messages you can use, see *Informational Message Examples* on page 143.

EMV 3-D Secure Services Logos

This table contains links to payment card company websites from which you can download logos and information about how to incorporate them into your online checkout process.

3-D Secure Services Logos Download Location

EMV 3-D Secure Service	Download Location
Visa Secure	https://usa.visa.com/run-your-business/small-business-tools/payment-technology/visa-secure.html This website contains information about Visa Secure and links to logos for download. The page also contains links to a best practice guide for implementing Visa Secure and a link to a Merchant Toolkit.
Mastercard Identity Check and Maestro	https://brand.mastercard.com/brandcenter.html This website contains information about Identity Check, links to logos for download, and information about integrating the Identity Check information into your website checkout page. For information about Maestro logos, go to: http://www.mastercardbrandcenter.com/us/howtouse/bms_mae.shtml
American Express SafeKey	https://network.americanexpress.com/uk/en/safekey/ This website contains information about SafeKey and links to logos for download.
JCB J/Secure	http://partner.jcbcard.com/security/jsecure/logo.html This website contains information about J/Secure and links to logos for download.
Diners Club ProtectBuy	https://www.dinersclubus.com/home/customer-service Contact Diners Club customer service for assistance.
Discover ProtectBuy	https://www.discovernetwork.com/en-us/business-resources/free-signage-logos This website contains information about Discover ProtectBuy and links to logos for download.

EMV 3-D Secure Service	Download Location
Elo Compra Segura	Contact Elo customer support to obtain log os.
China UnionPay	Contact China UnionPay customer support to obtain logos.

Informational Message Examples

Add a brief message next to the final buy button on your checkout page to inform customers that they might be prompted for their authentication password or to enroll in the authentication program for their card.
These examples might be used, but consult your specific card authentication program to make sure you conform to their messaging requirements.

Example

To help prevent unauthorized use of <card_type> cards online, <your_business_name> participates in <card_authentication_program>. When you submit your order, you might receive a <card_authentication_program> message from your <card_type> card issuer. If your card or issuer does not participate in the program, you are returned to our secure checkout to complete your order. Please wait while the transaction is processed. Do not click the **Back** button or close the browser window.

Example

Your card might be eligible Visa Secure, Mastercard, Maestro, American Express SafeKey, JCB J/Secure, Diners Club ProtectBuy, or Discover ProtectBuy programs. After you submit your order, your card issuer might prompt you to authenticate yourself. This authentication can be done through a one-time pass code sent to your phone or email, by biometrics, or some other form of authentication.

Alternate Methods for Device Data Collection

There are alternate methods for device data collection. You can also use the Payer Authentication Setup service described in *Implementing Direct API Payer Authentication*.

> **Important**
>
> If you are using tokenization, use the Direct API integration method and Payer Authentication Setup service.

Device Data Collection Overview

The device data collection collects the required browser data elements in order to make the EMV 3-D Secure 2.x request and to invoke the EMV 3-D Secure Method URL when it is available.

The Direct API places the required Method URL on the merchant site on behalf of the merchant. Per EMV 3-D Secure requirements, if the issuing bank uses a Method URL, it must run on the merchant site. This is done after a merchant passes in the card number on the POST to the device data collection URL. Options on how to include the BIN are described below.

The Method URL is a concept in the EMV 3-D Secure protocol that enables an issuing bank to obtain additional browser information before starting the authentication session to help facilitate risk-based authentication. The implementation techniques for obtaining the additional browser information are out of scope of the EMV 3-D Secure protocol.

Prerequisites

To support device data collection, you must complete one of these tasks:

- Obtain access to the card BIN (first eight digits or full card number of cardholder).
- Create an iframe on your website and send a POST request to the device data collection URL.

Endpoints

- Staging: `https://centinelapistag.cardinalcommerce.com/V1/Cruise/Collect`
- Production: `https://centinelapi.cardinalcommerce.com/V1/Cruise/Collect`

Collecting Device Data

The following options are available for device data collection:

- Card BIN in JWT: This option is the recommended approach and allows you to pass the card BIN (first eight digits or full card number) in the JWT.
- Card BIN as a POST parameter plus JWT: This option allows you to pass the card BIN directly from the web front end to the device data collection URL instead of the JWT. However, a JWT is still required in order to authenticate the session.

Card BIN in JWT

As part of the JWT generation, you add the card BIN to the payload within the transactional JWT. When the device data collection URL is invoked, the transactional JWT is sent to the URL.
The following example shows the return URL populated in the transactional JWT instead of a POST parameter.

1. Add the card BIN (first eight digits or full card number) to the transactional JWT.
2. Create a POST request to send the transactional JWT to the device data collection URL.
3. Handle the response from the device data collection URL on the return URL provided within the transactional JWT.

Card BIN in JWT

```
<iframe height="1" width="1" style="display: none;">
<form id="collectionForm" name="devicedata" method="POST" action="https://
centinelapistag.cardinalcommerce.com/V1/Cruise/Collect">
<input type="hidden" name="JWT" value="Transactional JWT generated per specification" />
</form>
<script>window.onload = function() {
// Auto submit form on page load
document.getElementById('collectionForm').submit();
}
</script>
</iframe>
```

Card BIN as a POST Parameter Plus JWT

This option allows you to post the card BIN as a POST parameter along with the transactional JWT. When the device data collection URL is invoked, the transactional JWT and the BIN are posted to the URL.

The following example shows the return URL populated in the transactional JWT along with a POST parameter.

1. Create a POST request to send the transactional JWT and the card BIN (first eight digits or full card number) to the device data collection URL.
2. Handle the response from the device data collection URL on the return URL provided within the transactional JWT.

Card BIN as a POST Parameter Plus JWT

```
<iframe height="1" width="1" style="display: none;">
<form id="collectionForm" name="devicedata" method="POST" action="https://
centinelapistag.cardinalcommerce.com/V1/Cruise/Collect">
<!-- POST Parameters: Bin=First eight digits to full pan of the payment card number. JWT=JWT generated
 per merchant spec -->
<input type="hidden" name="Bin" value="41000000" />
<input type="hidden" name="JWT" value="JWT generated per merchant spec" />
</form>
<script>window.onload = function() {
  // Auto submit form on page load
  document.getElementById('collectionForm').submit();
}
</script>
</iframe>
```

Upgrading Your Payer Authentication Implementation

This section describes how the benefits from upgrading to EMV 3-D Secure 2.x for merchants currently using Payer Authentication services.

Benefits

EMV 3-D Secure 2.x provides these benefits:

- Transactions that are more secure by providing additional data about the customer.
- Backward compatibility. Additional data is automatically sent to issuers as they upgrade to EMV 3-D Secure 2.x.
- Improved user-friendly shopping experience for customers, including frictionless authentication and shorter transaction times.
- Can result in higher authorization rates.
- Easier to upgrade to EMV 3-D Secure 2.2. Version 2.2 includes support for exemptions for PSD2. These exemptions that might allow frictionless authentication, include acquirer/issuer transactional risk assessment; white listing; low value, one leg out, and merchant-initiated transactions. These exemptions will be defined as they become available.

PSD2 Impact

If PSD2 affects you, you must upgrade to EMV 3-D Secure 2.x.
PSD2 requires additional security measures outlined in the Regulatory Technical Standards (RTS) that will apply in the future. PSD2 requires stronger identity checks for online payments, particularly for high-value transactions.

PSD2 means changes for all companies in Europe that deal with payments. Some of the implications for merchants include:

- Requiring two-factor authentication for all electronic payments although there are exemptions to allow a frictionless flow.
- Requiring EMV 3-D Secure e-commerce merchants to integrate dynamic authentication tools (such as EMV 3-D Secure 2.x).

Mandates

PSD2 includes mandates around strong customer authentication (SCA) and exemptions and challenges. For more information on the mandates, go to Cardinal's *consumer authentication demos page*, launch the EMV 3-D Secure information demo and click on the **Country Mandates** button at the upper right of the page.

Recommended Integration

Two types of integration are available for EMV 3-D Secure 2.x:

- Direct API
- SDK integration for your mobile application

If you are currently using Payer Authentication services in your business processes and need to upgrade to EMV 3-D Secure 2.x, we recommend using the Direct API integration. The Direct API integration most closely resembles the current process in which you request the Enrollment Check service to verify that the customer is enrolled in one of the card authentication programs and receive a response. With EMV 3-D Secure 2.x, that response includes a new value, the processor transaction ID.
For enrolled cards, include the Access Control Server (ACS) URL, payload, and processor transaction ID to proceed with the authentication session. Then, request the validation service, sending the processor transaction ID with your request, and receive a response with the e-commerce indicator and Cardholder Authentication Verification Value (CAVV) or Account Authentication Value (AAV).
For more information about the Direct API, see *Implementing Direct API for Payer Authentication* on page 24.
For details about the other integrations, see *Implementing SDK Payer Authentication* on page 51.

Important

If you are using tokenization, use the Direct API integration method for Payer Authentication.

Migrating from EMV 3-D Secure 1.x to 2.x FAQ

Q: Is a new JWT required for each transaction?
A: Yes, even though the JWT does not expire for two hours, you should send a new JWT with each new transaction.
Q: How do you link the device data to the transaction-level data?
A: There are two ways:

- You can create a reference ID in the original JWT and then pass that same value for the **payerAuthEnrollService_referenceID** request field for the Check Enrollment service.
- You can use the session ID returned from Payments.setupComplete for the **payerAuthEnrollService_referenceID** request field for the Check Enrollment service.

Q: When will the Payer Authentication reports include the new fields for EMV 3-D Secure 2.x?
A: They will be added in a future release.
Q: Will my current implementation continue to work while I am implementing and testing the newer version in parallel?
A: Yes, current implementation will continue to work.
Q: What testing should I conduct to ensure that my code is working correctly?
A: Use the test cases (*Test Cases for 3-D Secure 2.x* on page 70) to test your preliminary code and make the appropriate changes.
Q: How does EMV 3-D Secure 2.x authentication improve the experience for a customer who uses a mobile or tablet device?
A: EMV 3-D Secure 2.x works the same for each device, and you have control over the formatting of the authentication form. EMV 3-D Secure 2.x also supports newer, more secure authentication delivery tools, such as a one-time password (OTP) sent to a customer's mobile device or email.

Payer Authentication Transaction Details in the Business Center

This section describes how to search the Business Center for details of Payer Authentication transactions. Transaction data is stored for 12 months so that you can retrieve and send the data to payment card companies, if necessary.

Payer Authentication Search

You can search for transactions that used the payer authentication and card authorization services. When searching for transactions, consider the following:

- Search options:
  - Use the "PA Transaction ID" as a search parameter to find both parts of a transaction processed with an enrolled card.
  - The list of applications is simplified to facilitate searching for the relevant service requests.
  - Payer authentication information is available for 12 months after the transaction date.
- Search results: the results options include the Payer Authentication transaction ID and the customer's account number (PAN). Use the Payer Authentication transaction ID to find all parts of the transaction.
- Payer authentication details: all transaction details are discussed under Searching for Payer Authentication Details.

Storing Payer Authentication Data

Payment card companies permit only a certain number of days between the payer authentication and the authorization requests. If you settle transactions that are older

than the predetermined number of days, payment card companies might require that you send them the AAV, CAVV, or the XID when a chargeback occurs. The requirements depend on the card type and the region. For more information, refer to your agreement with your payment card company. After your transactions are settled, you can also use this data to update the statistics of your business.

Searching for Payer Authentication Details

The payer authentication data that is returned in API response fields can be searched by using the Transaction Search feature in the Business Center.
With other services, green means success, red means failure, and black means that the service request did not run. The result of the enrollment check is interpreted differently:

- If the application result appears in green, you do not need to authenticate the user. You can authorize the card immediately.
- If the application result appears in red, it means that authentication failed.
- If the application result appears in yellow, it means the transaction requires authentication.

Enrolled Card

Enrolling a card consists of two steps:

1. Checking for enrollment.
2. Authenticating the customer.

Enrollment Check

For the enrollment check for an enrolled card, payer authentication data is located in the Transaction Search Details window in these sections:

- Request Information section: The enrollment check service is shown in red because the card is enrolled. You receive the corresponding response information. If the card authorization service was requested at the same time, it did not run and appears in black.
- Order Information section: When authentication is required, American Express SafeKey requires that you save the XID for use later. You do not receive an ECI, AAV, or CAVV because the authentication is not complete.

If CAVV and ECI are not provided, and the enrollment transaction results in a challenge, authentication is required.

Authentication Validation

For a transaction in which the validation and the card authorization services were processed successfully, payer authentication data is located in the Transaction Search Details window in these sections:

- Request Information section: The validation service succeeded. A reason code 100 was returned with the corresponding response message. The necessary payer authentication information was passed to the card authorization service, which processed successfully. Both services are shown in green.
- Order Information section: You received a value for all three parameters because the validation was successful. You may not receive an ECI value when a system error prevents the card issuer from performing the validation or when the cardholder does not complete the process.

Card Not Enrolled

When the card is not enrolled, the result of the enrollment check service appears in green, and the card authorization request (if requested at the same time) proceeds normally.

Transaction Details

For a transaction in which the card is not enrolled, payer authentication data is located in the Transaction Search Details window in these sections:

- Request Information section: the service appears in green. You can obtain additional information about related orders by clicking the link on the right.
- Order Information section: the detailed information for the authorization service:
  - For Mastercard, the ECI value is 00: Authentication is not required because the customer's Mastercard card is not enrolled. Other cards will have an ECI value of 07.
  - The AAV/CAVV area is empty because you receive a value only when the customer is authenticated.
  - The XID area is empty because the card is not enrolled.

Payer Authentication Reports

This section describes the Payer Authentication reports that you can download from the Business Center.
All reports on the production servers are retained for 16 months, but the transaction history is only kept in the database for six months. All reports on the test servers are deleted after 60 days. Only transactions that were processed are reported. Those transactions that resulted in a system error or a time-out are not reported.
To get access to the reports, you must file a support ticket in the Support Center.

Payer Authentication Summary Report

This daily, weekly, and monthly summary report indicates the performance of the enrollment and validation services as a number of transactions and a total amount for groups of transactions. The report provides this information for each currency and type of card that you support. You can use this information to estimate how payer authentication screens your transactions: successful, attempted, and incomplete authentication. The cards reported are Visa, Mastercard, Maestro, American Express, JCB, Diners Club, Discover, China UnionPay, and Elo. This daily report is generally available by 7:00 a.m. EST. Data in this report remains available for 6 months.

Downloading the Report

To view the Payer Authentication Summary report:

1. In the left navigation panel, click the **Reports** icon.
2. Under Transaction Reports, click Payer Auth Summary. The Payer Auth Summary Report page appears.
3. In the search toolbar, select the Date Range you want to include in the report. Account level users must select a merchant as well.

4. Based on the date range selected, choose the specific day, week, or month you want to review.

Only months that have already occurred in the current year display in the Month list. To view all months of a previous year, select the year first, then choose the desired month. To view results before the selected period, below the search toolbar, click **Previous**. Click **Next** to see the previous period.

Matching the Report to the Transaction Search Results

The image below shows the search results that contain the transactions that appear in the report. For more information on search results, see *Searching for Payer Authentication Details* on page 151.

Payer Authentication Report Details

Mar 30 2020				
ubcvp1_2 Mar 30 2020 03:42:16 PM	1437540121000167904064 1143754012100	PATRICK MCMAHON null@cybersource.com	1.00 USD 0771	Credit Card Authorization Payer Authentication Validation
ubcvp1_2 Mar 30 2020 03:41:17 PM	1437543646410167904065 1143754364636	P MAN null@cybersource.com	101.00 USD 0771	Credit Card Authorization Payer Authentication Validation
ubcvp1_2 Mar 30 2020 03:40:09 PM	1437538846880167904064 1143753884687	PATRICK MCMAHON null@cybersource.com	16.00 USD 0771	Credit Card Authorization Payer Authentication Validation

Interpreting the Report

A report heading shows the title, the ID of the user who downloaded the report, the merchant ID, and the date or date range of the report. The report is organized by card type. In each section, currencies are reported alphabetically. For each currency, a summary of your payer authentication validation results displayed as total amount and number of transactions.

Payer Authentication Report Interpretation

Card Type	Interpretation	Protected?	Commerce Indicator	ECI
Visa, American Express, and JCB	No authentication	No	Internet	7
	Recorded attempt to authenticate	Yes	VbV, Desk, or JS Attempted	6
	Successful authentication	Yes	VbV, JS, or Aesk	5
Mastercard and Maestro	No authentication	No	Internet**	7*
	Recorded attempt to authenticate	Yes	SPA	1

Card Type	Interpretation	Protected?	Commerce Indicator	ECI
	Successful authentication	Yes	SPA	2
Diners Club and Discover	No authentication	No	Internet	7
	Recorded attempt to authenticate	Yes	PB or DIPB Attempted	6
	Successful authentication	Yes	PB or DIPB	5
China UnionPay, and Elo	No authentication	No	Internet	7
	Recorded attempt to authenticate	Yes	CS or Up3ds Attempted	6
	Successful authentication	Yes	CS or Up3ds	5

* Although the report heading is 7, you receive a collection indicator value of 1, or the response field is empty.
** Although the report heading is Internet, you receive `spa_failure` in the commerce indicator response field.
Transactions are divided into two groups: those for which you are protected and those for which you are not protected:

- For Visa, American Express, JCB, Diners Club, Discover, China UnionPay, and Elo: liability shift for VbV and VbV attempted.
- For Mastercard and Maestro: liability shift only for SPA.
- For all other results: no liability shift.

Comparing Payer Authentication and Payment Reports

There might be differences between the Payer Authentication report and the payment reports because an authenticated transaction might not be authorized.
The values (amounts and counts) in the Payer Authentication report might not match exactly your other sources of reconciliation. This report shows the transactions validated by payer authentication. There might be a different number of transactions that were authorized. Reconciliation discrepancies are more likely if you process your authorizations outside of <keyword keyref="company"/>.
The amounts and numbers can be higher in the Payer Authentication report than in the payment reports. In this example, it shows the results of the first two numbers in the Payer Authentication report and the last one in the payment reports.

To reconcile your reports more easily when using payer authentication, we recommend that you attempt to authenticate the same amount that you want to authorize.

Payer Authentication Reports Compared to Payment Reports

For 10,000 orders, you might receive these results:

- 9900 successful enrollment checks (Payer Authentication report)
- 9800 successful authentication checks (Payer Authentication report)
- 9500 successful authorization checks (Payment report)

Payer Authentication Detail Report

This section describes the elements of the Payer Authentication Detail report. Refer to the Business Center Reporting User Guide for instructions for downloading the report and additional report information. For more information about the

Report Element

The `Report` element is the root element of the report.

```
<Report>
<PayerAuthDetails>
 (PayerAuthDetail+)
</PayerAuthDetails>
</Report>
```

Child Elements of Report

Element Name	Description
PayerAuthDetail	Contains the transaction in the report. For a list of child elements, see *<PayerAuthDetail>*.

PayerAuthDetails Element

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE Report SYSTEM "https://api.cybersource.com/reporting/v3/dtds/padr">
<PayerAuthDetails>
 <PayerAuthDetail>
  ...
 </PayerAuthDetail>
</PayerAuthDetails>
```

PayerAuthDetail Element

The `PayerAuthDetail` element contains information about a single transaction.

```
<PayerAuthDetail>
 (RequestID)
 (MerchantID)
```

```
  (RequestDate)
  (TransactionType)
  (ProofXML)?
  (VEReq)?
  (VERes)?
  (PAReq)?
  (PARes)?
  (AuthInfo)?
</PayerAuthDetail>
```

Child Elements of PayerAuthDetail

Element Name	Description	Type & Length
RequestID	Unique identifier generated for the transac tion. This field corresponds to therequestID API field.	Numeric (26)
MerchantID	Merchant ID used for the transaction.	String (30)
RequestDate	Date on which the transaction was process ed.	DateTime (25)
ProofXML	Data that includes the date and time of the enrollment check and the VEReq and VERes elements. This field corresponds to the**AuthEnrollReply_proofXML** API field. For a list of child elements, see *<ProofXML>*.	String (1024)
VEReq	Verify Enrollment Request (VEReq) is sent by the merchant's server to the directory server. The directory server also sends it to the ACS to determine whether authentication is available for the customer's card number. For a list of child elements, see *<VEReq>*.	
VERes	Verify Enrollment Response (VERes) is sent by the directory server. For a list of child elements, see *<VERes>*.	
PAReq	Payer Authentication Request message tha t you send to the ACS through the payment card company. Corresponds to the payerAuthEnrollReply_paReq API fiel d. For a list of child elements, see <PAReq>.	
PARes	Payer Authentication Response message sent by the ACS. For a list of child elements, see *<PARes>*.	

Element Name	Description	Type & Length
AuthInfo	Address and card verification data. For a list of child elements, see *AuthInfo Element* on page 164.	

PayerAuthDetail Element

```
<PayerAuthDetail>
 <RequestID>0004223530000167905139</RequestID>
 <MerchantID>example_merchant</MerchantID>
 <RequestDate>2020-02-09T08:00:09-08:00</RequestDate>
 <TransactionType>ics_pa_enroll</TransactionType>
 <ProofXML>
  ...
 </ProofXML>
 <VEReq>
  ...
 </VEReq>
 <VERes>
  ...
 </VERes>
 <PAReq>
  ...
 </PAReq>
 <PARes>
  ...
 </PARes>
</PayerAuthDetail>
```

ProofXML Element

The `ProofXML` element contains data that includes the date and time of the enrollment check and the VEReq and VERes elements. This element corresponds to the **payerAuthEnrollReply_proofXML** API field.

```
<ProofXML>
 (Date)
 (DSURL)
 (PAN)
 (AcqBIN)
 (MerID)
 (Password)
 (Enrolled)
</ProofXML>
```

Child Elements of ProofXML

Element Name	Description	Type & Length
Date	Date when the proof XML is generated. (Although the date and time should appear sequentially during all stages of the processing of an order, they might not because of differing time zones and synchronization between servers.)	DateTime (25)
DSURL	URL for the directory server where the proof XML originated.	String (50)
PAN	Customer's masked account number. This element corresponds to the payerAuthEnrollReply_proxyPAN API field.	String (19)
AcqBIN	First six digits of the acquiring bank's identification number.	Numeric (6)
MerID	Identifier provided by your acquirer; used to login to the ACS URL.	String (24)
Password	Merchant's masked authentication password to the ACS; provided by your acquirer. Applies only to cards issued outside the U.S.	String (8)
Enrolled	Result of the enrollment check. This field can contain one of these values: Y: Authentication available. N: Cardholder not participating. U: Unable to authenticate regardless of the reason.	String (1)

ProofXML Element

```
<ProofXML>
  <Date>20200209 08:00:34</Date>
  <DSURL>https:123.456.789.01:234/DSMsgServlet</DSURL>
  <PAN>XXXXXXXXXXXX0771</PAN>
  <AcqBIN>123456</AcqBIN>
  <MerID>4444444</MerID>
  <Password />
  <Enrolled>Y</Enrolled>>
</ProofXML>
```

VEReq Element

The VEReq element contains the enrollment check request data.

```
<VEReq>
  (PAN)
```

```
  (AcqBIN)
  (MerID)
</VEReq>
```

Child Elements of VEReq

Element Name	Description	Type & Length
PAN	Customer's masked account number. This e lement corresponds to the payerAuthEnroll Reply_proxyPAN API field.	String (19)
AcqBIN	First six digits of the acquiring bank's identi fication number.	Numeric (6)
MerID	Identifier provided by your acquirer; used to log in to the ACS URL.	String (24)

VEReq Element

```
<VEReq>
  <PAN>XXXXXXXXXXXX0771</PAN>
  <AcqBIN>123456</AcqBIN>
  <MerID>example</MerID>
</VEReq>
```

VERes Element

The VERes element contains the enrollment check response data.

```
<VERes>
  (Enrolled)
  (AcctID)
  (URL)
</VERes>
```

Child Elements of VERes

Element Name	Description	Type & Length
Enrolled	Result of the enrollment check. This field c an contain one of these values: Y: Authentication available. N: Cardholder not participating. U: Unable to authenticate regardless of the reason.	String (1)
AcctID	Masked string used by the ACS.	String (28)

Element Name	Description	Type & Length
URL	URL of Access Control Server where to send the PAReq. This element corresponds to the payerAuthEnrollReply_acsURL API field.	String (1000)

VERes Element

```
<VERes>
 <Enrolled>Y</Enrolled>
 <AcctID>NDAxMjAwMTAxMTAwMDc3MQ==</AcctID>
 <URL>https://www.example_url.com</URL>
</VERes>
```

PAReq Element

The PAReq element contains the payer authentication request message. This element corresponds to the **payerAuthEnrollReply_paReq** API field.

```
<PAReq>
 (AcqBIN)
 (MerID)
 (Name)
 (Country)
 (URL)
 (XID)
 (Date)
 (PurchaseAmount)
 (AcctID)
 (Expiry)
</PAReq>
```

Child Elements of PAReq

Element Name	Description	Type & Length
AcqBIN	First six digits of the acquiring bank's identification number.	Numeric (6)
MerID	Identifier provided by your acquirer; used to log in to the ACS URL.	String (24)
Name	Merchant's company name.	String (25)
Country	Two-character code for the merchant's country of operation.	String (2)
URL	Merchant's business website.	String

Element Name	Description	Type & Length
XID	Unique transaction identifier generated for each Payment Authentication Request (PAR eq) message. The PARes sent back by the is suing bank contains the XID of the PAReq. T o ensure that both XIDs are the same, comp are it to the XID in the response. To find all r equests related to a transaction, you can al so search transactions for a specific XID.	String (28)
Date	Date and time of request. (Although the da te and time should appear sequentially duri ng all stages of the processing of an order, they may not because of differing time zone s and synchronization between servers.)	DateTime (25)
Purchase Amount	Authorization amount and currency for the transaction. This element corresponds to t he totals of the offer lines or from: ccAuth Reply_amount or purchaseTotals_grandTot alAmount from external data	Amount (15)
AcctID	Masked string used by the ACS.	String (28)
Expiry	Expiration month and year of the customer' s card.	Number (4)

PAReq Element

```
<PAReq>
  <AcqBIN>123456</AcqBIN>
  <MerID>444444</MerID>
  <Name>example</Name>
  <Country>US</Country>
  <URL>http://www.example.com</URL>
  <XID>fr2VCDrbEdyC37MOPfIzMwAHBwE=</XID>
  <Date>2020-02-09T08:00:34-08:00</Date>
  <PurchaseAmount>1.00 USD</PurchaseAmount>
  <AcctID>NDAxMjAwMTAxMTAwMDc3MQ==</AcctID>
  <Expiry>2309</Expiry>
</PAReq>
```

PARes Element

The PARes element contains the payer authentication response.

```
<PARes>
  (AcqBIN)
  (MerID)
```

```
  (XID)
  (Date)
  (PurchaseAmount)
  (PAN)
  (AuthDate)
  (Status)
  (CAVV)
  (ECI)
</PARes>
```

Child Elements of PARes

Element Name	Description	Type & Length
AcqBIN	First six digits of the acquiring bank's identification number.	Numeric (6)
MerID	Identifier provided by your acquirer; used to log in to the ACS URL.	String (24)
XID	XID value returned in the customer authentication response. This element corresponds to the payerAuthEnrollReply_xid and payerAuthValidateReply_xid API fields.	String (28)
Date	Date and time of request. (Although the date and time should appear sequentially during all stages of the processing of an order, they might not because of differing time zones and synchronization between servers.)	DateTime (25)
PurchaseAmount	Authorization amount and currency for the transaction. This element corresponds to the totals of the offer lines or from: ccAuthReply_amount or purchaseTotals_grandTotalAmount from external data.	Amount (15)
PAN	Customer's masked account number. This element corresponds to the payerAuthEnrollReply_proxyPAN API field.	String (19)
AuthDate	Date and time of request.(Although the date and time should appear sequentially during all stages of the processing of an order, they may not because of differing time zones and synchronization between servers.)	DateTime (25)

Element Name	Description	Type & Length
Status	Result of the authentication check. This field can contain one of these values: Y: Customer was successfully authenticated. N: Customer failed or cancelled authentication. Transaction denied. U: Authenticate not completed regardless of the reason. A: Proof of authentication attempt was generated.	String (1)
CAVV	CAVV (Visa, American Express, JCB, Diners Club, Discover, China UnionPay, and Elo) cards = * below) or AAV (Mastercard, and Maestro cards = ** below) returned in the customer authentication response. This element corresponds to the payerAuthValidateReply_cavv (*) and payerAuthValidateReply_ucafAuthenticationData (**) API fields.	String (50)
ECI	Electronic Commerce Indicator returned in the customer authentication response. This element corresponds to the payerAuthValidateReply_eci (*) and payerAuthValidateReply_ucafCollectionIndicator (**) API fields.	Numeric (1)

PARes Element

```
<PARes>
 <AcqBIN>123456</AcqBIN>
 <MerID>4444444</MerID>
 <XID>Xe5DcjrqEdyC37MOPfIzMwAHBwE=</XID>
 <Date>2020-02-09T07:59:46-08:00</Date>
 <PurchaseAmount>1002.00 USD</PurchaseAmount>
 <PAN>0000000000000771</PAN>
 <AuthDate>2020-02-09T07:59:46-08:00</AuthDate>
 <Status>Y</Status>
 <CAVV>AAAAAAAAAAAAAAAAAAAAAAAAAAA=</CAVV>
 <ECI>5</ECI>
</PARes>
```

AuthInfo Element

The `AuthInfo` element contains address and card verification information.

```
<AuthInfo>
 (AVSResult)
 (CVVResult)
</AuthInfo>
```

Child Elements of AuthInfo

Element Name	Description	Type & Length
AVSResult	Optional results of the address verification test.	String (1)
CVVResult	Optional results of the card verification nu mber test.	String (1)

AuthInfo Element

```
<AuthInfo>
 <AVSResult>Y</AVSResult>
 <CVVResult/>git
</AuthInfo>
```

Report Examples

These examples show a complete transaction: the failed enrollment check (enrolled card) and the subsequent successful authentication.
For transactions in India, use *https://ics2ws.in.ic3.com/commerce/1.x/transactionProcessor*.

Failed Enrollment Check

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE Result SYSTEM "https://api.cybersource.com/reporting/v3/dtd/padr">
<Report>
Name="Payer Authentication Detail"
Version="1.0”
xmlns="https://api.cybersource.com/reporting/v3/dtds/padr"
MerchantID="sample_merchant_id"
ReportStartDate="2022-02-09T08:00:00-08:00"
ReportEndDate="2022-02-10T08:00:00-08:00"
<PayerAuthDetails>
  <PayerAuthDetail>
    RequestID=”1895549430000167904548”
    TransactionType=”ics_pa_enroll
    RequestDate=”2022-02-09T08:00:02-08:00”
    <ProofXML>
      <Date>20220209 08:00:34</Date>
      <DSURL>https:123.456.789.01:234/DSMsgServlet</DSURL>
      <PAN>XXXXXXXXXXXX0771</PAN>
      <AcqBIN>123456</AcqBIN>
      <MerID>4444444</MerID>
      <Password />
      <Enrolled>Y</Enrolled>
    </ProofXML>
    <VEReq>
      <PAN>XXXXXXXXXXXX0771</PAN>
      <AcqBIN>123456</AcqBIN>
      <MerID>example</MerID>
    </VEReq>
```

```
    <VERes>
      <Enrolled>Y</Enrolled>
      <AcctID>NDAxMjAwMTAxMTAwMDc3MQ==</AcctID>
      <URL>https://www.sample_url.com</URL>
    </VERes>
    <PAReq>
      <AcqBIN>123456</AcqBIN>
      <MerID>example</MerID>
      <Name>Merchant Name</Name>
      <Country>US</Country>
      <URL>http://www.merchant_url.com</URL>
      <XID>2YNaNGDBEdydJ6WI6aFJWAAHBwE=</XID>
      <Date>2022-02-09T08:00:34-08:00</Date>
      <PurchaseAmount>1.00 USD</PurchaseAmount>
      <AcctID>NDAxMjAwMTAxMTAwMDc3MQ==</AcctID>
      <Expiry>2309</Expiry>
    </PAReq>
  </PayerAuthDetail>
</PayerAuthDetails>
</Report>
```


Successful Authentication


```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE Result SYSTEM "https://api.cybersource.com/reporting/v3/dtd/padr">
<Report>
<PayerAuthDetails>
  <PayerAuthDetail>
    RequestID=”1895549900000167904548”
    TransactionType=”ics_pa_validate”
    XID=”2YNaNGDBEdydJ6WI6aFJWAAHBwE=”
    RequestDate=”2022-02-09T08:00:02-08:00”
    <PARes>
      <AcqBIN>469216</AcqBIN>
      <MerID>6678516</MerID>
      <XID>2YNaNGDBEdydJ6WI6aFJWAAHBwE=</XID>
      <Date>2020-02-09T07:59:46-08:00</Date>
      <PurchaseAmount>1.00 USD</PurchaseAmount>
      <PAN>0000000000000771</PAN>
      <AuthDate>2022-02-09T07:59:46-08:00</AuthDate>
      <Status>Y</Status>
      <CAVV>AAAAAAAAAAAAAAAAAAAAAAAAAAA=</CAVV>
      <ECI>5</ECI>
    </PARes>
  </PayerAuthDetail>
</PayerAuthDetails>
</Report>
```

Reason Codes

This table lists the reason codes that are returned with the response. Cybersource reserves the right to add new reason codes at any time. If your error handler receives a reason code that it does not recognize, it should use the decision field to determine the result.

Reason Codes

Reason Code	Description
100	Successful transaction.
101	The request is missing one or more required fields. Possible action: See the response fields missingField_0 through missingField_N for the missing fields. Resend the request with the complete information.
102	One or more fields in the request contains invalid data. Possible action: See the response fields invalidField_0 through invalidField_N for the invalid fields. Resend the request with the correct information.
150	Error: General system failure. Possible action: Wait a few minutes and resend the request.
151	Error: The request was received, but a server time-out occurred. This error does not include time-outs between the client and the server. Possible action: Wait a few minutes and resend the request.
152	Error: The request was received, but a service time-out occurred. Possible action: Wait a few minutes and resend the request.

Reason Code	Description
234	A problem exists with your <keyword key ref="company"/> merchant configuration. Possible action: Do not resend the request. Contact customer support to correct the configuration problem.
475	The customer is enrolled in payer authentication. Authenticate the cardholder before continuing with the transaction.
476	The customer cannot be authenticated. Possible action: Review the customer's order.

Glossary

3RI Payments
An EMV 3-D Secure request for information. It is an EMVCo term for the EMV 3-D Secure service that can check a BIN without performing a complete authentication.

3-D Secure
Security protocol for online credit card and debit card transactions used by Visa Secure, Mastercard Identity Check, American Express SafeKey, JCB J/Secure, Diners Club ProtectBuy, Discover ProtectBuy, China UnionPay, and Elo.

AAV
Account Authentication Value. A unique 32-character transaction token for a 3-D Secure transaction. For Mastercard Identity Check, the AAV is named the UCAF. For Visa Secure, the AAV is named the CAVV.

acquirer
The financial institution that accepts payments for products or services on behalf of a merchant. Also referred to as “acquiring bank.” This bank accepts or acquires transactions that involve a credit card issued by a bank other than itself.

acquirer BIN
An eight-digit number that uniquely identifies the acquiring bank. There is a different acquirer BIN for for every participating acquirer. The Mastercard BIN starts with 5 and the Visa BIN starts with 4.

acquirer processor
Processor that provides credit card processing, settlement, and services to merchant banks.

ACS
Access Control Server. The card-issuing bank’s host for the payer authentication data.

ACS URL
The URL of the Access Control Server of the card-issuing bank that is returned in the response to the request to check enrollment. This is where you send the PAReq so that the customer can be authenticated.

American Express
A globally issued card type that starts with 3 and which is identified as card type 003. These cards participate in a card authentication service (SafeKey) provided by EMV 3-D Secure.

authentication result
Raw data sent by the card issuer that indicates the status of authentication. It is not required to pass this data into the authorization.

authorization
A request sent to the card issuing bank that ensures a cardholder has the funds available on their credit card for a specific purchase. A positive authorization causes an authorization code to be generated and the funds to be held. Following a payer authentication request, the authorization must contain payer authentication-specific fields containing card enrollment details. If these fields are not passed correctly to the bank, it can invalidate the liability shift provided by card authentication. Systemic failure can result in payment card company fines.

Base64
Standard encoding method for data transfer over the Internet.

BIN
Bank Identification Number. The eight-digit number at the beginning of the card that identifies the card issuer.

CAVV
Cardholder Authentication Verification Value. A Base64-encoded string sent back with Visa Secure-enrolled cards that specifically identifies the transaction with the issuing bank and Visa. Standard for collecting and sending AAV data for Visa Secure transactions. See AAV.

CAVV algorithm
A response passed back when the xPARes status is a `Y` or an `A`.

Compra Segura
Trademarked name for the Elo card authentication service.

CVV
Card Verification Value. Security feature for credit cards and debit cards. This feature consists of two values or codes: one that is encoded in the magnetic strip and one that is printed on the card. Usually the CVV is a three-digit number on the back of the card. The CVV for American Express cards is a 4-digit number on the front of the card. CVVs are used as an extra level of validation by issuing banks.

Diners Club
A globally issued card type that starts with a 3 or a 5. Diners Club cards are identified as card type 005. These cards participate in a card authentication service (ProtectBuy) provided by 3-D Secure.

Directory Servers (DS)
The Visa and Mastercard servers that are used to verify enrollment in a card authentication service.

Discover
Primarily, a U.S. card type that starts with a 6. Discover cards are identified as card type 004. These cards participate in a card authentication service (ProtectBuy) provided by 3-D Secure.

ECI (ECI Raw)
The numeric commerce indicator that indicates to the bank the degree of liability shift achieved during payer authentication processing.

E-Commerce Indicator
Alpha character value that indicates the transaction type, such as MOTO or INTERNET.

Elo
A globally issued card type that starts with a 5. Elo cards are identified as card type of 054. These cards participate in a card authentication service (Compra Segura) provided by 3-D Secure.

Enroll
A type of transaction used for verifying whether a card is enrolled in the Mastercard Identity Check or Visa Secure service.

HTTP
Hypertext Transfer Protocol. An application protocol used for data transfer on the Internet.

HTTP POST request

POST is one of the request methods supported by the HTTP protocol. The POST request method is used when the client sends data to the server as part of the request, such as when uploading a file or submitting a completed form.

HTTPS

Hypertext Transfer Protocol combines with SSL/TLS (Secure Sockets Layer/Transport Layer Security) to provide secure encryption of data transferred over the Internet.

J/Secure

The EMV 3-D Secure program of JCB.

issuer

The bank that issues the credit card.

JCB

Japan Credit Bureau. A globally issued card type that starts with a 3. JCB cards are identified as a card type of 007. These cards participate in a card authentication service (J/Secure) provided by EMV 3-D Secure.

Maestro.

A card brand owned by Mastercard that includes several debit card BINs within the U.K. and in Europe. Merchants who accept Maestro cards online are required to use SecureCode, Mastercard's card authentication service. Maestro cards are identified as 024 and 042 card types. Note that many international Maestro cards are not set up for online acceptance and cannot be used even if they participate in a Mastercard Identity Check authentication program.

Mastercard

A globally issued card that includes credit and debit cards. These cards start with a 5. These cards are identified as card type 002 for both credit and debit cards. These cards participate in a card authentication service (Mastercard Identity Check) provided by 3-D Secure.

Mastercard Identity Check

Trademarked name for Mastercard's payer authentication service.

MD

Merchant-defined Data that is posted as a hidden field to the ACS URL. You can use this data to identify the transaction on its return. This data is used to match the response from the card-issuing bank to a customer's specific order. Although payment card companies recommend that you use the XID, you can also use data such as an order number. This field

is required, but including a value is optional. The value has no meaning for the bank, and is returned to the merchant as is.

Merchant ID

Data that must be uploaded for the Mastercard and Visa card authentication process for each participating merchant. The Merchant ID is usually the bank account number or it contains the bank account number. The data is stored on the Directory Servers to identify the merchant during the enrollment check.

MPI

Merchant Plug-In. The software used to connect to Directory Servers and to decrypt the PARes.

PAN

Primary Account Number. Another term for the credit card number.

PAReq

Payer Authentication Request. Digitally signed Base64-encoded payer authentication request message, containing a unique transaction ID, that a merchant sends to the card-issuing bank. Send this data without alteration or decoding. Note that the field name has a lowercase “a” (PaReq), whereas the message name has an uppercase “A” (PAReq).

PARes

Payer Authentication Response. A compressed, Base64-encoded response from the card-issuing bank. This data is passed for validation.

PARes status

Payer Authentication Response status. One-character length status passed back by Visa and Mastercard that is required data for Asia, Middle East, and Africa Gateway authorizations.

processor

Financial entity that processes payments. Also see acquiring processor.

ProofXML

This field contains the VEReq and VERes for merchant storage. Merchants can use this data for future chargeback repudiation.

ProtectBuy

Trademarked name for the Diners Club and Discover card authentication services.

request ID
A 22- or 23-digit number that uniquely identifies each transaction. Merchants should store this number for future reference.

risk-based authentication
Risk-based authentication is provided by the card-issuing bank. The card-issuing bank gathers a cardholder's transaction data or leverages what data they have to silently authenticate the cardholder based on the perceived degree of risk. They base their risk assessment on factors such as cardholder spending habits, order or product velocity, the device IP address, order amount, and so on.

SafeKey
Trademarked name for the American Express card authentication service. (AESK)

SCMP API
A legacy name-value pair API that was superseded by the Simple Order API.

Simple Order API
An API, which provides three ways to access services: name-value pair (NVP), XML, and SOAP.

TermURL
Termination URL on a merchant's website where the card-issuing bank posts the payer authentication response (PARes) message.

UCAF
Universal Cardholder Authentication Field. A Base64-encoded string sent back with Mastercard Identity Check-enrolled cards specifically identifying the transaction with the issuing bank and Mastercard. Standard for collecting and sending AAV data for Mastercard Identity Check transactions. See AAV.

UCAF collection indicator
Value of `1` or `2` that indicates whether a Mastercard cardholder has authenticated themselves or not.

validate
A service that decodes and decrypts the PARes to determine success. The validate service returns the needed values for authorization.

VEReq
Verify Enrollment Request. Request sent to the Directory Servers to verify that a card is enrolled in a card authentication service.

VERes

Verify Enrollment Response. Response from the Directory Servers to the VEReq.

VERes enrolled

Verify Enrollment Response enrolled. One-character length status passed back by Visa and Mastercard that is required data for Asia, Middle East, and Africa Gateway authorizations.

Visa

A globally issued card that includes credit and debit cards. These cards start with a 4. These cards are identified as card type 001 for both credit and debit cards. These cards participate in a card authentication service (Visa Secure) provided by EMV 3-D Secure.

Visa Secure

(VbV) Trademarked name for Visa's card authentication service.

XID

String used by both Visa and Mastercard, which identifies a specific transaction on the Directory Servers. This string value should remain consistent throughout a transaction's history.