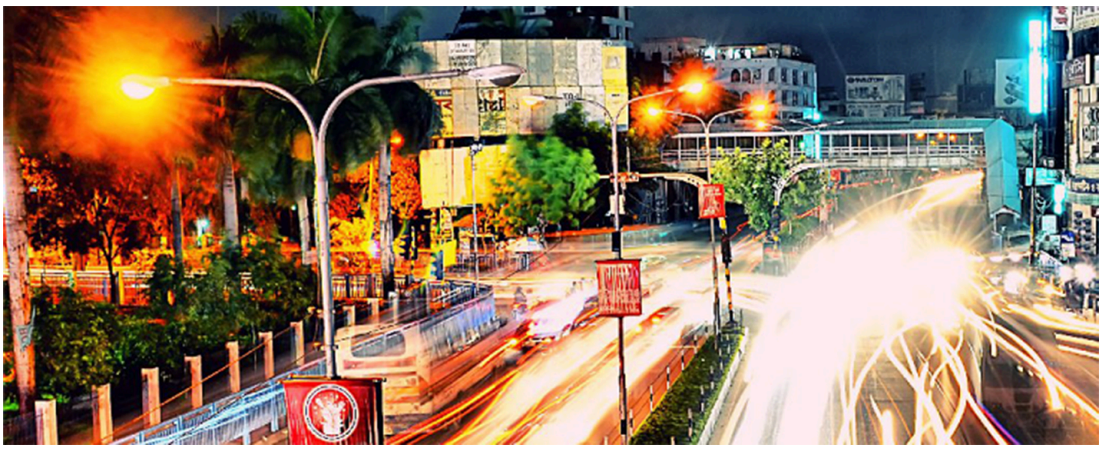
Payer Authentication

Simple Order API

Developer Guide

cybersource
A Visa Solution



Version: 22.02

Contents

Recent Revisions to This Document

22.02

Updated guide

Outdated content was removed and miscellaneous corrections made.

22.01

Added new optional API field names to the Check Enrollment service

See Optional Fields for Enrolling in Payer Authentication (on page 79).

Added MethodURL test case

See Test Case 2.14: Require MethodURL (on page 167).

Updated guide with miscellaneous corrections

Expanded BIN digits

Updated content to indicate that BINs now consist of eight digits instead of six digits.

21.08

Added missing challengeCode API field

Added **challengeCode** API field back to the optional fields for the Enrolled service in the Simple Order and REST versions of the guide after it was left out. See Optional Fields for Enrolling in Payer Authentication (on page 79).

21.07

Added Strong Authentication section.

See Strong Customer Authentication (on page 14).

Added exemption test case section

See Payer Authentication Exemption Test Cases (on page 169)

21.06

Created a REST version of guide.

- Reworked entire guide. Removed response API field descriptions from guide. Refer to API Field Reference Guides for field descriptions.

- Updated the Enrollment Check and Response Fields (on page 21) and Validation Check Response and Card Authorization Request (on page 31) tables.

21.05

API Key Credentials

Updated the process to obtain credentials to generate your API keys for the Cardinal Mobile SDK integration.

Test Case Updates

- Updated Test Case 38: American Express SafeKey Card Enrolled: Successful Authentication to fix a missing credit card digit.
- Added note that XID is not returned for Mastercard transactions.
- Updated Test Case 2.8: Time-Out (Cruise Direct and Hybrid Only) from PARes status = U to VERes enrolled = U.

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 following *Related Documents* section.

Conventions

The following 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 or both.

Related Documentation

Refer to the Support Center for complete technical documentation:

- *Getting Started with Cybersource Advanced for the Simple Order API* describes how to get started using the Simple Order API. (PDF)

- *Decision Manager Developer Guide Using the Simple Order API* describes how to integrate Decision Manager, a fraud detection service, with your order management system. (PDF)
- *Credit Card Services Using the Simple Order API* describes how to integrate payment processing services into your business. (PDF)
- *Digital Accept Hosted Checkout Integration Guide* describes how to create Digital Accept profiles, which enable you to integrate your order management system with the Digital Accept hosted checkout. (PDF)
- *Digital Accept Checkout API Guide* describes how to create Digital Accept profiles, which enable you to integrate your order management system with a website to process transactions. (PDF)
- *Reporting Developer Guide* describes how to view and configure Business Center reports. (HTML)
- The API Versions page provides information about the API versions.
- The *API Field Reference Guide* (HTML) provides information about the individual API fields.

Customer Support

For support information about any service, visit the Support Center:

http://www.cybersource.com/support

Introducing Payer Authentication

Payer Authentication services use front-end JavaScript and back-end API services to provide authentication. Payer Authentication services enable you to add support to your web store for card authentication services, including:

- Visa Secure℠
- Mastercard Identity Check®
- Maestro® (UK Domestic and international)
- American Express SafeKey℠
- JCB J/Secure™
- Diners Club ProtectBuy
- Discover ProtectBuy
- China UnionPay
- Elo Compra Segura

These card authentication services deter unauthorized card use and protect you from fraudulent chargeback activity referred to as liability shift. However, Payer Authentication is not a fraud management service, such as Decision Manager. It is recommended that you implement a comprehensive fraud management program in addition to payer authentication services.

You can use payer authentication services with specific payment processors. To find out if your payment processor supports this feature, see the "Payer Authentication" section in *Credit Card Services Using the SCMP API* (PDF | HTML).

Overview of Chargeback Protection

Visa, Mastercard, Maestro, American Express, JCB, Diners Club, Discover, China UnionPay, and Elo offer chargeback protection if merchants participate in 3-D Secure card authentication programs, such as Visa Secure or Mastercard Identity Check. Chargebacks occur after a transaction is processed, and how they are handled varies according to the region that issued the card. Payment card company rules might vary over time and across geographical regions. Contact your merchant account provider to learn how to interpret chargeback requirements and to discover which chargeback protections are offered.

PSD2

PSD2 (Payment Service Directive 2) is the new European regulatory framework that governs payment processing and customer security and authentication. PSD2 establishes a European Economic Area (EEA) single market for payments to encourage safer and more innovative payment services. PSD2 mandates Strong Customer Authentication (SCA) for all electronic payments. This requirement affects the way merchants receive payments from customers and how customers are authenticated. PSD2 stipulates that two-factor authentication must be applied for all electronic payments.

EMV 3-D Secure

EMV 3-D Secure, often referred to as 3-D Secure 2.x, is the authentication protocol provided by the card networks to support SCA. To comply with SCA, merchants must deploy 3-D Secure 2.x to their checkout page or use a compliant hosted checkout.

Additional data can be passed in now and is automatically sent to issuers as they upgrade to 3-D Secure 2.x. The Payer Authentication service is also backward-compatible with 3-D Secure 1.0.

Strong Customer Authentication

Strong customer authentication (SCA) increases the security of online payments. SCA is required for an online payment when the issuer and acquirer are in the European Economic Area, the U.K., or Gibraltar. To meet SCA requirements, perform payer authentication.

An SCA exemption enables you to bypass SCA requirements. SCA exemptions enable you to balance fraud reduction with a convenient payment experience for the customer. For some transactions, you must obtain approval from your acquirer before using a particular SCA exemption, especially when the acquirer is involved in the calculation of risk. You can request only one SCA exemption for a transaction.

You can request an SCA exemption when you make an authorization request, as documented in the Credit Card Services Developer Guide.

Prerequisites for Implementing Payer Authentication

To use the payer authentication services, you and your developers must be able to complete these tasks:

- Write code to enable a connection to the issuing bank.
- Add JavaScript to your website to facilitate the authentication.

- Add specific data to your API requests.
- Validate the necessary data.
- Provide the additional data to the authorization request.
- Modify your website to help the customer understand the process.

Payer Authentication Process Flow Overview

While each merchant's web browser-based flow will differ according to their individual circumstances, a general overview of the Payer Authentication process is described below.

1. When the cardholder checks out, and ***before*** the **Buy Now** button is pressed, a call is made to the **pa_setup** service. This call generates a JWT and returns a **sessionId** that is used for device collection in the next step.
2. The response from the **pa_setup** service call, returns a JWT and device data collection URL (DCC URL) to which the merchant must POST to the DDC URL in a hidden 10x10 frame including the JWT as a POST parameter; enabling EMVCo and issuer device collection.
3. After device collection is completed, a POST response is sent to the merchant confirming that device collection completed and that the user can press the **Buy Now** button to place the order. Provide an eight-second delay before the **Buy Now** button is enabled, so there is enough time for device collection to complete.
4. Pressing **Buy Now** triggers the **pa_enroll** service request that relays both, the order details (billing, email, phone information, etc.) and the initial **sessionId** returned from the **pa_setup** service. The **pa_enroll** initiates 3-D Secure and checks with the issuing bank to see if a bank challenge/step up requiring the cardholder to provide additional authentication is necessary.
5. The **pa_enroll** response can occur in two different ways:
    - If no step up or challenge is necessary (frictionless flow), the response includes the ECI, CAVV, DSTransactionId, and a PAResStatus. The PAResStatus can be Y or A for successful authentication or N, U, R for failed, unavailable, or rejected authentication. With successful authentication, the data points are added to the authorization message and the order is completed.
    - If the issuer must step up and challenge (friction flow), the response returns an ACS Url, a PAReq payload, a ParesStatus=C, a StepUpURL, a new JWT, and a TransactionId.
6. The JWT and the StepUpUrl received in the **pa_enroll** response, is passed to the client. A POST goes to the StepUpUrl in a viewable iframe with the JWT as a POST parameter to display a bank challenge screen. This enables the cardholder to view and respond to any bank challenge screen.

7. After the cardholder completes the bank challenge by providing additional authentication, a POST is sent to the merchant's returnUrl contained in the JWT. This POST is a notification that the challenge is complete and triggers the merchant to make a **pa_validate** call to obtain the final authentication outcome.

8. The challenge response triggers the **pa_validate** request with the TransactionId. The response to this **pa_validate** request contains the final authentication results including the final ECI, CAVV (if successful), DSTransactionId, ThreeDSVersion, and PAResStatus (Y or A = successful or N, U, R = failed, unavailable, or rejected).

9. If the authentication result is:

    - Successful—Proceed to authorization and append the 3-D Secure data points to the authorization message.

    - Failed, Unavailable, or Rejected—The cardholder is returned to the payment page to attempt payment with a different card.

Integrating Payer Authentication into Your Business

You can integrate payer authentication services into your existing business processes whether you are currently using 3-D Secure 1.0 or you are new to payer authentication.

There are three four types of integration available:

- Cardinal Cruise Direct Connection API (on page 20): This is the most recently developed integration method and is the method that is recommended for most merchants.

- SDK (on page 39): This integration method is used for implementing payer authentication in your Android and iOS mobile applications.

- Hybrid (on page 62): This older integration method is still supported for customers but is not recommended for new customers.

- Standard (on page 195): This older integration method is still supported for customers but is not recommended for new customers.

The SDK integration is designed for 3-D Secure 2.x transactions only; however, support for 3-D Secure 1.0 is available until October 2022. If you are currently using 3-D Secure 1.0, you need to upgrade to 3-D Secure 2.0. (See Upgrading Your Payer Authentication Implementation (on page 189).) Most card networks have announced their intention to stop supporting 3-D Secure 1.0 by October 2022. After that, no transactions can use 3-D Secure 1.0. We recommend updating to the Cardinal Cruise Direct Connection API solution to meet EMVCo 3-D Secure protocols and ensure a smooth experience for your cardholders.

You can also use Digital Accept to enable 3-D Secure 2.x for payer authentication services. For more information, see Using Digital Accept with Payer Authentication (on page 18).

Host Names

The host names that you use depend upon the environment receiving the request. Use a testing environment as a sandbox to test your system configuration by simulating transactions with various payment cards. Use the production environment to process real transactions.

Testing Environment

`https://apitest.cybersource.com`

Production Environment

`https://api.cybersource.com`

Endpoints

All requests to the Payer Authentication resource use a POST to the following endpoints.

Endpoints

Endpoint	Description
`/risk/v1/authentications`	Checks whether a card is enrolled in a card authentication program and if so, an authentication is requested from the issuer.
`/risk/v1/authentication-results`	Retrieves and validates the authentication results from the issuer and enables the merchant to proceed with processing the payment.
`/pts/v2/payments`	Bundles multiple payments together.

Overview of 3-D Secure 2.x Implementation

A broad overview of the process of implementing 3-D Secure 2.x is described below. Depending upon your current business arrangements, some of the steps may not apply to your situation.

1. Contact your account manager or sales manager to discuss how your business can implement 3-D Secure 2.x and PSD2.

2. If you are new to Payer Authentication, set up your merchant ID by contacting customer support to enable 3-D Secure 2.x for the desired card types, currencies, and acquiring bank. For additional details, see Required Merchant Information (on page 19).

3. Log in to the Business Center to obtain the API keys for implementation.

4. Implement 3-D Secure 2.x with the Simple Order API using the Cardinal Cruise Direct Connection API as well as the SDK if you want to implement a native mobile application. SDKs are available for iOS or Android. Implement the SDK to handle authentication steps within the native application. The SDKs are the CardinalCommerce JavaScript equivalent for mobile applications.

5. Configure your system to request the Check Enrollment and Validate Authentication services. Include the required API fields in your request and consider including optional fields based on your business needs. For more information, see the Required Merchant Information (on page 19) section and Implementing Cardinal Cruise Direct Connection API Payer Authentication (on page 20). You can configure your system to request payment services along with your payer authentication for 3-D Secure 2.x, but it is not required.

6. Test your 3-D Secure 2.x services. This testing ensures that you understand the possible use cases as part of implementation. Refer to Testing Payer Authentication Services (on page 88) and run the test cases in Test Cases for 3-D Secure 2.x (on page 139).

7. Configure your account for production by requesting a boarding form from customer support for your processor or acquirer.

8. If you are new to Payer Authentication, complete the boarding form with required information including your merchant ID, your acquirer merchant ID, and BIN information for all chosen card types. For details, see Required Merchant Information (on page 19).

Using Digital Accept with Payer Authentication

Digital Accept offers the ability to enable 3-D Secure 2.x for payer authentication services. You can choose when to upgrade by selecting the option in your Digital Accept profile in the Business Center.

For more information on implementing Digital Accept with payer authentication, see the *Digital Accept Hosted Checkout Integration Guide* (PDF) (HTML) or *Digital Accept Checkout API Guide* (PDF | HTML).

Required Merchant Information

Before using Payer Authentication services in production, you must contact customer support and provide information about your company and your acquiring bank so your account can be configured to implement these services.

You must provide the following information before payer authentication services can be enabled:

Information	Description
About your company	• Your merchant ID. • URL of your company's website, for example: http://www.example.com • Two-character ISO code for your country. • 3-D Secure requestor ID (optional) • 3-D Secure requestor name (optional) • Merchant category code • Name of your bank acquirer. • Complete name and address of your bank contact, including email address.
Bank information	• Name of your bank acquirer. • Complete name and address of your bank contact, including email address.
Visa, Mastercard, Maestro, American Express, JCB, Diners Club, Discover, China UnionPay, and Elo information Acquirer merchant ID	Information provided by your bank acquirer about each payment card company for which you are configured: • Eight-digit BIN numbers. • Acquirer merchant ID: merchant ID assigned by your acquirer. • All currencies that you are set up to process.

Implementing Cardinal Cruise Direct Connection API Payer Authentication

The Cardinal Cruise Direct Connection API supports 3-D Secure 2.x and is backward-compatible with 3-D Secure 1.0 when the issuer, acquirer, or both, are not ready for 2.x. This integration enables you to use an iframe to complete the device profiling and 3-D Secure authentication requirements without including third-party JavaScript directly on your site.

The implementation still requires the use of JavaScript on the page, and it uses CardinalCommerce JavaScript to leverage the authentication. However, the CardinalCommerce JavaScript is hosted and contained in the iframe and does not directly access your web page.

Prerequisites

Notify your account representative that you want to implement payer authentication (3-D Secure) using the Cardinal Cruise Direct Connection API. Provide the merchant ID that you will use for testing. For more information, see Required Merchant Information (on page 19).

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 Services. Testing ensures that your account is configured for production.

Step 1: Payer Authentication Setup Service

The Payer Authentication Setup service must be called on the server side before selecting the button to submit payment. Request the Payer Authentication Setup service separately without including other services.

Request Fields

When requesting the Payer Authentication Setup service, you must send either 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. The request fields may include any of the following:

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

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

Important Response Fields

payerAuthSetupReply_accessToken is used in Step 2: Device Data Collection Iframe (on page 22).

payerAuthSetupReply_deviceDataCollectionURL is used in Step 2: Device Data Collection Iframe (on page 22).

payerAuthSetupReply_referenceID is used in Step 3: Payer Authentication Check Enrollment Service (on page 24).

For further details on examples, see Request and Response Examples.

Step 2: Device Data Collection Iframe

Device Data Collection is initiated on the front end after you receive the server-side Payer Authentication Setup service response as described in Step 1: Payer Authentication Setup Service (on page 21) and pass payerAuthSetupReply_accessToken and DDC URL to the front end.

The hidden pixel iframe is rendered to the browser to profile the customer device. The response depends on the card-issuing bank and can take about eight seconds. If you proceed with the check enrollment service as described in Step 3: Payer Authentication Check Enrollment Service (on page 24) before a response is received, authentication reverts to 3-D Secure 1.0.

Building the Iframe

The iframe has the following parameters.

- Form POST Action: The POST goes to the URL opened within the iframe is from thepayerAuthSetupReply_deviceDataCollectionURL response field discussed in Step 1: Payer Authentication Setup Service (on page 21).
- JWT POST Parameter: Use the value from the payerAuthSetupReply_accessToken response field discussed in Step 1: Payer Authentication Setup Service (on page 21).

Initiate the Device Data Collection Iframe

Initiate a form POST in a hidden 10 x 10 iframe and send it to the device data collection URL. See the following example.

Place the following HTML anywhere inside the `<body>` element of the checkout page. You must dynamically replace the value of the form action attribute and JWT POST parameter with the response values discussed in Step 1: Payer Authentication Setup Service (on page 21)

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.eyJSZWZlcmVuY2VJZCI6ImE0NjVlYzU1LTMwN
TEtNGYwZC05MGE0LWZjMDNlMGE2MWQxOSIsIlJldHVyblVybCI6Imh0dHA6XC9cL2xvY2FsaG9zdDo4MDg
yXC9yZXF1ZXN0LWNhdGNoZXJcL2NhdGNoLXJlcXVlc3QucGhwIiwianRpIjoianRpXzVmMDVkM2VkY2U0M
jYzLjc5MjQwNzMzIiwiaWF0IjoxNTk0MjE3NDUzLCJpc3MiOiI1YjIzZjhjMGJmOWUyZjBkMzQ3ZGQ1YmE
iLCJPcmdVbml0SWQiOiI1NWVmM2YwY2Y3MjNhYTQzMWM5OWI0MzgifQ.Yw9cB9Hdrg71GPL40oAC0g3CVK
YElNGe0uvN9JAaw2E">
</form>
```

Add JavaScript to Submit 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 can also choose to submit the form at a different time, but 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>
```

Listen For 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 containing a message with the status of the device data collection process.

The `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 and 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);
```

Event Listener Callback Payload

This example shows a response payload from the event. None of the data that is returned needs to be stored for future use.

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

Step 3: Payer Authentication Check Enrollment Service

The device collection process must finish before the customer selects the option to buy. You must receive a response before requesting the Check Enrollment service.

Request Fields

The **payerAuthEnrollService_referenceID** field is mapped from the **payerAuthSetupReply_referenceID** field as discussed in Step 1: Payer Authentication Setup Service (on page 21).

payerAuthEnrollService_returnURL is set to the URL where the issuing bank will redirect the customer as discussed in Step 4: Step-Up IFrame (on page 28).

To request the Payer Authentication 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 may include any of the following:

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

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

- **billTo_city**
- **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**
- **payerAuthEnrollService_run**
- **purchaseTotals_currency**
- **purchaseTotals_grandTotalAmount**

You can send additional request data to reduce your issuer step-up authentication rates. It is recommended to send all available fields. You should include the 11 device information fields listed among the optional fields for the Check Enrollment service in your request as a backup, in case, Device Data Collection fails. If a failure does occur, adding these fields ensures a transaction is not downgraded to 3-D Secure 1.0. 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 28) can vary depending on the 3-D Secure version of the transaction (1.0 or 2.x). You can send 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 28)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.

Combining Services

You can use the enrollment check and card authorization services in the same request or in separate requests. Using the same request is recommended.

- Same request: Attempts to authorize the card after authentication are made if step-up payer authentication is not required. In this case, the field values that are required to prove that you attempted to check enrollment are passed automatically to the authorization service. With same request transactions, a different endpoint must be referenced and an additional element must be added to the JSON. If step-up authentication is required, processing stops to allow completion, and authorization is not called. This integration method is recommended.
- Separate requests: You must manually include the enrollment check result values (Enrollment Check Response Fields) in the authorization service request (Card Authorization Request Fields).

Depending on your card type and whether it is a 3-D Secure 1.0 or 2.x transaction, you might not receive the XID. If you receive this field back under a frictionless scenario, it is required for authorization.

The following table lists these fields.

Enrollment Check and Response Fields

Identifier	Enrollment Check Response Field	Card Authorization Request Field
E-commerce indicator (on page 221)	**payerAuthEnrollReply_commerceIndicator**	**ccAuthService_commerceIndicator**
Collection indicator (Mastercard only)	**payerAuthEnrollReply_ucafCollectionIndicator**	**ucaf_collectionIndicator**
CAVV	**payerAuthEnrollReply_cavv**	**ccAuthService_cavv**
AAV	**payerAuthEnrollReply_ucafAuthenticationData**	**ucaf_authenticationData**
XID	**payerAuthEnrollReply_xid**	**ccAuthService_xid**
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

The responses are similar for all card types.

- Enrolled cards: You receive reason code 475 if the customer's card is enrolled in a payer authentication program. When you receive this response, proceed to Step 4: Step-Up IFrame (on page 28).
- Cards not enrolled, or step-up authentication not required: You receive reason code 100 in the following cases:
  - When the account number is not eligible for a payer authentication program or when step-up authentication is not required. The other services in your request are processed normally. If you are making separate enrollment and authorization calls, you must include pertinent payer authentication data in the authorization request to receive liability shift protection.
  - When payer authentication is not supported by the card type. When you receive this response, you can proceed to card authorization. If you receive the authentication results along with reason code 100, you might receive liability shift protection.

An 476 status may occur that requires the merchant to display a card issuer message to the cardholder using the **pa_enroll_pa_cardholder_message** field.

The message text is provided by the ACS/issuer to the cardholder during a frictionless or decoupled transaction to convey information to cardholder. For example, “Additional authentication is needed for this transaction, contact (issuer name) at xxx-xxx-xxxx.”

The entry that appears in the log will be similar to this example:

"cardholderInfo":"You're unable to complete this purchase right now. For help call CommBank on 13 2221"

Important Response Fields

When you receive reason code 475, you also receive the following fields:

- **payerAuthEnrollReply_stepUpUrl** discussed in Step 4: Step-Up IFrame (on page 28).
- **payerAuthEnrollReply_accessToken** discussed in Step 4: Step-Up IFrame (on page 28).

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 24). You need to perform this step only when Step 3 indicates step-up authentication is required.

The iframe manages customer interaction with the card-issuing bank’s Access Control Server (on page 221) and 3-D Secure version compatibility for 3-D Secure 1.0 and 3-D Secure 2.x.

Building the Iframe Parameters

- Form POST Action: The POST is made to the URL within the iframe is from the **payerAuthEnrollReply_stepUpUrl** response field discussed in Step 3: Payer Authentication Check Enrollment Service (on page 24).
- JWT POST Parameter: Use the value from the **payerAuthEnrollReply_accessToken** field discussed in Step 3: Payer Authentication Check Enrollment Service (on page 24).
- MD POST Parameter: Merchant-defined data returned in the response. This field is optional.
- Iframe height and width:

- 3-D Secure 1.0 uses a standard size of 400 by 400 pixels.
- For 3-D Secure 2.x:
  - Use the **payerAuthEnrollService_acsWindowSize** request field to request (but not guarantee) a specific window size.
  - Use the **payerAuthEnrollReply_paReq** response field to determine iframe dimensions by Base64 decoding the string and cross-referencing the Challenge Window Size value with the corresponding size.

The following table lists these values.

Challenge Window Size Value and Corresponding Size

Challenge Window Size Value	Step-Up Iframe Dimensions (Width x Height)
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

Send a POST Request to the Step-Up URL

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

```
<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="eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiJmNmFmMTRmOS04YWRjLTRiNzktO
GVkYS04YWVlMTI2NTkzZTEiLCJpYXQiOjE1OTYwNTEyNzYsImlzcyI6IjVkZDgzYmYwMGU0MjNkMTQ5OGR
jYmFjYSIsImV4cCI6MTU5NjA1NDg3NiwiT3JnVW5pdElkIjoiNTVlZjNmNTZmNzIzYWE0MzFjOTlkNTRiI
iwiUGF5bG9hZCI6eyJBQ1NVcmwiOiJodHRwczovLzBtZXJjaGFudGFjc3N0YWcuY2FyZGluYWxjb21tZXJ
jZS5jb20vTWVyY2hhbnRBQ1NXZWIvY3JlcS5qc3AiLCJQYXlsb2FkIjoiZXlKdFpYTnpZV2RsVkhsd1pTS
TZJa05TWlhFaUxDSnRaWE56WVdkbFZtVnljMmx2YmlJNklqSXVNaTR3SWl3aWRHaHlaV1ZFVTFObGNuWmx
jbFJ5WVc1elNVUWlPaUpsTkdKaVpqazNNeTFqTW1FeUxUUTNOREF0T1RWak5DMWpNVGhoTVRNeE16TmlPR
FFpTENKaFkzTlVjbUZ1YzBsRUlqb2lNVGMzT0RFM016SXRORFk1TVMwME1HUmlMVGxoTkRndE1ESm1OREp
oTlRZd1lqYzVJaXdpWTJoaGJHeGxibWRsVjJsdVpHOTNVMmw2WlNJNklqQXlJbjAiLCJUcmFuc2FjdGlvb
klkIjoiQnh5a0hYVEp4M1JuNHBGWnF1bjAifSwiT2JqZWN0aWZ5UGF5bG9hZCI6dHJ1ZSwiUmV0dXJuVXJ
sIjoiaHR0cHM6Ly9taWNoYWVsdGF5bG9yLmlvL2N5YnMvc3RvcmVEZW1vL3B1YmxpYy9saXN0ZW5lci5we
SJ9.H8j-VYCJK_7ZEHxGz82_IwZGKBODzPaceJNNC99xZRo" /> <input type="hidden" name="MD"
 value="optionally_include_custom_data_that_will_be_returned_as_is”/> </form>
```

Using Javascript to invoke 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. You can also submit the form at a different time, but you must submit it 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 user is redirected back to the **payerAuthEnrollService_returnURL** within the iframe as specified in Step 3: Payer Authentication Check Enrollment Service (on page 24). The payload sent to the returnURL is URL-encoded and Base64-encoded (see the example below). The merchant hosting the returnURL can then close the iframe after redirection.

The response sent back to the return URL contains the following:

- Transaction ID: (**payerAuthEnrollReply_authenticationTransactionID** response field). This is used in Step 5: Payer Authentication Validation Service (on page 31).
- 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 response as discussed in Step 4: Step-Up IFrame (on page 28) make a validation call to verify that the customer successfully authenticated.

Request Fields

The `payerAuthValidateService_authenticationTransactionID` field is mapped from the `payerAuthEnrollReply_authenticationTransactionID` field in Step 4: Step-Up IFrame (on page 28).

For further details on examples, see Validation.

Combining Services or Mapping Authorization Fields

It is recommended that you request both payer authentication and card authorization services at the same time. When you do both services simultaneously, the correct information is automatically sent to your payment processor with the values of these fields being converted to the proper format required by your payment processor:

- E-commerce indicator: **payerAuthValidateReply_commerceIndicator**

- CAVV: **payerAuthValidateReply_cavv**
- AAV: **payerAuthValidateReply_ucafAuthenticationData**
- XID: **payerAuthValidateReply_xid**

If you request the services separately, you must 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 can 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 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, include the UCAF (universal cardholder authentication field) and the collection indicator.

Depending on your card type and whether it is a 3-D Secure 1.0 or 2.x transaction, you might not receive the XID. If you receive this field back, it is required for authorization.

The following table lists these fields.

Identifier	Validation Check Response Field	Card Authorization Request Field
E-commerce indicator	**payerAuthValidateReply_commerceIndicator**	**ccAuthService_commerceIndicator**
Collection indicator (Mastercard only)	**payerAuthValidateReply_ucafCollectionIndicator**	**ucaf_collectionIndicator**
CAVV (Visa and American Express only)	**payerAuthValidateReply_cavv**	**ccAuthService_cavv**
AAV (Mastercard only. Known as UCAF)	**payerAuthValidateReply_ucafAuthenticationData**	**ucaf_authenticationData**
XID	**payerAuthValidateReply_xid**	**ccAuthService_xid**
3-D Secure version	**payerAuthValidateReply_specificationVersion**	**ccAuthService_paSpecificationVersion**
Directory server transaction ID (Not required for 3-D Secure 1.0.)	**payerAuthValidateReply_directoryServerTransactionID**	**ccAuthService_directoryServerTransactionID**

Interpreting the Response

If the authentication is rejected (TransStatus R), Visa, American Express, JCB, Diners Club, Discover, China UnionPay, and Elo recommend that you should not proceed to AuthZ. Instead, you must ask the customer to use another payment method.

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

- Success: A status of reason code 100 indicates other service requests, including authorization, processed normally.
- Failure: A status of reason code 476 indicates the other services in your request did not process.
- 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 when appropriate.

Verify that the enrollment and validation checks are for the same transaction. One way to ensure that the enrollment check and validation responses are identical is by checking if a value such as the XID are the same.

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.
```

Cardinal Cruise Direct Connection Integration Examples

The following examples show a request and response for the Payer Authentication Setup, the Check Enrollment, and Validate Authentication services.

Setup Service Request and Response Examples

The following are examples of a Setup service request that is called after selecting the payment instrument and the corresponding response.

Payer Authentication Setup Service 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=XXXXXXXXXXXXXXX
card_cardType=001
card_expirationMonth=12
card_expirationYear=2030
merchantID=patest
merchantReferenceCode=0001
payerAuthSetupService_run=true
```

Payer Authentication Setup Service Response

```
decision=ACCEPT
merchantReferenceCode=0001
payerAuththSetupReply_deviceDataCollectionURL=https://centinelapistag.cardinalcomm
erce.com/V1/Cruise/Collect
payerAuthSetupReply_reasonCode=100
payerAuthSetupReply_referenceID=f13fe5e0-9b47-4ea1-a03a-ec360f4d0f9f
payerAuthSetupReply_accessToken=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiI1M
Dc4OTI0Mi0zYmEzLTRhZTItYWQwOS1kZjZkODk2NWQ5MjciLCJpYXQiOjE1OTgyOTk1MjQsImlzcyI6IjV
kZDgzYmYwMGU0MjNkMTQ5OGRjYmFjYSIsImV4cCI6MTU5ODMwMzEyNCwiT3JnVW5pdElkIjoiNTVlZjNmM
TBmNzIzYWE0MzFjOTliNWViIiwiUGF5bG9hZCI6eyJBQ1NVcmwiOiJodHRwczovLzBtZXJjaGFudGFjc3N
0YWcuY2FyZGluYWxjb21tZXJjZS5jb20vTWVyY2hhbnRBQ1NXZWIvY3JlcS5qc3AiLCJQYXlsb2FkIjoiZ
XlKdFpYTnpZV2RsVkhsd1pTSTZJa05TWlhFaUxDSnRaWE56WVdkbFZtVnljMmx2YmlJNklqSXVNaTR3SWl
3aWRHaHlaV1ZFVTFObGNuWmxjbFJ5WVc1elNVUWlPaUkzTkRNeVlUWXdNQzA0TXpNMkxUUm1PRGN0WVdKb
E9TMDJObVkzTkRFM01EaGhNV1FpTENKaFkzTlVjbUZ1YzBsRUlqb2lPR0U1TkRkaU9ETXRNRFJpTkMwMFl
tVmlMV0V5WWpNdFpHTmpNV0UxWmprMFlURXlJaXdpWTJoaGJHeGxibWRsVjJsdVpHOTNVMmw2WlNJNklqQ
XlJbjAiLCJUcmFuc2FjdGlvbklkIjoiVEQ1b1MwbzFGQzY1cWF2MHhzeDAifSwiT2JqZWN0aWZ5UGF5bG9
hZCI6dHJ1ZSwiUmV0dXJuVXJsIjoiaHR0cHM6Ly9leGFtcGxlLmNvbS9zdGVwLXVwLXJldHVybi11cmwua
nNwIn0.8wZ8XhLgOIIRvgEUugvYrRAi-efavZTNM0tWInYLTfE
payerAuthSetupReply_reasonCode=100
requestID=5982993692286989203011
```

```
requestToken=AxjzbwSTRFa3h+A4wXZDABEBURwlqraRpAy7gDthk0kyro9JLIYA8AAA2wK2
```

The Payer Authentication APIs are capable of handling encrypted digital payment payloads instead of the payment information. The following is an example of an Payer Authentication Validate request and its corresponding response using Google Pay as the digital payment option.

Check Enrollment Request and Response Examples

The following are examples of a Check Enrollment request that verifies whether a card is enrolled in a card authentication program and its corresponding response.

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=XXXXXXXXXXXXXXX
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
```

Check Enrollment Response

```
decision=REJECT
merchantReferenceCode=0001
payerAuthEnrollReply_accessToken=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiI1
MDc4OTI0Mi0zYmEzLTRhZTItYWQwOS1kZjZkODk2NWQ5MjciLCJpYXQiOjE1OTgyOTk1MjQsImlzcyI6Ij
VkZDgzYmYwMGU0MjNkMTQ5OGRjYmFjYSIsImV4cCI6MTU5ODMwMzEyNCwiT3JnVW5pdElkIjoiNTVlZjNm
```

```
MTBmNzIzYWE0MzFjOTliNWViIiwiUGF5bG9hZCI6eyJBQ1NVcmwiOiJodHRwczovLzBtZXJjaGFudGFjc3
N0YWcuY2FyZGluYWxjb21tZXJjZS5jb20vTWVyY2hhbnRBQ1NXZWIvY3JlcS5qc3AiLCJQYXlsb2FkIjoi
ZXlKdFpYTnpZV2RsVkhsd1pTSTZJa05TWlhFaUxDSnRaWE56WVdkbFZtVnljMmx2YmlJNklqSXVNaTR3SW
l3aWRHaHlaV1ZFVTFObGNuWmxjbFJ5WVc1elNVUWlPaUkzTkRNeVlUWXdNQzA0TXpNMkxUUm1PRGN0WVdK
bE9TMDJObVkzTkRFM01EaGhNV1FpTENKaFkzTlVjbUZ1YzBsRUlqb2lPR0U1TkRkaU9EQXRNRFJpTkMwME
ltVmlMV0V5WWpNdFpHTmpNV0UxWmprMFlURXlJaXdpWTJoaGJHeGxibWRsVjJsdVpHOTNVMmw2WlNJNklq
QXlJbjAiLCJUcmFuc2FjdGlvbklkIjoiVEQ1b1MwbzFGQzY1cWF2MHhzeDAifSwiT2JqZWN0aWZ5UGF5bG
9hZCI6dHJ1ZSwiUmV0dXJuVXJsIjoiaHR0cHM6Ly9leGFtcGxlLmNvbS9zdGVwLXVwLXJldHVybi11cmwu
anNwIn0.8wZ8XhLgOIIRvgEUugvYrRAi-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=eyJtZXNzYWdlVHlwZSI6IkNSZXEiLCJtZXNzYWdlVmVyc2lvbiI6IjIuMi4wIiwidGhyZWVEU1NlcnZlclRyYW5zSUQiOiI3NDMyYTYwMC04MzM2LTRmODctYWJlOS02NmY3NDE3MDhhMWQiLCJhY3NUcmFuc0lEIjoiOGE5NDdiODMtMDRiNC00YmViLWEyYjMtZGNjMWE1Zjk0YTEyIiwiY2hhbGxlbmdlV2luZG93U2l6ZSI6IjAyIn0
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
```

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

Validate Authentication Request and Response with Authorization

The following are examples of a Payer Authentication validate authentication request and its corresponding response.

Validate Authentication Request with Authorization

```
billTo_city=Mountain View
billTo_country=US
```

```
billTo_email=null@cybersource.com
billTo_firstName=John
billTo_lastName=Doe
billTo_postalCode=94043
billTo_state=CA
billTo_street1=1295 Charleston Road
card_accountNumber=XXXXXXXXXXXXXXX
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
```

Validate Authentication Response With Authorization

```
ccAuthReply_amount=30.00
ccAuthReply_authorizationCode=888888
ccAuthReply_authorizedDateTime=2020-08-24T20:06:35Z
ccAuthReply_avsCode=X
ccAuthReply_avsCodeRaw=I1
ccAuthReply_cvCode=M
ccAuthReply_cvCodeRaw=M
ccAuthReply_paymentNetworkTransactionID=123456789619999
ccAuthReply_processorResponse=100
ccAuthReply_reasonCode=100
ccAuthReply_reconciliationID=734426477E432MHS
merchantReferenceCode=0001
payerAuthValidateReply_acsTransactionID=8a947b83-04b4-4beb-a2b3-dcc1a5f94a12
payerAuthValidateReply_authenticationResult=0
payerAuthValidateReply_authenticationStatusMessage=Success
payerAuthValidateReply_cardBin=40000000
payerAuthValidateReply_cardTypeName=VISA
payerAuthValidateReply_cavv=MTIzNDU2Nzg5MDEyMzQ1Njc4OTA=
payerAuthValidateReply_commerceIndicator=vbv
payerAuthValidateReply_directoryServerTransactionID=395fb036-cfc6-462b-b28d-d6ed7c
970cdd
payerAuthValidateReply_eci=05
payerAuthValidateReply_eciRaw=05
payerAuthValidateReply_paresStatus=Y
payerAuthValidateReply_reasonCode=100
payerAuthValidateReply_specificationVersion=2.2.0
```

```
payerAuthValidateReply_threeDSServerTransactionID=7432a600-8336-4f87-abe9-66f74170
8a1d
payerAuthValidateReply_xid=MTIzNDU2Nzg5MDEyMzQ1Njc4OTA=
purchaseTotals_currency=USD
reasonCode=100
requestID=5982995945196028003011
requestToken=Axj/7wSTRFa/iOJ7uYLDABEg3ZtGjJs0bt4rRmymyKaiOEtVlSQFRHCWqypOkDLuAO1eD
STKuj0kshoD5NEVr+I4nu5gsMAA9ACm
```

The following is an example of an Payer Authentication Validate request and its corresponding response using Google Pay as the digital payment option.

Authorization with Enroll

The following are examples of an Payer Authentication Authorization with Enroll request and its corresponding response.

Authorization with Validation

The following are Payer Authentication examples of an Authorization with Validation request and its corresponding response.

Implementing SDK Payer Authentication

This chapter summarizes the process of integrating SDK Payer Authentication services into your mobile application. Payer Authentication services use the Cardinal Mobile SDK for iOS or Android to facilitate the authentication. New SDK versions are frequently released and you should ensure that you stay up-to-date with the current release. One way to stay informed on about new releases is to subscribe to Cardinal's 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 3-D Secure 2.x transactions. If a 3-D Secure 1.0 transaction occurs, you must include functionality to open up a WebView.

Implementation Overview

Notify your account representative that you want to implement payer authentication (3-D Secure). Give them the merchant ID that you will use for testing. For more information, see Required Merchant Information (on page 19).

Implementation tasks include:

- Download, import, and configure the Cardinal 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 the following 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 88).
- Ensure that your account is configured for production.

Process Flow for SDK Integration

The steps that are required to integrate Payer Authentication into a SDK mobile application are described below.

1. Contact Cybersource customer support to register for an API key.
2. Download and import the Cardinal 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 the following information in the Enrollment Check response:
    - E-commerce indicator
    - CAVV (all card types except Mastercard)
    - AAV (Mastercard only)
    - Transaction ID
    - 3-D Secure version
    - Directory server transaction ID
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 Cardinal Mobile SDK displays the authentication window, and the customer enters the authentication information.
10. The bank validates the customer credentials and a 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 58).

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 JSON Web Tokens (JWT) are described in this section.

Credentials/API Keys

API keys are required in order to create the JSON Web Token (JWT). For further information, contact customer support.

You will receive an email with your user name and a temporary password. Your user name will be in the following format:

```
cybersource_merchant name_contact name
```

For example, cybersource_petairways_peter.

Once your 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**.

Implementing SDK Payer Authentication

This chapter summarizes the process of integrating SDK Payer Authentication services into your mobile application. Payer Authentication services use the Cardinal Mobile SDK for iOS or Android to facilitate the authentication. New SDK versions are frequently released and you should ensure that you stay up-to-date with the current release. One way to stay informed on about new releases is to subscribe to Cardinal's 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 3-D Secure 2.x transactions. If a 3-D Secure 1.0 transaction occurs, you must include functionality to open up a WebView.

Using the Android SDK

A Cardinal Mobile SDK is available for integrating Payer Authentication services into mobile applications running on the Android platform. Since this Android SDK only works with 3-D Secure 2.x transactions, you must provide functionality to open a WebView to handle any 3-D Secure 1.0 transactions that occur.

Update 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.**
```

Configure the Android SDK

Get the instance of the cardinal object by *Cardinal.getInstance()*. It is recommended to 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);
}
```

Method	Description	Default Values
setEnableDFSync (boolean enableDFSync)	On setting true, onSetupCompleted is called after device data collected is sent to the server.	False
setEnableQuickAuth (boolean enableQuickAuth)	Sets enable quick auth false.	False
setEnvironment(Setting up Cardinal 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.	" "
setRenderType(org.json.JSONArray renderType)	Sets renderLists all UI 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

Method	Description	Default Values
setUiType(CardinalUiType uiType)	Sets all UI types that the device supports for displaying specific challenge user interfaces within the SDK.	CardinalUiType.BOTH

Set Up the Initial Call

Calling *Cardinal.init()* begins the communication process with Cardinal, authenticates your credentials (server JWT), and completes the data collection process. By the time the customer is ready to check out, all necessary pre-processing is complete. Use 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 the Running Payer Authentication in SDK (on page 52) section for next steps.

Using the iOS SDK

A Cardinal Mobile SDK is available for integrating Payer Authentication services into mobile applications running on the iOS platform. Since this iOS SDK only works with 3-D Secure 2.x transactions, you must provide functionality to open a WebView to handle any 3-D Secure 1.0 transactions that occur.

Download and Import the SDK

Download CardinalMobile.framework

Download the *CardinalMobile.framework* file using cURL in the following example.

```
curl -L -u <USER_NAME>
        :<API_KEY>
 https://cardinalcommerceprod.jfrog.io/artifactory/ios/<VERSION>-<BUILD_NUMBER>/ca
rdinalmobilesdk.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 CardinalMobile.xcframework

Download the *CardinalMobile.xcframework* file using the cURL in the following example.

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

#Example:
curl -L -u UserName:ApiKey
 "https://cardinalcommerceprod.jfrog.io/artifactory/ios/2.2.5-1/CardinalMobileiOSX
C.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.

Set Up 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.

Configure the iOS SDK

CardinalSession new (iOS SDK - Objective-C)

Create a new instance of the cardinal object by *CardinalSession* new. It is recommended to use the default configuration options. See the following examples to complete the iOS SDK configuration.

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

```
#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)
}
```

Method	Description	Default Values	Possible Values
deploymentEnvironment	The environment to which the SDK connects.	CardinalSessionEnvironmentProduction	CardinalSessionEnvironment Staging CardinalSessionEnvironment Production

Method	Description	Default Values	Possible Values
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.	CardinalSessionUITypeBoth	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
proxyServerURL	Proxy server through which the Cardinal SDK Session operates.	nil	
enableQuickAuth	Enable Quick Authentication	false	

Method	Description	Default Values	Possible Values
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	

Set Up the Initial Call

Calling *cardinal session setup* begins the communication process with Cardinal, authenticates your credentials (server JWT), and completes the data collection process. By the time the customer is ready to check out, all necessary pre-processing is complete. Refer to the following code examples for completing 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 in SDK

The payer authentication process in SDK requires that you check 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. Follow the procedures described below 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 now' 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 the following 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.

You can 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: You must manually include the enrollment check result values (Enrollment Check response fields) in the authorization service request (Card Authorization request fields).

These fields are listed in the following table.

Enrollment Check and Response Fields

Identifier	Enrollment Check Response Field	Card Authorization Request Field
E-commerce indicator (on page 221)	**payerAuthEnrollReply_commerceIndicator**	**ccAuthService_commerceIndicator**
Collection indicator (Mastercard only)	**payerAuthEnrollReply_ucafCollectionIndicator**	**ucaf_collectionIndicator**
CAVV	**payerAuthValidateReply_cavv**	**ccAuthService_cavv**
AAV	**payerAuthValidateReply_ucafAuthenticationData**	**ucaf_authenticationData**
XID	**payerAuthEnrollReply_xid** and **payerAuthValidateReply_xid**	**ccAuthService_xid**
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. You must challenge the cardholder and display an authentication challenge window to the cardholder so the cardholder can send a validation request and receive a validation response.

Authenticating Enrolled Cards

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

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

These values identify whether it is a 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.

Call 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 the following 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
 }
```

Call Cardinal session continue (iOS 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 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. The following examples show an example of 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
```

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

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

```
[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!) {

    }

}
```

Cardinal.continue Call in the Same Class (Swift)

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

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

The SDK displays the authentication window if necessary 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 both payer authentication and card authorization services at the same time. When you do this, the correct information is automatically sent to your payment processor and the values of these fields are converted to the proper format required by your payment processor:

- E-commerce indicator (on page 221): `payerAuthEnrollReply_commerceIndicator`
- CAVV (on page 221): `payerAuthValidateReply_cavv`
- AAV (on page 221): `payerAuthValidateReply_ucafAuthenticationData`
- XID (on page 221): `payerAuthEnrollReply_xid` and `payerAuthValidateReply_xid`

If you request the services separately, you must 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 may 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 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, include the UCAF (universal cardholder authentication field) and the collection indicator.

Validation Check and Response Fields

Identifier	Validation Check Response Field	Card Authorization Request Field
E-commerce indicator (on page 221)	`payerAuthValidateReply_commerceIndicator`	**e_commerce_indicator**
Collection indicator (Mastercard only)	`payerAuthValidateReply_ucafCollectionIndicator`	**ucaf_collection_indicator**
CAVV (on page 221) (Visa and American Express only)	`payerAuthValidateReply_cavv`v	`ccAuthService_cavv`
AAV (on page 221) (Mastercard only. Known as UCAF (on page 221))	`payerAuthValidateReply_ucafAuthenticationData`	`ucaf_authenticationData`
XID (on page 221)	`payerAuthValidateReply_xid`	`ccAuthService_xid`
3-D Secure version	`payerAuthValidateReply_specificationVersion`	`ccAuthService_paSpecificationVersion`

Validation Check and Response Fields (continued)

Identifier	Validation Check Response Field	Card Authorization Request Field
Directory server transaction ID (Not required for 3-D Secure 1.0.)	`payerAuthValidateReply_directoryServerTransactionID`	`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 that you receive. 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 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 all messages that display to customers are accurate, complete, and that they address all possible scenarios for enrolled and non-enrolled cards. For example, if the authentication fails, a message such as the following should be displayed to the customer:

```
Authentication Failed
```

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

Hybrid Payer Authentication

If you are just getting started with Payer Authentication, you should use the Cardinal Direct Connection API implementation method. It is the newest and most advanced method of integrating 3-D Secure into your transaction process and is the method the majority of our customers choose to use. The Hybrid implementation is still available when customer support determines that it best fits your company's business needs.

The same prerequisites (on page 41) involving API keys and JSON Web Tokens necessary for the SDK integration are also required for the Hybrid integration. In addition, you need to add JavaScript to your checkout page and setup BIN detection as explained below. Complete these prerequisites before continuing with your Hybrid implementation. Additional information about configuring the Hybrid version of payer authentication is available at Cardinal Cruise documentation.

Implementation Overview

Notify your account representative that you want to implement payer authentication (3-D Secure). Give them the merchant ID that you will use for testing. For more information, see Required Merchant Information (on page 19).

Implementation tasks include:

- Add the JavaScript code to your checkout page
- For each purchase request
    - Build the authentication request
    - Call the Payer Authentication Setup service
    - Allow device collection
    - Handle declines
    - Call Payer Authentication Enroll
    - Call these services:
        - Payer Authentication Validation (only for Hybrid integration)
        - Card Authorization service (optional)

- Use the test cases to test your preliminary code and make appropriate changes. You can change to the test environment by changing the URL in your JavaScript code. See Testing Payer Authentication (on page 88).
- Ensure that your account is configured for production.

Process Flow for Hybrid Integration

1. Call Setup service.
2. Add the JavaScript tag to your checkout page.
3. Call Cardinal.setup (init).
4. Run BIN detection. If the BIN is eligible for 3-D Secure 2.x, it gathers the proper Method URL JavaScript required by the issuer to collect additional device data.
5. User selects the **Submit Payment** button.
6. You 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 the following information in the Enrollment Check response:
    - E-commerce indicator
    - CAVV (all card types except Mastercard)
    - AAV (Mastercard only)
    - Transaction ID
    - 3-D Secure version
    - Directory server transaction ID
8. If the issuing bank requires authentication, you receive a response with the ACS URL of the issuing bank, the payload, and the transaction ID that you include in the *Cardinal.continue* JavaScript call.
9. The JavaScript displays the authentication window, and the customer enters the authentication information.
10. The bank validates the customer credentials, and a JWT is returned that the merchant is required to validate server-side for security reasons.

11. You 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 68).

Payer Authentication Setup

Run the Payer Authentication Setup service on the server side before selecting the button to submit payment. Request the Payer Authentication Setup service without including other services.

When requesting the Payer Authentication Setup service, you must send either 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. The request fields may include any of the following:

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

When the card type is Cartes Bancaires or UPI, the **card_cardType** field is required.

Add the JavaScript

Add Songbird.js to your checkout page and complete the additional steps:

1. **Configure it**: create the configuration object and pass it to *Cardinal.configure()*.
2. **Listen for Events**: subscribe to events with *Cardinal.on()* and set up callback functions for:
    - payments.setupComplete: this optional event triggers when the JavaScript successfully initializes, after calling *Cardinal.setup()*.
    - payments.validated: this event triggers when the transaction completes.
3. **Initialize it**: call *Cardinal.setup()* to trigger and pass your JWT to the JavaScript for each transaction.

To complete these steps, see the JavaScript Documentation.

BIN Detection

BIN detection is required and enables the card-issuing bank's ACS provider to collect additional device data. It speeds up the authentication process by collecting this data before the checkout page launches. This step occurs prior to authentication. For additional information, see the JavaScript Documentation.

Requesting the Check Enrollment Service

Request the Check Enrollment service to verify that the card is enrolled in a card authentication program. The following fields are required:

- **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**

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

You can use the enrollment check and card authorization services in the same request or in separate requests:

- Same request: Cybersource attempts to authorize the card when 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: You must manually include the enrollment check result values (Enrollment Check response fields) in the authorization service request (Card Authorization request fields).

The following table lists these fields.

Enrollment Check and Response Fields

Identifier	Enrollment Check Response Field	Card Authorization Request Field
E-commerce indicator (on page 221)	**payerAuthEnrollReply_commerceIndicator**	**ccAuthService_commerceIndicator**
Collection indicator (Mastercard only)	**payerAuthEnrollReply_ucafCollectionIndicator**	**ucaf_collectionIndicator**
CAVV	**payerAuthValidateReply_cavv**	**ccAuthService_cavv**
AAV	**payerAuthValidateReply_ucafAuthenticationData**	**ucaf_authenticationData**
XID	**payerAuthEnrollReply_xid** and **payerAuthValidateReply_xid**	**ccAuthService_xid**
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

After you receive this response, you can proceed to card authorization.

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. You must challenge the cardholder and display an authentication challenge window to the cardholder so the cardholder can send a validation request and receive a validation response.

Authenticating Enrolled Cards

Cardinal.continue

When you have verified that a customer's card is enrolled in a card authentication program, you must include the URL of the card-issuing bank's Access Control Server (ACS), the payload, and the **payerAuthEnrollReply_authenticationTransactionID** response field in the *Cardinal.continue* function in order to proceed with the authentication session as shown in the following example.

```
Cardinal.continue('cca',
{

 "AcsUrl":"https://testcustomer34.cardinalcommerce.com/merchantacsfrontend/pareq.j
sp?vaa=b&amp;gold=AAAAAAAA...AAAAAAA",

 "Payload":"eNpVUk1zgjAQvedXME7PJEFBVdKt1CECeDkVCk2PcfcnNjv8Kr+7tx4nlbGOcz/se6G1uM
ENPTPeeIz1G37WGEUth7YnpO21TfTvF3wDCBqspQ=="
 },
 {
    "OrderDetails":{
         "TransactionId" :"123456abc"

                        }
        }
);
```

Cardinal.continue displays the authentication window if necessary and automatically redirects the customer's session over to the ACS URL for authentication. The customer's browser displays the authentication window with the option to enter their password.

Decoded Response JWT

Next, *payments.validated* launches, and returns the authentication results and response JWT along with the processor transaction ID as shown in the following example.

```
{
  "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 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 both payer authentication and card authorization services at the same time. When you do so, the correct information is automatically sent to your payment processor and the values of these fields are converted to the proper format required by your payment processor:

- E-commerce indicator: **payerAuthEnrollReply_commerceIndicator**
- CAVV: **payerAuthValidateReply_cavv**

- AAV: **payerAuthValidateReply_ucafAuthenticationData**
- XID: **payerAuthEnrollReply_xid** and **payerAuthValidateReply_xid**

If you request the services separately, you must 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 may 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 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, include the UCAF (universal cardholder authentication field) and the collection indicator.

The following table lists these fields.

Identifier	Validation Check Response Field	Card Authorization Request Field
E-commerce indicator	**payerAuthValidateReply_commerceIndicator**	**ccAuthService_commerceIndicator**
Collection indicator (Mastercard only)	**payerAuthValidateReply_ucafCollectionIndicator**	**ucaf_collectionIndicator**
CAVV (Visa and American Express only)	**payerAuthValidateReply_cavv**	**ccAuthService_cavv**
(Mastercard only. Known as UCAF)	**payerAuthValidateReply_ucafAuthenticationData**	**ucaf_authenticationData**
XID	**payerAuthValidateReply_xid**	**ccAuthService_xid**
3-D Secure version	**payerAuthValidateReply_specificationVersion**	**ccAuthService_paSpecificationVersion**
Directory server transaction ID (Not required for 3-D Secure 1.0.)	**payerAuthValidateReply_directoryServerTransactionID**	**ccAuthService_directoryServerTransactionID**

Interpreting the Response

Important: If a TransStatus R is returned, you should not authorize the transaction. Ask the customer to use another payment method.

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

- Success: You will receive a status of reason code 100, and other service requests, including authorization, are processed normally.
- Failure: You will receive a status of 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 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 all messages that display to customers are accurate, complete, and that they address all possible scenarios for enrolled and unenrolled cards. For example, if the authentication fails, a message such as the following should be displayed to the customer:

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

Hybrid Integration Examples

The following examples show a request and response for the check enrollment service and a request and response for the validate authentication service.

Check Enrollment

The following is an example of an Payer Authentication Check Enrollment request and its corresponding response using the Hybrid implementation.

Validate

The following is an example of an Payer Authentication Validate request and its corresponding response using the Hybrid implementation.

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

Prod URL: https://api.cybersource.com/pts/v2/payments

Enrollment and Authorization

The following is an example of an Payer Authentication Check Enrollment and Authorization request and its corresponding response using the Hybrid implementation.

Validation and Authorization Request

The following is an example of an Payer Authentication Validation and Authorization request and its corresponding response using the Hybrid implementation.

Validate and Authorization Request

```
payerAuthValidateService_run=true
merchantID=patest
merchantReferenceCode=23AEE8CB6B62EE2AF07
item_0_unitPrice=19.99
purchaseTotals_currency=USD
card_expirationMonth=01
card_expirationYear=2020
card_accountNumber=xxxxxxxxxxxxxxx
card_cardType=001
payerAuthValidateService_authenticationTransactionID=
UhGFMeW6IPZbgt9diHK0
referenceID=CybsTester-cc719e84
```

Validate and Authorization Response

```
requestID=0340290070000167905080
merchantReferenceCode=23AEE8CB6B62EE2AF07
purchaseTotals_currency=USD
decision=ACCEPT
reasonCode=100
payerAuthValidateReply_reasonCode=100
payerAuthValidateReply_authenticationResult=0
payerAuthValidateReply_authenticationStatusMessage=Success
payerAuthValidateReply_cavv=Y2FyZGluYWxjb21tZXJjZWF1dGg=
payerAuthValidateReply_commerceIndicator=vbv
payerAuthValidateReply_eci=5
payerAuthValidateReply_eciRaw=05
payerAuthValidateReply_paresStatus=Y
payerAuthValidateReply_reasonCode=100
payerAuthValidateReply_specificationVersion=2.0.1
request_token=AhjzbwSTHCfKtXsaE6elEQJP+BFNcZtIHTiD9au3tjijj5Uar+AuAAAAkAY5
```

API Fields

This section describes the Simple Order API fields that you can use to access Payer Authentication services. The API and client toolkits can be downloaded from the website at the following URL:https://developer.cybersource.com/api/soap-developer-guides.html

Formatting Restrictions

Unless otherwise noted, all field names are case sensitive and all fields accept special characters such as `@`, `#`, and `%`.

The values of the **item_#_** fields must not contain carets (^) or colons (:) because these characters are reserved for use by the services.

Values for request-level and item-level fields must not contain new lines or carriage returns. However, they can contain embedded spaces and any other printable characters. All leading and trailing spaces are removed.

Numbered Elements

The Cybersource XML schema includes several numbered elements. You can include these complex elements more than once in a request. For example, when a customer order includes more than one item, you must include multiple `<item>` elements in your request. Each item is numbered, starting with 0. The XML schema uses an `id` attribute in the item's opening tag to indicate the number. For example:

```
<item id="0">
```

As a name-value pair field name, this tag is called `item_0`. In this portion of the field name, the underscore before the number does not indicate hierarchy in the XML schema. Each item field is generically referred to as `item_#_<element name>` in the documentation.

Below is an example of the numbered `<item>` element and the corresponding name-value pair field names. If you are using the Simple Object Access Protocol (SOAP), the client contains a corresponding item class.

XML Schema Element Names	Corresponding Name-Value Pair Field Names

`<item id="0">`
`  <unitPrice>`
`  <quantity>`
`</item>`
	`item_0_unitPrice`
	`item_0_quantity`

XML Schema Element Names	Corresponding Name-Value Pair Field Names
<item id="1"> <unitPrice> <quantity> </item>	item_1_unitPrice item_1_quantity

Important: When a request in XML format includes an `<item>` element, the element must include an `id` attribute. For example: `<item id="0">`.

Required Fields for Setting Up Payer Authentication

The following fields are required when requesting the Payer Authentication Setup service:

billTo_city

billTo_country

billTo_email

billTo_firstName

billTo_lastName

billTo_postalCode

Required only when the **billTo_country**field is `US` or `CA`.

billTo_street1

card_accountNumber

card_cardType

This field is required when the card type is Cartes Bancaires or UPI.

card_expirationMonth

card_expirationYear

encryptedPayment_data

merchantID

merchantReferenceCode

payerAuthSetupService_run

billTo_city

billTo_country

billTo_email

billTo_firstName

billTo_lastName

billTo_postalCode

Required when the **billTo_country**field is `US` or `CA`.

billTo_street1

card_accountNumber

card_cardType

This field is required when card type is Cartes Bancaires or UPI.

card_expirationMonth

card_expirationYear

encryptedPayment_data

merchantID

merchantReferenceCode

payerAuthSetupService_run

Related information

API Field Reference for the Simple Order API

Optional Fields for Setting Up Payer Authentication

The following fields are optional when requesting the Payer Authentication Setup service:

billTo_state

Required for U.S., Canada, and China. For U.S. and Canada, use the two-character state, province, or territory codes. For China, use the ISO 3166-2 format. When the value is not in the correct format, you may experience authentication errors. It is better to not send a value rather than format the value incorrectly.

card_cardType

Required for the Payer Authentication Check Enrollment Service and Payer Authentication Validation Service. A value must be included in your request. For the Payer Authentication Setup service, the field is required when the card type is Cartes Bancaires.

encryptedPayment_descriptor

recurringSubscriptionInfo_subscriptionID

tokenSource_transientToken

Required Fields for Checking Enrollment in Payer Authentication

These fields are required when requesting the Payer Authentication Enrollment service:

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_city

billTo_country

billTo_email

billTo_firstName

billTo_httpBrowserColorDepth

billTo_httpBrowserJavaEnabled

billTo_httpBrowserJavaScriptEnabled

billTo_httpBrowserLanguage

billTo_httpBrowserScreenHeight

billTo_httpBrowserScreenWidth

billTo_httpBrowserTimeDifference

billTo_ipAddress

billTo_lastName

billTo_postalCode

Required if the **billTo_country** field is `US` or `CA`.

billTo_state

Required for U.S., Canada, and Mainland China. For Mainland China, use the ISO 3166-2 format.

card_accountNumber

card_cardType

card_expirationMonth

card_expirationYear

encryptedPayment_data

item_#_unitPrice

Optional when the **purchaseTotals_grandTotalAmount** field is used.

merchantReferenceCode

payerAuthEnrollService_customerCCAlias

Required if tokenization is enabled in the merchant profile settings.

payerAuthEnrollService_deviceChannel

Required for SDK integration.

payerAuthEnrollService_httpAccept

When the customer's browser provides a value, include that value in your request.

payerAuthEnrollService_httpUserAccept

payerAuthEnrollService_httpUserAgent

When the customer's browser provides a value, include that value in your request.

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_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

payerAuthEnrollService_returnURL

payerAuthEnrollService_run

payerAuthEnrollService_sdkMaxTimeout

Required for 3D Secure 2.x.

purchaseTotals_currency

purchaseTotals_grandTotalAmount

Optional when you use the **item_#_unitPrice** field.

shipTo_city

Required if any shipping address information is included. Required for American Express SafeKey (U.S.).

shipTo_country

Required only for American Express SafeKey (U.S.).

shipTo_postalCode

Required if the **shipTo_country** field value is `US` or `CA`. Required for American Express SafeKey (U.S.).

shipTo_shippingMethod

Required only for American Express SafeKey (U.S.).

shipTo_state

Required if the **shipTo_country** field value is `CA`, `US`, or `Mainland China`. Required for American Express SafeKey (U.S.).

shipTo_street1

Required if any shipping address information is included. Required for American Express SafeKey (U.S.).

shipTo_street2

Required only for American Express SafeKey (U.S.).

Related information

API Field Reference for the Simple Order API

Optional Fields for Enrolling in Payer Authentication

These fields are optional when requesting the Payer Authentication Enrollment service:

airlineData_leg_#_carrierCode

Required for each leg. Required for American Express SafeKey (U.S.) for travel-related requests.

airlineData_leg_#_departureDate

Required for American Express SafeKey (U.S.) for travel-related requests.

The numbered element name must contain 0 instead of #. Payer Authentication services only use the first leg of the trip.

airlineData_leg_#_destination

Required for each leg. Required for American Express SafeKey (U.S.) for travel-related requests.

airlineData_leg_#_originatingAirportCode

Required for American Express SafeKey (U.S.) for travel-related requests.

airlineData_numberOfPassengers

When this field is not included in your request, a default value of `1` is used. Required for American Express SafeKey (U.S.) for travel-related requests.

airlineData_passenger_#_firstName

Required for American Express SafeKey (U.S.) for travel-related requests.

airlineData_passenger_#_lastName

Required for American Express SafeKey (U.S.) for travel-related requests.

billTo_city

billTo_customerAccountChangeDate

Recommended for Discover ProtectBuy.

billTo_customerAccountCreateDate

Recommended for Discover ProtectBuy.

billTo_customerAccountPasswordChangeDate

Recommended for Discover ProtectBuy.

billTo_email

billTo_firstName

billTo_httpBrowserColorDepth

billTo_httpBrowserJavaEnabled

billTo_httpBrowserJavaScriptEnabled

billTo_httpBrowserLanguage

billTo_httpBrowserScreenHeight

billTo_httpBrowserScreenWidth

billTo_httpBrowserTimeDifference

billTo_ipAddress

billTo_lastName

billTo_passportCountry

Recommended for Discover ProtectBuy.

billTo_passportNumber

Recommended for Discover ProtectBuy.

billTo_phoneNumber

billTo_postalCode

billTo_state

Required for U.S., Canada, and Mainland China. For U.S. and Canada, use the two-character state, province, or territory codes. For Mainland China, use the ISO 3166-2 format.

billTo_street1

billTo_street2

billTo_street3

buyerInformation.workPhone

card_accountNumber

card_cardType

card_expirationMonth

card_expirationYear

ccAuthService_paChallengeCode

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_#_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 without consulting with customer support.

payerAuthEnrollService_customerCCAlias

Required when tokenization is enabled in the merchant profile settings.

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_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_merchantName

Required for Visa Secure travel.

payerAuthEnrollService_merchantNewCustomer

payerAuthEnrollService_merchantScore

Required for transactions processed in France.

payerAuthEnrollService_merchantURL

payerAuthEnrollService_messageCategory

payerAuthEnrollService_mobilePhone

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.).

payerAuthEnrollService_recurringEndDate

Required for recurring transactions.

payerAuthEnrollService_recurringFrequency

Required for recurring transactions.

payerAuthEnrollService_recurringOriginalPurchaseDate

When this field is empty, the current date is used.

payerAuthEnrollService_referenceID

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_scoreRequest

payerAuthEnrollService_sdkMaxTimeout

Required for 3-D Secure 2.x. When you do not send a value in this field, the value defaults to `15`.

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

paymentNetworkToken_transactionType

purchaseTotals_currency

requestID

shipTo_city

Required for American Express SafeKey (U.S.).

shipTo_country

Required for American Express SafeKey (U.S.).

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 for American Express SafeKey (U.S.).

shipTo_state

Required when **shipTo_country** value is `CA`, `US`, or `China`. Required for American Express SafeKey (U.S.).

shipTo_street1

Required for American Express SafeKey (U.S.).

shipTo_street2

Required for American Express SafeKey (U.S.).

shipTo_street3

Required for American Express SafeKey (U.S.).

Required Fields for Validating Payer Authentication

These fields are required when requesting the Payer Authentication Validation service:

card_accountNumber

card_cardType

card-expirationMonth

Required only if **card_accountNumber** is included.

card-expirationYear

Required only if **card_accountNumber** is included.

item_#_unitPrice

Optional if the **purchaseTotals_grandTotalAmount** field is used.

merchantID

merchantReference Code

purchaseTotals_grandTotalAmount

Optional if the **item_#_unitPrice** field is used.

payerAuthValidateService_authenticationTransactionID

payerAuthValidateService_run

purchaseTotals_currency

purchaseTotals_grandTotalAmount

Related information

API Field Reference for the Simple Order API

Optional Fields for Validating Payer Authentication

These fields are optional when requesting the Payer Authentication Validation service:

card_expirationMonth

Required when **card_accountNumber** is included.

card_expirationYear

Required when **card_accountNumber** is included.

merchantReferenceCode

payerAuthValidateService_authenticationTransactionID

Required for Hybrid integration.

payerAuthValidateService_credentialEncrypted

payerAuthValidateService_responseAccessToken

Required for Hybrid integration when you use the Cybersource-generated access token.

payerAuthValidateService_signedPARes

The field is in Base64. Remove all carriage returns and line feeds before adding the PARes to the request.

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.

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 2021, use 2024. You also need the minimum required fields for an order.

Enrollment Check

For some of the enrolled cards, an authentication window appears after the enrollment check completes.

To view the authentication window, you must enable your browser to display popup windows.

The test password is 1234.

Depending on the user's action, two results are possible:

- If the user submits the correct password for the enrolled card, authentication is successful.
- If the user clicks the **Exit** link and clicks **OK** in the verification window, authentication does not occur.

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

Response Fields Used in the Enrollment Check Test Cases

Name Used in Test Cases	API Field
ACS URL	**payerAuthEnrollReply_acsURL**
E-commerce indicator	**payerAuthEnrollReply_commerceIndicator**
ECI	
PAReq	

Response Fields Used in the Enrollment Check Test Cases (continued)

Name Used in Test Cases	API Field
proofXML	**payerAuthEnrollReply_proofXML**
Reason code	**payerAuthEnrollReply_reasonCode**
VERes enrolled	**payerAuthEnrollReply_veresEnrolled**
XID	**payerAuthEnrollReply_xid**

Enrollment Check Response Fields

Name Used in Test Cases	API Field
ACS URL	l**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 Test Case 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_authenticationResult**
E-commerce indicator	**payerAuthValidateReply_commerceIndicator**
AAV (Mastercard only)	**payerAuthValidateReply_ucafAuthenticationData**
CAVV ((all card types except Mastercard)	**payerAuthValidateReply_cavv**

Response Fields Used in the Authentication Validation Test Cases (continued)

Name Used in Test Cases	API Field
Collection indicator	**payerAuthValidateReply_ucafCollectionIndicator**
ECI	**payerAuthValidateReply_eci**
PARes status	**payerAuthValidateReply_authenticationStatusMessage**
Reason code	**payerAuthValidateReply_reasonCode**
XID	**payerAuthValidateReply_xid**

Expected Results

These flowcharts summarize the payer authentication process based on the enrollment status of the card and the subsequent customer experience with the authentication path.

Use this information with the test cases to determine how to process orders.

3-D Secure 1.0 Testing

The following test cases can be used to test 3-D Secure authentication for each supported processor.

Visa Secure 3-D Secure 1.0 Test Cases

These test cases can be used to test 3-D Secure authentication with Visa.

Visa Secure Test Cases for 3-D Secure 1.0

You can use Payer Authentication services with 16- and 19-digit Visa cards if they are supported by your processor.

Remove spaces in card numbers when testing.

Possible Values for Visa Secure Response Fields

Result and Interpretation	Validate Authentication Response

Possible Values for Visa Secure Response Fields (continued)

		Authentication Result	ECI	Commerce Indicator	Reason Code
Success	Successful authentication.	0	05	`vbv`	100
	Recorded attempt to authenticate.	1	06	`vbv_attempted`	100
Failure (Customer not responsible)	System error that prevents the completion of authentication: you can proceed with authorization, but there is no liability shift.	6	1	— [2]	100
	Issuer unable to perform authentication.	6	07	`internet`	100
	No response from the Directory Servers or Issuer because of a problem.		07	`internet` `vbv_failure` (processors: AIBMS, Barclays, Streamline, or FDC Germany)	
	Invalid PARes.	-1	—	—	476
Failure (Customer responsible)	Authentication failed or cardholder did not complete authentication. If the authentication fails, Visa suggests that you do not accept the card. You must ask the	9	—	—	476

Possible Values for Visa Secure Response Fields (continued)

	customer to use another payment method.				

1. The ECI value can vary depending on the reason for the failure.
2. A dash (—) indicates that the field is blank or absent.

Test Case 1: Visa Secure Card Enrolled: Successful Authentication

Card Number	445653 00 0000 0007 445653 00 0000 0000 025	With authentication window With 19-digit PAN		
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	100
	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		**ics_pa_validate** service was successful.	
Details	ACS URL	URL	Authentication result	0
	PAReq	PAReq value	CAVV	CAVV value
	proofXML	proofXML value	E-commerce indicator	vbv
	VERes enrolled	Y	ECI	05
	XID	XID value	PARes status	Y
			XID	XID value
Action	1. Add the signed PARes to the Validate Authentication request. 2. Ensure that the XID from the enrollment check matches that from the authentication validation. 3. Add the CAVV and ECI values to your authorization request.			

Test Case 2: Visa Secure Card Enrolled: Successful Authentication but Invalid PARes

Card Number	445653 00 0000 0015	With authentication window		
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	476
	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		We encountered a payer authentication problem: PARes signature digest value mismatch. PARes message has been modified.	
Details	ACS URL	URL value	Authentication result	-1
	PAReq	PAReq value	XID	XID value
	proofXML	proofXML value		
	VERes enrolled	Y		
	XID	XID value		
Action	Do not proceed with authorization. Instead, ask the customer for another form of payment.			

Test Case 4: Visa Secure Card Enrolled: Incomplete Authentication

Card Number	445653 00 0000 0031			
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	100
	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		• Issuer unable to perform authentication. • **ics_pa_validate** service was successful.	
Details	ACS URL	URL value	Authentication result	6

Test Case 4: Visa Secure Card Enrolled: Incomplete Authentication (continued)

	PAReq	PAReq value	E-commerce indicator	internet or vbv_failure
	proofXML	proofXML value	ECI	07
	VERes enrolled	Y	PARes status	U
	XID	XID value	XID	XID value
Action	Ask the customer for another form of payment, or submit the transaction. No liability shift.			

Test Case 5: Visa Secure Card Enrolled: Unsuccessful Authentication

Card Number	445653 00 0000 0023	With authentication window		
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	476
	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		• User failed authentication. • Payer cannot be authenticated.	
Details	ACS URL	URL value	Authentication result	9
	PAReq	PAReq value	PARes status	N
	proofXML	proofXML value	XID	XID value
	VERes enrolled	Y		
	XID	XID value		
Action	You are not permitted to submit this transaction for authorization. Instead ask the customer for another form of payment.			

Test Case 6: Visa Secure Card Enrolled: Unsuccessful Authentication (Customer Exited)

Card Number	445653 00 0000 0023	

Test Case 6: Visa Secure Card Enrolled: Unsuccessful Authentication (Customer Exited) (continued)

Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	476
	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		• Customer prevents authentication. • **ics_pa_validate** service was successful.	
Details	ACS URL	URL value	Authentication result	9
	PAReq	PAReq value	PARes status	N
	proofXML	proofXML value	XID	XID value
	VERes enrolled	Y		
	XID	XID value		
Action	You are not permitted to submit this transaction for authorization. Instead ask the customer for another form of payment.			

Test Case 7: Visa Secure Card Enrolled: Unavailable Authentication

Card Number	445653 00 0000 0064			
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	100		
	ics_pa_enroll service was successful.			
Details	E-commerce indicator	internet or vbv_failure		
	proofXML	proofXML value		
	VERes enrolled	U		
Action	Submit your authorization request. No liability shift.			

Test Case 8: Visa Secure Card Enrolled: Authentication Error

Card Number	445653 00 0000 0098			
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	476
	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		We encountered a payer authentication problem: Error Processing PARes.	
Details	ACS URL	URL value	E-commerce indicator	internet or vbv_failure
	PAReq	PAReq value	ECI	07
	proofXML	proofXML value		
	VERes enrolled	Y		
	XID	XID value		
Action	Ask the customer for another form of payment. No liability shift.			

Test Case 9: Visa Secure Card Not Enrolled

Card Number	445653 00 0000 0056			
Auth. Type	Non-participating bank			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	100		
	ics_pa_enroll service was successful.			
Details	E-commerce indicator	internet		
	ECI	07		
	proofXML	proofXML value		
	VERes enrolled	B		

Test Case 9: Visa Secure Card Not Enrolled (continued)

Action	Submit your authorization request.

Test Case 10: Visa Secure Enrollment Check: Time-Out

Card Number	445653 00 0000 0049			
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	100		
	i **ics_pa_enroll** service was successful.			
Details	E-commerce indicator	internet or vbv_failure		
	proofXML	proofXML value		
Action	After 10-12 seconds, proceed with the authorization request. No liability shift.			

Test Case 11: Visa Secure Enrollment Check Error

Card Number	445653 00 0000 0080	Error response		
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	100		
	ics_pa_enroll service was successful.			
Details	E-commerce indicator	internet or vbv_failure		
	proofXML	proofXML value		
	VERes enrolled	U		

Test Case 11: Visa Secure Enrollment Check Error (continued)

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.

Mastercard Identity Check 3-D Secure 1.0 Test Cases

The following test cases can be used to test 3-D Secure authentication with Mastercard.

Mastercard Identity Check Test Cases for 3-D Secure 1.0

Possible Values for Mastercard Identity Check Response Fields

Result and Interpretation		**Validate Authentication Response**			
		Authentication Result	**ECI**	**Commerce Indicator**	**Reason Code**
Success	Successful authentication.	0	2	`spa`	100
	Recorded attempt to authenticate.	1	1	`spa`	100
	Authentication not completed.	1	0	spa	100
Failure (Customer not responsible)	System error (Issuer unable to perform authentication): you cannot authorize this card; no liability shift.	6	0	`internet`	100
	Invalid PARes.	-1	0		476
Failure (Customer responsible)	Authentication failed or cardholder did not complete authentication.	9	0	–	476

Test Case 16: Mastercard Identity Check Card Enrolled: Successful Authentication

Card Number	520000 00 0000 0007 520000 00 0000 0114	With authentication window Without authentication window		
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	100
	The card is enrolled in payer authentication. Please authenticate before proceeding with authorization.		**ics_pa_validate** service was successful.	
Details	ACS URL	URL	Authentication result	0
	PAReq	PAReq value	AAV	AAV value
	proofXML	proofXML value	Collection indicator	2
	VERes enrolled	Y	E-commerce indicator	spa
	XID	XID value	PARes status	Y
			XID	XID value
Action	1. Add the signed PARes to the Validate Authentication request. 2. Ensure that the XID from the enrollment check matches that from the authentication validation. 3. Add the required payer authentication values to your authorization request.			

Test Case 17: Mastercard Identity Check Card Enrolled: Successful Authentication but Invalid PARes

Card Number	520000 00 0000 0015	With authentication window		
Auth. Type	Active authentication			

Test Case 17: Mastercard Identity Check Card Enrolled: Successful Authentication but Invalid PARes (continued)

Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	476
	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		Payer authentication problem: PARes signature digest value mismatch. PARes message has been modified.	
Details	ACS URL	URL	Authentication result	-1
	PAReq	PAReq value	XID	XID value
	proofXML	proofXML value		
	VERes enrolled	Y		
	XID	XID value		
Action	Do not process the authorization request. Instead ask the customer for another form of payment.			

Test Case 19: Mastercard Identity Check Card Enrolled: Incomplete Authentication

Card Number	520000 00 0000 0031	Without authentication window		
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	100
	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		• **ics_pa_validate** service was successful. • Issuer unable to perform authentication.	
Details	ACS URL	URL value	Authentication result	6
	PAReq	PAReq value	Collection indicator	0
	proofXML	proofXML value	E-commerce indicator	internet
	VERes enrolled	Y	PARes status	U

Test Case 19: Mastercard Identity Check Card Enrolled: Incomplete Authentication (continued)

	XID	XID value	XID	XID value
Action	Ask the customer for another form of payment, or submit the transaction. No liability shift.			

Test Case 20: Mastercard Identity Check Card Enrolled: Unsuccessful Authentication

Card Number	520000 00 0000 0023	With authentication window		
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	476
	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		• User failed authentication • Payer could not be authenticated.	
Details	ACS URL	URL value	Authentication result	9
	PAReq	PAReq value	PARes status	N
	proofXML	proofXML value	Collection indicator	0
	VERes enrolled	Y	XID	XID value
	XID	XID value		
Action	You are not permitted to submit this transaction for authorization. Instead ask the customer for another form of payment.			

Test Case 21: Mastercard Identity Check Card Enrolled: Unsuccessful Authentication (Customer Exited)

Card Number	564182 10 0001 0028			
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	476

Test Case 21: Mastercard Identity Check Card Enrolled: Unsuccessful Authentication (Customer Exited) (continued)

	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		• Customer prevents authentication. • **ics_pa_validate** service was successful.	
Details	ACS URL	URL value	Authentication result	9
	PAReq	PAReq value	PARes status	N
	proofXML	proofXML value	XID	XID value
	VERes enrolled	Y	Collection Indicator	0
	XID	XID value		
Action	You are not permitted to submit this transaction for authorization. Instead ask the customer for another form of payment.			

Test Case 22: Mastercard Identity Check Card Enrolled: Unavailable Authentication

Card Number	520000 00 0000 0064			
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	100		
	ics_pa_validate service was successful.			
Details	Collection indicator	0		
	E-commerce indicator	internet		
	proofXML	proofXML value		
	VERes enrolled	U		
Action	Submit the transaction. No liability shift.			

Test Case 23: Mastercard Identity Check Card Enrolled: Authentication Error

Card Number	520000 00 0000 0098	Without authentication window		
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	476
	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		We encountered a payer authentication problem: Error Processing PARes.	
Details	ACS URL	URL value	Collection indicator	0
	PAReq	PAReq value	E-commerce indicator	internet
	proofXML	proofXML value		
	VERes enrolled	Y		
	XID	XID value		
Action	Ask the customer for another form of payment. No liability shift.			

Test Case 24: Mastercard Identity Check Enrollment Check Time-Out

Card Number	520000 00 0000 0049			
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	100		
	ics_pa_enroll service was successful.			
Details	Collection indicator	0		
	E-commerce indicator	internet		
	proofXML	proofXML value		

Test Case 24: Mastercard Identity Check Enrollment Check Time-Out (continued)

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

Test Case 25: Mastercard Identity Check Enrollment Check Error

Card Number	520000 00 0000 0080			
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	100		
	ics_pa_enroll service was successful.			
Details	Collection indicator	0		
	E-commerce indicator	internet		
	proofXML	proofXML value		
	VERes enrolled	U		
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.			

Maestro 3-D Secure 1.0 Test Cases

The following test cases can be used test 3-D Secure authentication with Maestro.

Maestro Test Cases for 3-D Secure 1.0

Possible Values for Maestro Response Fields

Result and Interpretation	Validate Authentication Response			
	Authentication Result	**ECI**	**Commerce Indicator**	**Reason Code**

Possible Values for Maestro Response Fields (continued)

Result and Interpretation		Validate Authentication Response			
Success	Successful authentication.	0	2	`spa`	100
	Recorded attempt to authenticate.	1	1	`spa`	100
	Authentication not completed.	1	0	spa	100
Failure (Customer not responsible)	System error (Issuer unable to perform authentication): you cannot authorize this card; no liability shift.	6	0	`internet`	100
	Invalid PARes.	-1	0		476
Failure (Customer responsible)	Authentication failed or cardholder did not complete authentication.	9	0	–	476

Test Case 30: Maestro Card Enrolled: Successful Authentication

Card Number	675941 11 0000 0008 675941 00 0000 6404	Without authentication window With authentication window		
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	100
	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		**ics_pa_validate** service was successful.	
Details	ACS URL	URL	Authentication result	0

Test Case 30: Maestro Card Enrolled: Successful Authentication (continued)

	PAReq	PAReq value	AAV	AAV value
	proofXML	proofXML value	Collection indicator	2
	VERes enrolled	Y	E-commerce indicator	spa
	XID	XID value	PARes status	Y
			XID	XID value
Action	1. Add the signed PARes to the validation request. 2. In the response, ensure that the XID from the enrollment check matches that from the validation. 3. Add the required payer authentication values to your authorization request.			

Test Case 31: Maestro Card Enrolled: Successful Authentication but Invalid PARes

Card Number	633110 12 3456 7892	Without authentication window		
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	476
	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		Payer authentication problem: PARes signature digest value mismatch. PARes message has been modified.	
Details	ACS URL	URL	Authentication result	-1
	PAReq	PAReq value	XID	XID value
	proofXML	proofXML value		
	VERes enrolled	Y		
	XID	XID value		
Action	Do not process the authorization request. Instead ask the customer for another form of payment.			

Test Case 32: Maestro Card Enrolled: Attempts Processing - *Deprecated This test case is no longer used*.

Card Number	560000 00 0000 00 0193	Card enrollment option during purchase process		
Auth. Type	Maestro stand in attempts service			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	100
	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		**ics_pa_validate** service was successful.	
Details	ACS URL	URL	Authentication result	1
	PAReq	PAReq value	AAV	AAV value
	proofXML	proofXML value	E-commerce indicator	spa
	VERes enrolled	Y	PARes status	A
	XID	XID value	XID	XID value
Action	This test card enables you to reproduce the process by which the customer enrolls the card during the purchase. If the card is not enrolled, a card enrollment option windows appears in the customer's browser after the enrollment check. The customer can activate the card at that time or later. In both cases, the card is authenticated, and validation is successful. 1. Add the signed PARes to the Validate Authentication request. 2. Ensure that the XID from the enrollment check matches that from the authentication validation. 3. Add the required payer authentication values to your authorization request.			

Test Case 33: Maestro Card Enrolled: Incomplete Authentication

Card Number	633110 12 5035 3227	Without authentication window		
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	

Test Case 33: Maestro Card Enrolled: Incomplete Authentication (continued)

Summary	Reason Code	475	Reason Code	100
	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		Issuer unable to perform authentication.	
Details	ACS URL	URL value	Authentication result	6
	PAReq	PAReq value	Collection indicator	0
	proofXML	proofXML value	E-commerce indicator	spa
	VERes enrolled	Y	PARes status	U
	XID	XID value	XID	XID value
Action	Ask the customer for another form of payment, or submit the transaction. No liability shift.			

Test Case 34: Maestro Card Enrolled: Unsuccessful Authentication

Card Number	633110 06 1019 4313	Without authentication window		
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	476
	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		User failed authentication	
Details	ACS URL	URL value	Authentication result	9
	PAReq	PAReq value	PARes status	N
	proofXML	proofXML value	XID	XID value
	VERes enrolled	Y	Collection Indicator	0
	XID	XID value		

Test Case 34: Maestro Card Enrolled: Unsuccessful Authentication (continued)

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

Test Case 35: Maestro Card Enrolled: Unavailable Authentication

Card Number	633110 02 6697 7839			
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	100		
	ics_pa_enroll service was successful.			
Details	Collection indicator	0		
	E-commerce indicator	spa		
	proofXML	proofXML value		
Action	Submit the transaction. No liability shift.			

Test Case 36: Maestro Card Enrolled: Authentication Error

Card Number	560000 51 1607 57 7094	Without authentication window		
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	476
	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		We encountered a payer authentication problem: Error Processing PARes.	
Details	ACS URL	URL value	Collection indicator	0
	PAReq	PAReq value	E-commerce indicator	internet

Test Case 36: Maestro Card Enrolled: Authentication Error (continued)

	proofXML	proofXML value		
	VERes enrolled	Y		
	XID	XID value		
Action	Do not request authorization. Instead ask the customer for another form of payment. No liability shift.			

Test Case 37: Maestro Enrollment Check Error

Card Number	560000 84 1211 09 2515			
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	100		
	ics_pa_enroll service was successful.			
Details	Collection indicator	0		
	E-commerce indicator	internet		
	proofXML	proofXML value		
	VERes enrolled	U		
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.			

American Express SafeKey 3-D Secure 1.0 Test Cases

The following test cases can be used to test 3-D Secure authentication with American Express.

Possible Values for American Express SafeKey Response Fields

Result and Interpretation		Validate Authentication Response			
		Authentication Result	**ECI**	**Commerce Indicator**	**Reason Code**
Success	Successful authentication.	0	05	`aesk`	100
	Recorded attempt to authenticate.	1	06	`aesk_attempted`	100
Failure (Customer not responsible)	System error that prevents the completion of authentication: you can proceed with authorization, but there is no liability shift.	6	1	— [2]	100
	Issuer unable to perform authentication.	6	07	`internet`	100
	Incomplete or unavailable authentication.		07	`internet`	
	Invalid PARes.	-1	—	—	476
Failure (Customer responsible)	Authentication failed or cardholder did not complete authentication. If the authentication fails, Visa requires that you do not accept the card. You must ask the customer to use another payment method.	9	—	—	100

1. The ECI value can vary depending on the reason for the failure.

Possible Values for American Express SafeKey Response Fields (continued)

Result and Interpretation	Validate Authentication Response
2. A dash (—) indicates that the field is blank or absent.	

Test Case 38: American Express SafeKey Card Enrolled: Successful Authentication

Card Number	340000 00 0003 961 371449 11 1020 228	Without authentication window With authentication window		
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	100
	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		**ics_pa_validate** service was successful.	
Details	ACS URL	URL value	Authentication result	0
	PAReq	PAReq value	CAVV	CAVV value
	proofXML	proofXML value	E-commerce indicator	aesk
	VERes enrolled	Y	ECI	05
	XID	XID value	PARes status	Y
			XID	XID value
Action	1. Add the signed PARes to the Validate Authentication request. 2. Ensure that the XID from the enrollment check matches that from the authentication validation. 3. Add the CAVV and ECI values to your authorization request.			

Test Case 39: American Express SafeKey Card Enrolled: Successful Authentication but Invalid PARes

Card Number	340000

Test Case 39: American Express SafeKey Card Enrolled: Successful Authentication but Invalid PARes (continued)

	00 0006 022			
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	476
	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		We encountered a payer authentication problem: PARes signature digest value mismatch. PARes message has been modified.	
Details	ACS URL	URL value	Authentication result	-1
	PAReq	PAReq value	XID	XID value
	proofXML	proofXML value		
	VERes enrolled	Y		
	XID	XID value		
Action	Do not proceed with authorization. Instead, ask the customer for another form of payment.			

Test Case 41: American Express SafeKey Card Enrolled: Incomplete Authentication

Card Number	340000 00 0002 302	Without authentication window		
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	100
	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		**ics_pa_validate** service was successful.	
Details	ACS URL	URL value	Authentication result	6
	PAReq	PAReq value	E-commerce indicator	internet
	proofXML	proofXML value	ECI	07

Test Case 41: American Express SafeKey Card Enrolled: Incomplete Authentication (continued)

	VERes enrolled	Y	PARes status	U
	XID	XID value	XID	XID value
Action	Ask the customer for another form of payment, or submit the transaction. No liability shift.			

Test Case 42: American Express SafeKey Card Enrolled: Unsuccessful Authentication

Card Number	340000 00 0000 033	Without authentication window		
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	476
	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		• User failed authentication. • Payer cannot be authenticated.	
Details	ACS URL	URL value	Authentication result	9
	PAReq	PAReq value	PARes status	N
	proofXML	proofXML value	ECI	07
	VERes enrolled	Y	XID	XID value
	XID	XID value		
Action	You are not permitted to submit this transaction for authorization. Instead ask the customer for another form of payment.			

Test Case 43: American Express SafeKey Card Enrolled: Unavailable Authentication

Card Number	340000 00 0007 780			
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	100		

Test Case 43: American Express SafeKey Card Enrolled: Unavailable Authentication (continued)

	ics_pa_enroll service was successful.			
Details	E-commerce indicator	internet		
	proofXML	proofXML value		
	VERes enrolled	U		
Action	Submit your authorization request. No liability shift.			

Test Case 44: American Express SafeKey Card Enrolled: Authentication Error

Card Number	340000 00 0009 299			
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	476
	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		We encountered a payer authentication problem: Error Processing PARes.	
Details	ACS URL	URL value	ECI	07
	PAReq	PAReq value	E-commerce Indicator	internet
	proofXML	proofXML value		
	VERes enrolled	Y		
	XID	XID value		
Action	Ask the customer for another form of payment. No liability shift.			

Test Case 45: American Express SafeKey Card Not Enrolled

Card Number	340000 00 0008 135			
Auth. Type	Non-participating bank			
Results	Check Enrollment		Validate Authentication	

Test Case 45: American Express SafeKey Card Not Enrolled (continued)

Summary	Reason Code	100		
	ics_pa_enroll service was successful.			
Details	E-commerce indicator	internet		
	ECI	07		
	proofXML	proofXML value		
	VERes enrolled	N		
Action	Submit the transaction.			

Test Case 46: American Express SafeKey Enrollment Check: Time-Out

Card Number	340000 00 0008 309			
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	100		
	ics_pa_enroll service was successful.			
Details	E-commerce indicator	internet		
	ECI	07		
	proofXML	proofXML value		
Action	After 10-12 seconds, proceed with the authorization request. No liability shift.			

Test Case 47: American Express SafeKey Enrollment Check Error

Card Number	340000 00 0007 244			
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	

Test Case 47: American Express SafeKey Enrollment Check Error (continued)

Summary	Reason Code	100		
	ics_pa_enroll service was successful.			
Details	E-commerce indicator	internet		
	proofXML	proofXML value		
	VERes enrolled	U		
Action	Proceed with the authorization request, and contact your support representative to resolve the issue. If you requested payer authentication and authorization together, the authorization is processed automatically. No liability shift.			

JCB J/Secure 3-D Secure 1.0 Test Cases

The following test cases can be used to test 3-D Secure authentication with JCB.

JCB J/Secure Test Cases for 3-D Secure 1.0

Possible Values for JCB J/Secure Response Fields

Result and Interpretation		**Validate Authentication Response_**			
		Authentication Result	**ECI**	**Commerce Indicator**	**Reason Code**
Success	Successful authentication.	0	05	`js`	100
	Recorded attempt to authenticate	1	06	`js_attempted`	100
Failure (Customer not responsible)	System error that prevents the completion of authentication: you can proceed with authorization, but no liability shift.	6	[1]	—[2]	

Possible Values for JCB J/Secure Response Fields (continued)

Result and Interpretation		Validate Authentication Response_			
	Issuer unable to perform authentication	6	07	`internet`	100

	Incomplete or unavailable authentication.		07	`internet`
				
				`js_failure`	

	Invalid PARes.	-1	00		476
Failure (Customer responsible)	Authentication failed or cardholder did not complete authentication. If the authentication fails, Visa requires that you do not accept the card. You must ask the customer to use another payment method.	9	—	—	476
1 The ECI value can vary depending on the reason for the failure. 2 A dash (—) indicates that the field is blank or absent.					

Test Case 48: JCB J/Secure Card Enrolled: Successful Authentication

Card Number	356999 00 1008 3722	Without authentication window		
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	100
	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		**ics_pa_validate** service was successful.	
Details	ACS URL	URL	Authentication result	0

Test Case 48: JCB J/Secure Card Enrolled: Successful Authentication (continued)

	PAReq	PAReq value	CAVV	CAVV value
	proofXML	proofXML value	E-commerce indicator	js
	VERes enrolled	Y	ECI	05
	XID	XID value	PARes status	Y
			XID	XID value
Action	1. Add the signed PARes to the Validate Authentication request. 2. Ensure that the XID from the enrollment check matches that from the authentication validation. 3. Add the CAVV and ECI values to your authorization request.			

Test Case 49: JCB J/Secure Card Enrolled: Successful Authentication but Invalid PARes

Card Number	356999 00 1008 3748	Without authentication window		
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	476
	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		Payer authentication problem: PARes signature digest value mismatch. PARes message has been modified.	
Details	ACS URL	URL	Authentication result	-1
	PAReq	PAReq value	XID	XID value
	VERes enrolled	Y		
Action	Do not process the authorization request. Instead ask the customer for another form of payment.			

Test Case 50: JCB J/Secure Card Enrolled: Attempted Authentication

Card Number	356996 00 1008 3758	

Test Case 50: JCB J/Secure Card Enrolled: Attempted Authentication (continued)

Auth. Type	Activation during shopping			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	100
	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		**ics_pa_validate** service was successful.	
Details	ACS URL	URL value	Authentication result	1
	PAReq	PAReq value	CAVV	CAVV value
	proofXML	proofXML value	E-commerce indicator	js_attempted
	VERes enrolled	Y	ECI	06
	XID	XID value	PARes status	A
			XID	XID value
Action	If you request Validate Authentication and authorization services separately, follow these steps: 1. Add the signed PARes to the validation request. 2. In the response, ensure that the XID from the enrollment check matches that from the validation. 3. Add the CAVV and ECI values to your authorization request. If you request the Validate Authentication and authorization services together, the process described above occurs automatically.			

Test Case 51: JCB J/Secure Card Enrolled: Incomplete Authentication (Unavailable)

Card Number	354159 99 9810 3643	Without authentication window		
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	100

Test Case 51: JCB J/Secure Card Enrolled: Incomplete Authentication (Unavailable) (continued)

	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		• Issuer unable to perform authentication. • **ics_pa_validate** service was successful.	
Details	ACS URL	URL value	Authentication result	6
	PAReq	PAReq value	E-commerce indicator	internet
	proofXML	proofXML value	ECI	07
	VERes enrolled	Y	PARes status	U
	XID	XID value	XID	XID value
Action	Ask the customer for another form of payment, or submit the transaction. No liability shift.			

Test Case 52: JCB J/Secure Card Enrolled: Failed Authentication

Card Number	356999 01 1008 3721	Without authentication window		
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	476
	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		• User failed authentication. • Payer cannot be authenticated.	
Details	ACS URL	URL value	Authentication result	9
	PAReq	PAReq value	PARes status	N
	proofXML	proofXML value	XID	XID value
	VERes enrolled	Y	ECI	07
	XID	XID value		
Action	You are not permitted to submit this transaction for authorization. Instead ask the customer for another form of payment.			

Test Case 53: JCB J/Secure Card Enrolled: Unavailable Authentication

Card Number	354159 99 9910 3865			
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	100		
	ics_pa_enroll service was successful.			
Details	E-commerce indicator	internet		
	proofXML	proofXML value		
	VERes enrolled	U		
Action	Submit your authorization request. No liability shift.			

Test Case 54: JCB J/Secure Card Enrolled: Authentication Error Processing PARes

Card Number	354159 99 9910 3881			
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	476
	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		We encountered a payer authentication problem: Error Processing PARes.	
Details	ACS URL	URL value	ECI	07
	PAReq	PAReq value	E-commerce indicator	internet
	proofXML	proofXML value		
	VERes enrolled	Y		
	XID	XID value		
Action	Ask the customer for another form of payment. No liability shift.			

Test Case 55: JCB J/Secure Card Not Enrolled

Card Number	356997 00 1008 3724			
Auth. Type	Non-participating bank			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	100		
	ics_pa_enroll service was successful.			
Details	E-commerce indicator	js_attempted		
	ECI	06		
	proofXML	proofXML value		
	VERes enrolled	N		
Action	Submit your authorization request.			

Test Case 56: JCB J/Secure Enrollment Check: Time-Out

Card Number	356998 00 1008 3723			
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	100		
	ics_pa_enroll service was successful.			
Details	E-commerce indicator	internet		
	proofXML	proofXML value		
Action	After 10-12 seconds, proceed with the authorization request. No liability shift.			

Test Case 57: JCB J/Secure Enrollment Check: Lookup Error Processing Message Request

Card Number	354159	

Test Case 57: JCB J/Secure Enrollment Check: Lookup Error Processing Message Request (continued)

	99 6910 3614			
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	100		
	ics_pa_enroll service was successful.			
Details	E-commerce indicator	internet		
	VERes enrolled	U		
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.			

Diners Club Protect Buy 3-D Secure 1.0 Test Cases

The following test cases can be used to test 3-D Secure authentication with Diners Club.

Diners Club Protect Buy Test Cases for 3-D Secure 1.0

Possible Values for Diners Club ProtectBuy Response Fields

Result and Interpretation		Validate Authentication Response			
		Authentication Result	**ECI**	**Commerce Indicator**	**Reason Code**
Success	Successful authentication.	0	05	`pb`	100
	Recorded attempt to authenticate.	1	06	`pb_attempted`	100
Failure (Customer not responsible)	System error that prevents the completion of authentication: you can proceed with	6	[1]	— [2]	100

Possible Values for Diners Club ProtectBuy Response Fields (continued)

Result and Interpretation		Validate Authentication Response			
	authorization, but there is no liability shift.				
	Issuer unable to perform authentication.	6	07	`internet`	100
	Incomplete or unavailable authentication.		07	`internet`	
	Invalid PARes.	-1	—	—	476
Failure (Customer responsible)	Authentication failed or cardholder did not complete authentication. If the authentication fails, Visa requires that you do not accept the card. You must ask the customer to use another payment method.	9	—	—	476

1. The ECI value can vary depending on the reason for the failure.

2. A dash (—) indicates that the field is blank or absent.

Test Case 58: Diners Club ProtectBuy Card Enrolled: Successful Authentication

Card Number	300500 00 0000 6246			
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	100

Test Case 58: Diners Club ProtectBuy Card Enrolled: Successful Authentication (continued)

	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		**ics_pa_validate** service was successful.	
Details	ACS URL	URL	Authentication result	0
	PAReq	PAReq value	CAVV	CAVV value
	proofXML	proofXML value	E-commerce indicator	pb
	VERes enrolled	Y	ECI	05
	XID	XID value	PARes status	Y
			XID	XID value
Action	1. Add the signed PARes to the Validate Authentication request. 2. Ensure that the XID from the enrollment check matches that from the authentication validation. 3. Add the CAVV and ECI values to your authorization request.			

Test Case 59: Diners Club ProtectBuy Card Enrolled: Successful Authentication but Invalid PARes

Card Number	300500 00 0000 4373			
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	476
	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		We encountered a payer authentication problem: PARes signature digest value mismatch. PARes message has been modified.	
Details	ACS URL	URL value	Authentication result	-1
	PAReq	PAReq value	XID	XID value

Test Case 59: Diners Club ProtectBuy Card Enrolled: Successful Authentication but Invalid PARes (continued)

	proofXML	proofXML value		
	VERes enrolled	Y		
	XID	XID value		
Action	Do not proceed with authorization. Instead, ask the customer for another form of payment.			

Test Case 60: Diners Club ProtectBuy Card Enrolled: Attempts Processing

Card Number	300500 00 0000 5271	Card enrollment option during purchase process		
Auth. Type	Activation during shopping			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	100
	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		**ics_pa_validate** service was successful.	
Details	ACS URL	URL value	Authentication result	1
	PAReq	PAReq value	CAVV	CAVV value
	proofXML	proofXML value	E-commerce indicator	pb_attempted
	VERes enrolled	Y	ECI	06
	XID	XID value	PARes status	A
			XID	XID value
Action	If you request Validate Authentication and authorization services separately, follow these steps: 1. Add the signed PARes to the validation request. 2. Ensure that the XID from the enrollment check matches that from the authentication validation. 3. Add the CAVV and ECI values to your authorization request.			

Test Case 60: Diners Club ProtectBuy Card Enrolled: Attempts Processing (continued)

	If you request the Validate Authentication and authorization services together, the process described above occurs automatically.

Test Case 61: Diners Club ProtectBuy Card Enrolled: Incomplete Authentication

Card Number	300500 00 0000 7376			
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	100
	The cardholder is enrolled in payer authentication. Authenticate before proceeding with authorization.		• Issuer unable to perform authentication. • **ics_pa_validate** service was successful.	
Details	ACS URL	URL value	Authentication result	6
	PAReq	PAReq value	E-commerce indicator	internet
	proofXML	proofXML value	ECI	07
	VERes enrolled	Y	PARes status	U
	XID	XID value	XID	XID value
Action	Ask the customer for another form of payment, or submit the transaction. No liability shift.			

Test Case 62: Diners Club ProtectBuy Card Enrolled: Unsuccessful Authentication

Card Number	300500 00 0000 5925			
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	476

Test Case 62: Diners Club ProtectBuy Card Enrolled: Unsuccessful Authentication (continued)

	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		• User failed authentication. • Payer cannot be authenticated.	
Details	ACS URL	URL value	Authentication result	9
	PAReq	PAReq value	PARes status	N
	proofXML	proofXML value	XID	XID value
	VERes enrolled	Y		
	XID	XID value	ECI	07
Action	You are not permitted to submit this transaction for authorization. Instead ask the customer for another form of payment.			

Test Case 63: Diners Club ProtectBuy Card Enrolled: Unavailable Authentication

Card Number	300500 00 0000 6030			
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	100		
	ics_pa_enroll service was successful.			
Details	E-commerce indicator	internet		
	proofXML	proofXML value		
	VERes enrolled	U		
Action	Submit your authorization request. No liability shift.			

Test Case 64: Diners Club ProtectBuy Card Enrolled: Authentication Error

Card Number	300500 00 0000 5602			
Auth. Type	Active authentication			

Test Case 64: Diners Club ProtectBuy Card Enrolled: Authentication Error (continued)

Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	476
	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		We encountered a payer authentication problem: Error Processing PARes.	
Details	ACS URL	URL value	E-commerce indicator	internet
	PAReq	PAReq value	ECI	07
	proofXML	proofXML value		
	VERes enrolled	Y		
	XID	XID value		
Action	Ask the customer for another form of payment. No liability shift.			

Test Case 65: Diners Club ProtectBuy Card Not Enrolled

Card Number	300500 00 0000 7269			
Auth. Type	Non-participating bank			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	100		
	ics_pa_enroll service was successful.			
Details	E-commerce indicator	internet		
	ECI	07		
	proofXML	proofXML value		
	VERes enrolled	N		
Action	Submit the transaction.			

Test Case 66: Diners Club ProtectBuy Enrollment Check: Time-Out

Card Number	300500	

Test Case 66: Diners Club ProtectBuy Enrollment Check: Time-Out (continued)

	00 0000 1890			
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	100		
	ics_pa_enroll service was successful.			
Details	E-commerce indicator	internet		
	proofXML	proofXML value		
Action	After 10-12 seconds, proceed with the authorization request. No liability shift.			

Test Case 67: Diners Club ProtectBuy Enrollment Check Error

Card Number	300500 00 0000 9877 300500 00 0000 4837	Error response Incorrect Configuration: Unable to Authenticate		
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	100		
	ics_pa_enroll service was successful.			
Details	E-commerce indicator	internet		
	proofXML	proofXML value		
	VERes enrolled	U		
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.			

Discover Protect Buy 3-D Secure 1.0 Test Cases

The following test cases can be used to test 3-D Secure authentication with Discover.

Discover Protect Buy Test Cases for 3-D Secure 1.0

Possible Values for Discover ProtectBuy Response Fields

Result and Interpretation		Validate Authentication Response			
		Authentication Result	**ECI**	**Commerce Indicator**	**Reason Code**
Success	Successful authentication.	0	05	`dipb`	100
	Recorded attempt to authenticate.	1	06	`dipb_attempted`	100
Failure (Customer not responsible)	System error that prevents the completion of authentication: you can proceed with authorization, but there is no liability shift.	6	[1]	— [2]	100
	Issuer unable to perform authentication.	6	07	`internet`	100
	Incomplete or unavailable authentication.		07	`internet`	
	Invalid PARes.	-1	—	—	476
Failure (Customer responsible)	Authentication failed or cardholder did not complete authentication. If the authentication fails, Visa requires that you do not accept the card. You must ask the	9	—	—	476

Possible Values for Discover ProtectBuy Response Fields (continued)

Result and Interpretation		Validate Authentication Response			
	customer to use another payment method.				
1 The ECI value can vary depending on the reason for the failure. 2 A dash (—) indicates that the field is blank or absent.					

Test Case 68: Discover ProtectBuy Card Enrolled: Successful Authentication

Card Number	601100 00 0000 0004			
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	100
	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		**ics_pa_validate** service was successful.	
Details	ACS URL	URL	Authentication result	0
	PAReq	PAReq value	CAVV	CAVV value
	proofXML	proofXML value	E-commerce indicator	dipb
	VERes enrolled	Y	ECI	05
	XID	XID value	PARes status	Y
			XID	XID value
Action	1. Add the signed PARes to the Validate Authentication request. 2. Ensure that the XID from the enrollment check matches that from the authentication validation. 3. Add the CAVV and ECI values to your authorization request.			

Test Case 69: Discover ProtectBuy Card Enrolled: Successful Authentication but Invalid PARes

Card Number	601100 00 0000 0012			
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	476
	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		We encountered a payer authentication problem: PARes signature digest value mismatch. PARes message has been modified.	
Details	ACS URL	URL value	Authentication result	-1
	PAReq	PAReq value	XID	XID value
	proofXML	proofXML value		
	VERes enrolled	Y		
	XID	XID value		
Action	Do not proceed with authorization. Instead, ask the customer for another form of payment.			

Test Case 70: Discover ProtectBuy Card Enrolled: Attempts Processing - *Deprecated This test case is no longer used*.

Card Number	601100 00 0000 0038	Card enrollment option during purchase process		
Auth. Type	Discover stand in attempts service			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	100
	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		**ics_pa_validate** service was successful.	
Details	ACS URL	URL value	Authentication result	1
	PAReq	PAReq value	CAVV	CAVV value

Test Case 70: Discover ProtectBuy Card Enrolled: Attempts Processing - *Deprecated This test case is no longer used.* (continued)

	proofXML	proofXML value	E-commerce indicator	dipb_attempted
	VERes enrolled	Y	ECI	06
	XID	XID value	PARes status	A
			XID	XID value
Action	If you request Validate Authentication and authorization services separately, follow these steps: 1. Add the signed PARes to the validation request. 2. Ensure that the XID from the enrollment check matches that from the authentication validation. 3. Add the CAVV and ECI values to your authorization request. If you request the Validate Authentication and authorization services together, the process described above occurs automatically.			

Test Case 71: Discover ProtectBuy Card Enrolled: Incomplete Authentication

Card Number	601100 00 0000 0103			
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	100
	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		• Issuer unable to perform authentication. • **ics_pa_validate** service was successful.	
Details	ACS URL	URL value	Authentication result	6
	PAReq	PAReq value	E-commerce indicator	internet
	proofXML	proofXML value	ECI	07
	VERes enrolled	Y	PARes status	U

Test Case 71: Discover ProtectBuy Card Enrolled: Incomplete Authentication (continued)

	XID	XID value	XID	XID value
Action	Ask the customer for another form of payment, or submit the transaction. No liability shift.			

Test Case 72: Discover ProtectBuy Card Enrolled: Unsuccessful Authentication

Card Number	601100 00 0000 0020			
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	476
	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		• User failed authentication. • Payer cannot be authenticated.	
Details	ACS URL	URL value	Authentication result	9
	PAReq	PAReq value	PARes status	N
	proofXML	proofXML value	XID	XID value
	VERes enrolled	Y		
	XID	XID value		
Action	You are not permitted to submit this transaction for authorization. Instead ask the customer for another form of payment.			

Test Case 73: Discover ProtectBuy Card Enrolled: Unavailable Authentication

Card Number	601100 00 0000 0061			
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	100		
	ics_pa_enroll service was successful.			

Test Case 73: Discover ProtectBuy Card Enrolled: Unavailable Authentication (continued)

Details	E-commerce indicator	internet		
	proofXML	proofXML value		
	VERes enrolled	U		
Action	Submit your authorization request. No liability shift.			

Test Case 74: Discover ProtectBuy Card Enrolled: Authentication Error

Card Number	601100 00 0000 0095			
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	475	Reason Code	476
	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		We encountered a payer authentication problem: Error Processing PARes.	
Details	ACS URL	URL value	E-commerce indicator	internet
	PAReq	PAReq value	ECI	07
	proofXML	proofXML value		
	VERes enrolled	Y		
	XID	XID value		
Action	Ask the customer for another form of payment. No liability shift.			

Test Case 75: Discover ProtectBuy Card Not Enrolled

Card Number	601100 00 0000 0053			
Auth. Type	Non-participating bank			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	100		

Test Case 75: Discover ProtectBuy Card Not Enrolled (continued)

	ics_pa_enroll service was successful.			
Details	E-commerce indicator	internet		
	ECI	07		
	proofXML	proofXML value		
	VERes enrolled	N		
Action	Submit the transaction.			

Test Case 76: Discover ProtectBuy Enrollment Check: Time-Out

Card Number	601100 00 0000 0046			
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	
Summary	Reason Code	100		
	ics_pa_enroll service was successful.			
Details	E-commerce indicator	internet		
	proofXML	proofXML value		
Action	After 10-12 seconds, proceed with the authorization request. No liability shift.			

Test Case 77: Discover ProtectBuy Enrollment Check Error

Card Number	601100 00 0000 0087 601100 00 0000 0079	Error response Incorrect Configuration: Unable to Authenticate		
Auth. Type	Active authentication			
Results	Check Enrollment		Validate Authentication	

Test Case 77: Discover ProtectBuy Enrollment Check Error (continued)

Summary	Reason Code	100		
	ics_pa_enroll service was successful.			
Details	E-commerce indicator	internet		
	proofXML	proofXML value		
	VERes enrolled	U		
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 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 2022, use 2025. You also need the minimum required fields for an order.

Be sure to remove spaces in card numbers when testing.

The 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

Card Numbers

Card Type	Test Card Number
Visa CardType = 001	445653 00 0000 1005
Visa 19-digit PAN CardType = 001	445653 00 0000 1007
Cartes Bancaires Visa CardType = 036	445653 00 0000 3001
Mastercard CardType = 002	520000 00 0000 1005
Cartes Bancaires Mastercard CardType = 036	520000 00 0000 3001
American Express CardType = 003	340000 00 0000 1007
Discover CardType = 004	601100 00 0000 1002
Diners Club CardType = 005	601100 00 0000 1002
JCB J/Secure CardType = 007	333700 00 0000 0008
Elo CardType = 054	650505 00 0000 1000
China UnionPay CardType = 062	620001 00 0020 0000

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>

E-Commerce Indicator Values

The following table lists the expected e-commerce indicator value for each network.

Network	Expected E-Commerce Indicator
Visa	vbv
Cartes Bancaires Visa	vbv
Mastercard	spa
Cartes Bancaires Mastercard	spa
American Express	aesk
Discover	dipb
Diners Club	pb
JCB J/Secure	js
Elo	cs
China UnionPay	up3ds

ECI/Collection Indicator Values

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

Network	Expected ECI
Visa	05
Cartes Bancaires Visa	05

Network	Expected ECI
American Express	05
Discover	05
Diners Club	05
JCB J/Secure	05
Elo	05
China UnionPay	05
	Expected Collection Indicator
Mastercard	2
Cartes Bancaires Mastercard	2

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 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

Card Numbers

Card Type	Test Card Number
Visa CardType = 001	445653 00 0000 1013
Cartes Bancaires Visa CardType = 036	445653 00 0000 3019
Mastercard	520000

Card Type	Test Card Number
CardType = 002	00 0000 1013
Cartes Bancaires Mastercard CardType = 036	520000 00 0000 3019
American Express CardType = 003	340000 00 0000 1015
Discover CardType = 004	601100 00 0000 1010
Diners Club CardType = 005	601100 00 0000 1010
JCB J/Secure CardType = 007	333700 00 0000 0990
Elo CardType = 054	650505 00 0000 1018
China UnionPay CardType = 062	620001 00 0010 0010

Results for the Check Enrollment Service

Reason code = 476

- User failed authentication.
- Payer cannot be authenticated.

VERes enrolled = Y

PARes status = N

ECI/Collection Indicator Values

Network	Expected ECI
Visa	07

Network	Expected ECI
Cartes Bancaires Visa	07
American Express	07
Discover	07
Diners Club	07
JCB J/Secure	07
Elo	07
China UnionPay	07
	Expected Collection Indicator
Mastercard	0
Cartes Bancaires Mastercard	0

Results for the Validation Authentication Service

No results are returned.

Action

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

Card Numbers

Card Type	Test Card Number
Visa CardType = 001	445653 00 0000 1021
Cartes Bancaires Visa CardType = 036	445653 00 0000 3027
Mastercard CardType = 002	520000 00 0000 1021

Card Type	Test Card Number
Cartes Bancaires Mastercard CardType = 036	520000 00 0000 3027
American Express CardType = 003	340000 00 0000 1023
Discover CardType = 004	601100 00 0000 1028
Diners Club CardType = 005	601100 00 0000 1028
JCB J/Secure CardType = 007	333700 00 0000 7045
Elo CardType = 054	650505 00 0000 1026
China UnionPay CardType = 062	620001 00 0000 0020

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>

E-Commerce Indicator Values

The following table lists the expected e-commerce indicator value for each network.

Network	Expected E-Commerce Indicator
Visa	vbv_attempted
Cartes Bancaires Visa	vbv_attempted
Mastercard	spa
Cartes Bancaires Mastercard	spa
American Express	aesk_attempted
Discover	dipb_attempted
Diners Club	pb_attempted
JCB J/Secure	js_attempted
Elo	cs_attempted
China UnionPay	up3ds_attempted

ECI/Collection Indicator Values

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

Network	Expected ECI
Visa	06
Cartes Bancaires Visa	06
American Express	06
Discover	06
Diners Club	06
JCB J/Secure	06
Elo	06
China UnionPay	06
	Expected Collection Indicator
Mastercard	1
Cartes Bancaires Mastercard	1

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 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

Card Numbers

Card Type	Test Card Number
Visa CardType = 001	445653 00 0000 1039
Cartes Bancaires Visa CardType = 036	445653 00 0000 3035
Mastercard CardType = 002	520000 00 0000 1039
Cartes Bancaires Mastercard CardType = 036	520000 00 0000 3035
American Express CardType = 003	340000 00 0000 1031
Discover CardType = 004	601100 00 0000 1036
Diners Club CardType = 005	601100 00 0000 1036
JCB J/Secure CardType = 007	333700 00 0000 0735
Elo CardType = 054	650505 00 0000 1034
China UnionPay	620001

Card Type	Test Card Number
CardType = 062	00 0040 0030

Results for the Check Enrollment Service

Reason code = 100

ics_pa_enroll service was successful.

VERes enrolled = Y

PARes status = U

E-Commerce Indicator Values

The following table lists the expected e-commerce indicator value for each network.

Network	Expected E-Commerce Indicator
Visa	internet or vbv_failure
Cartes Bancaires Visa	internet or vbv_failure
Mastercard	internet
Cartes Bancaires Mastercard	internet
American Express	internet
Discover	internet
Diners Club	internet
JCB J/Secure	internet
Elo	internet
China UnionPay	internet

ECI/Collection Indicator Values

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

Network	Expected ECI
Visa	07
Cartes Bancaires Visa	07

Network	Expected ECI
American Express	07
Discover	07
Diners Club	07
JCB J/Secure	07
Elo	07
China UnionPay	07
	Expected Collection Indicator
Mastercard	0
Cartes Bancaires Mastercard	0

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

Card Numbers

Card Type	Test Card Number
Visa CardType = 001	445653 00 0000 1047
Cartes Bancaires Visa CardType = 036	445653 00 0000 3043
Mastercard CardType = 002	520000 00 0000 1047
Cartes Bancaires Mastercard	520000

Card Type	Test Card Number
CardType = 036	00 0000 3043
American Express CardType = 003	340000 00 0000 1049
Discover CardType = 004	601100 00 0000 1044
Diners Club CardType = 005	601100 00 0000 1044
JCB J/Secure CardType = 007	333700 00 0000 0321
Elo CardType = 054	650505 00 0000 1042
China UnionPay CardType = 062	620001 00 0030 0040

Results for the Check Enrollment Service

Reason code = 476

- User failed authentication.
- Payer cannot be authenticated.

VERes enrolled = Y

PARes status = R

ECI/Collection Indicator Values

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

Network	Expected ECI
Visa	07
Cartes Bancaires Visa	07

Network	Expected ECI
American Express	07
Discover	07
Diners Club	07
JCB J/Secure	07
Elo	07
China UnionPay	07
	Expected Collection Indicator
Mastercard	0
Cartes Bancaires Mastercard	0

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

Card Numbers

Card Type	Test Card Number
Visa CardType = 001	445653 00 0000 1054
Cartes Bancaires Visa CardType = 036	445653 00 0000 3050
Mastercard CardType = 002	520000 00 0000 1054
Cartes Bancaires Mastercard	520000

Card Type	Test Card Number
CardType = 036	00 0000 3050
American Express CardType = 003	340000 00 0000 1056
Discover CardType = 004	601100 00 0000 1051
Diners Club CardType = 005	601100 00 0000 1051
JCB J/Secure CardType = 007	333700 00 0000 6765
Elo CardType = 054	650505 00 0000 1059
China UnionPay CardType = 062	620001 00 0060 0050

Results for the Check Enrollment Service

Reason code = 100

ics_pa_enroll service was successful.

VERes enrolled = U

E-Commerce Indicator Values

Network	Expected E-Commerce Indicator
Visa	internet or vbv_failure
Cartes Bancaires Visa	internet or vbv_failure
Mastercard	internet
Cartes Bancaires Mastercard	internet
American Express	internet
Discover	internet

Network	Expected E-Commerce Indicator
Diners Club	internet
JCB J/Secure	internet
Elo	internet
China UnionPay	internet

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

Card Numbers

Card Type	Test Card Number
Visa CardType = 001	445653 00 0000 1062
Cartes Bancaires Visa CardType = 036	445653 00 0000 3068
Mastercard CardType = 002	520000 00 0000 1062
Cartes Bancaires Mastercard CardType = 036	520000 00 0000 3068
American Express CardType = 003	340000 00 0000 1064
Discover	601100

Card Type	Test Card Number
CardType = 004	00 0000 1069
Diners Club CardType = 005	601100 00 0000 1069
JCB J/Secure CardType = 007	333700 00 0000 0016
Elo CardType = 054	650505 00 0000 1067
China UnionPay CardType = 062	620001 00 0050 0060

Results for the Check Enrollment Service

Reason code = 100

ics_pa_enroll service was successful.

VERes enrolled = U

E-Commerce Indicator Values

Network	Expected E-Commerce Indicator
Visa	internet or vbv_failure
Cartes Bancaires Visa	internet or vbv_failure
Mastercard	internet
Cartes Bancaires Mastercard	internet
American Express	internet
Discover	internet
Diners Club	internet
JCB J/Secure	internet
Elo	internet
China UnionPay	internet

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

Card Numbers

Card Type	Test Card Number
Visa CardType = 001	445653 00 0000 1070
Cartes Bancaires Visa CardType = 036	445653 00 0000 3076
Mastercard CardType = 002	520000 00 0000 1070
Cartes Bancaires Mastercard CardType = 036	520000 00 0000 3076
American Express CardType = 003	340000 00 0000 1072
Discover CardType = 004	601100 00 0000 1077
Diners Club CardType = 005	601100 00 0000 1077

Card Type	Test Card Number
JCB J/Secure CardType = 007	333700 00 0000 0081
Elo CardType = 054	650505 00 0000 1075
China UnionPay CardType = 062	620001 00 0090 0070

Results for the Check Enrollment Service

Reason code = 100

ics_pa_enroll service was successful.

VERes enrolled = U

E-Commerce Indicator Values

Network	Expected E-Commerce Indicator
Visa	internet or vbv_failure
Cartes Bancaires Visa	internet or vbv_failure
Mastercard	internet
Cartes Bancaires Mastercard	internet
American Express	internet
Discover	internet
Diners Club	internet
JCB J/Secure	internet
Elo	internet
China UnionPay	internet

ECI/Collection Indicator Values

Network	Expected ECI
Visa	07
Cartes Bancaires Visa	07
American Express	07
Discover	07
Diners Club	07
JCB J/Secure	07
Elo	07
China UnionPay	07
	Expected Collection Indicator
Mastercard	0
Cartes Bancaires Mastercard	0

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

Card Numbers

Card Type	Test Card Number
Visa CardType = 001	445653 00 0000 1088
Cartes Bancaires Visa	445653

Card Type	Test Card Number
CardType = 036	00 0000 3084
Mastercard CardType = 002	520000 00 0000 1088
Cartes Bancaires Mastercard CardType = 036	520000 00 0000 3084
American Express CardType = 003	340000 00 0000 1080
Discover CardType = 004	601100 00 0000 1085
Diners Club CardType = 005	601100 00 0000 1085
JCB J/Secure CardType = 007	333700 00 0000 0537
Elo CardType = 054	650505 00 0000 1083
China UnionPay CardType = 062	620001 00 0080 0080

Results for the Check Enrollment Service

Reason code = 100

ics_pa_enroll service was successful.

VERes enrolled = B

E-Commerce Indicator Values

The following table lists the expected e-commerce indicator value for each network.

Network	Expected E-commerce Indicator
Visa	internet

Network	Expected E-commerce Indicator
	ECI = 07
Cartes Bancaires Visa	internet ECI = 07
Mastercard	internet ECI = 0
Cartes Bancaires Mastercard	internet ECI = 07
American Express	internet ECI = 07
Discover	internet ECI = 07
Diners Club	internet ECI = 07
JCB J/Secure	internet ECI = 07
Elo	internet ECI = 07
China UnionPay	internet ECI = 07

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

Card Numbers

Card Type	Test Card Number
Visa CardType = 001	445653 00 0000 1096
Cartes Bancaires Visa CardType = 036	445653 00 0000 3134
Mastercard CardType = 002	520000 00 0000 1096
Cartes Bancaires Mastercard CardType = 036	520000 00 0000 3092
American Express CardType = 003	340000 00 0000 1098
Discover CardType = 004	601100 00 0000 1093
Diners Club CardType = 005	601100 00 0000 1093
JCB J/Secure CardType = 007	333700 00 0020 0004
Elo CardType = 054	650505 00 0000 1091
China UnionPay CardType = 062	620001 99 9980 0019

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

PAReq = <PAReq value>

ACS URL = <URL value>

Results for the Validation Authentication Service

Reason code = 100

ics_pa_validate service was successful.

PARes status = Y

CAVV = <CAVV value>

AVV = <AVV value> (Mastercard only)

XID = <XID value>

E-Commerce Indicator Values

The following table lists the expected e-commerce indicator value for each network.

Network	Expected E-Commerce Indicator
Visa	vbv
Cartes Bancaires Visa	vbv
Mastercard	spa
Cartes Bancaires Mastercard	spa
American Express	aesk
Discover	dipb
Diners Club	pb
JCB J/Secure	js
Elo	cs
China UnionPay	up3ds

ECI/Collection Indicator Values

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

Network	Expected ECI
Visa	05
Cartes Bancaires Visa	05
American Express	05
Discover	05
Diners Club	05
JCB J/Secure	05
Elo	05
China UnionPay	05
	Expected Collection Indicator
Mastercard	2
Cartes Bancaires Mastercard	2

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.

Test Case 2.11a: Unsuccessful Step-Up Authentication

Card Numbers

Card Type	Test Card Number
Visa CardType = 001	445653 00 0000 1104
Cartes Bancaires Visa CardType = 036	445653 00 0000 3092
Mastercard	520000

Card Type	Test Card Number
CardType = 002	00 0000 1104
Cartes Bancaires Mastercard CardType = 036	520000 00 0000 3100
American Express CardType = 003	340000 00 0000 1106
Discover CardType = 004	601100 00 0000 1101
Diners Club CardType = 005	601100 00 0000 1101
JCB J/Secure CardType = 007	333700 00 0020 0087
Elo CardType = 054	650505 00 0000 1109
China UnionPay CardType = 062	620001 99 9970 0029

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

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

ECI/Collection Indicator Values

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

Network	Expected ECI
Visa	07
Cartes Bancaires Visa	07
American Express	07
Discover	07
Diners Club	07
JCB J/Secure	07
Elo	07
China UnionPay	07
	Expected Collection Indicator
Mastercard	0
Cartes Bancaires Mastercard	0

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

Card Numbers

Card Type	Test Card Number
Visa CardType = 001	445653 00 0000 1112
Cartes Bancaires Visa	445653

Card Type	Test Card Number
CardType = 036	00 0000 3100
Mastercard CardType = 002	520000 00 0000 1112
Cartes Bancaires Mastercard CardType = 036	520000 00 0000 3118
American Express CardType = 003	340000 00 0000 1114
Discover CardType = 004	601100 00 0000 1119
Diners Club CardType = 005	601100 00 0000 1119
JCB J/Secure CardType = 007	333700 00 0020 0079
Elo CardType = 054	650505 00 0000 1117
China UnionPay CardType = 062	620001 99 9960 0039

Results for the Check Enrollment Service

Reason code = 475

The cardholder is enrolled in payer authentication. Authenticate before proceeding with authorization.

VERes enrolled = Y

PAReq = <PAReq value>

ACS URL = <URL value>

Results for the Validation Authentication Service

Reason code = 100 **ics_pa_validate** service was successful.

PARes status = U

E-Commerce Indicator Values

The following table lists the expected e-commerce indicator value for each network.

Network	Expected E-Commerce Indicator
Visa	internet or vbv_failure
Cartes Bancaires Visa	internet or vbv_failure
Mastercard	internet
Cartes Bancaires Mastercard	internet
American Express	internet
Discover	internet
Diners Club	internet
JCB J/Secure	internet
Elo	internet
China UnionPay	internet

ECI/Collection Indicator Values

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

Network	Expected ECI
Visa	07
Cartes Bancaires Visa	07
American Express	07
Discover	07
Diners Club	07
JCB J/Secure	07
Elo	07
China UnionPay	07

Network	Expected ECI
	Expected Collection Indicator
Mastercard	0
Cartes Bancaires Mastercard	0

Action

Retry authentication or process without liability shift.

Test Case 2.14: Require MethodURL

Card Numbers

The Method URL test case provides a URL associated with the issuer's BIN range. This Method runs before the authentication request to gather information about the location and type of device being used in the transaction to better assess the risk of a transaction. If device data can't be collected, an older version of the 3-D Secure protocol is used and the order is assessed as a higher risk.

Card Type	Test Card Number
Visa CardType = 001	400001 00 0000 0001
Cartes Bancaires Visa CardType = 036	400000 00 0000 3212
Mastercard CardType = 002	520001 00 0000 0006
Cartes Bancaires Mastercard CardType = 036	520000 00 0000 3017
American Express CardType = 003	340001 00 0000 0007
Discover CardType = 004	601101 00 0000 0003

Card Type	Test Card Number
Diners Club CardType = 005	601101 00 0000 0003
JCB J/Secure CardType = 007	333700 00 0000 0388
Elo CardType = 054	650505 00 0000 1208
China UnionPay CardType = 062	620001 00 0000 0205

Results for the Check Enrollment Service

Reason code = 100

VERes enrolled = Y

PARes status = Y

CAVV = <CAVV value>

AVV = <AVV value> (Mastercard only)

XID = <XID value> (American Express only)

ECI/Collection Indicator Values

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

Network	Expected ECI
Visa	05
Cartes Bancaires Visa	05
American Express	05
Discover	05
Diners Club	05
JCB J/Secure	05

Network	Expected ECI
Elo	05
China UnionPay	05
	Expected Collection Indicator
Mastercard	02
Cartes Bancaires Mastercard	02

Results for the Validation Authentication Service

If device data cannot be collected or you do not run the 3-D Secure Method (or you do not provide the browser fields needed for an 3-D Secure transaction on the lookup request), the transaction reverts to using 3-D Secure 1.0.2 with an PARes status of U.

If you provide the browser fields on the Enroll request but do not run the 3-D Secure Method, you get the following result:

Reason code = 100 **ics_pa_validate** service was successful.

PARes status = U

Action

If your device data collection method implements correctly and 3-D Secure Method processing occurs, the test transaction produces a Frictionless Success result.

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 (on page 221) (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 CardType = 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

ECI = 00

Results for the Validation Authentication Service

Reason code = 100 **ics_pa_validate** service was successful.

PARes status = Y

CAVV = <CAVV>

ECI = 02

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 CardType = 002	445653 00 0000 2045

Results for the Check Enrollment Service

Reason code = 100

VERes enrolled = Y

PARes status = N

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 CardType = 001	445653 00 0000 2144

Results for the Check Enrollment Service

Reason code = 100

VERes enrolled = Y

PARes status = U

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 CardType = 001	445653 00 0000 2169

Results for the Check Enrollment Service

Reason code = 100

VERes enrolled = Y

PARes status = R

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 CardType = 001	445653 00 0000 2177

Results for the Check Enrollment Service

Reason code = 100

VERes enrolled = Y

PARes status = U

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 CardType = 001	445653 00 0000 2235

Results for the Check Enrollment Service

Reason code = 100

VERes enrolled = Y

PARes status = U

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 CardType = 001	445653 00 0000 2243

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

ECI = 07

Reason code = 21

Test Case 3a: Transaction Risk Analysis Exemption - Low Value - Mastercard

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.

Card Type	Test Card Number
Mastercard CardType = 002	520000 00 0000 1161

Required Fields for Check Enrollment

Challenge code = 05

Results for the Check Enrollment Service

Reason code = 100

VERes enrolled = Y

PARes status = I

CAVV = <CAVV value>

ECI = 06

Reason code = 81

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 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 3-D Secure. The PARes Status will differ between the 3-D Secure versions.

Card Type	Test Card Number
Mastercard	(version 2.1.0) 5200 0000 0000 1161

Card Type	Test Card Number
CardType = 002	(version 2.2.0) 5200 0000 0000 2052

Required Fields for Check Enrollment

Challenge code = 05

Results for the Check Enrollment Service

Reason code = 100

VERes enrolled = Y

PARes status (version 2.1.0) = N

(version 2.2.0) = I

CAVV = <CAVV value>

ECI = 06

Reason code = 81

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 CardType = 001	445653 00 0000 2029

Required Fields for Check Enrollment

Challenge code = 05

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 CardType = 004	601100 00 0000 1002

Required Fields for Check Enrollment

Challenge code = 04

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

You have 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
Cartes Bancaires Visa CardType = 036	400000 00 0000 3006
Cartes Bancaires Mastercard CardType = 036	520000 00 0000 3001

Required Fields for Check Enrollment

Challenge code = 02

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 could be returned in production based on issuer rules surrounding co-branding with Visa or Mastercard BINs.)

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 CardType = 002	445653 00 0000 2003

Required Fields for Check Enrollment

Challenge code = 09

Results for the Check Enrollment Service

VERes enrolled = Y

PARes status = C

ECI = 07

Results for the Authenticate Response

PARes status = Y

CAVV = <CAVV value>

ECI = 05

WhiteListStatus = Y

WhiteListStatusSource = 03

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 CardType = 002	445653 00 0000 2011

Required Fields for Check Enrollment

Challenge code = 08

Results for the Check Enrollment Service

Reason code = 100

PARes status = Y

CAVV = <CAVV value>

ECI = 05

WhiteListStatus = Y

WhiteListStatusSource = 03

ThreeDSVersion = 2.2.0

Action

You should 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 CardType = 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

ThreeDSVersion = 2.1.0

Reason code = 100

Action

You should 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 CardType = 002	520000 00 0000 1005

Required Fields for Check Enrollment

Message Category = 80

Score Request = Y

Merchant Reason Code = A

Results for the Check Enrollment Service

Reason code = 100

PAResStatus = U

CAVV = <CAVV value>

ECI = 04

ThreeDSVersion = 2.1.0

Results for the Authentication Result

Reason code = 100

Action

You should append the ECI and DS Transaction ID values to the authorization message.

Website Modification Reference

This section contains information about modifying 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. Use session checks throughout the authentication process to prevent authenticating transactions twice. Avoid confusing messages, such as warnings about expired pages.

Add appropriate logos:

- Download the appropriate logos of the cards that you support and place these logos next to the card information entry fields on your checkout pages. For more information about obtaining logos and using them, see 3-D Secure Services Logos (on page 183).

Add informational message:

- Add a message next to the final "buy" button and the card logo to inform your customers that they may be prompted to provide their authentication password. For examples of messages you can use, see Informational Message Examples (on page 185).

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

3-D Secure Service	Download Location
Visa Secure	https://usa.visa.com/run-your-business/small-business-tools/payment-technology/visa-secure.html

3-D Secure Services Logos Download Location (continued)

3-D Secure Service	Download Location
	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.
Elo Compra Segura	Contact Elo Customer Support to obtain logos.
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 may be prompted for their authentication password or to enroll in the authentication program for their card.

These examples may 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 may 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 may 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 may 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 Cardinal Cruise Direct Connection API Payer Authentication.

Important: If you are using tokenization, use the Cardinal Cruise Direct Connection 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 3-D Secure 2.x request and invoke the 3-D Secure Method URL when it is available.

The Cardinal Cruise Direct Connection 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, then it must run on the merchant site. This is done after a merchant passes in the BIN (at least the first eight digits of the card number) into Cardinal on the POST to the device data collection URL. Options on how to include the BIN are below.

The Method URL is a concept in the EMV 3-D Secure protocol that enables an issuing bank to obtain additional browser information prior to 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. This process also occurred in 3-D Secure 1.0 when the customer's browser was redirected to the ACS URL. The Method URL step provides a better user experience.

Prerequisites

To support device data collection, you must complete one of the following:

- Obtain access to the card BIN (first eight digits or full card number of cardholder).
- Create an iframe on your website and POST 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 provides information about upgrading to 3-D Secure 2.x for merchants currently using Payer Authentication services.

Benefits

3-D Secure 2.x provides these benefits:

- More secure transactions as additional data is available.
- Backward compatibility. Additional data is automatically sent to issuers as they upgrade to 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 3-D Secure 2.2. Version 2.2 includes support for exemptions for PSD2, which might allow frictionless authentication, including 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 3-D Secure 2.x.

PSD2 requires additional security measures outlined in the Regulatory Technical Standards (RTS) that become applicable 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 3-D Secure e-commerce merchants to integrate dynamic authentication tools (such as 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:

https://demos.cardinalcommerce.com/3DS_Info/Country_Mandates/index.html

Recommended Integration

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

- Cardinal Cruise Direct Connection API
- SDK integration for your mobile application
- Hybrid integration

If you are currently using Payer Authentication services in your existing business processes and need to upgrade to 3-D Secure 2.x, we recommend using the Cardinal Cruise Direct Connection API integration. The Cardinal Cruise Direct Connection 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 3-D Secure 2.x, that response includes a new value, the processor transaction ID.

For enrolled cards, include the 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 CAVV or AAV.

For more information about the Cardinal Cruise Direct Connection API, see Implementing Cardinal Cruise Direct Connection API Payer Authentication (on page 20).

For details about the other integrations, see Implementing SDK Payer Authentication (on page 39) or Implementing Hybrid Payer Authentication (on page 62).

Important: If you are using tokenization, use the Cardinal Cruise Direct Connection API integration method for Payer Authentication.

Migration 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 to do this:

- 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 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 139)) to test your preliminary code and make the appropriate changes.

Q: How does 3-D Secure 2.x authentication improve the experience for a customer who uses a mobile or tablet device?

A: 3-D Secure 2.x is agnostic to the device and you have control over the formatting of the authentication form. 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 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 PA Transaction ID and the customer's account number (PAN). Use the PA 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

When a card is enrolled, the process 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 or AAV_CAVV because the authentication is not complete.

If CAVV and ECI are not provided and the Enrollment transaction results in a challenge, then 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 is returned with the corresponding response message. The necessary payer authentication information is 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 if the customer is authenticated.
    - The XID area is empty because the card in not enrolled.

Standard Payer Authentication Implementation Overview

If you are just getting started with Payer Authentication, in most cases, you should use the Cardinal Direct Connection API implementation method. It is the newest and best developed method of integrating 3D Secure into your transaction process and is the method the majority of our customers choose to use. The Standard implementation is still available when customer support determines that it best fits your company's business needs.

The same prerequisites (on page 41) involving JSON Web Tokens and BIN detection that are necessary for the SDK integration are also required for the Standard integration. Complete these prerequisites before continuing with your Standard implementation.

Give customer support the merchant ID that you will use for testing. For more information, see Required Merchant Information (on page 19).

Implementation tasks include:

- Add the JavaScript code to your checkout page
- For each purchase request
    - Build the authentication request
    - Invoke the authentication
    - Handle declines
    - Call the following services:
        - **payerAuthEnrollService**: Payer Authentication Enrollment Check
        - **ccAuthService**: Card Authorization service (optional)
- Use the test cases to test your preliminary code and make appropriate changes. You can change to the test environment by changing the URL in your JavaScript code. See Testing Payer Authentication (on page 88).
- Ensure that your account is configured for production.

Process Flow for Standard Integration

1. You generate a JSON Web Token (JWT).
2. You add the JavaScript tag to your checkout page.
3. Call *Cardinal.setup()*.
4. Run BIN detection. If the BIN is eligible for 3D Secure 2.x, it gathers the proper Method URL JavaScript required by the issuer to collect additional device data.
5. When the customer places an order on your website, you call the *cardinal.start* function to pass in the transaction level data including the full PAN and order details.
6. The JavaScript verifies with the bank that the card is enrolled in a 3D Secure card authentication program by using a server-to-server call.
7. If the issuing bank requires authentication, the JavaScript displays the authentication window.
8. If required, the customer enters the authentication information.
9. The bank validates the customer credentials, and a JWT is returned that the merchant is required to validate server-side for security reasons.
10. You request the ICS Enrollment Check service, extracting the processor transaction ID value from the JWT and sending it in the **payerAuthEnrollService_authenticationTransactionID** request field. You receive this information:
    - E-commerce indicator
    - CAVV (all card types except Mastercard)
    - AAV (Mastercard only)
    - Transaction ID
    - 3D Secure version
    - 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 Check Enrollment Service (Standard)," page 244.

Starting Authentication

The JavaScript handles the device data collection, initiates the transaction for authentication, displays the authentication window if required, and returns the authentication results.

You initiate this authentication process, usually when the customer clicks the Place Order or Submit Order button, by triggering *Cardinal.start()*. *Cardinal.start()* invokes the authentication and authenticates the customer.

Create an order object to pass to the *Cardinal.start()* event. The more fields you include, the less likely the cardholder will be challenged to provide credentials.

Initiate *Cardinal.start()* before the authorization as shown in the following example. The second argument of data is a Request Order Object. You can construct this object ahead of time or pass it directly as shown.

Cardinal.start with Request Order Object

```
Cardinal.start("cca", {
OrderDetails: {
OrderNumber: "1234567890"
},
Consumer: {
Account: {
AccountNumber: "4000000000001000",
ExpirationMonth: "01",
ExpirationYear: "2099"
                ...
<Other 2.x required/optional fields>

}
}
...
});
```

Decoded Response JWT

Payments.validated returns the authentication results and response JWT along with the processor transaction ID as shown in the following example.

```
{
  "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"
  }
}
```

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.
```

Requesting the Check Enrollment Service (Standard)

Once the validation is complete, use the Check Enrollment service to obtain the values needed for authorization.

To request the Check Enrollment service, extract the processor transaction ID value from the JWT and send it in the payerAuthEnrollService_authenticationTransactionIDpa_authentication_transaction_id request field. The following fields are also required:

- **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**

You can send additional request data in order to reduce your issuer step-up authentication rates. It is best to send all available fields.

For further details on required and optional fields, see "Request Fields," page 149"Request-Level Fields," page 149.

It is recommended that you request both payer authentication and card authorization services at the same time. When you do so, the correct information is automatically sent to your payment processor. The values of these fields are converted to the proper format required by your payment processor:

E-commerce indicator: **payerAuthEnrollReply_commerceIndicator**

CAVV: **payerAuthValidateReply_cavv**

AAV: **payerAuthValidateReply_ucafAuthenticationData**

XID: **payerAuthEnrollReply_xid** and **payerAuthValidateReply_xid**

If you request the services separately, you must manually include the enrollment check result values (Enrollment 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 may invalidate your liability shift for that transaction. Include the electronic commerce indicator (ECI), the transaction ID (XID), the 3D Secure version, the directory server transaction ID, and 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, include the UCAF (universal cardholder authentication field) and the collection indicator.

The following table lists these fields.

Enrollment Check and Response Fields

Identifier	Enrollment Check Response Field	Card Authorization Request Field
E-commerce indicator	**payerAuthEnrollReply_commerceIndicator**	**ccAuthService_commerceIndicator**
CAVV (Visa and American Express only)	**PayerAuthEnrollReply_cavv**	**ccAuthService_cavv**
AAV (Mastercard only. Known as UCAF)	**payerAuthEnrollReply_ucafAuthenticationData**	**ucaf_authenticationData**
XID	**payerAuthEnrollReply_xid**	**ccAuthService_xid**
3D Secure version	**payerAuthEnrollReply_specificationVersion**	**ccAuthService_paSpecificationVersion**
Directory server transaction ID (Not required for 3D Secure 1.0.)	**payerAuthEnrollReply_directoryServerTransactionID**	**ccAuthService_directoryServerTransactionID**

In most cases, you request card authorization only once for each purchase. However, you must send multiple authorization requests if the original authorization expires before it is captured, which can occur when order fulfillment is split or delayed. In these cases, you must include in subsequent authorization requests the same payer authentication data contained in the original request so that your acquiring bank can track all related requests if a dispute occurs. Authentication data can only be used for one authorization and cannot be used multiple times or on subsequent authorizations.

Standard Integration Examples

The following examples show a request and response for the check enrollment service.

Standard: Check Enrollment

Standard Integration Check Enrollment Request

```
payerAuthEnrollService_run=true
merchantID=patest
merchantReferenceCode=23AEE8CB6B62EE2AF07
item_0_unitPrice=19.99
purchaseTotals_currency=USD
card_expirationMonth=01
card_expirationYear=2020
card_accountNumber=xxxxxxxxxxxxxxx
card_cardType=001
payerAuthEnrollService_authenticationTransactionID=Fl8d1UW9VwTyawKTdex0
```

Standard Integration Transaction Response for Visa Card with Visa Secure

```
requestID=0340290070000167905080
merchantReferenceCode=23AEE8CB6B62EE2AF07
purchaseTotals_currency=USD
decision=ACCEPT
reasonCode=100
payerAuthEnrollReply_reasonCode=100
payerAuthEnrollReply_authenticationResult=0
payerAuthEnrollReply_authenticationStatusMessage=Success
payerAuthEnrollReply_authenticationTransactionID=Fl8d1UW9VwTyawKTdex0
payerAuthEnrollReply_cavv=Y2FyZGluYWxjb21tZXJjZWF1dGg=
payerAuthEnrollReply_commerceIndicator=vbv
payerAuthEnrollReply_eci=5
payerAuthEnrollReply_eciRaw=05
payerAuthEnrollReply_paresStatus=Y
payerAuthEnrollReply_reasonCode=100
payerAuthEnrollReply_specificationVersion=2.0.1
payerAuthEnrollReply_veresEnrolled=Y
```

Standard Integration Transaction Response for Visa Card with Visa Secure

```
ics_decision_reason_code=100
ics_rcode=1
ics_return_code=1000000
ics_rflag=SOK
ics_rmsg=Request was processed successfully.
merchant_ref_number=cybs_test
pa_enroll_authentication_result=0
pa_enroll_authentication_status_msg=Success
pa_enroll_authentication_transaction_id=S2mJR91kjL3jkt1vCW30
pa_enroll_cavv=Y2FyZGluYWxjb21tZXJjZWF1dGg=
```

```
pa_enroll_e_commerce_indicator=vbv
pa_enroll_eci=05
pa_enroll_eci_raw=05
pa_enroll_pares_status=Y
pa_enroll_rcode=1
pa_enroll_return_code=1050000
pa_enroll_rflag=SOK
pa_enroll_rmsg=ics_pa_enroll service was successful
pa_enroll_specification_version=2.0.1
pa_enroll_veres_enrolled=Y
request_id=5241736293196265601541
request_token=Ahj77wSTHCe0OsC8NroFEU/4EU07jAU/4EU07j0gV9D9ajLtjijj5Uar+ALJjhPaHWBe
G10CgAAA+gzb
```

Payer Authentication Reports

This section describes the Payer Authentication reports, Payer Authentication Summary Report and Payer Authentication Detail Report, that you can download from the Business Center. More information about the Summary Report is provided in a following section. For information about the report elements contained in the Detail Report, refer to the Reports section in the Payer Authentication, Using the Simple Order API guide.

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 obtain 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 am EST. Data in this report remains available for six 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 which 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 193).

Payer Authentication Report Details

Mar 30 2020				
ubcvp1_2 Mar 30 2020 03:42:16 PM	14375401210000167904064 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	14375436464100167904065 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	14375388468800167904064 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?	Reported	
			Commerce Indicator	ECI
Visa, American Express, and JCB	No authentication Recorded attempt to authenticate Successful authentication	No Yes Yes	Internet VbV, Desk, or JS Attempted VbV, JS, or Aesk	7 6 5
Mastercard and Maestro	No authentication Yes	No	Internet[2] SPA	7[1] 1

Payer Authentication Report Interpretation (continued)

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

1. Although the report heading is 7, you receive a collection indicator value of 1, or the response field is empty.
2. 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 may be differences between the Payer Authentication report and the payment reports because an authenticated transaction may not be authorized.

The values (amounts and counts) in the Payer Authentication report may not match exactly your other sources of reconciliation. This report shows the transactions validated by payer authentication. There may be a different number of transactions that were authorized. There are more likely to be reconciliation discrepancies 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 may receive these results:

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

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 Elements

The Payer Authentication Detail report consists of these elements:

- <Report> (on page 207)
- <PayerAuthDetail> (on page 207)
- <ProofXML> (on page 209)

- <VEReq> (on page 211)
- <VERes> (on page 212)
- <PAReq> (on page 212)
- <PARes> (on page 214)
- <AuthInfo> (on page 216)

Report

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> (on page 207).

<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

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 transaction. This field corresponds to the**requestID** API field.	Numeric (26)
<MerchantID>	Merchant ID used for the transaction.	String (30)
<RequestDate>	Date on which the transaction was processed.	DateTime (25)
<ProofXML>	Data that includes the date and time of the enrollment check and the VEReq and VERes elements. This field corresponds to theL**AuthEnrollReply_proofXML** API field. For a list of child elements, see <ProofXML> (on page 209).	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> (on page 211).	
<VERes>	Verify Enrollment Response (VERes) is sent by the directory server. For a list of child elements, see <VERes> (on page 212).	
<PAReq>	Payer Authentication Request message that you send to the ACS through the payment card company. Corresponds to the q**payerAuthEnrollReply_paReq** API field. For a list of child elements, see <PAReq> (on page 212).	

Child Elements of <PayerAuthDetail> (continued)

Element Name	Description	Type & Length
<PARes>	Payer Authentication Response message sent by the ACS. For a list of child elements, see <PARes> (on page 214).	
<AuthInfo>	Address and card verification data. For a list of child elements, see AuthInfo (on page 216).	

<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

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 may 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 log into 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

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 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 log in to the ACS URL.	String (24)

<VEReq> Element

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

VERes

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 can 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)
<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

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
<XID>	Unique transaction identifier generated for each Payment Authentication Request (PAReq) message. The PARes sent back by the issuing bank contains the XID of the PAReq. To ensure that both XIDs are the same, compare it to the XID in the response. To find all requests related to a transaction, you can also search transactions for a specific XID.	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 may not because of differing time zones and synchronization between servers.)	DateTime (25)
<Purchase Amount>	Authorization amount and currency for the transaction. This element corresponds to the totals of the offer lines or from these fields: **ccAuthReply_amount** (see *Credit Card Services Using the*	Amount (15)

Child Elements of <PAReq> (continued)

Element Name	Description	Type & Length
	Simple Order API [PDF | HTML]) or **purchaseTotals_grandTotalAmount** from external data.	
<AcctID>	Masked string used by the ACS.	String (28)
<Expiry>	Expiration month and year of the customer's card.	Number (4)

```
<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

The <PARes> element contains the payer authentication response.

```
<PARes>
  (AcqBIN)
  (MerID)
  (XID)
  (Date)
  (PurchaseAmount)
  (PAN)
  (AuthDate)
  (Status)
  (CAVV)
  (ECI)
</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 may 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 this field: **ccAuthReply_amount** (see *Credit Card Services Using the Simple Order API* [PDF | HTML]) 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)
<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.	String (1)

Element Name	Description	Type & Length
	• U: Authenticate not completed regardless of the reason. • A: Proof of authentication attempt was generated.	
<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

The <AuthInfo> element contains address and card verification information.

```
<AuthInfo>
  (AVSResult)
```

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

Element Name	Description	Type & Length
<AVSResult>	Optional results of the address verification test. See **ccAuthReply_avsCode** or **afsService_avsCode** (if from external data) in *Credit Card Services Using the Simple Order API* (PDF).	String (1)
<CVVResult>	Optional results of the card verification number test. See **ccAuthReply_cvvCode** or **afsService_cvCode** (if from external data) in *Credit Card Services Using the Simple Order API* (PDF).	String (1)

<AuthInfo> Element

```
<AuthInfo>
  <AVSResult>Y</AVSResult>
  <CVVResult/>
</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="2020-02-09T08:00:00-08:00"
ReportEndDate="2020-02-10T08:00:00-08:00"
<PayerAuthDetails>
   <PayerAuthDetail>
      RequestID=”189554943000016790454 8”
      TransactionType=”ics_pa_enroll
      RequestDate=”2020-02-09T08:00:02-08:00”
      <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>
         <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>2020-02-09T08:00:34-08:00</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=”189554990000167904548”
      TransactionType=”ics_pa_validate”
      XID=”2YNaNGDBEdydJ6WI6aFJWAAHBwE=”
      RequestDate=”2020-02-09T08:00:02-08:00”
      <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>2020-02-09T07:59:46-08:00</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.
234	A problem exists with your <keyword keyref="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

A 3-D Secure request for information. It is an EMVCo term for a 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. 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 Mastercard (starts with 5) and Visa (starts with 4) for every participating acquirer.

acquiring 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 3-D Secure.

API

Application Programming Interface. A specification used by software components to communicate with each other.

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 one-digit 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

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.

HTTP

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

Enroll

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

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 3-D Secure program of JCB.

issuer

The bank that issued a 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 3-D Secure.

Maestro.

A card brand owned by Mastercard that includes several debit card BINs within the U.K. (where it was formerly known as Switch), 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 card 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 card 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 a 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. 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.

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.

Solo

A debit card type owned by Maestro. It was permanently discontinued March 31, 2011.

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.

Switch

See Maestro.

validate

Cybersource service for decoding and decrypting 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 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.