

CyberSource Payer Authentication

Using the Simple Order API

March 2020



CyberSource Contact Information

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

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

For support information about any CyberSource Service, visit the Support Center at <http://www.cybersource.com/support>.

Copyright

© 2020 CyberSource Corporation. All rights reserved. CyberSource Corporation ("CyberSource") furnishes this document and the software described in this document under the applicable agreement between the reader of this document ("You") and CyberSource ("Agreement"). You may use this document and/or software only in accordance with the terms of the Agreement. Except as expressly set forth in the Agreement, the information contained in this document is subject to change without notice and therefore should not be interpreted in any way as a guarantee or warranty by CyberSource. CyberSource assumes no responsibility or liability for any errors that may appear in this document. The copyrighted software that accompanies this document is licensed to You for use only in strict accordance with the Agreement. You should read the Agreement carefully before using the software. Except as permitted by the Agreement, You may not reproduce any part of this document, store this document in a retrieval system, or transmit this document, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written consent of CyberSource.

Restricted Rights Legends

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

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 Unpublished rights reserved under the copyright laws of the United States.

Trademarks

Authorize.Net, eCheck.Net, and The Power of Payment are registered trademarks of CyberSource Corporation.

CyberSource, CyberSource Payment Manager, CyberSource Risk Manager, CyberSource Decision Manager, and CyberSource Connect are trademarks and/or service marks of CyberSource Corporation.

All other brands and product names are trademarks or registered trademarks of their respective owners.

Contents

Recent Revisions to This Document 8

About This Guide 12

Audience and Purpose 12

Scope 12

Conventions 13

Note, Important, and Warning Statements 13

Text and Command Conventions 13

Related Documents 14

Customer Support 14

Chapter 1 Introducing Payer Authentication 15

Overview of Chargeback Protection 15

PSD2 16

3D Secure 2.x 16

Prerequisites for Implementing Payer Authentication 16

Integrating Payer Authentication into Your Business 17

Implementing 3D Secure 2.x 18

Scenario 1: You are a New Merchant 18

Scenario 2: You Use the CyberSource Simple Order API and Payer Authentication

Services for 3D Secure 1.0 19

Scenario 3: You Want to Integrate Using an SDK for your Mobile Application 20

Using Secure Acceptance with Payer Authentication 21

Required Merchant Information 21

Chapter 2 Implementing Cardinal Cruise Direct Connection API Payer Authentication 22

Implementation Overview 22

Process Flow for Cardinal Cruise Direct Connection API 23

Before You Begin 24

Credentials/API Keys 24

Create the JSON Web Token (JWT) 24

JWT Claims 24

JWT Examples 25

Choose Your Device Data Collection Option 26

Prerequisites	26
Endpoints	26
Device Data Collection Options	27
Option 1: Card BIN in JWT	27
Option 2: Card BIN as a POST Parameter Plus JWT	28
Implementing Cardinal Cruise Direct Connection API Payer Authentication	29
Requesting the Check Enrollment Service (Cardinal Cruise Direct Connection API)	29
Interpreting the Reply	30
Authenticating Enrolled Cards	31
Receiving the Authentication Results	33
Requesting the Validation Service	33
Interpreting the Reply	35
Redirecting Customers to Pass or Fail Message Page	35

Chapter 3	Implementing SDK Payer Authentication	36
Implementation Overview	36	
Process Flow for SDK Integration	37	
Before You Begin	38	
Credentials/API Keys	38	
Create the JSON Web Token (JWT)	38	
JWT Claims	39	
JWT Examples	40	
Using the Android SDK	41	
Update the Gradle Build Properties	41	
Configure the Android SDK	42	
Set Up the Initial Call	44	
Using the iOS SDK	45	
Download and Import the SDK	45	
Set Up Your Build Environment	45	
Configure the iOS SDK	45	
Set Up the Initial Call	48	
Implementing SDK Payer Authentication	50	
Requesting the Check Enrollment Service (SDK)	50	
Interpreting the Reply	51	
Authenticating Enrolled Cards	52	
Call Cardinal.cca_continue (Android SDK)	53	
Call Cardinal session continue (iOS SDK)	54	
Receiving the Authentication Results	56	
Requesting the Validation Service	56	
Interpreting the Reply	58	
Redirecting Customers to Pass or Fail Message Page	58	

Chapter 4 Upgrading Your Payer Authentication Implementation 59

Upgrading to 3D Secure 2.x 59

Benefits 59

PSD2 Impact 60

Mandates 60

Timelines 60

Recommended Integration 60

Migration FAQ 61

Chapter 5 Testing Payer Authentication Services 63

Testing Process 63

Enrollment Check 63

Authentication Validation 64

Expected Results 65

Test Cases for 3D Secure 1.0 67

Visa Secure 67

Mastercard Identity Check 76

Maestro 84

American Express SafeKey 88

JCB J/Secure 94

Diners Club ProtectBuy 100

Discover ProtectBuy 106

Test Cases for 3D Secure 2.x 112

Visa Secure 112

Mastercard Identity Check 119

American Express SafeKey 127

Discover ProtectBuy and Diners Club ProtectBuy 135

Appendix A API Fields 142

Formatting Restrictions 142

Data Type Definitions 142

Numbered Elements 143

Request Fields 144

Reply Fields 164

Appendix B Reason Codes 183

Appendix C	Request and Reply Examples	184
	Standard Integration Examples	184
	Check Enrollment Request Example	184
	Check Enrollment Reply Example	185
	Hybrid Integration Examples	186
	Check Enrollment Request Example	186
	Check Enrollment Reply Example	187
	Validate Authentication Request Example	188
	Validate Authentication Reply Example	189

Appendix D	Web Site Modification Reference	190
	Web Site Modification Checklist	190
	3D Secure Services Logos	191
	Informational Message Examples	192

Appendix E	Payer Authentication Transaction Details in the Business Center	193
	Searching for Payer Authentication Details	193
	Enrolled Card	193
	Enrollment Check	193
	Authentication Validation	194
	Card Not Enrolled	195
	Transaction Details	195
	Payer Authentication Search	195
	Storing Payer Authentication Data	196

Appendix F	Payer Authentication Reports	197
	Payer Authentication Summary Report	197
	Downloading the Report	198
	Matching the Report to the Transaction Search Results	198
	Interpreting the Report	199
	Comparing Payer Authentication and Payment Reports	200
	Payer Authentication Detail Report	201
	Report Elements	201
	<Report>	201
	<PayerAuthDetail>	201
	<ProofXML>	203
	<VReq>	204
	<VRes>	205
	<PReq>	206
	<PRes>	207
	<AuthInfo>	209
	Examples	210
	Failed Enrollment Check	210
	Successful Authentication	211

Appendix G	Rules-Based Payer Authentication	212
	Available Rules	212
	API Replies	213
	Bypassed Authentication Transactions	213
	Risk-Based Bank Transactions	214

Appendix H	Implementing Hybrid or Standard Payer Authentication	215
	Hybrid Payer Authentication	215
	Implementation Overview	215
	Process Flow for Hybrid Integration	216
	Before You Begin	217
	Credentials/API Keys	217
	Create the JSON Web Token (JWT)	217
	Add the JavaScript	220
	BIN Detection	220
	Implementing Hybrid Payer Authentication	220
	Requesting the Check Enrollment Service (Hybrid)	220
	Authenticating Enrolled Cards	223
	Requesting the Validation Service	224
	Standard Payer Authentication	226
	Implementation Overview	226
	Process Flow for Standard Integration	227
	Before You Begin	228
	Credentials/API Keys	228
	Create the JSON Web Token (JWT)	228
	Add the JavaScript	230
	BIN Detection	230
	Implementing Standard Payer Authentication	231
	Starting Authentication	231
	Requesting the Check Enrollment Service (Standard)	232

Glossary	235
-----------------	------------

Recent Revisions to This Document

Release	Changes
March 2020	<ul style="list-style-type: none"> ■ Temporarily removed the Payer Authentication Setup service from Chapter 2, "Implementing Cardinal Cruise Direct Connection API Payer Authentication," on page 22. ■ Temporarily removed new fields for the Payer Authentication Setup service. ■ Temporarily moved the device data collection section from Appendix I to Chapter 2, "Implementing Cardinal Cruise Direct Connection API Payer Authentication," on page 22. ■ Updated Chapter 4, "Upgrading Your Payer Authentication Implementation," on page 59.
February 2020	<ul style="list-style-type: none"> ■ Added the Payer Authentication Setup service. See "Request the Payer Authentication Setup Service for Device Data Collection," page 25. ■ Added "Create the JSON Web Token (JWT)" section. ■ Added new reply fields for the Payer Authentication Setup service: <ul style="list-style-type: none"> • payerAuthSetupReply_deviceDataCollection URL • payerAuthSetupReply_reasonCode • payerAuthSetupReply_referenceID ■ Updated the Payer Authentication Detail Report sections in Appendix E, Payer Authentication Transaction Details in the Business Center and "Payer Authentication Detail Report." ■ Moved the Hybrid and Standard implementation chapters to Appendix H, Implementing Hybrid or Standard Payer Authentication. ■ Moved the device data collection section to Appendix I, Alternate Methods for Device Data Collection.
December 2019	<ul style="list-style-type: none"> ■ Added a high-level process description of Payer Authentication for the Cardinal Cruise Direct Connection API integration. ■ Added a new chapter for the Cardinal Cruise Direct Connection API integration. See Chapter 2, "Implementing Cardinal Cruise Direct Connection API Payer Authentication," on page 22. ■ Finalized the 3D Secure 2.x. fields. See Appendix A, "API Fields," on page 142. ■ Updated links for several of the card brands in "3D Secure Services Logos."

Release	Changes
October 2019	<p>Added 3D Secure 2.x fields. See Appendix A, "API Fields," on page 142.</p> <p>Note: The 3D Secure 2.x. fields are in beta release. The field descriptions, values, and lengths are subject to change.</p> <p>Request fields:</p> <ul style="list-style-type: none"> ■ billTo_httpBrowserColorDepth ■ billTo_httpBrowserJavaEnabled ■ billTo_httpBrowserJavaScriptEnabled ■ billTo_httpBrowserLanguage ■ billTo_httpBrowserScreenHeight ■ billTo_httpBrowserScreenWidth ■ billTo_httpBrowserTimeDifference ■ item_#_shippingDestinationTypes ■ payerAuthEnrollService_acquirerCountry ■ payerAuthEnrollService_acsWindowSize ■ payerAuthEnrollService_authenticationIndicator ■ payerAuthEnrollService_deviceChannel ■ payerAuthEnrollService_merchantFraudRate ■ payerAuthEnrollService_merchantScore ■ payerAuthEnrollService_priorAuthenticationData ■ payerAuthEnrollService_priorAuthenticationMethod ■ payerAuthEnrollService_priorAuthenticationReferenceID ■ payerAuthEnrollService_priorAuthenticationTime ■ payerAuthEnrollService_requestorInitiatedAuthenticationIndicator ■ payerAuthEnrollService_sdkMaxTimeout ■ payerAuthEnrollService_secureCorporatePaymentIndicator ■ payerAuthEnrollService_totalOffersCount ■ payerAuthEnrollService_whiteListStatus ■ shipTo_destinationTypes

Release	Changes
October 2019 (continued)	Reply fields: <ul style="list-style-type: none"> ■ card_bin ■ card_cardTypeName ■ payerAuthEnrollReply_acsRenderingType ■ payerAuthEnrollReply_acsTransactionID ■ payerAuthEnrollReply_authenticationStatusReason ■ payerAuthEnrollReply_authenticationType ■ payerAuthEnrollReply_authorizationPayload ■ payerAuthEnrollReply_cardholderMessage ■ payerAuthEnrollReply_challengeCancelCode ■ payerAuthEnrollReply_challengeRequired ■ payerAuthEnrollReply_directoryServerErrorCode ■ payerAuthEnrollReply_directoryServerErrorDescription ■ payerAuthEnrollReply_effectiveAuthenticationType ■ payerAuthEnrollReply_ivrEnabledMessage ■ payerAuthEnrollReply_ivrEncryptionKey ■ payerAuthEnrollReply_ivrEncryptionMandatory ■ payerAuthEnrollReply_ivrEncryptionType ■ payerAuthEnrollReply_ivrLabel ■ payerAuthEnrollReply_ivrPrompt ■ payerAuthEnrollReply_ivrStatusMessage ■ payerAuthEnrollReply_networkScore ■ payerAuthEnrollReply_sdkTransactionID ■ payerAuthEnrollReply_stepUpUrl ■ payerAuthEnrollReply_threeDSServerTransactionID ■ payerAuthEnrollReply_whiteListStatus ■ payerAuthEnrollReply_whiteListStatusSource ■ payerAuthValidateReply_acsRenderingType ■ payerAuthValidateReply_acsTransactionID ■ payerAuthValidateReply_authenticationStatusReason ■ payerAuthValidateReply_authenticationType ■ payerAuthValidateReply_authorizationPayload ■ payerAuthValidateReply_challengeCancelCode ■ payerAuthValidateReply_directoryServerErrorCode ■ payerAuthValidateReply_directoryServerErrorDescription

Release	Changes
October 2019 (continued)	<p>Reply fields: (continued)</p> <ul style="list-style-type: none"> ■ payerAuthValidateReply_effectiveAuthenticationType ■ payerAuthValidateReply_interactionCounter ■ payerAuthValidateReply_sdkTransactionID ■ payerAuthValidateReply_threeDSServerTransactionID ■ payerAuthValidateReply_whiteListStatus ■ payerAuthValidateReply_whiteListStatusSource
September 2019	<ul style="list-style-type: none"> ■ Added a new chapter for SDK integration. See Chapter 3, "Implementing SDK Payer Authentication," on page 36. ■ Removed the Action Codes section from Chapter 3, "Implementing Standard Payer Authentication," on page 34. ■ Updated the "Test Cases for 3D Secure 2.x" for the following: <ul style="list-style-type: none"> • Added the e-commerce indicator or ECI to several test cases where it was missing. • Removed the ECI from "Test Case 2.6: Visa Secure Card Enrolled: Authentication not Available on Lookup" and corresponding test cases for Mastercard, American Express, and Discover. • Removed the XID from the Check Enrollment results for "Test Case 2.10a: Visa Secure Card Enrolled: Successful Step-Up Authentication (Cruise Direct and Hybrid)" and corresponding test cases for Mastercard, American Express, and Discover. • In test cases where ECI=7, added vbv_failure as a possible value along with internet for the e-commerce indicator (Visa test cases only). • Updated test case titles to include <i>Hybrid</i> and <i>Standard</i> where applicable. • Removed the <i>Validate Authentication</i> column from Standard-only test cases. • Added a note and e-commerce indicator values to enable 2.x Discover test cases to be used for Diners Club ProtectBuy. • Removed the Step Up Authentication with Merchant Bypass test cases. • Removed the Incomplete Authentication test cases. • Renumbered the 2.x test cases.
August 2019	Added a high-level process description of Payer Authentication for the SDK integration. See "Required Merchant Information," page 21.

About This Guide

Audience and Purpose

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

Implementing CyberSource Payer Authentication services requires software development skills. You must write code that uses the API request and reply 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 CyberSource services other than payer authentication. For that information, see ["Related Documents," page 14](#).

Conventions

Note, Important, and Warning Statements



A *Note* contains helpful suggestions or references to material not contained in this document.



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



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

Text and Command Conventions

Convention	Usage
bold	<ul style="list-style-type: none"> Field and service names in text. For example: Include the ics_applications field. Items that you are instructed to act upon. For example: Click Save.
<i>italic</i>	<ul style="list-style-type: none"> Filenames and pathnames. For example: Add the filter definition and mapping to your <i>web.xml</i> file. Placeholder variables for which you supply particular values.
Screen text	<ul style="list-style-type: none"> XML elements. Code examples and samples. Text that you enter in an API environment. For example: Set the davService_run field to <code>true</code>.

Related Documents

- *Getting Started with CyberSource Advanced for the Simple Order API* describes how to get started using the Simple Order API. ([PDF](#) | [HTML](#))
- *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](#) | [HTML](#))
- *Credit Card Services Using the Simple Order API* describes how to integrate CyberSource payment processing services into your business. ([PDF](#) | [HTML](#))
- *Secure Acceptance Hosted Checkout Integration Guide* describes how to create Secure Acceptance profiles, which enable you to integrate your order management system with the Secure Acceptance web/mobile checkout. ([PDF](#) | [HTML](#))
- *Secure Acceptance Checkout API Integration Guide* describes how to create Secure Acceptance profiles, which enable you to integrate your order management system with a web site to process transactions. ([PDF](#) | [HTML](#))
- *Reporting Developer Guide* describes how to view and configure Business Center reports. ([PDF](#) | [HTML](#))
- The [CyberSource API Versions page](#) provides information about the CyberSource API versions.

Refer to the Support Center for complete CyberSource technical documentation:

http://www.cybersource.com/support_center/support_documentation

Customer Support

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

<http://www.cybersource.com/support>

Introducing Payer Authentication

CyberSource Payer Authentication services use JavaScript and the ICS services to provide authentication.

Payer Authentication services enable you to add support to your web store for card authentication services, including Visa SecureSM, Mastercard Identity Check[®], Maestro[®] (UK Domestic and international), American Express SafeKeySM, JCB J/Secure[™], Diners Club ProtectBuy, and Discover ProtectBuy.

These card authentication services deter unauthorized card use and protect you from fraudulent chargeback activity referred to as *liability shift*. However, CyberSource Payer Authentication is not a fraud management service, such as Decision Manager with Advanced Fraud Screen. CyberSource recommends 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 Simple Order API* ([PDF](#) | [HTML](#)).

Overview of Chargeback Protection

Visa, Mastercard, Maestro, American Express, JCB, Diners Club, and Discover offer chargeback protection if merchants participate in [3D 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. CyberSource recommends that you contact your merchant account provider to find out exactly how to interpret chargeback requirements and 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 (including the UK even if departed from the EEA) 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 be applied for all electronic payments.

3D Secure 2.x

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

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

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 web site to facilitate the authentication.
- Add specific data to your API requests to CyberSource.
- Validate the necessary data.
- Provide the additional data to the authorization request.
- Modify your web site to help the customer understand the process.

Integrating Payer Authentication into Your Business

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

Four types of integration are available:

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



If you are using tokenization, you must use the Hybrid integration method.



The SDK integration is designed for 3D Secure 2.x transactions only.

The Cardinal Cruise Direct Connection API is the recommended integration method.

If you are currently using 3D Secure 1.0, see [Chapter 4, "Upgrading Your Payer Authentication Implementation," on page 59](#).

You can also use Secure Acceptance to enable 3D Secure 2.x for payer authentication services. For more information, see ["Using Secure Acceptance with Payer Authentication."](#)

Implementing 3D Secure 2.x

Scenario 1: You are a New Merchant

-
- Step 1** Contact your CyberSource account manager or sales manager to learn more about 3D Secure 2.x and PSD2.
- Step 2** Set up your merchant ID with CyberSource.
- a** Contact Customer Support to enable 3D Secure 2.x for the desired card type, currencies, and acquiring bank. For details, see ["Required Merchant Information."](#)
You must request additional services such as payments using Secure Acceptance at this time; you can request additional services such as token management as well.
 - b** Log in to the Business Center to obtain the API keys for implementation.
- Step 3** Implement 3D Secure 2.x.
- Using CyberSource Simple Order API:
 - a** Use the Cardinal Cruise Direct Connection API.
 - b** 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"](#) section and [Chapter 2, "Implementing Cardinal Cruise Direct Connection API Payer Authentication," on page 22.](#)

You can configure your system to request payment services through CyberSource along with your payer authentication for 3D Secure 2.x; however, it is not required.
- Step 4** Test your 3D Secure 2.x services.
This testing ensures that you understand the possible use cases as part of implementation.
Refer to [Chapter 5, "Testing Payer Authentication Services," on page 63](#) and run the test cases in ["Test Cases for 3D Secure 2.x."](#)
- Step 5** Configure your account for production.
- a** Request a boarding form from Customer Support for your processor or acquirer.
 - b** Complete the boarding form with required information including your CyberSource merchant ID, your acquirer merchant ID, and BIN information for all chosen card types. For details, see ["Required Merchant Information."](#)
-

Scenario 2: You Use the CyberSource Simple Order API and Payer Authentication Services for 3D Secure 1.0

- Step 1** Contact your CyberSource account manager, sales manager, or technical account manager to learn more about 3D Secure 2.x and PSD2.
- Step 2** Configure your test merchant ID with CyberSource.
- a** Contact Customer Support to make the necessary configuration changes to enable 3D Secure 2.x for the desired card type and currencies.
 - b** Log in to the Business Center to obtain the API keys for implementation.
- Step 3** Implement 3D Secure 2.x by migrating to the Hybrid integration.
- a** Add the CardinalCommerce JavaScript code to your checkout page.
 - b** 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 [Chapter 4, "Upgrading Your Payer Authentication Implementation," on page 59](#).
- You can configure your system to request payment services through CyberSource along with your payer authentication for 3D Secure 2.x; however, it is not required.
- Step 4** Test your 3D Secure 2.x services.
- This testing ensures that you understand the possible use cases as part of implementation.
- Refer to [Chapter 5, "Testing Payer Authentication Services," on page 63](#) and run the test cases in ["Test Cases for 3D Secure 2.x."](#)
- Step 5** Configure your account for production. Repeat Steps 2-4 for the production environment.
-

Scenario 3: You Want to Integrate Using an SDK for your Mobile Application

- Step 1** Contact your CyberSource account manager, sales manager, or technical account manager to learn more about 3D Secure 2.x and PSD2.
- Step 2** Configure your test merchant ID with CyberSource.
- a** Contact Customer Support to make the necessary configuration changes to enable 3D Secure 2.x for the desired card type and currencies. For details, see ["Required Merchant Information."](#)
 - b** Log in to the Business Center to obtain the API keys for implementation.
- Step 3** Implement 3D Secure 2.x.
- You must use the CyberSource Simple Order API as well as the SDK in order to implement a native mobile application. SDKs are available for iOS or Android.
- a** Implement the SDK to handle authentication steps within the native application. The SDKs are the CardinalCommerce JavaScript equivalent for mobile applications. For more information see [Chapter 3, "Implementing SDK Payer Authentication," on page 36.](#)
 - b** 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.
- You can configure your system to request payment services through CyberSource along with your payer authentication for 3D Secure 2.x; however, it is not required.
- Step 4** Test your 3D Secure 2.x services.
- This testing ensures that you understand the possible use cases as part of implementation.
- Refer to [Chapter 5, "Testing Payer Authentication Services," on page 63](#) and run the test cases in ["Test Cases for 3D Secure 2.x."](#)
- Step 5** Configure your account for production. Repeat Steps 2-4 for the production environment.
-

Using Secure Acceptance with Payer Authentication

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

For more information on implementing Secure Acceptance with payer authentication, see the *Secure Acceptance Hosted Checkout Integration Guide* or *Secure Acceptance Checkout API Integration Guide*.

Required Merchant Information

Before using CyberSource Payer Authentication services in production, you must contact Customer Support to provide information about your company and your acquiring bank so that CyberSource can configure your account to implement these services.

You must provide the information listed in [Table 1](#) to CyberSource before payer authentication services can be enabled:

Table 1 Merchant Information Required for Payer Authentication Services

Information	Description
About your company	<ul style="list-style-type: none"> Your CyberSource merchant ID. URL of your company's web site, for example: http://www.example.com Two-character ISO code for your country. 3D Secure requestor ID 3D Secure requestor name Merchant category code
Bank information	<ul style="list-style-type: none"> Name of your bank acquirer. Complete name and address of your bank contact, including email address.
Visa, Mastercard, Maestro, American Express, JCB, Diners Club, and Discover information	Information provided by your bank acquirer about each payment card company for which you are configured: <ul style="list-style-type: none"> 6-digit BIN numbers.
Acquirer merchant ID	<ul style="list-style-type: none"> 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 is a non-JavaScript implementation that supports 3D Secure 2.x and is backward-compatible with 3D Secure 1.0 when the issuer, acquirer, or both, are not ready for 2.x. This integration allows you to use an iframe to complete the device profiling and 3D 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 webpage.

Implementation Overview

Notify your CyberSource account representative that you want to implement payer authentication (3D Secure). Give them the CyberSource merchant ID that you will use for testing. For more information, see ["Required Merchant Information," page 21](#).

Implementation tasks include:

- Choose your device data collection option and use an iframe to complete device profiling
- For each purchase request
 - Build the authentication request
 - Call the **payerAuthEnrollService**: Payer Authentication Enrollment Check service
 - If required, use an iframe to complete the 3D Secure authentication requirements
 - Handle the authentication response (in the return URL in the step-up JWT)
 - Call the following services in the same request:
 - **payerAuthValidateService**: Payer Authentication Validation
 - **ccAuthService**: Card Authorization service (optional)

- Use the test cases to test your preliminary code and make appropriate changes. See [Chapter 5, "Testing Payer Authentication Services,"](#) on page 63.
- Ensure that your account is configured for production.

Process Flow for Cardinal Cruise Direct Connection API

- 1 You generate a JSON Web Token (JWT).
- 2 You choose your device data collection option.
- 3 You request the Enrollment Check service, passing in transaction details and the payerAuthEnrollService_referenceID request field.
- 4 If the issuing bank does not require step-up authentication, you receive the following information in the Enrollment Check reply:
 - E-commerce indicator
 - CAVV (all card types except Mastercard)
 - AAV (Mastercard only)
 - Transaction ID
 - 3D Secure version
 - Directory server transaction ID
- 5 If the issuing bank requires step-up authentication, you receive a response with the ACS URL of the issuing bank, the payload, the transaction ID, and the step-up URL. Create a new step-up JWT and include the ACS URL, payload, transaction ID, and the merchant's return URL.
- 6 Pass the JWT to your web front-end and create an iframe to POST the step-up JWT to the step-up URL.
- 7 The iframe displays the authentication window, and the customer enters the authentication information.
- 8 You get the session back through the return URL in the step-up JWT sent by the merchant. The response posted back to the return URL contains the transaction ID and the URL-encoded and Base64-encoded payload.
- 9 You request the Validate Authentication service, sending the transaction ID in the payerAuthValidateService_authenticationTransactionID request field. You receive the e-commerce indicator, CAVV or AAV, transaction ID, 3D 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," page 33](#).

Before You Begin

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.

Credentials/API Keys

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

Create the JSON Web Token (JWT)

The Cardinal Cruise Direct Connection API integration uses JWTs as the method of authentication.



Note

For security reasons, all JWT creation must be done on the server side.

When creating the JWT, use your company API Key as the JWT secret. You can use any JWT library that supports JSON Web Signature (JWS). For further information about JWTs, see <https://jwt.io/>.

JWT Claims

[Table 2](#) lists the standard claims that can be used in a JWT claim set.

Table 2 JWT Claims

Claim Name	Description
Required	Note Each claim key is case sensitive.
jti	JWT ID - unique identifier for the JWT. This field should change each time a JWT is generated.
iat	Issued at - the epoch time in seconds beginning when the JWT is issued. This value indicates how long a JWT has existed and can be used to determine if it is expired.

Table 2 JWT Claims (Continued)

Claim Name		Description
	iss	Issuer - identifier of who is issuing the JWT. Contains the API key identifier or name.
	OrgUnitId	The merchant SSO Org Unit Id.
	Payload	The JSON data object being sent. This object is usually an order object.
Optional	ReferenceId	Merchant-supplied identifier that can be used to match up data collected from the Cardinal Cruise Direct Connection API and enrollment check service.
	ObjectifyPayload	Boolean flag that indicates how the API should consume the payload claim. If set to true, the payload claim is an object. If set to false, the payload claim is a stringified object. Some JWT libraries do not support passing objects as claims; this allows those who only allow strings to use their libraries without customization.
	exp	Expiration - the numeric epoch time in which the JWT should be considered expired. This value is ignored if it is more than 4 hours.

JWT Examples

Example 1 shows the JSON content of a basic JWT payload that passes an object within the payload claim.

Example 1 Raw JWT

```
{
  "jti": "a5a59bfb-ac06-4c5f-be5c-351b64ae608e",
  "iat": 1448997865,
  "iss": "56560a358b946e0c8452365ds",
  "OrgUnitId": "565607c18b946e058463ds8r",
  "Payload": {
    "OrderDetails": {
      "OrderNumber": "0e5c5bf2-ea64-42e8-9ee1-71fff6522e15",
      "Amount": "1500",
      "CurrencyCode": "840"
    }
  },
  "ObjectifyPayload": true,
  "ReferenceId": "c88b20c0-5047-11e6-8c35-8789b865ff15",
  "exp": 1449001465,
}
```

[Example 2](#) shows the JSON content of a basic JWT payload that passes a string within the payload claim.

Example 2 Stringified JWT

```
{
  "jti": "29311a10-5048-11e6-8c35-8789b865ff15",
  "iat": 1448997875,
  "iss": "56560a358b946e0c8452365ds",
  "OrgUnitId": "565607c18b946e058463ds8r",
  "Payload": "{\"OrderDetails\":{\"OrderNumber\":\"19ec6910-5048-11e6-8c35-8789b865ff15\", \"Amount\":\"1500\", \"CurrencyCode\":\"840\"}}",
  "ObjectifyPayload" false
  "ReferenceId": "074fda80-5048-11e6-8c35-8789b865ff15"
  "exp": 1449001465,
}
```

Choose Your Device Data Collection Option

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

The Cardinal Cruise Direct Connection API places the required Method URL on the merchant's site within an iframe if the issuing bank chooses to use one. (According to EMV 3D Secure requirements, a merchant must place and run the Method URL on their web site if the issuing bank uses one.)

The Method URL is a concept in the EMV 3D Secure protocol that allows 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 3D Secure protocol. This process also occurred in 3D 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 8 digits or full card number of cardholder).
- Create an iframe on your web site 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>

Device Data Collection Options

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 8 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.

Option 1: 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.

-
- Step 1** Add the card BIN (first 8 digits or full card number) to the transactional JWT.
 - Step 2** Create a POST request to send the transactional JWT to the device data collection URL.
 - Step 3** Handle the response from the device data collection URL on the return URL provided within the transactional JWT.

[Example 3](#) shows the return URL populated in the transactional JWT instead of a POST parameter.

Example 3 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>
```

Option 2: 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.

Step 1 Create a POST request to send the transactional JWT and the card BIN (first 8 digits or full card number) to the device data collection URL.

Step 2 Handle the response from the device data collection URL on the return URL provided within the transactional JWT.

[Example 4](#) shows the return URL populated in the transactional JWT along with a POST parameter.

Example 4 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 8 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>
```

Implementing Cardinal Cruise Direct Connection API Payer Authentication

Requesting the Check Enrollment Service (Cardinal Cruise Direct Connection API)

After device collection completes, and when the customer clicks the 'buy now' button, you must request the Enrollment Check 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_mobilePhone`
- `payerAuthEnrollService_referenceID`
- `payerAuthEnrollService_run`
- `purchaseTotals_currency`
- `purchaseTotals_grandTotalAmount`

**Note**

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 144](#).

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 step-up payer authentication is not required. In this case, the field values that are required in order 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 Reply Fields) in the authorization service request (Card Authorization Request Fields).

[Table 3](#) lists these fields.

Table 3 Enrollment Check and Reply Fields

Identifier	Enrollment Check Reply Field	Card Authorization Request Field
E-commerce indicator	payerAuthEnrollReply_commerceIndicator	ccAuthService_commerceIndicator
Collection indicator (Mastercard only)	payerAuthEnrollReply_ucafCollectionIndicator	ucaf_collectionIndicator
Result of the enrollment check for Asia, Middle East, and Africa Gateway	payerAuthEnrollReply_veresEnrolled	ccAuthService_veresEnrolled
3D Secure version	payerAuthEnrollReply_specificationVersion	ccAuthService_paSpecificationVersion
Directory server transaction ID Note Not required for 3D Secure 1.0.	payerAuthEnrollReply_directoryServerTransactionID	ccAuthService_directoryServerTransactionID

Interpreting the Reply

The replies are similar for all card types. See [Appendix C, "Request and Reply Examples," on page 184](#) for examples of enrollment replies.

- Enrolled cards

You receive reason code 475 if the customer's card is enrolled in a payer authentication program. When you receive this reply, you can proceed to validate authentication.

- Cards not enrolled or step-up authentication not required

You receive reason code 100 in the following cases:

- When the account number is not enrolled in a payer authentication program or when step-up authentication is not required. The other services in your request are processed normally.
- When payer authentication is not supported by the card type.

When you receive this reply, you can proceed to card authorization. If you receive the authentication results along with reason code 100, you receive liability shift protection, and you do not need to pass these results to authorization.

Authenticating Enrolled Cards

The Cardinal Cruise Direct Connection API integration uses the **payerAuthEnrollReply_stepUpUrl**reply field to manage customer interaction with the URL of the card-issuing bank's [Access Control Server](#) (ACS) and 3D Secure version compatibility for 3D Secure 1.0 and 3D Secure 2.x.



Contact Customer Support to make the necessary configuration changes to receive the step-up URL reply field.

You might receive the ACS URL, payload, **payerAuthEnrollReply_authenticationTransactionID** reply field, and **payerAuthEnrollReply_stepUpUrl**reply field. When these fields are populated, you can use them to decide whether to present the authentication session to the customer.

When a frictionless response is returned, which means step-up authentication is not required, then the ACS URL, payload, and step-up URL fields are not populated.

If you need to present the authentication session to the customer, you must generate a new step-up JWT with the ACS URL, payload, transaction ID, and merchant's return URL (see [Example 5](#)). When adding these values to the JWT, you will securely sign them with a secret key provided during the onboarding process.

Example 5 Step-Up JWT

```
{
  "jti": "4595beb0-a4a9-11e8-8fd8-bdf5ff435fec",
  "iat": 1534790755,
  "iss": "Midas-XXXXX-Key",
  "OrgUnitId": "59c2745f2f3e7357b4aa516a",
  "ReturnUrl": "http://localhost:8189/cart/enterprise/term",
  "ReferenceId": "c88b20c0-5047-11e6-8c35-8789b865ff15",
  "Payload": {
    "ACSTUrl": "https://merchantacsdev.cardinalabs.com/MerchantACSWeb/
pareq.jsp?vaa=b&gold=AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA",
    "Payload":
    "eNpVUV1PwjAUfe+vIMRX13ZAMsmlCTKJmIAwBzw3XeMaWTe6TfHf25ZNtE/
3nNv7cc6FNDdSxm9StEYyWMu65u9yoLLZse7W2+NiV+7MkzpcZJ7qlgwZbOeJPDP4lKZWpW
Y0IEEIuIfItjAi57phwMX5cbVh1D3AHUJQSLOKr6xPXDECzQvJlirj9X36VcZl85oD9iQCU
ba6Md8sGhPAPUDQmhPLm6aaYtFfj61VIwNRFoBdEGG+7bNtXVRbmReVMUffNvtlFu+PD4n4
OFQHnZ/
4MmmzdD4D7H4gyHgJWUhoRKKQDGG0HU+mI6vW8wh44TZhd3QSEltXBxFUbtD8iujEpf4yVk
1rjNSil9MjBPJS1VraP3bGb2w13DZfPDtfrWMDg9HQ2+qBr1XWlJCQks9W3iHsCnB3M9yd1
0b/zv4DGhKoaw==",
    "TransactionId": "sRMPWCQoQrEiVxehTnu0"
  },
  "ObjectifyPayload": true
}
```

Once you have the step-up JWT, you must pass it to your web front-end and create an iframe to POST the step-up JWT to the step-up URL (see [Example 6](#)).

The size of the iframe can vary depending on the 3D Secure version of the transaction (1.0 or 2.x). For 3D Secure 2.x transactions, the size of the challenge window can be sent in the **payerAuthEnrollService_acsWindowSize** request field.

Example 6 POST to Step-Up URL

```
<iframe height="250" width="400">

  <form name="stepup" method="POST" action="*_StepUpURL returned in
the response*_ "> <input type="hidden" name="JWT" value="JWT generated
by merchant per spec" /> <input type="hidden" name="MD" value="Any
merchant specific data that will be passed back to the return URL AS-IS"
/> </form>

</iframe>
```

Receiving the Authentication Results

After the customer interacts with the issuer's ACS, you must get the session back through the return URL in the step-up JWT sent by the merchant. The payload sent to the return URL is URL-encoded and Base64-encoded (see [Example 7](#)).

Example 7 POST to Return URL

```
TransactionId=BwNsDeDPsQV4q8uy1Kq1&Response=eyJtZXNzYWdlVHlwZSI6IkdNSZXM  
iLcJtZXxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxcHJvdGVjdGlvbiEifQ%3D%3  
D
```

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

- **TransactionId**: you must send the transaction ID value in the **payerAuthValidateService_authenticationTransactionID** request field when you request the Validate Authentication service.
- **Response**: contains the payload from the issuer ACS and the PAREs. You might include these values when you request the Validate Authentication service.

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 reply that you receive contains the validation result.

CyberSource recommends that you request both payer authentication and card authorization services at the same time. When you do so, CyberSource automatically sends the correct information to your payment processor; CyberSource converts the values of these fields 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 Reply 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, and Discover 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 3D Secure 1.0 or 2.x transaction, you may not receive the XID.

Table 4 lists these fields.

Table 4 Validation Check and Reply Fields

Identifier	Validation Check Reply 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
3D Secure version	payerAuthValidateReply_specificationVersion	ccAuthService_paSpecificationVersion
Directory server transaction ID	payerAuthValidateReply_directoryServerTransactionID	ccAuthService_directoryServerTransactionID
Note Not required for 3D Secure 1.0.		

Interpreting the Reply



Important

If the authentication fails, Visa, American Express, JCB, Diners Club, and Discover require that you do 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 replies are similar for all card types:

- *Success:*

You receive the 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 CyberSource 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 replies 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 nonenrolled 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.
```

Implementing SDK Payer Authentication

This chapter summarizes the process of integrating SDK Payer Authentication services into your mobile application. CyberSource Payer Authentication services use the Cardinal Mobile SDK for iOS or Android to facilitate the authentication.

The SDK is only designed to handle 2.0 transactions. If a 1.0 transaction occurs, you must include functionality to open up a WebView.

Implementation Overview

Notify your CyberSource account representative that you want to implement payer authentication (3D Secure). Give them the CyberSource merchant ID that you will use for testing. For more information, see ["Required Merchant Information," page 21](#).

Implementation tasks include:

- Download, import, and configure the Cardinal Mobile SDK for either iOS or Android
- For each purchase request:
 - Build the authentication request
 - Make a back-end, server-to-server call to request the **payerAuthEnrollService**: Payer Authentication Enrollment Check service
 - 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 [Chapter 5, "Testing Payer Authentication Services," on page 63](#).
- Ensure that your account is configured for production.

Process Flow for SDK Integration

- 1 You generate a JSON Web Token (JWT).
- 2 Contact CardinalCommerce Customer Support for instructions to register for an API key.
- 3 Download and import the Cardinal Mobile SDK for either iOS or Android.
- 4 Set up your build environment.
- 5 Configure your SDK.
- 6 Call *Cardinal session.setup()*.
- 7 Create an API call to your merchant server to request the Enrollment Check service, passing in transaction details and the **payerAuthEnrollService_referenceID** request field.
- 8 If the issuing bank does not require authentication, you receive the following information in the Enrollment Check reply:
 - E-commerce indicator
 - CAVV (all card types except Mastercard)
 - AAV (Mastercard only)
 - Transaction ID
 - 3D Secure version
 - Directory server transaction ID
- 9 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.
- 10 The Cardinal Mobile SDK displays the authentication window, and the customer enters the authentication information.
- 11 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.
- 12 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, 3D 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," page 56](#).

Before You Begin

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.

Implementing the SDK in your mobile application requires either Android or iOS platform application programming skills. Android API 19 or iOS 8 and XCode 8 are required.

Credentials/API Keys

API keys are required in order to create the JSON Web Token (JWT). For instructions to register for your Bintray user name and API keys, contact CyberSource Customer Support.

You will receive an email invitation from `noreply@bintray.com` to join JFrog Bintray. When you receive the email, click to join the organization and complete your registration.

To create your registration user name, include your merchant name and Cybersource in the following format:

`{Insert Merchant Name Here}_Cybersource`

When you are fully registered, `@cardinalcommerce` is added to the end of your user name. The final format of your user name will be:

`{Insert Merchant Name Here}_Cybersource@cardinalcommerce`

Once you complete your registration, you will be able to access your API keys.

Create the JSON Web Token (JWT)

The Cardinal Cruise Direct Connection API integration uses JWTs as the method of authentication.



For security reasons, all JWT creation must be done on the server side.

When creating the JWT, use your company API Key as the JWT secret. You can use any JWT library that supports JSON Web Signature (JWS). For further information about JWTs, see <https://jwt.io/>.

JWT Claims

Table 5 lists the standard claims that can be used in a JWT claim set.

Table 5 JWT Claims

Claim Name	Description
Required	Note Each claim key is case sensitive.
jti	JWT ID - unique identifier for the JWT. This field should change each time a JWT is generated.
iat	Issued at - the epoch time in seconds beginning when the JWT is issued. This value indicates how long a JWT has existed and can be used to determine if it is expired.
iss	Issuer - identifier of who is issuing the JWT. Contains the API key identifier or name.
OrgUnitId	The merchant SSO Org Unit Id.
Payload	The JSON data object being sent. This object is usually an order object.
Optional	
ReferenceId	Merchant-supplied identifier that can be used to match up data collected from the Cardinal Cruise Direct Connection API and enrollment check service.
ObjectifyPayload	Boolean flag that indicates how the API should consume the payload claim. If set to true, the payload claim is an object. If set to false, the payload claim is a stringified object. Some JWT libraries do not support passing objects as claims; this allows those who only allow strings to use their libraries without customization.
exp	Expiration - the numeric epoch time in which the JWT should be considered expired. This value is ignored if it is more than 4 hours.

JWT Examples

[Example 8](#) shows the JSON content of a basic JWT payload that passes an object within the payload claim.

Example 8 Raw JWT

```
{
  "jti": "a5a59bfb-ac06-4c5f-be5c-351b64ae608e",
  "iat": 1448997865,
  "iss": "56560a358b946e0c8452365ds",
  "OrgUnitId": "565607c18b946e058463ds8r",
  "Payload": {
    "OrderDetails": {
      "OrderNumber": "0e5c5bf2-ea64-42e8-9ee1-71fff6522e15",
      "Amount": "1500",
      "CurrencyCode": "840"
    }
  },
  "ObjectifyPayload": true,
  "ReferenceId": "c88b20c0-5047-11e6-8c35-8789b865ff15",
  "exp": 1449001465,
}
```

[Example 9](#) shows the JSON content of a basic JWT payload that passes a string within the payload claim.

Example 9 Stringified JWT

```
{
  "jti": "29311a10-5048-11e6-8c35-8789b865ff15",
  "iat": 1448997875,
  "iss": "56560a358b946e0c8452365ds",
  "OrgUnitId": "565607c18b946e058463ds8r",
  "Payload": "{\"OrderDetails\":{\"OrderNumber\":\"19ec6910-5048-11e6-8c35-8789b865ff15\",\"Amount\":\"1500\",\"CurrencyCode\":\"840\"}}",
  "ObjectifyPayload": false,
  "ReferenceId": "074fda80-5048-11e6-8c35-8789b865ff15",
  "exp": 1449001465,
}
```

Using the Android SDK

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. Make sure that you edit the Gradle file that is located in the app directory. Add the contents shown in [Example 10](#) to the Gradle file.

Example 10 Gradle Build Properties

```
repositories {
    ...
    maven {
        url "https://cardinalcommerce.bintray.com/android"
        credentials {
            username '//Bintray username
            password '//Bintray user API Key
        }
    }
}

dependencies {
    ...
    //Cardinal Mobile SDK
    implementation
    'org.jfrog.cardinalcommerce.gradle:cardinalmobilesdk:2.1.4-1'

    // Required libraries
    implementation group: 'com.nimbusds', name: 'nimbus-jose-jwt',
version: '7.0.1'
    implementation group: 'org.bouncycastle', name: 'bcprov-jdk15on',
version: '1.61'
}
```

Configure the Android SDK

Get the instance of the cardinal object by *Cardinal.getInstance()*. You can set multiple configuration options in the SDK; if not specified, they are set to default. See [Example 11](#) to complete *Cardinal.configure()*.

For more details on configuration options, see [Table 6](#).

Example 11 Cardinal.configure

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

    CardinalConfigurationParameters cardinalConfigurationParameters = new
    CardinalConfigurationParameters();

    cardinalConfigurationParameters.setEnvironment(CardinalEnvironment.STAG
    ING);

        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(yourUICustomizationO
    bject);

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

Table 6 Android SDK Configuration Options

Method	Description	Default Values
setEnabledDFSync (boolean enableDFSync)	On setting true, onSetupCompleted will be called after device data collected is sent to the server.	False
setEnabledQuickAuth (boolean enableQuickAuth)	Sets enable quick auth false.	False

Table 6 Android SDK Configuration Options (Continued)

Method	Description	Default Values
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(CardinalRenderType.OTP); rType.put(CardinalRenderType.SINGLE_SELECT); rType.put(CardinalRenderType.MULTI_SELECT); rType.put(CardinalRenderType.OOB); rType.put(CardinalRenderType.HTML);
setTimeout(int timeout)	Sets the maximum amount of time (in milliseconds) for all exchanges.	8000
setUICustomization (UiCustomization UI Customization)	Sets UICustomization	Device Default Values
setUiType(CardinalUiType uiType)	Sets all 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 in [Example 12](#) for completing the *cardinal.init()*.

Example 12 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 ["Implementing SDK Payer Authentication"](#) section for next steps.

Using the iOS SDK

Download and Import the SDK

Download the *CardinalMobile.framework* file using the cURL in [Example 13](#).

Example 13 Download cURL

```
curl -L -u<USER_NAME>
      :<API_KEY> https://cardinalcommerce.bintray.com/ios/<VERSION>-
<BUILD_NUMBER>/cardinalmobilesdk.zip
      -o <LOCAL_FILE_NAME.EXT>
```

#Example:

```
curl -L -uUserName:ApiKey "https://cardinalcommerce.bintray.com/ios/
2.1.4-2/cardinalmobilesdk.zip" -o cardinalmobile2.1.4-2.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

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

Configure the iOS SDK

Create a new instance of the cardinal object by *CardinalSession new*. You can set multiple configuration options in the SDK; if not specified, they are set to default. See [Example 14](#) or [Example 15](#) to complete the iOS SDK configuration.

For more details on configuration options, see [Table 7](#).

Objective-C Example

Example 14 CardinalSession new (iOS SDK - Objective-C)

```

#import <CardinalMobile/CardinalMobile.h>

CardinalSession *session;

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

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

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

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

```

Swift Example

Example 15 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)
}
```

Table 7 iOS SDK Configuration Options

Method	Description	Default Values	Possible Values
deploymentEnvironment	The environment to which the SDK connects.	CardinalSessionEnvironmentProduction	CardinalSessionEnvironmentStaging CardinalSessionEnvironmentProduction
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

Table 7 iOS SDK Configuration Options (Continued)

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

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. Use the code example shown in [Example 16](#) or [Example 17](#) for completing the *cardinal session setup*. The function call must be placed in your Checkout ViewController.

Objective-C Example

Example 16 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
}];

```

Swift Example

Example 17 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
}

```

Implementing SDK Payer Authentication

Requesting the Check Enrollment Service (SDK)

After the SDK completes the device collection from your mobile application and once 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 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_mobilePhone`
- `payerAuthEnrollService_referenceID`
- `payerAuthEnrollService_run`
- `purchaseTotals_currency`
- `purchaseTotals_grandTotalAmount`

**Note**

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 144](#).

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 in order 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 Reply Fields) in the authorization service request (Card Authorization Request Fields).

Table 8 lists these fields.

Table 8 Enrollment Check and Reply Fields

Identifier	Enrollment Check Reply Field	Card Authorization Request Field
E-commerce indicator	payerAuthEnrollReply_commerceIndicator	ccAuthService_commerceIndicator
Collection indicator (Mastercard only)	payerAuthEnrollReply_ucafCollectionIndicator	ucaf_collectionIndicator
Result of the enrollment check for Asia, Middle East, and Africa Gateway	payerAuthEnrollReply_veresEnrolled	ccAuthService_veresEnrolled
3D Secure version	payerAuthEnrollReply_specificationVersion	ccAuthService_paSpecificationVersion
Directory server transaction ID Note Not required for 3D Secure 1.0.	payerAuthEnrollReply_directoryServerTransactionID	ccAuthService_directoryServerTransactionID

Interpreting the Reply

The replies are similar for all card types. See [Appendix C, "Request and Reply Examples," on page 184](#) for examples of enrollment replies.

■ Enrolled Cards

You receive reason code 475 if the customer's card is enrolled in a payer authentication program. When you receive this reply, you can proceed to validate authentication.

- *Cards Not Enrolled*

You receive reason code 100 in the following cases:

- When the account number is not enrolled in a payer authentication program. The other services in your request are processed normally.
- When payer authentication is not supported by the card type.

When you receive this reply, you can proceed to card authorization.

Authenticating Enrolled Cards

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

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

These values identify whether it is a 2.x transaction and that a challenge is required. If the 3D 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) in order 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** reply field and include them in the *Cardinal.cca_continue* function in order to proceed with the authentication session as shown in [Example 18](#).

Example 18 *Cardinal.cca_continue* (Android SDK)

```

/**
 * 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** reply field and include them in the *Cardinal session continue* function in order to proceed with the authentication session as shown in [Example 19](#).

In *Continue*, you should pass a class conforming to a protocol *CardinalValidationDelegate* (and implement a method *stepUpDidValidate*) as a parameter. [Example 19](#) or [Example 21](#) show an example of class conforming to *CardinalValidationDelegate* protocol.

Objective-C Examples

Example 19 Cardinal session continue (iOS SDK - Objective-C)

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

}
@end

@implementation YourViewController

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

    }

@end
```

If *Continue* is called in the same class, call the method shown in [Example 20](#) to start *StepUpFlow*.

Example 20 Cardinal.continue Call in the Same Class (Objective-C)

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

Swift Examples

Example 21 Cardinal session continue (iOS SDK - Swift)

```
class YourViewController:CardinalValidationDelegate {

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

    }

}
```

If *Continue* is called in the same class, call the method shown in [Example 22](#) to start *StepUpFlow*.

Example 22 Cardinal.continue Call in the Same Class (Swift)

```
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 [Example 23](#).

Example 23 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 reply that you receive contains the validation result.

CyberSource recommends that you request both payer authentication and card authorization services at the same time. When you do so, CyberSource automatically sends the correct information to your payment processor; CyberSource converts the values of these fields 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 Reply 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, and Discover include the CAVV (cardholder authentication verification value).
- For Mastercard, include the UCAF (universal cardholder authentication field) and the collection indicator.

Table 9 lists these fields.

Table 9 Validation Check and Reply Fields

Identifier	Validation Check Reply 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
3D Secure version	payerAuthValidateReply_specificationVersion	ccAuthService_paSpecificationVersion
Directory server transaction ID	payerAuthValidateReply_directoryServerTransactionID	ccAuthService_directoryServerTransactionID
Note Not required for 3D Secure 1.0.		

Interpreting the Reply



Important

If the authentication fails, Visa, American Express, JCB, Diners Club, and Discover require that you do 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 replies are similar for all card types:

- *Success:*

You receive the 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 CyberSource 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 replies 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 nonenrolled 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.
```

Upgrading Your Payer Authentication Implementation

This chapter provides information about upgrading to 3D Secure 2.x for merchants currently using CyberSource Payer Authentication services.

Upgrading to 3D Secure 2.x

Benefits

3D Secure 2.x provides the following benefits:

- More secure transactions as additional data is available.
- 3D Secure 2.x is backward compatible. Additional data will automatically be sent to issuers as they upgrade to 3D 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 3D Secure 2.2 when it becomes available. 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

Upgrading to 3D Secure 2.x is necessary if you are affected by PSD2.

PSD2 requires additional security measures outlined in the Regulatory Technical Standards (RTS) due to 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. It has implications for merchants including:

- Two-factor authentication will be required for all electronic payments, although there are exemptions to allow a frictionless flow.
- 3-D Secure e-commerce merchants will have to integrate dynamic authentication tools (such as 3D Secure 2.x).

If you are impacted by PSD2 changes, it is very important that you upgrade to 3D Secure 2.x.

Mandates

PSD2 includes mandates around strong customer authentication (SCA) and exemptions and challenges. See the following page for more information on the mandates:

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

Timelines

See the following page for PSD2 compliance timelines:

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

Recommended Integration

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

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

If you are currently using Payer Authentication services in your existing business processes and need to upgrade to 3D Secure 2.x, CyberSource recommends 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 3D Secure 2.x, the 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 [Chapter 2, Implementing Cardinal Cruise Direct Connection API Payer Authentication](#).

For details about the other integrations, see [Chapter 3, Implementing SDK Payer Authentication](#) or [Appendix H, Implementing Hybrid or Standard Payer Authentication](#).



Important

If you are using tokenization, you must use the Hybrid integration method.

Migration FAQ

Q: Do I need to send in a new JWT 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.

Payments.setupComplete returns a session ID and you can use that value for the **payerAuthEnrollService_referenceID** request field for the Check Enrollment service.

Q: When will the Payer Authentication reports include the new fields for 3D 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 3D Secure 2.x](#)," [page 112](#)) to test your preliminary code and make the appropriate changes.

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

A: 3D Secure 2.x is agnostic to the device and you have control over the formatting of the authentication form. 3D 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.

Testing Payer Authentication Services

After you have completed 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 reply.

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 2019, use 2022. 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.

**Note**

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

The test password is always 1234.

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

- If the user submits the password for the enrolled card, you receive the URL of the [Access Control Server](#) (ACS) where the customer can be authenticated.
- If the user clicks the Exit link and clicks OK in the verification window, authentication does not occur.

Table 10 lists the reply fields used in the test cases.

Table 10 Reply 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	payerAuthEnrollReply_eci
PAReq	payerAuthEnrollReply_paReq
proofXML	payerAuthEnrollReply_proofXML
Reason code	payerAuthEnrollReply_reasonCode
VERes enrolled	payerAuthEnrollReply_veresEnrolled
XID	payerAuthEnrollReply_xid

Authentication Validation

Table 11 lists only the reply fields used in the test cases.

Table 11 Reply 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
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.

Figure 1 Authentication Path for Visa, American Express, JCB, Diners Club, and Discover

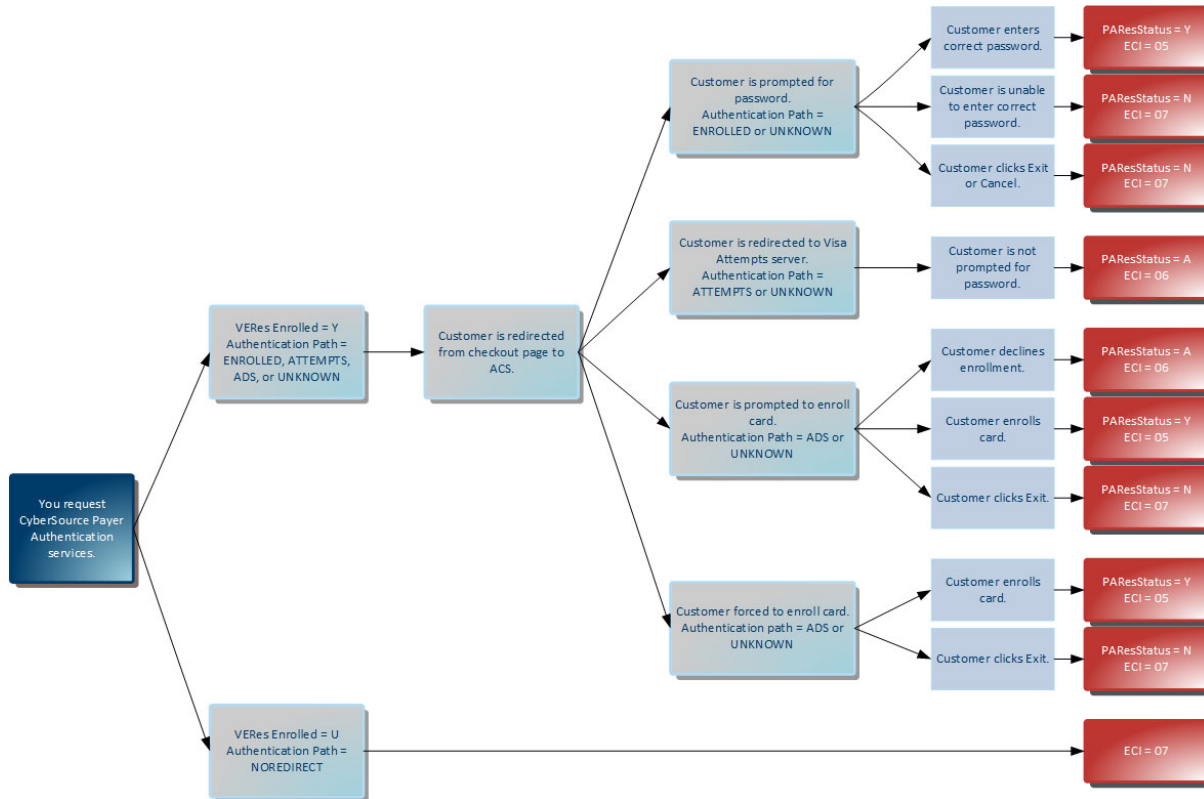
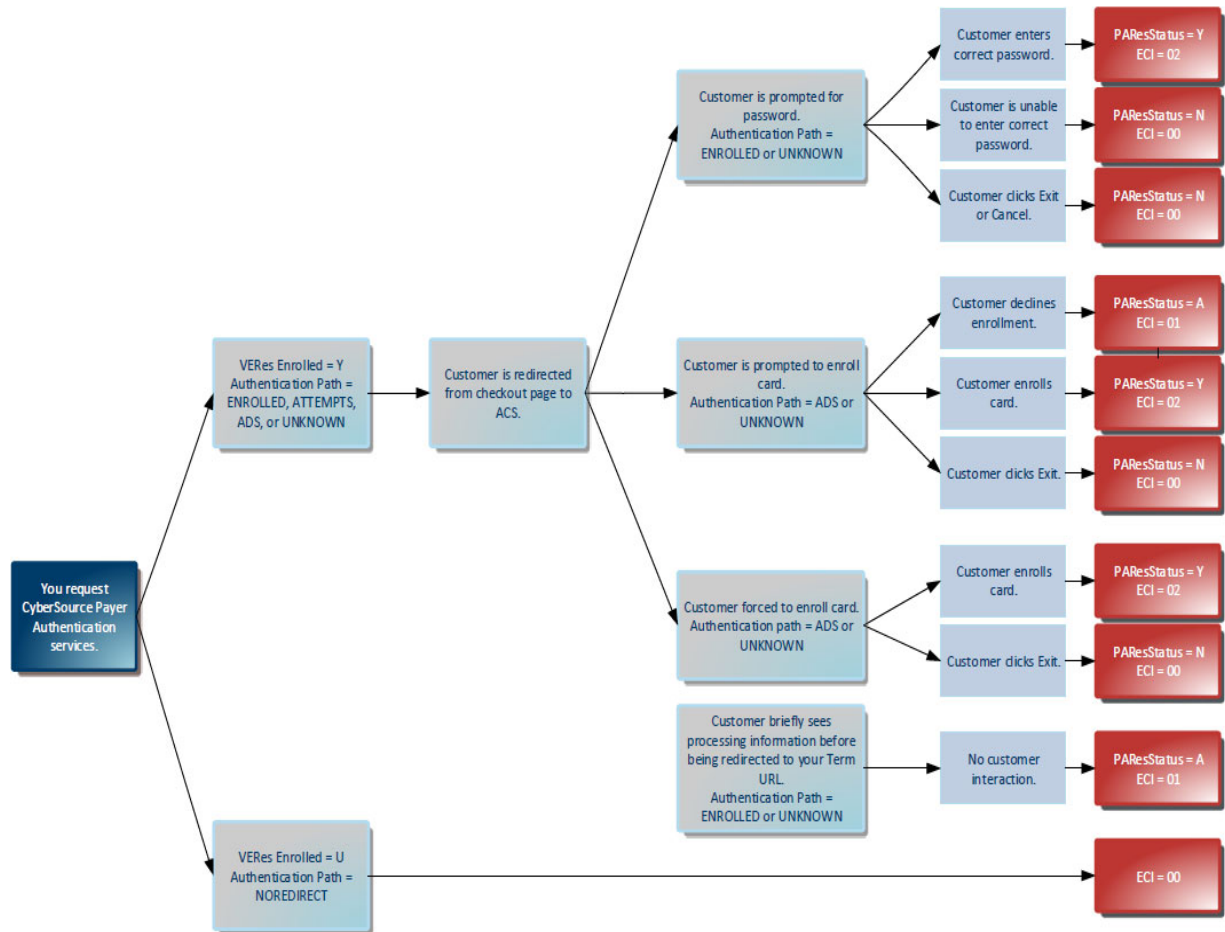


Figure 2 Authentication Path for Mastercard and Maestro

Test Cases for 3D Secure 1.0

Visa Secure

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

Table 12 Possible Values for Visa Secure Reply Fields

Result and Interpretation		Validate Authentication Reply			
		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 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 1: Visa Secure Card Enrolled: Successful Authentication

Card Number	4000000000000002 With authentication window	
	4000000000000000022	
	40000000000000119 Without authentication window	
Auth. Type	Active authentication	
Results	Check Enrollment	Validate Authentication
	Reason code 475	Reason code 100
Summary	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.	
	ics_pa_validate service was successful.	
	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	4000000000000010 With authentication window	
	4000000000000000071	
Auth. Type	Active authentication	
Results	Check Enrollment	Validate Authentication
	Reason code 475	Reason code 476
Summary	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.	
	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 3: Visa Secure Card Enrolled: Attempts Processing

Card Number	4000000000000101	Without authentication window
	400000000000000063	With authentication window
	4000000000000127	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 vbv_attempted
	VERes enrolled Y	ECI 06
	XID XID value	PARes status A
		XID XID value
Action	<p>If you request Validate Authentication and authorization services separately, follow these steps:</p> <ol style="list-style-type: none"> 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. <p>If you request the Validate Authentication and authorization services together, the process described above occurs automatically. Test card 4000000000000127 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.</p>	

Test Case 4: Visa Secure Card Enrolled: Incomplete Authentication

Card Number	40000000000000036 400000000000000055	
Auth. Type	Active authentication	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 475	Reason code 100
Details	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.	<ul style="list-style-type: none"> ■ Issuer unable to perform authentication. ■ ics_pa_validate service was successful.
	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 5: Visa Secure Card Enrolled: Unsuccessful Authentication

Card Number	40000000000000028 With authentication window 400000000000000048	
Auth. Type	Active authentication	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 475	Reason code 476
Details	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.	<ul style="list-style-type: none"> ■ User failed authentication. ■ Payer cannot be authenticated.
	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	4000008531947799		
Auth. Type	Active authentication		
Results	Check Enrollment	Validate Authentication	
Summary	Reason code	475	Reason code 476
Details	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		<ul style="list-style-type: none"> Customer prevents authentication. ics_pa_validate service was successful.
	ACS URL	URL value	Authentication result 9
	PARReq	PARReq 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	40000000000000069 4000000000000000014		
Auth. Type	Active authentication		
Results	Check Enrollment	Validate Authentication	
Summary	Reason code	100	
Details	ics_pa_enroll service was successful.		
	E-commerce indicator	internet	
	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	40000000000000093 400000000000000006	
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 9: Visa Secure Card Not Enrolled

Card Number	40000000000000051 4000000000000000030	
Auth. Type	Non-participating bank	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 100	
	ics_pa_enroll service was successful.	
Details	E-commerce indicator vbv_attempted	
	ECI 06	
	proofXML proofXML value	
	VERes enrolled N	
Action	Submit your authorization request. Liability shift.	

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

Card Number	4000000000000044	
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 11: Visa Secure Enrollment Check Error

Card Number	4000000000000085	Error response
	4000000000000077	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.	

Test Case 12: Visa Secure Enrollment RIBA_PASS

Card Number	4000180000000002	
Auth. Type	Passive authentication	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 475	Reason code 100
Details	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.	ics_pa_validate service was successful.
	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 13: Visa Secure Enrollment RIBA_PASS: Unsuccessful Authentication

Card Number	4000180000000028	
Auth. Type	Passive authentication	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 475	Reason code 476
Details	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.	<ul style="list-style-type: none"> ■ User failed authentication. ■ Payer cannot be authenticated.
	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 14: Visa Secure Enrollment RIBA

Card Number	4000260000000002 With authentication window	
Auth. Type	Risk-based bank	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 475	Reason code 100
Details	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.	ics_pa_validate service was successful.
	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 15: Visa Secure Enrollment RIBA: Unsuccessful Authentication

Card Number	40002600000000028 With authentication window	
Auth. Type	Risk-based bank	
Results	Reason code 475	Reason code 476
Summary	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.	<ul style="list-style-type: none"> 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.	

Mastercard Identity Check

Table 13 Possible Values for Mastercard Identity Check and Maestro Reply Fields

Result and Interpretation		payerAuthValidateReply_			
		authentication Result	ucafCollection Indicator	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	5200000000000007 5200000000000114		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	5200000000000015 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.	
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 18: Mastercard Identity Check Card Enrolled: Attempts Processing

Card Number	5200000000000122 Card enrollment option during purchase process 5200000000000106	
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.	
Details	ACS URL URL	Authentication result 1
	PAREq PAREq value	AAV AAV value
	proofXML proofXML value	Collection indicator 1
	VERes enrolled Y	E-commerce indicator spa
	XID XID value	PAREs status A
		XID XID value
Action	<p>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.</p> <ol style="list-style-type: none"> 1 Add the signed PAREs to the validation request. 2 In the reply, 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 19: Mastercard Identity Check Card Enrolled: Incomplete Authentication

Card Number	5200000000000031 Without authentication window	
Auth. Type	Active authentication	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 475	Reason code 100
Details	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.	<ul style="list-style-type: none"> ■ ics_pa_validate service was successful. ■ Issuer unable to perform authentication.
	ACS URL URL value	Authentication result 6
	PARReq PARReq value	Collection indicator 0
	proofXML proofXML value	E-commerce indicator internet
	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 20: Mastercard Identity Check Card Enrolled: Unsuccessful Authentication

Card Number	5200000000000023 With authentication window	
Auth. Type	Active authentication	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 475	Reason code 476
Details	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.	<ul style="list-style-type: none"> ■ User failed authentication ■ Payer could not be authenticated.
	ACS URL URL value	Authentication result 9
	PARReq PARReq value	PARes status N
	proofXML proofXML value	
	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	5641821000010028	
Auth. Type	Active authentication	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 475	Reason code 476
Details	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.	<ul style="list-style-type: none"> Customer prevents authentication. ics_pa_validate service was successful.
	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 22: Mastercard Identity Check Card Enrolled: Unavailable Authentication

Card Number	5200000000000064	
Auth. Type	Active authentication	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 100	
Details	ics_pa_enroll service was successful.	
	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	5200000000000098 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	5200000000000049	
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	After 10-12 seconds, proceed with the authorization message. No liability shift.	

Test Case 25: Mastercard Identity Check Enrollment Check Error

Card Number	5200000000000080	
Auth. Type	Active authentication	
Results	Check Enrollment	Validate Authentication
Summary	Reason code	100
	ics_pa_enroll service was successful.	
	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.	

Test Case 26: Mastercard Identity Check RIBA_PASS

Card Number	5200180000000007	
Auth. Type	Passive authentication	
Results	Check Enrollment	Validate Authentication
Summary	Reason code	475
	The card is enrolled in payer authentication. Please authenticate before proceeding with authorization.	
	ics_pa_validate service was successful.	
	ACS URL	URL
	PAReq	PAReq value
	proofXML	proofXML value
Details	VERes enrolled	Y
	XID	XID value
	Authentication result	0
	AAV	AAV value
Action	Collection indicator	2
	E-commerce indicator	spa
	PARes status	Y
Action	XID	XID value
	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 27: Mastercard Identity Check RIBA_PASS: Unsuccessful Authentication

Card Number	5200180000000023		
Auth. Type	Passive authentication		
Results	Check Enrollment	Validate Authentication	
Summary	Reason code	475	Reason code 476
Details	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		<ul style="list-style-type: none"> ■ User failed authentication ■ Payer could not be authenticated.
	ACS URL	URL value	Authentication result 9
	PAReq	PAReq value	PARes status N
	proofXML	proofXML value	
	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 28: Mastercard Identity Check RIBA

Card Number	5200260000000007	With authentication window	
Auth. Type	Risk-based bank		
Results	Check Enrollment		Validate Authentication
	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.
	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 29: Mastercard Identity Check RIBA: Unsuccessful Authentication

Card Number	5200260000000023 With authentication window	
Auth. Type	Risk-based bank	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 475	Reason code 476
Details	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.	<ul style="list-style-type: none"> ■ User failed authentication ■ Payer could not be authenticated.
	ACS URL URL value	Authentication result 9
	PAReq PAReq value	PARes status N
	proofXML proofXML value	
	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.	

Maestro

Test Case 30: Maestro Card Enrolled: Successful Authentication

Card Number	6759411100000008	Without authentication window
	6759410000006404	With authentication window
Auth. Type	Active authentication	
Results	Check Enrollment	Validate Authentication
	Reason code 475	Reason code 100
Summary	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.	
	ics_pa_validate service was successful.	
	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	<ol style="list-style-type: none"> 1 Add the signed PARes to the validation request. 2 In the reply, 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	6331101234567892	Without authentication window
Auth. Type	Active authentication	
Results	Check Enrollment	Validate Authentication
	Reason code 475	Reason code 476
Summary	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.	
	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

Card Number	560000000000000193 Card enrollment option during purchase process	
Auth. Type	Activation during shopping	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 475	Reason code 100
Details	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.	ics_pa_validate service was successful.
	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	<p>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.</p> <ol style="list-style-type: none"> 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	6331101250353227 Without authentication window	
Auth. Type	Active authentication	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 475	Reason code 100
Details	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.	Issuer unable to perform authentication.
	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	6331100610194313 Without authentication window	
Auth. Type	Active authentication	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 475	Reason code 476
Details	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.	User failed authentication
	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 35: Maestro Card Enrolled: Unavailable Authentication

Card Number	6331100266977839	
Auth. Type	Active authentication	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 100	
Details	ics_pa_enroll service was successful.	
	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	560000511607577094 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
	PARReq PARReq value	E-commerce indicator internet
	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	560000841211092515	
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

Table 14 Possible Values for American Express SafeKey Reply Fields

Result and Interpretation		Validate Authentication Reply			
		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	—	—	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 38: American Express SafeKey Card Enrolled: Successful Authentication

Card Number	340000000003961	Without authentication window
	371449111020228	With authentication window
Auth. Type	Active authentication	
Results	Check Enrollment	Validate Authentication
	Reason code 475	Reason code 100
Summary	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.	
	ics_pa_validate service was successful.	
	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	340000000006022	
Auth. Type	Active authentication	
Results	Check Enrollment	Validate Authentication
	Reason code 475	Reason code 476
Summary	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.	
	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 40: American Express SafeKey Card Enrolled: Attempts Processing

Card Number	340000000003391	Without authentication window	
	344400000000569	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 aesk_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 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. If you request the validation and authorization services together, the process described above occurs automatically.		

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

Card Number	340000000002302	Without authentication window	
Auth. Type	Active authentication		
Results	Check Enrollment		Validate Authentication
	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.
	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 42: American Express SafeKey Card Enrolled: Unsuccessful Authentication

Card Number	340000000000033 Without authentication window	
Auth. Type	Active authentication	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 475	Reason code 476
Details	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.	<ul style="list-style-type: none"> ■ User failed authentication. ■ Payer cannot be authenticated.
	ACS URL URL value	Authentication result 9
	PARReq PARReq 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	3400000000007780	
Auth. Type	Active authentication	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 100	
Details	ics_pa_enroll service was successful.	
	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	340000000009299	
Auth. Type	Active authentication	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 475	Reason code 476
Details	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.	We encountered a payer authentication problem: Error Processing PAREs.
	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	340000000008135	
Auth. Type	Non-participating bank	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 100	
Details	ics_pa_enroll service was successful.	
	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	340000000008309	
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	340000000007244	
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. If you requested payer authentication and authorization together, the authorization is processed automatically. No liability shift.	

JCB J/Secure

Table 15 Possible Values for JCB J/Secure Reply Fields

Result and Interpretation		payerAuthValidateReply_			
		authentication Result	eci	commerceIndicator	reasonCode
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	
	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	3569990010083722 Without authentication window	
Auth. Type	Active authentication	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 475	Reason code 100
Details	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.	ics_pa_validate service was successful.
	ACS URL URL	Authentication result 0
	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 (Signature Failure)

Card Number	3569990010083748	
Auth. Type	Active authentication	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 475	Reason code 476
Details	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.
	ACS URL URL value	Authentication result -1
	PAReq PAReq value	XID XID value
	VERes enrolled Y	
Action	Do not proceed with authorization. Instead ask the customer for another form of payment.	

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

Card Number	3569960010083758		
Auth. Type	Activation during shopping		
Results	Check Enrollment	Validate Authentication	
Summary	Reason code	475	Reason code 100
Details	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		ics_pa_validate service was successful.
	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	<p>If you request Validate Authentication and authorization services separately, follow these steps:</p> <ol style="list-style-type: none"> 1 Add the signed PARes to the validation request. 2 In the reply, ensure that the XID from the enrollment check matches that from the validation. 3 Add the CAVV and ECI values to your authorization request. <p>If you request the Validate Authentication and authorization services together, the process described above occurs automatically.</p>		

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

Card Number	3541599998103643	Without authentication window	
Auth. Type	Active authentication		
Results Summary	Check Enrollment		Validate Authentication
	Reason code	475	Reason code 100
	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		<div><div>■ Issuer unable to perform authentication.</div><div>■ ics_pa_validate service was successful.</div></div>
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	3569990110083721 Without authentication window	
Auth. Type	Active authentication	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 475	Reason code 476
Details	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.	<ul style="list-style-type: none"> ■ User failed authentication. ■ Payer cannot be authenticated.
	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 53: JCB J/Secure Card Enrolled: Unavailable Authentication

Card Number	3541599999103865	
Auth. Type	Active authentication	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 100	
Details	ics_pa_enroll service was successful.	
	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	3541599999103881	
Auth. Type	Active authentication	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 475	Reason code 476
Details	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.	We encountered a payer authentication problem: Error Processing PAREs.
	ACS URL URL value	ECI 07
	PARReq PARReq 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	3569970010083724	
Auth. Type	Non-participating bank	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 100	
Details	ics_pa_enroll service was successful.	
	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	3569980010083723	
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	3541599969103614	
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.	

Diners Club ProtectBuy

Table 16 Possible Values for Diners Club ProtectBuy Reply Fields

Result and Interpretation		Validate Authentication Reply			
		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 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	—	—	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	3005000000006246	
Auth. Type	Active authentication	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 475	Reason code 100
Details	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.	ics_pa_validate service was successful.
	ACS URL URL	Authentication result 0
	PARReq PARReq 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	3005000000004373	
Auth. Type	Active authentication	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 475	Reason code 476
Details	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.
	ACS URL URL value	Authentication result -1
	PARReq PARReq 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 60: Diners Club ProtectBuy Card Enrolled: Attempts Processing

Card Number	3005000000005271 Card enrollment option during purchase process	
Auth. Type	Activation during shopping	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 475	Reason code 100
Details	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.	ics_pa_validate service was successful.
	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	<p>If you request Validate Authentication and authorization services separately, follow these steps:</p> <ol style="list-style-type: none"> 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. <p>If you request the Validate Authentication and authorization services together, the process described above occurs automatically.</p>	

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

Card Number	3005000000007376	
Auth. Type	Active authentication	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 475	Reason code 100
Details	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.	<ul style="list-style-type: none"> ■ Issuer unable to perform authentication. ■ ics_pa_validate service was successful.
	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	3005000000005925	
Auth. Type	Active authentication	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 475	Reason code 476
Details	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.	<ul style="list-style-type: none"> ■ User failed authentication. ■ Payer cannot be authenticated.
	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 63: Diners Club ProtectBuy Card Enrolled: Unavailable Authentication

Card Number	3005000000006030	
Auth. Type	Active authentication	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 100	
Details	ics_pa_enroll service was successful.	
	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	3005000000005602	
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
	PARReq PARReq 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	3005000000007269	
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	3005000000001890	
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	3005000000009877	Error response
	3005000000004837	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 ProtectBuy

Table 17 Possible Values for Discover ProtectBuy Reply Fields

Result and Interpretation		Validate Authentication Reply			
		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 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 68: Discover ProtectBuy Card Enrolled: Successful Authentication

Card Number	6011000000000004	
Auth. Type	Active authentication	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 475	Reason code 100
Details	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.	ics_pa_validate service was successful.
	ACS URL URL	Authentication result 0
	PARReq PARReq 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	6011000000000012	
Auth. Type	Active authentication	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 475	Reason code 476
Details	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.
	ACS URL URL value	Authentication result -1
	PARReq PARReq 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

Card Number	6011000000000038 Card enrollment option during purchase process	
Auth. Type	Activation during shopping	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 475	Reason code 100
Details	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.	ics_pa_validate service was successful.
	ACS URL URL value	Authentication result 1
	PAReq PAReq value	CAVV CAVV value
	proofXML proofXML value	E-commerce indicator dipb_attempted
	VERes enrolled Y	ECI 06
	XID XID value	PARes status A
		XID XID value
Action	<p>If you request Validate Authentication and authorization services separately, follow these steps:</p> <ol style="list-style-type: none"> 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. <p>If you request the Validate Authentication and authorization services together, the process described above occurs automatically.</p>	

Test Case 71: Discover ProtectBuy Card Enrolled: Incomplete Authentication

Card Number	6011000000000103	
Auth. Type	Active authentication	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 475	Reason code 100
Details	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.	<ul style="list-style-type: none"> ■ Issuer unable to perform authentication. ■ ics_pa_validate service was successful.
	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 72: Discover ProtectBuy Card Enrolled: Unsuccessful Authentication

Card Number	6011000000000020	
Auth. Type	Active authentication	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 475	Reason code 476
Details	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.	<ul style="list-style-type: none"> ■ User failed authentication. ■ Payer cannot be authenticated.
	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	6011000000000061	
Auth. Type	Active authentication	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 100	
Details	ics_pa_enroll service was successful.	
	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	6011000000000095	
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
	PARReq PARReq 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	6011000000000053	
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 76: Discover ProtectBuy Enrollment Check: Time-Out

Card Number	6011000000000046	
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	6011000000000087	Error response
	6011000000000079	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.	

Test Cases for 3D 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 2019, use 2022. You also need the minimum required fields for an order.



Note

XID values are included in 3D Secure 2.x test cases for legacy reasons.

The 3D Secure version and directory server transaction ID fields are returned for the Check Enrollment and Validate Authentication services but are not included in the 3D Secure 2.x test cases.

Visa Secure

Test Case 2.1: Visa Secure Card Enrolled: Successful Frictionless Authentication

Card Number	4000000000001000	
Results	Check Enrollment	Validate Authentication
	Reason code	100
	Summary	NA
	ics_pa_enroll service was successful.	
	Details	
	VERes enrolled	Y
	PARes status	Y
	CAVV	CAVV value
	E-commerce indicator	vbv
Action	ECI	05
	XID	XID value
	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 2.2: Visa Secure Card Enrolled: Unsuccessful Frictionless Authentication

Card Number	4000000000001018	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 476	NA
Details	<ul style="list-style-type: none"> ■ User failed authentication. ■ Payer cannot be authenticated. 	
	VERes enrolled Y	
	PARes status N	
	E-commerce indicator internet or vbv_failure	
	ECI 07	
Action	It is not recommended to submit this transaction for authorization. Instead ask the customer for another form of payment.	

Test Case 2.3: Visa Secure Card Enrolled: Attempts Processing Frictionless Authentication

Card Number	4000000000001026	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 100	NA
Details	ics_pa_enroll service was successful.	
	VERes enrolled Y	
	PARes status A	
	CAVV CAVV value	
	E-commerce indicator vbv_attempted	
	ECI 06	
	XID XID value	
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 2.4: Visa Secure Card Enrolled: Unavailable Frictionless Authentication

Card Number	4000000000001034	
Results	Check Enrollment	Validate Authentication
Summary	Reason code	100
	ics_pa_enroll service was successful.	NA
Details	VERes enrolled	Y
	PARes status	U
	E-commerce indicator	internet or vbv_failure
	ECI	07
Action	Submit your authorization request. No liability shift.	

Test Case 2.5: Visa Secure Card Enrolled: Rejected Frictionless Authentication

Card Number	4000000000001042	
Results	Check Enrollment	Validate Authentication
Summary	Reason code	476
	<ul style="list-style-type: none"> User failed authentication. Payer cannot be authenticated. 	NA
Details	VERes enrolled	Y
	PARes status	R
	E-commerce indicator	internet or vbv_failure
	ECI	07
Action	You are not permitted to submit this transaction for authorization. Instead ask the customer for another form of payment.	

Test Case 2.6: Visa Secure Card Enrolled: Authentication not Available on Lookup

Card Number	4000000000001059	
Results	Check Enrollment	Validate Authentication
Summary	Reason code	100
	ics_pa_enroll service was successful.	NA
Details	VERes enrolled	U
	E-commerce indicator	internet or vbv_failure
Action	Submit your authorization request. No liability shift.	

Test Case 2.7: Visa Secure Enrollment Check Error

Card Number	4000000000001067	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 100	NA
Details	ics_pa_enroll service was successful.	
	VERes enrolled U	
	E-commerce indicator internet or vbv_failure	
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: Visa Secure Enrollment Check: Time-Out (Cruise Direct and Hybrid only)

Card Number	4000000000001075	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 100	NA
Details	ics_pa_enroll service was successful.	
	PARes status U	
	E-commerce indicator internet or vbv_failure	
	ECI 07	
Action	After 10-12 seconds, proceed with the authorization request. No liability shift.	

Test Case 2.9: Visa Secure Bypassed Authentication

Card Number	4000000000001083	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 100	NA
Details	ics_pa_enroll service was successful.	
	VERes enrolled B	
	E-commerce indicator internet	
Action	Submit your authorization request. No liability shift.	

Test Case 2.10a: Visa Secure Card Enrolled: Successful Step-Up Authentication (Cruise Direct and Hybrid)

Card Number	4000000000001091	
Results	Check Enrollment	Validate Authentication
	Reason code 475	Reason code 100
Summary	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.	
	ics_pa_validate service was successful.	
	VERes enrolled Y	
	PARes status Y	
	PARes enrolled Y	
	PARes status Y	
	CAVV CAVV value	
	E-commerce indicator vbv	
Details	ECI 05	
	XID XID value	
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 2.10b: Visa Secure Card Enrolled: Successful Step-Up Authentication (Standard)

Card Number	4000000000001091	
Results	Check Enrollment	
	Reason code 100	
Summary	ics_pa_enroll service was successful.	
	VERes enrolled Y	
	PARes status Y	
	CAVV CAVV value	
	E-commerce indicator vbv	
	ECI 05	
	XID XID value	
Details		
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: Visa Secure Card Enrolled: Unsuccessful Step-Up Authentication (Cruise Direct and Hybrid)

Card Number	4000000000001109	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 475	Reason code 476
Details	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.	<ul style="list-style-type: none"> ■ User failed authentication. ■ Payer cannot be authenticated.
	VERes enrolled Y	
	PARes status N	
	ECI 07	
Action	You are not permitted to submit this transaction for authorization. Instead ask the customer for another form of payment.	

Test Case 2.11b: Visa Secure Card Enrolled: Unsuccessful Step-Up Authentication (Standard)

Card Number	4000000000001109	
Results	Check Enrollment	
Summary	Reason code 476	
Details	<ul style="list-style-type: none"> ■ User failed authentication. ■ Payer cannot be authenticated. 	
	VERes enrolled Y	
	PARes status N	
	ECI 07	
Action	You are not permitted to submit this transaction for authorization. Instead ask the customer for another form of payment.	

Test Case 2.12a: Visa Secure Card Enrolled: Unavailable Step-Up Authentication (Cruise Direct and Hybrid)

Card Number	4000000000001117		
Results	Check Enrollment		Validate Authentication
Summary	Reason code	475	Reason code 100
Details	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.		ics_pa_validate service was successful.
	VERes enrolled	Y	
	PARes status	U	
	PARes status	U	
	ACS URL	URL value	E-commerce indicator internet or vbv_failure
			ECI 07
Action	Retry authentication or process without liability shift.		

Test Case 2.12b: Visa Secure Card Enrolled: Unavailable Step-Up Authentication (Standard)

Card Number	4000000000001117		
Results	Check Enrollment		
Summary	Reason code	100	
Details	ics_pa_enroll service was successful.		
	VERes enrolled	Y	
	PARes status	U	
	E-commerce indicator	internet or vbv_failure	
	ECI	07	
Action	Retry authentication or process without liability shift.		

Mastercard Identity Check



Note

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

Test Case 2.13: Mastercard Identity Check Card Enrolled: Successful Frictionless Authentication

Card Number	5200000000001005	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 100	NA
Details	ics_pa_enroll service was successful.	
	VERes enrolled	Y
	PARes status	Y
	AAV	AAV value
	E-commerce indicator	spa
	Collection indicator	2
	XID	XID value
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 2.14: Mastercard Identity Check Card Enrolled: Unsuccessful Frictionless Authentication

Card Number	5200000000001013	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 476	NA
Details	<ul style="list-style-type: none"> ■ User failed authentication. ■ Payer cannot be authenticated. 	
	VERes enrolled	Y
	PARes status	N
	E-commerce indicator	internet
	Collection indicator	0
Action	It is not recommended to submit this transaction for authorization. Instead ask the customer for another form of payment.	

Test Case 2.15: Mastercard Identity Check Card Enrolled: Attempts Processing Frictionless Authentication

Card Number	5200000000001021	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 100	NA
Details	ics_pa_enroll service was successful.	
	VERes enrolled Y	
	PARes status A	
	AAV AAV value	
	E-commerce indicator spa	
	Collection indicator 1	
	XID XID value	
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 2.16: Mastercard Identity Check Card Enrolled: Unavailable Frictionless Authentication

Card Number	5200000000001039	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 100	NA
Details	ics_pa_enroll service was successful.	
	VERes enrolled Y	
	PARes status U	
	E-commerce indicator internet	
	Collection indicator 0	
Action	Submit your authorization request. No liability shift.	

Test Case 2.17: Mastercard Identity Check Card Enrolled: Rejected Frictionless Authentication

Card Number	5200000000001047	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 476	NA
Details	<ul style="list-style-type: none"> ■ User failed authentication. ■ Payer cannot be authenticated. 	
	VERes enrolled Y	
	PARes status R	
	E-commerce indicator internet	
	Collection indicator 0	
Action	You are not permitted to submit this transaction for authorization. Instead ask the customer for another form of payment.	

Test Case 2.18: Mastercard Identity Check Card Enrolled: Authentication not Available on Lookup

Card Number	5200000000001054	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 100	NA
Details	ics_pa_enroll service was successful.	
	VERes enrolled U	
	E-commerce indicator internet	
	Collection indicator 0	
Action	Submit your authorization request. No liability shift.	

Test Case 2.19: Mastercard Identity Check Enrollment Check Error

Card Number	5200000000001062	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 100	NA
Details	ics_pa_enroll service was successful.	
	VERes enrolled U	
	E-commerce indicator internet	
	Collection indicator 0	
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 2.20: Mastercard Identity Check Enrollment Check: Time-Out (Cruise Direct and Hybrid only)

Card Number	5200000000001070	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 100	NA
Details	ics_pa_enroll service was successful.	
	PARes status U	
	E-commerce indicator internet	
	Collection indicator 0	
Action	After 10-12 seconds, proceed with the authorization request. No liability shift.	

Test Case 2.21: Mastercard Identity Check Bypassed Authentication

Card Number	5200000000001088	
Results	Check Enrollment	Validate Authentication
Summary	Reason code	100
	NA	
Details	ics_pa_enroll service was successful.	
	VERes enrolled	B
	E-commerce indicator	internet
	Collection indicator	0
Action	Submit your authorization request. No liability shift.	

Test Case 2.22a: Mastercard Identity Check Card Enrolled: Successful Step Up Authentication (Cruise Direct and Hybrid)

Card Number	5200000000001096		
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	VERes enrolled Y		
	PAReq PAReq value		PARes status Y
	ACS URL URL value		AAV AAV value
			E-commerce indicator spa
			Collection indicator 2
			XID XID value
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 2.22b: Mastercard Identity Check Card Enrolled: Successful Step-Up Authentication (Standard)

Card Number	5200000000001096	
Results	Check Enrollment	
Summary	Reason code	100
Details	ics_pa_enroll service was successful.	
	VERes enrolled	Y
	PARes status	Y
	AAV	AAV value
	E-commerce indicator	spa
	Collection indicator	2
	XID	XID value
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 2.23a: Mastercard Identity Check Card Enrolled: Unsuccessful Step-Up Authentication (Cruise Direct and Hybrid)

Card Number	5200000000001104	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 475	Reason code 476
Details	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.	<ul style="list-style-type: none"> ■ User failed authentication. ■ Payer cannot be authenticated.
	VERes enrolled Y	
	PAReq PAReq value	PARes status N
	ACS URL URL value	Collection indicator 0
Action	You are not permitted to submit this transaction for authorization. Instead ask the customer for another form of payment.	

Test Case 2.23b: Mastercard Identity Check Card Enrolled: Unsuccessful Step-Up Authentication (Standard)

Card Number	52000000000001104		
Results	Check Enrollment		
Summary	Reason code	476	
	<div><div>■ User failed authentication.</div><div>■ Payer cannot be authenticated.</div></div>		
Details	VERes enrolled	Y	
	PARes status	N	
	Collection indicator	0	
Action	You are not permitted to submit this transaction for authorization. Instead ask the customer for another form of payment.		

Test Case 2.24a: Mastercard Identity Check Card Enrolled: Unavailable Step-Up Authentication (Cruise Direct and Hybrid)

Card Number	5200000000001112		
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	VERes enrolled	Y	
	PAReq	PAReq value	PARes status U
	ACS URL	URL value	E-commerce indicator internet
			Collection indicator 0
Action	Retry authentication or process without liability shift.		

Test Case 2.24b: Mastercard Identity Check Card Enrolled: Unavailable Step-Up Authentication (Standard)

Card Number	5200000000001112	
Results	Check Enrollment	
Summary	Reason code	100
	ics_pa_enroll service was successful.	
Details	VERes enrolled	Y
	PARes status	U
	E-commerce indicator	internet
	Collection indicator	0
Action	Retry authentication or process without liability shift.	

American Express SafeKey

Test Case 2.25: American Express SafeKey Card Enrolled: Successful Frictionless Authentication

Card Number	340000000001007	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 100	NA
Details	ics_pa_enroll service was successful.	
	VERes enrolled Y	
	PARes status Y	
	CAVV CAVV value	
	E-commerce indicator aesk	
	ECI 05	
	XID XID value	
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 2.26: American Express SafeKey Card Enrolled: Unsuccessful Frictionless Authentication

Card Number	340000000001015	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 476	NA
Details	<ul style="list-style-type: none"> ■ User failed authentication. ■ Payer cannot be authenticated. 	
	VERes enrolled Y	
	PARes status N	
	E-commerce indicator internet	
	ECI 07	
Action	It is not recommended to submit this transaction for authorization. Instead ask the customer for another form of payment.	

Test Case 2.27: American Express SafeKey Card Enrolled: Attempts Processing Frictionless Authentication

Card Number	340000000001023	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 100	NA
Details	ics_pa_enroll service was successful.	
	VERes enrolled Y	
	PARes status A	
	CAVV CAVV value	
	E-commerce indicator aesk_attempted	
	ECI 06	
	XID XID value	
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 2.28: American Express SafeKey Card Enrolled: Unavailable Frictionless Authentication

Card Number	340000000001031	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 100	NA
Details	ics_pa_enroll service was successful.	
	VERes enrolled Y	
	PARes status U	
	E-commerce indicator internet	
	ECI 07	
Action	Submit your authorization request. No liability shift.	

Test Case 2.29: American Express SafeKey Card Enrolled: Rejected Frictionless Authentication

Card Number	340000000001049	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 476	NA
Details	<ul style="list-style-type: none"> ■ User failed authentication. ■ Payer cannot be authenticated. 	
	VERes enrolled Y	
	PARes status R	
	E-commerce indicator internet	
	ECI 07	
Action	You are not permitted to submit this transaction for authorization. Instead ask the customer for another form of payment.	

Test Case 2.30: American Express SafeKey Card Enrolled: Authentication not Available on Lookup

Card Number	340000000001056	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 100	NA
Details	ics_pa_enroll service was successful.	
	VERes enrolled U	
	E-commerce indicator internet	
Action	Submit your authorization request. No liability shift.	

Test Case 2.31: American Express SafeKey Enrollment Check Error

Card Number	340000000001064	
Results	Check Enrollment	Validate Authentication
Summary	Reason code	100
	NA	
Details	ics_pa_enroll service was successful.	
	VERes enrolled	U
	E-commerce indicator	internet
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 2.32: American Express SafeKey Enrollment Check: Time-Out (Cruise Direct and Hybrid only)

Card Number	340000000001072	
Results	Check Enrollment	Validate Authentication
Summary	Reason code	100
	NA	
Details	ics_pa_enroll service was successful.	
	VERes enrolled	U
	E-commerce indicator	internet
Action	ECI	07
	After 10-12 seconds, proceed with the authorization request. No liability shift.	

Test Case 2.33: American Express SafeKey Bypassed Authentication

Card Number	340000000001080	
Results	Check Enrollment	Validate Authentication
	Summary	
	Reason code 100	NA
	ics_pa_enroll service was successful.	
	Details	
	VERes enrolled B	
	E-commerce indicator internet	
	ECI 07	
Action	Submit your authorization request. No liability shift.	

Test Case 2.34a: American Express SafeKey Card Enrolled: Successful Step-Up Authentication (Cruise Direct and Hybrid)

Card Number	340000000001098	
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	
	VERes enrolled Y	
	PAReq PAReq value	PARes status Y
	ACS URL URL value	CAVV CAVV value
		E-commerce indicator aesk
		ECI 05
		XID XID value
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 2.34b: American Express SafeKey Card Enrolled: Successful Step-Up Authentication (Standard)

Card Number	340000000001098	
Results	Check Enrollment	
Summary	Reason code 100	
Details	ics_pa_enroll service was successful.	
	VERes enrolled	Y
	PARes status	Y
	CAVV	CAVV value
	E-commerce indicator	aesk
	ECI	05
	XID	XID value
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 2.35a: American Express SafeKey Card Enrolled: Unsuccessful Step-Up Authentication (Cruise Direct and Hybrid)

Card Number	340000000001106	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 475	Reason code 476
Details	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.	<ul style="list-style-type: none"> User failed authentication. Payer cannot be authenticated.
	VERes enrolled Y	
	PAReq PAREq value	PARes status N
	ACS URL URL 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 2.35b: American Express SafeKey Card Enrolled: Unsuccessful Step-Up Authentication (Standard)

Card Number	340000000001106		
Results	Check Enrollment		
Summary	Reason code 476		
	<ul style="list-style-type: none"> User failed authentication. Payer cannot be authenticated. 		
	VERes enrolled Y		
	PAREs status N		
	ECI 07		
Action	You are not permitted to submit this transaction for authorization. Instead ask the customer for another form of payment.		

Test Case 2.36a: American Express SafeKey Card Enrolled: Unavailable Step-Up Authentication (Cruise Direct and Hybrid)

Card Number	340000000001114		
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.
	VERes enrolled Y		
	PAREq	PAREq value	PAREs status U
	ACS URL	URL value	E-commerce indicator internet
Details			ECI 07
Action	Retry authentication or process without liability shift.		

Test Case 2.36b: American Express SafeKey Card Enrolled: Unavailable Step-Up Authentication (Standard)

Card Number	340000000001114	
Results	Check Enrollment	
Summary	Reason code	100
	ics_pa_enroll service was successful.	
Details	VERes enrolled	Y
	PARes status	U
	E-commerce indicator	internet
	ECI	07
Action	Retry authentication or process without liability shift.	

Discover ProtectBuy and Diners Club ProtectBuy



Note

To test Diners Club ProtectBuy, use the Discover card numbers in the following section, but use card type of 005 (Diners) rather than 004 (Discover).

Test Case 2.37: Discover/Diner ProtectBuy Card Enrolled: Successful Frictionless Authentication

Card Number	6011000000001002	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 100	NA
Details	ics_pa_enroll service was successful.	
	VERes enrolled	Y
	PARes status	Y
	CAVV	CAVV value
	E-commerce indicator	dipb (Discover) pb (Diners Club)
	ECI	05
	XID	XID value
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 2.38: Discover/Diner ProtectBuy Card Enrolled: Unsuccessful Frictionless Authentication

Card Number	6011000000001010	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 476	NA
Details	<ul style="list-style-type: none"> ■ User failed authentication. ■ Payer cannot be authenticated. 	
	VERes enrolled	Y
	PARes status	N
	E-commerce indicator	internet
	ECI	07
Action	It is not recommended to submit this transaction for authorization. Instead ask the customer for another form of payment.	

Test Case 2.39: Discover/Diner ProtectBuy Card Enrolled: Attempts Processing Frictionless Authentication

Card Number	6011000000001028	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 100	NA
Details	ics_pa_enroll service was successful.	
	VERes enrolled Y	
	PARes status A	
	CAVV CAVV value	
	E-commerce indicator	dipb_attempted (Discover) pb_attempted (Diners Club)
	ECI 06	
	XID XID value	
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 2.40: Discover/Diner ProtectBuy Card Enrolled: Unavailable Frictionless Authentication

Card Number	6011000000001036	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 100	NA
Details	ics_pa_enroll service was successful.	
	VERes enrolled Y	
	PARes status U	
	E-commerce indicator	internet
	ECI 07	
Action	Submit your authorization request. No liability shift.	

Test Case 2.41: Discover/Diner ProtectBuy Card Enrolled: Rejected Frictionless Authentication

Card Number	6011000000001044	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 476	NA
Details	<ul style="list-style-type: none"> ■ User failed authentication. ■ Payer cannot be authenticated. 	
	VERes enrolled Y	
	PARes status R	
	E-commerce indicator internet	
	ECI 07	
Action	You are not permitted to submit this transaction for authorization. Instead ask the customer for another form of payment.	

Test Case 2.42: Discover/Diner ProtectBuy Card Enrolled: Authentication not Available on Lookup

Card Number	6011000000001051	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 100	NA
Details	ics_pa_enroll service was successful.	
	VERes enrolled U	
	E-commerce indicator internet	
Action	Submit your authorization request. No liability shift.	

Test Case 2.43: Discover/Diner ProtectBuy Enrollment Check Error

Card Number	6011000000001069	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 100	NA
Details	ics_pa_enroll service was successful.	
	VERes enrolled U	
	E-commerce indicator internet	
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 2.44: Discover/Diner ProtectBuy Enrollment Check: Time-Out (Cruise Direct and Hybrid only)

Card Number	6011000000001077	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 100	NA
	ics_pa_enroll service was successful.	
Details	VERes enrolled U	
	E-commerce indicator internet	
	ECI 07	
Action	After 10-12 seconds, proceed with the authorization request. No liability shift.	

Test Case 2.45: Discover/Diner ProtectBuy Bypassed Authentication

Card Number	6011000000001085	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 100	NA
	ics_pa_enroll service was successful.	
Details	VERes enrolled B	
	E-commerce indicator internet	
	ECI 07	
Action	Submit your authorization request. No liability shift.	

Test Case 2.46a: Discover/Diner ProtectBuy Card Enrolled: Successful Step-Up Authentication (Cruise Direct and Hybrid)

Card Number	6011000000001093	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 475	Reason code 100
Details	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.	
	ics_pa_validate service was successful.	
	VERes enrolled Y	
	PARes status Y	
	CAVV CAVV value	
	E-commerce indicator	dipb (Discover) pb (Diners Club)
	ECI 05	
Action	XID XID value	
	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 2.46b: Discover/Diner ProtectBuy Card Enrolled: Successful Step-Up Authentication (Standard)

Card Number	6011000000001093	
Results	Check Enrollment	
Summary	Reason code 100	
Details	ics_pa_enroll service was successful.	
	VERes enrolled Y	
	PARes status Y	
	CAVV CAVV value	
	E-commerce indicator	dipb (Discover) pb (Diners Club)
	ECI 05	
	XID XID value	
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 2.47a: Discover/Diner ProtectBuy Card Enrolled: Unsuccessful Step-Up Authentication (Cruise Direct and Hybrid)

Card Number	6011000000001101	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 475	Reason code 476
Details	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.	<ul style="list-style-type: none"> ■ User failed authentication. ■ Payer cannot be authenticated.
	VERes enrolled Y	
	PAReq PAReq value	PARes status N
	ACS URL URL 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 2.47b: Discover/Diner ProtectBuy Card Enrolled: Unsuccessful Step-Up Authentication (Standard)

Card Number	6011000000001101	
Results	Check Enrollment	
Summary	Reason code 476	
Details	<ul style="list-style-type: none"> ■ User failed authentication. ■ Payer cannot be authenticated. 	
	VERes enrolled Y	
	PAReq status N	
	ECI 07	
Action	You are not permitted to submit this transaction for authorization. Instead ask the customer for another form of payment.	

Test Case 2.48a: Discover/Diner ProtectBuy Card Enrolled: Unavailable Step-Up Authentication (Cruise Direct and Hybrid)

Card Number	6011000000001119	
Results	Check Enrollment	Validate Authentication
Summary	Reason code 475	Reason code 100
Details	The cardholder is enrolled in payer authentication. Please authenticate before proceeding with authorization.	
	VERes enrolled Y	ics_pa_validate service was successful.
	PARes status U	
	PARes status U	
	ACS URL URL value	E-commerce indicator internet
		ECI 07
Action	Retry authentication or process without liability shift.	

Test Case 2.48b: Discover/Diner ProtectBuy Card Enrolled: Unavailable Step-Up Authentication (Standard)

Card Number	6011000000001119	
Results	Check Enrollment	
Summary	Reason code 100	
Details	ics_pa_enroll service was successful.	
	VERes enrolled Y	
	PARes status U	
	E-commerce indicator internet	
	ECI 07	
Action	Retry authentication or process without liability shift.	

API Fields

This appendix describes the Simple Order API fields that you can use to access CyberSource Payer Authentication services. The API and client toolkits can be downloaded from the CyberSource web site at the following URL:

http://www.cybersource.com/developers/develop/integration_methods/simple_order_and_soap_toolkit_api/

Formatting Restrictions

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



Note

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

For Atos, the **billTo_** fields must not contain colons (:).

The values of all request 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 will be removed.

Data Type Definitions

For more information about these data types, see the World Wide Web Consortium (W3C) *XML Schema Part 2: Datatypes* specification:

<http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/>

Data Type	Description
Integer	Whole number {..., -3, -2, -1, 0, 1, 2, 3, ...}.
String	Sequence of letters, numbers, spaces, and special characters, such as @ and #.

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. The item fields are 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.

Example 24 Numbered XML Schema Element Names and Name-Value Pair Field Names

XML Schema Element Names	Corresponding Name-Value Pair Field Names
<pre><item id="0"> <unitPrice> <quantity> </item></pre>	<pre>item_0_unitPrice item_0_quantity</pre>
<pre><item id="1"> <unitPrice> <quantity> </item></pre>	<pre>item_1_unitPrice item_1_quantity</pre>



Important

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

Request Fields

See *Credit Card Services Using the Simple Order API* ([PDF](#) | [HTML](#)) and *Getting Started with CyberSource Advanced* ([PDF](#) | [HTML](#)) for more information about using the Simple Order API to access CyberSource services using either name-value pairs or XML.



Note

The fields in the following table refer to the enroll and validate services only. Please review *Credit Card Services Using the Simple Order API* ([PDF](#) | [HTML](#)) for more information about the fields specific to the authorization.

Table 18 Request Fields

Field Name	Description	Required/ Optional	Type & Length
airlineData_leg_#_ carrierCode	International Air Transport Association (IATA) code for the carrier for this leg of the trip. Required for each leg. Required for American Express SafeKey (U.S.) for travel-related requests.	Enroll (O)	String (2)
airlineData_leg_#_ departureDate	Departure date for the first leg of the trip. Format: YYYYMMDD. Required for American Express SafeKey (U.S.) for travel-related requests. Note The numbered element name should contain 0 instead of #. Payer Authentication services only use the first leg of the trip. See "Numbered Elements," page 143 .	Enroll (O)	Integer (8)
airlineData_leg_#_ destination	International Air Transport Association (IATA) code for the destination airport for this leg of the trip. Required for each leg. Required for American Express SafeKey (U.S.) for travel-related requests.	Enroll (O)	String (5)
airlineData_leg_#_ originatingAirportCode	International Air Transport Association (IATA) code for the originating airport for the first leg of the trip. Required for American Express SafeKey (U.S.) for travel-related requests.	Enroll (O)	String (5)
airlineData_ numberOfPassengers	Number of passengers for whom the ticket was issued. If you do not include this field in your request, CyberSource uses a default value of 1. Required for American Express SafeKey (U.S.) for travel-related requests.	Enroll (O)	Integer (3)

Table 18 Request Fields (Continued)

Field Name	Description	Required/ Optional	Type & Length
airlineData_passenger_#_ firstName	<p>First name of the passenger to whom the ticket was issued. If there are multiple passengers, include all listed on the ticket.</p> <p>Do not include special characters such as commas, hyphens, or apostrophes. Only ASCII characters are supported.</p> <p>Required for American Express SafeKey (U.S.) for travel-related requests.</p>	Enroll (O)	String (60)
airlineData_passenger_#_ lastName	<p>Last name of the passenger to whom the ticket was issued. If there are multiple passengers, include all listed on the ticket.</p> <p>Do not include special characters such as commas, hyphens, or apostrophes. Only ASCII characters are supported.</p> <p>Required for American Express SafeKey (U.S.) for travel-related requests.</p>	Enroll (O)	String (60)
billTo_city	City of the billing address.	Enroll (R)	String (50)
billTo_country	Billing country for the account. Use the two-character country codes .	Enroll (R)	String (2)
billTo_ customerAccountChange Date	<p>Date the cardholder's account was last changed. This includes changes to the billing or shipping address, new payment accounts or new users added.</p> <p>This field can contain one of these values:</p> <ul style="list-style-type: none"> ■ -1: Guest account ■ 0: Changed during this transaction <p>If neither applies, enter the date in YYYYMMDD format.</p> <p>Recommended for Discover ProtectBuy.</p>	Enroll (O)	Integer (8)
billTo_ customerAccountCreate Date	<p>Date the cardholder opened the account.</p> <p>This field can contain one of these values:</p> <ul style="list-style-type: none"> ■ -1: Guest account ■ 0: Opened during this transaction <p>If neither applies, enter the date in YYYYMMDD format.</p> <p>Recommended for Discover ProtectBuy.</p>	Enroll (O)	Integer (8)

Table 18 Request Fields (Continued)

Field Name	Description	Required/ Optional	Type & Length
billTo_ customerAccount PasswordChange Date	<p>Date the cardholder last changed or reset password on account.</p> <p>This field can contain one of these values:</p> <ul style="list-style-type: none"> ■ -1: Guest account ■ 0: Changed during this transaction <p>If neither applies, enter the date in YYYYMMDD format.</p> <p>Recommended for Discover ProtectBuy.</p>	Enroll (O)	Integer (8)
billTo_email	Customer's email address, including the full domain name. Use the following format: name@host.domain (for example, jdoe@example.com).	Enroll (R)	String (255)
billTo_firstName	Customer's first name. The value should be the same as the value that appears on the card.	Enroll (R)	String (60)
billTo_ httpBrowserColorDepth	<p>Indicates the bit depth of the color palette for displaying images, in bits per pixel.</p> <p>Example 24</p> <p>For more information, see https://en.wikipedia.org/wiki/Color_depth.</p>	Enroll (O)	String (2)
billTo_ httpBrowserJavaEnabled	<p>Indicates the ability of the cardholder browser to execute Java. The value is returned from the navigator.javaEnabled property. This field can contain one of these values:</p> <ul style="list-style-type: none"> ■ true ■ false 	Enroll (O)	String (5)
billTo_ httpBrowserJavaScript Enabled	<p>Indicates the ability of the cardholder browser to execute JavaScript. This value is available from the fingerprint details of the cardholder's browser. This field can contain one of these values:</p> <ul style="list-style-type: none"> ■ true ■ false 	Enroll (O)	String (5)
billTo_ httpBrowserLanguage	<p>Indicates the browser language as defined in IETF BCP47.</p> <p>Example en-US</p> <p>For more information, see https://en.wikipedia.org/wiki/IETF_language_tag.</p>	Enroll (O)	String (8)
billTo_ httpBrowserScreenHeight	<p>Total height of the cardholder's screen in pixels.</p> <p>Example 864</p>	Enroll (O)	String (6)

Table 18 Request Fields (Continued)

Field Name	Description	Required/ Optional	Type & Length
billTo_ httpBrowserScreenWidth	Total width of the cardholder's screen in pixels. Example 1536	Enroll (O)	String (6)
billTo_httpBrowserTime Difference	Time difference between UTC time and the cardholder browser local time, in minutes. Example 300	Enroll (O)	String (5)
billTo_ipAddress	Customer's IP address, such as 10.1.27.63, reported by your web server via socket information.	Enroll (O)	String (45)
billTo_lastName	Customer's last name. The value should be the same as the value that appears on the card.	Enroll (R)	String (60)
billTo_passportCountry	Issuing country for the cardholder's passport. Recommended for Discover ProtectBuy.	Enroll (O)	Integer (3)
billTo_passportNumber	The cardholder's passport number. Recommended for Discover ProtectBuy.	Enroll (O)	String (40)
billTo_phoneNumber	Telephone number of the customer. For countries other than US or CA, add the country code at the beginning of the phone number, if possible. Otherwise, the billing country is used to determine the country code.	Enroll (O)	String (15)
billTo_postalCode	Postal code for the billing address. The postal code must consist of 5 to 9 digits. When the billing country is the U.S., the 9-digit postal code must follow this format: [5 digits][dash][4 digits] Example 12345-6789 When the billing country is Canada, the 6-digit postal code must follow this format: [alpha][numeric][alpha][space] [numeric][alpha][numeric] Example A1B 2C3 Required only if the billTo_country field is US or CA.	Enroll (R)	String (10)
billTo_state	State or province of the customer. Required for U.S. and Canada. Use the two-character state, province, or territory codes .	Enroll (R)	String (2)
billTo_street1	First line of the billing street address as it appears on the credit card issuer's records.	Enroll (R)	String (60)
billTo_street2	Additional address information, for example: <i>Attention: Accounts Payable</i>	Enroll (O)	String (60)

Table 18 Request Fields (Continued)

Field Name	Description	Required/ Optional	Type & Length
card_accountNumber	Customer's card number.	Enroll (R) Validate (O)	Integer (20)
card_cardType	Type of card. For more information, see <i>Credit Card Services Using the Simple Order API</i> (PDF HTML). This field contain one of these values: <ul style="list-style-type: none"> ■ 001: Visa ■ 002: Mastercard ■ 003: American Express ■ 004: Discover ■ 005: Diners Club ■ 007: JCB ■ 024: Maestro (UK Domestic) ■ 036: Cartes Bancaires ■ 042: Maestro (International) 	Enroll (R) Validate (R)	String (3)
card_expirationMonth	Expiration month (MM) of the card. Required for the Validate service if card_accountNumber is included.	Enroll (R) Validate (O)	String (2)
card_expirationYear	Expiration year (YYYY) of the card. Required for the Validate service if card_accountNumber is included.	Enroll (R) Validate (O)	String (4)
item_#_passengerFirstName	Passenger's first name. See "Numbered Elements," page 143.	Enroll (O)	String (60)
item_#_passengerLastName	Passenger's last name. See "Numbered Elements," page 143.	Enroll (O)	String (60)
item_#_productDescription	Brief description of item.	Enroll (O)	String (256)
item_#_productName	Name of the product. See "Numbered Elements," page 143.	Enroll (O)	String (255)
item_#_productSKU	Merchant's product identifier code. See "Numbered Elements," page 143.	Enroll (O)	String (255)
item_#_quantity	Quantity of the product being purchased. The default value is 1. See "Numbered Elements," page 143.	Enroll (O)	Non-negative integer (10)
item_#_shippingDestinationTypes	Destination to which the item is shipped. Example Commercial, residential, store	Enroll (O)	String (50)

Table 18 Request Fields (Continued)

Field Name	Description	Required/ Optional	Type & Length
item_#_unitPrice	<p>Per-item price of the product. This value cannot be negative. The amount will be truncated to the correct number of decimal places. You can include a decimal point (.) in this field, but you cannot include any other special characters.</p> <p>Note The <code>item_#_unitPrice</code> field is optional if the <code>purchaseTotals_grandTotalAmount</code> field is used.</p> <p>See "Numbered Elements," page 143.</p>	<p>Enroll (R)</p> <p>Validate (R)</p>	String (15)
merchantDefinedData_ mddField_1 to merchantDefinedData_ mddField_5	<p>Fields that you can use to store information. The value appears in the Case Management Details window in the Business Center. The first four fields are the same fields that are used by the Secure Data services. These fields can only be used if you are referencing target API 1.75 or higher.</p> <p>Important These fields override the old merchant-defined data fields. For example, if you use the obsolete field <code>merchantDefinedData_field5</code> and the new field <code>merchantDefinedData_mddField_5</code> in the same request, the new field value overwrites the value specified in the obsolete field.</p> <p>Warning Merchant-defined data fields are not intended to and <i>must not</i> 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). In the event CyberSource discovers that a merchant is capturing and/or transmitting personally identifying information via the merchant-defined data fields, whether or not intentionally, CyberSource will immediately suspend the merchant's account, which will result in a rejection of any and all transaction requests submitted by the merchant after the point of suspension.</p>	Enroll (O)	String (255)

Table 18 Request Fields (Continued)

Field Name	Description	Required/ Optional	Type & Length
merchantID	Your CyberSource merchant ID.	Enroll (R) Validate (R)	String (30)
merchantReference Code	Merchant-generated order reference or tracking number.	Enroll (R) Validate (R)	String (50)
payerAuthEnrollService_ accountPurchases	Number of purchases with this cardholder account during the previous six months. Recommended for Discover ProtectBuy.	Enroll (O)	Integer (4)
payerAuthEnrollService_ acquirerCountry	Issuers should be aware of the acquirer's country code when the acquirer country differs from the merchant country, and the acquirer is in the EEA (European Economic Area).	Enroll (O)	String (2)
payerAuthEnrollService_ acsWindowSize	You can send this override field to set the challenge window size to display to the cardholder. The Access Control Server (ACS) replies with content that is formatted appropriately for this window size to allow for the best user experience. The sizes are width x height in pixels of the window displayed in the cardholder browser. Possible values: <ul style="list-style-type: none"> ■ 01: 250x400 ■ 02: 390x400 ■ 03: 500x600 ■ 04: 600x400 ■ 05: Full page 	Enroll (O)	Integer (2)
payerAuthEnrollService_ addCardAttempts	Number of add card attempts in the last 24 hours. Recommended for Discover ProtectBuy.	Enroll (O)	Integer (3)
payerAuthEnrollService_ alternateAuthentication Data	Data that documents and supports a specific authentication process.	Enroll (O)	String (2048)
payerAuthEnrollService_ alternateAuthentication Date	Date and time in UTC of the cardholder authentication. Format: YYYYMMDDHHMM	Enroll (O)	Integer (12)

Table 18 Request Fields (Continued)

Field Name	Description	Required/ Optional	Type & Length
payerAuthEnrollService_ alternateAuthentication Method	<p>Mechanism used by the cardholder to authenticate to the 3D Secure requestor.</p> <p>Possible values:</p> <ul style="list-style-type: none"> ■ 01: No authentication occurred ■ 02: Login using merchant system credentials ■ 03: Login using Federated ID ■ 04: Login using issuer credentials ■ 05: Login using third-party authenticator ■ 06: Login using FIDO Authenticator 	Enroll (O)	Integer (2)
payerAuthEnrollService_ authenticationIndicator	<p>Indicates the type of authentication request.</p> <p>Possible values:</p> <ul style="list-style-type: none"> ■ 01: Payment transaction ■ 02: Recurring transaction ■ 03: Installment transaction ■ 04: Add card ■ 05: Maintain card ■ 06: Cardholder verification as part of EMV token ID&V (identity and verification) 	Enroll (O)	Integer (2)
payerAuthEnrollService_ authentication TransactionID	<p>Payer authentication transaction identifier passed to link the check enrollment and validate authentication messages.</p> <p>Note Required for Standard integration.</p>	Enroll (O)	String (20)

Table 18 Request Fields (Continued)

Field Name	Description	Required/ Optional	Type & Length
payerAuthEnrollService_ challengeCode	<p>Possible values:</p> <ul style="list-style-type: none"> ■ 01: No preference ■ 02: No challenge request ■ 03: Challenge requested (3D Secure requestor preference) ■ 04: Challenge requested (mandate) ■ 05: No challenge requested (transactional risk analysis is already performed) ■ 06: No challenge requested (Data share only) ■ 07: No challenge requested (strong consumer authentication is already performed) ■ 08: No challenge requested (utilize whitelist exemption if no challenge required) ■ 09: Challenge requested (whitelist prompt requested if challenge required) <p>Note This field will default to 01 on merchant configuration and can be overridden by the merchant. EMV 3D Secure version 2.1.0 supports values 01–04. Version 2.2.0 supports values 01–09.</p>	Enroll (O)	String (2)
payerAuthEnrollService_ customerCCAlias	<p>An alias that uniquely identifies the customer's account and credit card on file.</p> <p>Note This field is required if Tokenization is enabled in the merchant profile settings.</p>	Enroll (O)	String (128)
payerAuthEnrollService_ defaultCard	<p>Indicates that the card being used is the one designated as the primary payment card for purchase. This field can contain one of these values:</p> <ul style="list-style-type: none"> ■ true ■ false <p>Recommended for Discover ProtectBuy.</p>	Enroll (O)	String (5)

Table 18 Request Fields (Continued)

Field Name	Description	Required/ Optional	Type & Length
payerAuthEnrollService_ deviceChannel	<p>Indicates the channel used for the transaction.</p> <p>Note Required for SDK integration.</p> <p>Possible Values:</p> <ul style="list-style-type: none"> ■ SDK ■ Browser ■ 3RI (3D Secure Integrator Request) <p>Note If you use the SDK integration, this field is dynamically set to <code>SDK</code>. If you use the JavaScript code, this field is dynamically set to <code>Browser</code>. For merchant-initiated or 3RI transactions, you must set the field to <code>3RI</code>. If you use this field in addition to JavaScript code, you must set the field to <code>Browser</code>.</p>	Enroll (O)	String (10)
payerAuthEnrollService_ fraudActivity	<p>Indicates whether the merchant experienced suspicious activity (including previous fraud) on the account. This field can contain one of these values:</p> <ul style="list-style-type: none"> ■ 01: No suspicious activity ■ 02: Suspicious activity observed <p>Recommended for Discover ProtectBuy.</p>	Enroll (O)	String (2)
payerAuthEnrollService_ giftCardAmount	<p>The purchase amount total for prepaid gift cards in major units</p> <p>Example:</p> <p>123.45 USD= 123</p>	Enroll (O)	Integer (15)
payerAuthEnrollService_ giftCardCount	Total count of individual prepaid gift cards purchased.	Enroll (O)	Integer (2)
payerAuthEnrollService_ giftCardCurrency	Currency used for the gift card purchase. Use the standard ISO codes located in the Support Center .	Enroll (O)	Integer (3)
payerAuthEnrollService_ httpAccept	<p>Value of the <code>Accept</code> header sent by the customer's web browser.</p> <p>Note If the customer's browser provides a value, you must include it in your request.</p>	Enroll (O)	String (255)
payerAuthEnrollService_ httpUserAccept	The exact content of the HTTP accept header.	Enroll (O)	String (256)
payerAuthEnrollService_ httpUserAgent	<p>Value of the <code>User-Agent</code> header sent by the customer's web browser.</p> <p>Note If the customer's browser provides a value, you must include it in your request.</p>	Enroll (O)	String (255)

Table 18 Request Fields (Continued)

Field Name	Description	Required/ Optional	Type & Length
payerAuthEnrollService_ installmentTotalCount	An integer value greater than 1 indicating the maximum number of permitted authorizations for installment payments. Note This value is required if the merchant and cardholder have agreed to installment payments.	Enroll (O)	Integer (4)
payerAuthEnrollService_ marketingOptIn	Indicates whether the customer has opted in for marketing offers. This field can contain one of these values: <ul style="list-style-type: none"> ■ true ■ false Recommended for Discover ProtectBuy.	Enroll (O)	String (5)
payerAuthEnrollService_ marketingSource	Indicates origin of the marketing offer. Recommended for Discover ProtectBuy.	Enroll (O)	String (40)
payerAuthEnrollService_ MCC	Merchant category code. Important Required only for Visa Secure transactions in Brazil. Do not use this request field for any other types of transactions.	Enroll (O)	Integer (4)
payerAuthEnrollService_ merchantFraudRate	Calculated by merchants according to Payment Service Directive 2 (PSD2) and Regulatory Technical Standards (RTS). European Economic Area (EEA) card fraud divided by all EEA card volumes. Possible Values: <ul style="list-style-type: none"> ■ 1: Represents fraud rate <=1 ■ 2: Represents fraud rate >1 and <=6 ■ 3: Represents fraud rate >6 and <=13 ■ 4: Represents fraud rate >13 and <=25 ■ 5: Represents fraud rate >25 	Enroll (O)	Integer (2)
payerAuthEnrollService_ merchantName	Your company's name as you want it to appear to the customer in the issuing bank's authentication form. This value overrides the value specified by your merchant bank. Required for Visa Secure travel.	Enroll (O)	String (25)
payerAuthEnrollService_ merchantNewCustomer	Indicates whether the consumer is a new or existing customer with the merchant. This field can contain one of these values: <ul style="list-style-type: none"> ■ true ■ false 	Enroll (O)	String (5)/

Table 18 Request Fields (Continued)

Field Name	Description	Required/ Optional	Type & Length
payerAuthEnrollService_ merchantScore	Risk score provided by merchants. Used for Cartes Bancaires transactions.	Enroll (O)	String (20)
payerAuthEnrollService_ merchantURL	Address of your company's web site, for example, http://www.example.com. This value overrides the value specified by your merchant bank.	Enroll (O)	String (100)
payerAuthEnrollService_ messageCategory	Category of the message for a specific use case. Possible values: <ul style="list-style-type: none"> ■ 01: PA (payment authentication) ■ 02: NPA (non-payment authentication) ■ 03–79: Reserved for EMVCo future use (values invalid until defined by EMVCo) ■ 80–99: Reserved for DS (directory server) use 	Enroll (O)	String (2)
payerAuthEnrollService_ mobilePhone	Cardholder's mobile phone number. Used for Visa Secure transactions in Brazil.	Enroll (R)	Integer (25)
payerAuthEnrollService_ overridePaymentMethod	Specifies the Brazilian payment account type used for the transaction. This field overrides other payment types that might be specified in the request. Use one of the following values for this field: <ul style="list-style-type: none"> ■ NA: Not applicable. Do not override other payment types that are specified in the request. ■ CR: Credit card ■ DB: Debit card ■ VSAVR: Visa Vale Refeicao ■ VSAVA: Visa Vale Alimentacao <p>Important Required only for Visa Secure transactions in Brazil. Do not use this request field for any other types of transactions.</p>	Enroll (O)	String (10)

Table 18 Request Fields (Continued)

Field Name	Description	Required/ Optional	Type & Length
payerAuthEnrollService_ paymentAccountDate	<p>Date the payment account was added to the cardholder account.</p> <p>This field can contain one of these values:</p> <ul style="list-style-type: none"> ■ -1: Guest account ■ 0: Added during this transaction <p>If neither applies, enter the date in YYYYMMDD format.</p> <p>Recommended for Discover ProtectBuy.</p>	Enroll (O)	Integer (8)
payerAuthEnrollService_ preorder	<p>Indicates whether cardholder is placing an order with a future availability or release date.</p> <p>This field can contain one of these values:</p> <ul style="list-style-type: none"> ■ 01: Merchandise available ■ 02: Future availability 	Enroll (O)	String (2)
payerAuthEnrollService_ preorderDate	<p>Expected date that a pre-ordered purchase will be available.</p> <p>Format:</p> <p>YYYYMMDD</p>	Enroll (O)	Integer (8)
payerAuthEnrollService_ priorAuthenticationData	<p>This field contains data that the ACS can use to verify the authentication process.</p>	Enroll (O)	String (2048)
payerAuthEnrollService_ priorAuthenticationMethod	<p>Method the cardholder used previously to authenticate to the 3D Secure requestor.</p> <p>Possible values:</p> <ul style="list-style-type: none"> ■ 01: Frictionless authentication occurred by ACS ■ 02: Cardholder challenge occurred by ACS ■ 03: AVS verified ■ 04: Other issuer methods ■ 05–79: Reserved for EMVCo future use (values invalid until defined by EMVCo) ■ 80–99: Reserved for directory server use 	Enroll (O)	Integer (2)
payerAuthEnrollService_ priorAuthentication ReferenceID	<p>This field contains the ACS transaction ID for a prior authenticated transaction. For example, the first recurring transaction that was authenticated with the cardholder.</p>	Enroll (O)	String (36)
payerAuthEnrollService_ priorAuthenticationTime	<p>Date and time in UTC of the prior cardholder authentication.</p> <p>Format:</p> <p>YYYYMMDDHHMM</p>	Enroll (O)	Integer (12)

Table 18 Request Fields (Continued)

Field Name	Description	Required/ Optional	Type & Length
payerAuthEnrollService_ productCode	<p>Specifies the product code, which designates the type of transaction. Specify one of the following values for this field:</p> <ul style="list-style-type: none"> ■ AIR: Airline purchase Important Required for American Express SafeKey (U.S.). ■ ACC: Accommodation Rental ■ ACF: Account funding ■ CHA: Check acceptance ■ DIG: Digital Goods ■ DSP: Cash Dispensing ■ GAS: Fuel ■ GEN: General Retail ■ LUX: Luxury Retail ■ PAL: Prepaid activation and load ■ PHY: Goods or services purchase ■ QCT: Quasi-cash transaction ■ REN: Car Rental ■ RES: Restaurant ■ SVC: Services ■ TBD: Other ■ TRA: Travel <p>Important Required for Visa Secure transactions in Brazil. Do not use this request field for any other types of transactions.</p>	Enroll (O)	String (3)
payerAuthEnrollService_ recurringEndDate	<p>The date after which no further recurring authorizations should be performed. Format: YYYYMMDD.</p> <p>Note This field is required for recurring transactions.</p>	Enroll (O)	Integer (8)

Table 18 Request Fields (Continued)

Field Name	Description	Required/ Optional	Type & Length
payerAuthEnrollService_ recurringFrequency	<p>Integer value indicating the minimum number of days between recurring authorizations. A frequency of monthly is indicated by the value 28. Multiple of 28 days will be used to indicate months.</p> <p>Example:</p> <p>6 months= 168</p> <p>Example values accepted (31 days):</p> <ul style="list-style-type: none"> ■ 31 ■ 031 ■ 0031 <p>Note This field is required for recurring transactions.</p>	Enroll (O)	Integer (4)
payerAuthEnrollService_ recurringOriginalPurchase Date	<p>Date of original purchase. Required for recurring transactions.</p> <p>Format:</p> <p>YYYYMMDDHHMMSS</p> <p>Note If this field is empty, the current date is used.</p>	Enroll (O)	String (17)
payerAuthEnrollService_ referenceID	<p>Reference ID that corresponds to the device fingerprinting data that was collected previously.</p> <p>Note Required for Hybrid integration.</p>	Enroll (R)	String (50)

Table 18 Request Fields (Continued)

Field Name	Description	Required/ Optional	Type & Length
payerAuthEnrollService_ requestorInitiated AuthenticationIndicator	<p>Indicates the type of 3RI request (3D Secure Integrator Request).</p> <p>Possible Values:</p> <ul style="list-style-type: none"> ■ 01: Recurring transaction ■ 02: Installment transaction ■ 03: Add card ■ 04: Maintain card ■ 05: Account verification ■ 06: Split/delayed shipment ■ 07: Top-up ■ 08: Mail Order ■ 09: Telephone Order ■ 10: Whitelist status check ■ 11: Other payment <p>Note EMV 3D Secure version 2.1.0 supports values 01–05. Version 2.2.0 supports values 01–11.</p>	Enroll (O)	Integer (2)
payerAuthEnrollService_ reorder	<p>Indicates whether the cardholder is reordering previously purchased merchandise.</p> <p>This field can contain one of these values:</p> <ul style="list-style-type: none"> ■ 01: First time ordered ■ 02: Reordered 	Enroll (O)	String (2)
payerAuthEnrollService_ run	<p>Whether to include payerAuthEnrollService in your request. The field has one of these values:</p> <ul style="list-style-type: none"> ■ <code>true</code>: Include the service in your request. ■ <code>false</code> (default): Do not include the service in your request. 	Enroll (R)	String (5)
payerAuthEnrollService_ sdkMaxTimeout	<p>This field indicates the maximum amount of time for all 3D Secure 2.x messages to be communicated between all components (in minutes).</p> <p>Possible Values:</p> <ul style="list-style-type: none"> ■ Greater than or equal to 05 (05 is the minimum timeout to set) ■ Default is set to 15 <p>Note This field is a required for 3D Secure 2.x. If you do not send a value in this field, it defaults to 15.</p>	Enroll (O)	String (1)

Table 18 Request Fields (Continued)

Field Name	Description	Required/ Optional	Type & Length
payerAuthEnrollService_ secureCorporatePayment Indicator	Indicates that dedicated payment processes and procedures were used. Potential secure corporate payment exemption applies. Possible Values: <ul style="list-style-type: none">■ 0■ 1	Enroll (O)	String (1)
payerAuthEnrollService_ shipAddressUsageDate	Date when the shipping address for this transaction was first used. This field can contain one of these values: <ul style="list-style-type: none">■ -1: Guest account■ 0: First used during this transaction If neither applies, enter the date in YYYYMMDD format. Recommended for Discover ProtectBuy.	Enroll (O)	Integer (8)
payerAuthEnrollService_ totalOffersCount	Total number of articles or items in the order as a numeric decimal count. Possible values: 00–99	Enroll (O)	Integer (2)
payerAuthEnrollService_ transactionCountDay	Number of transaction (successful or abandoned) for this cardholder account within the last 24 hours. Recommended for Discover ProtectBuy.	Enroll (O)	Integer (3)
payerAuthEnrollService_ transactionCountYear	Number of transactions (successful and abandoned) for this cardholder account within the last year. Recommended for Discover ProtectBuy.	Enroll (O)	Integer (3)
payerAuthEnrollService_ transactionMode	Transaction mode identifier. Identifies the channel from which the transaction originates. Possible values: <ul style="list-style-type: none">■ M: MOTO (Mail Order Telephone Order)■ R: Retail■ S: eCommerce■ P: Mobile Device■ T: Tablet	Enroll (O)	String (1)

Table 18 Request Fields (Continued)

Field Name	Description	Required/ Optional	Type & Length
payerAuthEnrollService_ whiteListStatus	Enables the communication of trusted beneficiary and whitelist status among the ACS, the directory server, and the 3D Secure requestor. Possible Values: <ul style="list-style-type: none">■ Y: 3D Secure requestor is whitelisted by cardholder■ N: 3D Secure requestor is not whitelisted by cardholder	Enroll (O)	String (1)
payerAuthValidateService_ _authentication TransactionID	Payer authentication transaction identifier passed to link the check enrollment and validate authentication messages. Note Required for Hybrid integration.	Validate (O)	String (20)
payerAuthValidate Service_signedPARES	Payer authentication result (PARES) message returned by the card-issuing bank. If you need to show proof of enrollment checking, you may need to decrypt and parse the string for the information required by the payment card company. For more information, see "Storing Payer Authentication Data," page 196 . Note The field is in Base64. You must remove all carriage returns and line feeds before adding the PARES to the request.	Validate (O)	String (no limit, may be very large)
payerAuthValidate Service_run	Whether to include payerAuthValidateService in your request. The field can contain one of these values: <ul style="list-style-type: none">■ <code>true</code>: Include the service in your request.■ <code>false</code> (default): Do not include the service in your request.	Validate (R)	String (5)
purchaseTotals_currency	Currency used for the order. Use the standard ISO codes located in the Support Center .	Enroll (R) Validate (R)	String (5)
purchaseTotals_ grandTotalAmount	Grand total for the order. In your request, you must include either this field or item_#_unitPrice . For more information, see <i>Credit Card Services Using the Simple Order API</i> (PDF HTML). Note The purchaseTotals_grandTotalAmount field is optional if you use the item_#_unitPrice field.	Enroll (R) Validate (R)	String (15)

Table 18 Request Fields (Continued)

Field Name	Description	Required/ Optional	Type & Length
shipTo_city	City of the shipping address. Required if any shipping address information is included. Required for American Express SafeKey (U.S.).	Enroll (O)	String (50)
shipTo_country	Country of the shipping address. Use the two-character ISO Standard Country Codes . Required for American Express SafeKey (U.S.).	Enroll (O)	String (2)
shipTo_destinationCode	Indicates destination chosen for the transaction. Possible values: <ul style="list-style-type: none"> ■ 01: Ship to cardholder billing address ■ 02: Ship to another verified address on file with merchant ■ 03: Ship to address that is different than billing address ■ 04: Ship to store (store address should be populated on request) ■ 05: Digital goods ■ 06: Travel and event tickets, not shipped ■ 07: Other 	Enroll (O)	Integer (2)
shipTo_destinationTypes	Shipping destination. Example Commercial, residential, store	Enroll (O)	String (25)
shipTo_firstName	First name of the recipient.	Enroll (O)	String (60)
shipTo_lastName	Last name of the recipient.	Enroll (O)	String (60)
shipTo_phoneNumber	Phone number for the shipping address. For information on formatting, see billTo_phoneNumber .	Enroll (O)	String (15)

Table 18 Request Fields (Continued)

Field Name	Description	Required/ Optional	Type & Length
shipTo_postalCode	<p>Postal code for the shipping address. The postal code must consist of 5 to 9 digits.</p> <p>When the shipping country is the U.S., the 9-digit postal code must follow this format: [5 digits][dash][4 digits]</p> <p>Example 12345-6789</p> <p>When the shipping country is Canada, the 6-digit postal code must follow this format: [alpha][numeric][alpha][space] [numeric][alpha][numeric]</p> <p>Example A1B 2C3</p> <p>Required if the shipTo_country field value is US or CA.</p> <p>Required for American Express SafeKey (U.S.).</p>	Enroll (O)	String (10)
shipTo_shippingMethod	<p>Shipping method for the product. Possible values:</p> <ul style="list-style-type: none"> ■ lowcost: Lowest-cost service ■ sameday: Courier or same-day service ■ oneday: Next-day or overnight service ■ twoday: Two-day service ■ threeday: Three-day service ■ pickup: Store pick-up ■ other: Other shipping method ■ none: No shipping method because product is a service or subscription <p>Required for American Express SafeKey (U.S.).</p>	Enroll (O)	String (10)
shipTo_state	<p>State or province of the shipping address. Use the State, Province, and Territory Codes for the United States and Canada.</p> <p>Required if shipTo_country value is CA or US.</p> <p>Required for American Express SafeKey (U.S.).</p>	Enroll (O)	String (2)
shipTo_street1	<p>First line of the shipping address.</p> <p>Required if any shipping address information is included.</p> <p>Required for American Express SafeKey (U.S.).</p>	Enroll (O)	String (60)
shipTo_street2	<p>Second line of the shipping address.</p> <p>Required for American Express SafeKey (U.S.).</p>	Enroll (O)	String (60)

Reply Fields

Table 19 Reply Fields

Field Name	Description	Returned By	Type & Length
card_bin	Six-digit card issuer bank identification number.	Enroll Validate	String (6)
card_cardTypeName	The card brand name associated with the cardholder's card number.	Enroll Validate	String (25)
decision	Summarizes the result of the overall request. The field can contain one of these values: ■ ACCEPT ■ ERROR ■ REJECT	Enroll Validate	String (255)
invalidField_0 through invalidField_N	Fields in the request that contained invalid data.	Enroll Validate	String (255)
merchantReferenceCode	Your order reference or tracking number.	Enroll Validate	String (255)
missingField_0 through missingField_N	Required fields that were missing from the request.	Enroll Validate	String (255)

Table 19 Reply Fields (Continued)

Field Name	Description	Returned By	Type & Length
payerAuthEnrollReply_ acsRenderingType	<p>Identifies the UI type that the ACS will use to complete the challenge.</p> <p>Note Available only for mobile application transactions using the Cardinal Mobile SDK.</p> <p>This field is a JSON object that comprises the following two fields, each 2 characters in length.</p> <ul style="list-style-type: none"> ■ ACS Interface <p>Field Name: acsInterface</p> <p>This is the ACS interface the challenge presents to the cardholder.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • 01: Native UI • 02: HTML UI ■ ACS UI Template <p>Field Name: acsUiTemplate</p> <p>Identifies the UI template format that the ACS first presents to the consumer.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • 01: Text • 02: Single select • 03: Multi select • 04: OOB (Out of Band) • 05: HTML other <p>Valid values for each interface:</p> <ul style="list-style-type: none"> ■ Native UI: 01–04 ■ HTML UI: 01–05 <p>HTML other is valid only when combined with HTML UI. If used with Native UI, it results in error=203.</p> <p>JSON Object Example:</p> <pre>{ "acsRenderingType": { "acsInterface": "02", "acsUiTemplate": "03" } }</pre>	Enroll	String (see description)
payerAuthEnrollReply_ acsTransactionID	<p>Unique transaction identifier assigned by the ACS to identify a single transaction.</p>	Enroll	String (36)

Table 19 Reply Fields (Continued)

Field Name	Description	Returned By	Type & Length
payerAuthEnrollReply_acsURL	URL for the card-issuing bank's authentication form that you receive when the card is enrolled. The field length can be very large.	Enroll	String (no length limit)
payerAuthEnrollReply_authenticationPath	<p>Indicates what displays to the customer during the authentication process. This field can contain one of these values:</p> <ul style="list-style-type: none"> ■ ADS: (Card not enrolled) customer prompted to activate the card during the checkout process. ■ ATTEMPTS: (Attempts processing) <i>Processing...</i> briefly displays before the checkout process is completed. ■ ENROLLED: (Card enrolled) the card issuer's authentication window displays. ■ UNKNOWN: Card enrollment status cannot be determined. ■ NOREDIRECT: (Card not enrolled, authentication unavailable, or error occurred) nothing displays to the customer. <p>The following values can be returned if you are using rules-based payer authentication. See "Rules-Based Payer Authentication," page 212:</p> <ul style="list-style-type: none"> ■ RIBA: The card-issuing bank supports risk-based authentication, but whether the cardholder is likely to be challenged cannot be determined. ■ RIBA_PASS: The card-issuing bank supports risk-based authentication and it is likely that the cardholder will not be challenged to provide credentials, also known as <i>silent authentication</i>. 	Enroll	String (255)

Table 19 Reply Fields (Continued)

Field Name	Description	Returned By	Type & Length
payerAuthEnrollReply_authenticationResult	Raw authentication data that comes from the card-issuing bank. Primary authentication field that indicates if authentication was successful and if liability shift occurred. You should examine first the result of this field. This field contains one of these values: <ul style="list-style-type: none"> ■ -1: Invalid PAREs. ■ 0: Successful validation. ■ 1: Cardholder is not participating, but the attempt to authenticate was recorded. ■ 6: Issuer unable to perform authentication. ■ 9: Cardholder did not complete authentication. 	Enroll	String w/ numbers only (255)
payerAuthEnrollReply_authenticationStatus Message	Message that explains the content of payerAuthEnrollReply_authenticationResult .	Enroll	String (255)
payerAuthEnrollReply_authenticationStatus Reason	Provides additional information about the PAREs status value.	Enroll	Integer (2)
payerAuthEnrollReply_authentication TransactionID	Payer authentication transaction identifier passed to link the check enrollment and validate authentication messages.	Enroll	String (20)
payerAuthEnrollReply_authenticationType	Indicates the type of authentication that is used to challenge the card holder. Possible Values: <ul style="list-style-type: none"> ■ 01: Static ■ 02: Dynamic ■ 03: OOB (Out of Band) Note EMV 3D Secure version 2.1.0 supports values 01–03. Version 2.2.0 supports values 01–03.	Enroll	Integer (2)
payerAuthEnrollReply_authorizationPayload	The Base64-encoded JSON Payload of Cartes Bancaires Authorization Values returned in the challenge flow.	Enroll	String (255)
payerAuthEnrollReply_cardholderMessage	Text provided by the AC or issuer or both to the cardholder during a frictionless or decoupled transaction. The issuer can provide information to the cardholder. For example, "Additional authentication is needed for this transaction. Please contact (Issuer Name) at xxx-xxx-xxxx.". The issuing bank can choose to support this value.	Enroll	String (128)

Table 19 Reply Fields (Continued)

Field Name	Description	Returned By	Type & Length
payerAuthEnrollReply_cavv	<p>Unique identifier generated by the card-issuing bank after the customer is authenticated. The value is in Base64. When you request the card authorization service, CyberSource automatically converts the value, not the field name, to the format required by your payment processor.</p> <p>Note This field is generated only for Visa Secure, American Express SafeKey, JCB, Diners Club, and Discover transactions.</p>	Enroll	String (255)
payerAuthEnrollReply_cavvAlgorithm	<p>Field returned when payerAuthEnrollReply_paresStatus contains the values Y (successful authentication) or A (attempted authentication). This field contains one of these values:</p> <ul style="list-style-type: none"> 2: Visa, American Express, JCB, Diners Club, and Discover 3: Mastercard and Maestro <p>Note This field only applies if you use the Atos processor. If you use Atos, send the value of this field in the ccAuthService_cavvAlgorithm request field of the authorization service.</p>	Enroll	Integer (1)
payerAuthEnrollReply_challengeCancelCode	<p>Indicates why the transaction was canceled. Possible Values:</p> <ul style="list-style-type: none"> 01: Cardholder selected Cancel 02: Reserved for future EMVCo use (values invalid until defined by EMVCo). 03: Transaction timed out—Decoupled Authentication 04: Transaction timed out at ACS—other timeouts 05: Transaction timed out at ACS—First CReq not received by ACS 06: Transaction Error 07: Unknown 08: Transaction timed out at SDK 	Enroll	Integer (2)

Table 19 Reply Fields (Continued)

Field Name	Description	Returned By	Type & Length
payerAuthEnrollReply_challengeRequired	<p>Indicates whether a challenge is required in order to complete authentication.</p> <p>Note Regional mandates might determine that a challenge is required.</p> <p>Possible values:</p> <ul style="list-style-type: none"> ■ Y: Challenge required ■ N: Challenge not required <p>Note Used by the Hybrid integration.</p>	Enroll	String (1)
payerAuthEnrollReply_commerceIndicator	<p>Commerce indicator for cards not enrolled. This field contains one of these values:</p> <ul style="list-style-type: none"> ■ <code>internet</code>: Card not enrolled, or card type not supported by payer authentication. No liability shift. ■ <code>js_attempted</code>: JCB card not enrolled, but attempt to authenticate is recorded. Liability shift. ■ <code>js_failure</code>: You receive this result if JCB's directory service is not available. No liability shift. ■ <code>spa</code>: Mastercard card not enrolled. No liability shift. ■ <code>vbv_attempted</code>: Visa card not enrolled, but attempt to authenticate is recorded. Liability shift. ■ <code>vbv_failure</code>: For the payment processors Barclays, Streamline, AIBMS, or FDC Germany, you receive this result if Visa's directory service is not available. No liability shift. 	Enroll	String (255)
payerAuthEnrollReply_directoryServerErrorCode	The directory server error code indicating a problem with the transaction.	Enroll	Integer (3)
payerAuthEnrollReply_directoryServerErrorDescription	Directory server text and additional detail about the error for the transaction.	Enroll	String (4096)
payerAuthEnrollReply_directoryServerTransactionID	The directory server transaction ID is generated by the directory server during authentication and returned with the authentication results. Your card brand might require you to send this field in the authorization service request.	Enroll	String (36)

Table 19 Reply Fields (Continued)

Field Name	Description	Returned By	Type & Length
payerAuthEnrollReply_eci	<p>Note This field applies only to non U.S.-issued cards.</p> <p>Numeric electronic commerce indicator (ECI) returned only for Visa, American Express, JCB, Diners Club, and Discover transactions when the card is not enrolled.</p> <p>If you are not using the CyberSource payment services, you must send this value to your payment processor in the subsequent request for card authorization. This field contains one of these values:</p> <ul style="list-style-type: none"> 06: The card can be enrolled. Liability shift. 07: The card cannot be enrolled. No liability shift. 	Enroll	String (255)
payerAuthEnrollReply_eciRaw	<p>ECI value that can be returned for Visa, Mastercard, American Express, JCB, Diners Club, and Discover. The field is absent when authentication fails. If your payment processor is Streamline, you must pass the value of this field instead the value of payerAuthEnrollReply_eci or payerAuthEnrollReply_ucafCollectionIndicator.</p> <p>This field can contain one of these values:</p> <ul style="list-style-type: none"> 01: Authentication attempted (Mastercard) 02: Successful authentication (Mastercard) 05: Successful authentication (Visa, American Express, JCB, Diners Club, and Discover) 06: Authentication attempted (Visa, American Express, JCB, Diners Club, and Discover) 	Enroll	String (255)
payerAuthEnrollReply_effectiveAuthenticationType	<p>The type of 3D Secure transaction flow that occurred. It can be one of the following:</p> <ul style="list-style-type: none"> CH: Challenge FR: Frictionless FD: Frictionless with delegation, (challenge not generated by the issuer but by the scheme on behalf of the issuer). <p>Used for Cartes Bancaires transactions.</p>	Enroll	String (2)

Table 19 Reply Fields (Continued)

Field Name	Description	Returned By	Type & Length
payerAuthEnrollReply_ivrEnabledMessage	Indicates whether a valid Interactive Voice Response (IVR) transaction was detected.	Enroll	String (5)
payerAuthEnrollReply_ivrEncryptionKey	Encryption key to be used in the event the ACS requires encryption of the credential field.	Enroll	String (16)
payerAuthEnrollReply_ivrEncryptionMandatory	Indicates whether the ACS requires the credential to be encrypted.	Enroll	String (5)
payerAuthEnrollReply_ivrEncryptionType	An indicator from the ACS to inform the type of encryption that should be used in the event the ACS requires encryption of the credential field.	Enroll	String (20)
payerAuthEnrollReply_ivrLabel	An ACS-provided label that can be presented to the cardholder. Recommended use with an application.	Enroll	String (20)
payerAuthEnrollReply_ivrPrompt	An ACS-provided string that can be presented to the cardholder. Recommended use with an application.	Enroll	String (80)
payerAuthEnrollReply_ivrStatusMessage	An ACS-provided message that can provide additional information.	Enroll	String (80)
payerAuthEnrollReply_networkScore	The global score calculated by the Cartes Bancaires scoring platform and returned to the merchant.	Enroll	Integer (2)
payerAuthEnrollReply_paReq	Payer authentication request (PAReq) message that you need to forward to the ACS. The field length can be very large. The value is in Base64.	Enroll	String (No length limit)
payerAuthEnrollReply_paresStatus	<p>Raw result of the authentication check. This field can contain one of these values:</p> <ul style="list-style-type: none"> ■ A: Proof of authentication attempt was generated. ■ B: Bypassed authentication. ■ N: Customer failed or canceled authentication. Transaction denied. ■ R: Authentication rejected (used for 3D Secure 2.x transactions only) ■ U: Authentication not completed regardless of the reason. ■ Y: Customer was successfully authenticated. <p>Note If you are configured for Asia, Middle East, and Africa Gateway Processing, you must send the value of this field in your authorization request.</p>	Enroll	String (255)

Table 19 Reply Fields (Continued)

Field Name	Description	Returned By	Type & Length
payerAuthEnrollReply_ proofXML	<p>Date and time of the enrollment check combined with the <code>VEReq</code> and <code>VERes</code> elements. If you ever need to show proof of enrollment checking, you will need to parse the string for the information required by the payment card company. For more information, see "Storing Payer Authentication Data," page 196. The value can be very large.</p> <ul style="list-style-type: none"> ■ For cards issued in the U.S. or Canada, Visa may require this data for specific merchant category codes. ■ For cards not issued in the U.S. or Canada, your bank may require this data as proof of enrollment checking for any payer authentication transaction that you re-present because of a chargeback. 	Enroll	String (no length limit)
payerAuthEnrollReply_ proxyPAN	Encrypted version of the card number used in the payer authentication request message.	Enroll	String (255)
payerAuthEnrollReply_ reasonCode	Numeric value corresponding to the result of the Enrollment Check service request. For a list of possible values, see Appendix B, "Reason Codes," on page 183 .	Enroll	Integer (5)
payerAuthEnrollReply_ sdkTransactionID	SDK unique transaction identifier that is generated on each new transaction.	Enroll	String (36)
payerAuthEnrollReply_ specificationVersion	This field contains the 3D Secure version that was used to process the transaction. For example, 1.0.2 or 2.0.0.	Enroll	String (20)
payerAuthEnrollReply_ stepUpUrl	<p>The fully qualified URL that the merchant uses to post a form to the cardholder in order to complete the Consumer Authentication transaction for the Cardinal Cruise Direct Connection API integration.</p> <p>Note Used by the Cardinal Cruise Direct Connection API integration.</p>	Enroll	String (2048)
payerAuthEnrollReply_ threeDSServer TransactionID	Unique transaction identifier assigned by the 3D Secure Server to identify a single transaction.	Enroll	String (36)
payerAuthEnrollReply_ ucafAuthenticationData	<p>AAV is a unique identifier generated by the card-issuing bank after the customer is authenticated. The value is in Base64. Include the data in the card authorization request.</p> <p>Note This field is returned for only Mastercard Identity Check transactions.</p>	Enroll	String (255)

Table 19 Reply Fields (Continued)

Field Name	Description	Returned By	Type & Length
payerAuthEnrollReply_ ucafCollectionIndicator	Indicates that authentication is not required because the customer is not enrolled. Add the value of this field to the authorization field ucaf_collectionIndicator . This field can contain these values: 0, 1. Note This field is returned for only Mastercard Identity Check transactions.	Enroll	String (255)
payerAuthEnrollReply_ veresEnrolled	Result of the enrollment check. This field can contain one of these values: <ul style="list-style-type: none"> Y: Card enrolled; you must authenticate. Liability shift. N: Card not enrolled; proceed with authorization. Liability shift. U: Unable to authenticate regardless of the reason. No liability shift. Note This field only applies to the Asia, Middle East, and Africa Gateway. If you are configured for this processor, you must send the value of this field in your authorization request. The following value can be returned if you are using rules-based Payer Authentication. See "Rules-Based Payer Authentication," page 212 : <ul style="list-style-type: none"> B: Indicates that authentication was bypassed. 	Enroll	String (255)
payerAuthEnrollReply_ whiteListStatus	Enables the communication of trusted beneficiary and whitelist status among the ACS, the directory server, and the 3D Secure requestor. Possible Values: <ul style="list-style-type: none"> Y: 3D Secure requestor is whitelisted by cardholder N: 3D Secure requestor is not whitelisted by cardholder 	Enroll	String (1)
payerAuthEnrollReply_ whiteListStatusSource	This field is populated by the system setting Whitelist Status. Possible Values: <ul style="list-style-type: none"> 01: 3D Secure Server 02: Directory server 03: ACS 	Enroll	Integer (2)

Table 19 Reply Fields (Continued)

Field Name	Description	Returned By	Type & Length
payerAuthEnrollReply_xid	Transaction identifier generated by CyberSource for successful enrollment checks. Use this value to match an outgoing PAREq with an incoming PAREs. If your payment processor is Barclays, CyberSource forwards the XID with your card authorization service request. The value is in Base64.	Enroll	String (255)

Table 19 Reply Fields (Continued)

Field Name	Description	Returned By	Type & Length
payerAuthValidateReply_ acsRenderingType	<p>Identifies the UI type that the ACS will use to complete the challenge.</p> <p>Note Available only for mobile application transactions using the Cardinal Mobile SDK.</p> <p>This field is a JSON object that comprises the following two fields, each 2 characters in length.</p> <ul style="list-style-type: none"> ■ ACS Interface <p>Field Name: acsInterface</p> <p>This is the ACS interface the challenge presents to the cardholder.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • 01: Native UI • 02: HTML UI ■ ACS UI Template <p>Field Name: acsUiTemplate</p> <p>Identifies the UI template format that the ACS first presents to the consumer.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • 01: Text • 02: Single select • 03: Multi select • 04: OOB (Out of Band) • 05: HTML other <p>Valid values for each interface:</p> <ul style="list-style-type: none"> ■ Native UI: 01–04 ■ HTML UI: 01–05 <p>HTML other is valid only when combined with HTML UI. If used with Native UI, it results in error=203.</p> <p>JSON Object Example:</p> <pre>{ "acsRenderingType": { "acsInterface": "02", "acsUiTemplate": "03" } }</pre>		String (see description)
payerAuthValidateReply_ acsTransactionID	<p>Unique transaction identifier assigned by the ACS to identify a single transaction.</p>		String (36)

Table 19 Reply Fields (Continued)

Field Name	Description	Returned By	Type & Length
payerAuthValidateReply_authenticationResult	<p>Raw authentication data that comes from the card-issuing bank. Primary authentication field that indicates if authentication was successful and if liability shift occurred. You should examine first the result of this field. This field contains one of these values:</p> <ul style="list-style-type: none"> ■ -1: Invalid PARES. ■ 0: Successful validation. ■ 1: Cardholder is not participating, but the attempt to authenticate was recorded. ■ 6: Issuer unable to perform authentication. ■ 9: Cardholder did not complete authentication. 	Validate	String w/ numbers only (255)
payerAuthValidateReply_authenticationStatus Message	<p>Message that explains the content of payerAuthValidateReply_authenticationResult.</p>	Validate	String (255)
payerAuthValidateReply_authenticationStatus Reason	<p>Provides additional information about the PARES status value.</p>	Validate	Integer (2)
payerAuthValidateReply_authenticationType	<p>Indicates the type of authentication that is used to challenge the card holder.</p> <p>Possible Values:</p> <ul style="list-style-type: none"> ■ 01: Static ■ 02: Dynamic ■ 03: OOB (Out of Band) <p>Note EMV 3D Secure version 2.1.0 supports values 01–03. Version 2.2.0 supports values 01–03.</p>	Validate	Integer (2)
payerAuthValidateReply_authorizationPayload	<p>The Base64-encoded JSON Payload of Cartes Bancaires Authorization Values returned in the challenge flow.</p>	Validate	String (255)
payerAuthValidateReply_cavv	<p>Unique identifier generated by the card-issuing bank after the customer is authenticated. The value is in Base64. When you request the card authorization service, CyberSource automatically converts the value, not the field name, to the format required by your payment processor.</p> <p>Note This field is generated only for Visa Secure, American Express SafeKey, JCB, Diners Club, and Discover transactions.</p>	Validate	String (255)

Table 19 Reply Fields (Continued)

Field Name	Description	Returned By	Type & Length
payerAuthValidateReply_cavvAlgorithm	<p>Field returned when payerAuthValidateReply_paresStatus contains the values Y (successful authentication) or A (attempted authentication). This field contains one of these values:</p> <ul style="list-style-type: none"> ■ 2: Visa, American Express, JCB, Diners Club, and Discover ■ 3: Mastercard and Maestro <p>Note This field only applies if you use the Atos processor. If you use Atos, send the value of this field in the ccAuthService_cavvAlgorithm request field of the authorization service.</p>	Validate	Integer (1)
payerAuthValidateReply_challengeCancelCode	<p>Indicates why the transaction was canceled. Possible Values:</p> <ul style="list-style-type: none"> ■ 01: Cardholder selected Cancel ■ 02: Reserved for future EMVCo use (values invalid until defined by EMVCo). ■ 03: Transaction timed out—Decoupled Authentication ■ 04: Transaction timed out at ACS—other timeouts ■ 05: Transaction timed out at ACS—First CReq not received by ACS ■ 06: Transaction Error ■ 07: Unknown ■ 08: Transaction timed out at SDK 	Validate	Integer (2)

Table 19 Reply Fields (Continued)

Field Name	Description	Returned By	Type & Length
payerAuthValidateReply_commerceIndicator	<p>Indicator used to differentiate different types of transactions. The authentication failed if this field is not returned. The value of this field is passed automatically to the authorization service if you request the services together. This field contains one of these values:</p> <ul style="list-style-type: none"> ■ aesk: American Express SafeKey authentication verified successfully. ■ aesk_attempted: Card not enrolled in American Express SafeKey, but the attempt to authenticate was recorded. ■ dipb: Discover ProtectBuy authentication verified successfully. ■ dipb_attempted: Card not enrolled in Discover ProtectBuy, but the attempt to authenticate was recorded. ■ internet: Authentication failed. ■ js: J/Secure authentication verified successfully. ■ js_attempted: Card not enrolled in J/Secure, but the attempt to authenticate was recorded. ■ moto: Mail or telephone order. ■ pb_attempted: Card not enrolled in Diners Club ProtectBuy, but the attempt to authenticate was recorded. ■ recurring: Recurring transaction. ■ spa: Mastercard Identity Check authentication verified successfully. ■ spa_failure: Mastercard Identity Check failed authentication. ■ vbv: Visa Secure authentication verified successfully. ■ vbv_attempted: Card not enrolled in Visa Secure, but the attempt to authenticate was recorded. ■ vbv_failure: Visa Secure authentication unavailable. <p>Note For Visa, if the payment processor is Streamline, Barclays, AIBMS, or FDC Germany, you receive vbv_failure instead of internet when payerAuthValidateReply_ecl is 07.</p>	Validate	String (255)
payerAuthValidateReply_directoryServerErrorCode	The directory server error code indicating a problem with the transaction.	Validate	Integer (3)

Table 19 Reply Fields (Continued)

Field Name	Description	Returned By	Type & Length
payerAuthValidateReply_directoryServerErrorDescription	Directory server text and additional detail about the error for the transaction.	Validate	String (4096)
payerAuthValidateReply_directoryServerTransactionID	The Directory server transaction ID is generated by the directory server during authentication and returned with the authentication results. Your card brand might require you to send this field in the authorization service request.	Validate	String (36)
payerAuthValidateReply_eci	Numeric electronic commerce indicator (ECI) returned only for Visa, American Express, JCB, Diners Club, and Discover transactions. You must send this value to your payment processor in the subsequent request for card authorization. This field contains one of these values: <ul style="list-style-type: none"> ■ 05: Successful authentication ■ 06: Authentication attempted ■ 07: Failed authentication (No response from the merchant because of a problem.) 	Validate	String (255)
payerAuthValidateReply_eciRaw	ECI value that can be returned for Visa, Mastercard, American Express, JCB, Diners Club, and Discover. The field is absent when authentication fails. If your payment processor is Streamline, you must pass the value of this field instead the value of payerAuthValidateReply_eci or payerAuthValidateReply_ucafCollectionIndicator . <p>This field can contain one of these values:</p> <ul style="list-style-type: none"> ■ 01: Authentication attempted (Mastercard) ■ 02: Successful authentication (Mastercard) ■ 05: Successful authentication (Visa, American Express, JCB, Diners Club, and Discover) ■ 06: Authentication attempted (Visa, American Express, JCB, Diners Club, and Discover) 	Validate	String (255)

Table 19 Reply Fields (Continued)

Field Name	Description	Returned By	Type & Length
payerAuthValidateReply_effectiveAuthenticationType	<p>The type of 3D Secure transaction flow that occurred. It can be one of the following:</p> <ul style="list-style-type: none"> ■ CH: Challenge ■ FR: Frictionless ■ FD: Frictionless with delegation, (challenge not generated by the issuer but by the scheme on behalf of the issuer). <p>Used for Cartes Bancaires transactions</p>	Validate	String (2)
payerAuthValidateReply_interactionCounter	<p>Indicates the number of authentication cycles that the cardholder attempted. It is tracked by the issuing bank's ACS.</p> <p>Example When the customer receives the challenge window, enters their one-time password, and clicks submit, the interaction counter equals 1. When the customer receives the challenge window, receives the bank message asking if they want the one-time password sent to their phone or email, and they choose before going to the next screen to enter their one-time password, the interaction count equals 2. One count is to choose how to have their one-time password delivered. The second count is for entering the one-time password and clicking Submit.</p>	Validate	Integer (2)
payerAuthValidateReply_paresStatus	<p>Raw result of the authentication check. This field can contain one of these values:</p> <ul style="list-style-type: none"> ■ A: Proof of authentication attempt was generated. ■ N: Customer failed or canceled authentication. Transaction denied. ■ U: Authentication not completed regardless of the reason. ■ Y: Customer was successfully authenticated. <p>Note If you are configured for Asia, Middle East, and Africa Gateway Processing, you must send the value of this field in your authorization request.</p>	Validate	String (255)
payerAuthValidateReply_reasonCode	<p>Numeric value corresponding to the result of the validation request. For a list of possible values, see Appendix B, "Reason Codes," on page 183.</p>	Validate	Integer (5)

Table 19 Reply Fields (Continued)

Field Name	Description	Returned By	Type & Length
payerAuthValidateReply_sdkTransactionID	SDK unique transaction identifier that is generated on each new transaction.	Validate	String (36)
payerAuthValidateReply_specificationVersion	This field contains the 3D Secure version that was used to process the transaction. For example, 1.0.2 or 2.0.0.	Validate	String (20)
payerAuthValidateReply_threeDSServerTransactionID	Unique transaction identifier assigned by the 3D Secure Server to identify a single transaction.	Validate	String (36)
payerAuthValidateReply_ucafAuthenticationData	<p>AAV is a unique identifier generated by the card-issuing bank after the customer is authenticated. The value is in Base64. Include the data in the card authorization request.</p> <p>Note This field is returned for only Mastercard Identity Check transactions.</p>	Validate	String (255)
payerAuthValidateReply_ucafCollectionIndicator	<p>Numeric electronic commerce indicator (ECI). The field is absent when authentication fails. You must send this value to your payment processor in the request for card authorization. This field contain one of these values:</p> <ul style="list-style-type: none"> 0: UCAF collection is not supported at your web site. Customer authentication was not completed. 1: UCAF collection is supported at your web site, and UCAF was populated. Customer authentication was not completed. 2: UCAF collection is supported at your web site, and UCAF was populated. Customer completed authentication. <p>Note This field is returned for only Mastercard Identity Check transactions.</p>	Validate	String (255)
payerAuthValidateReply_whiteListStatus	<p>Enables the communication of trusted beneficiary and whitelist status among the ACS, the directory server, and the 3D Secure requestor.</p> <p>Possible Values:</p> <ul style="list-style-type: none"> Y: 3D Secure requestor is whitelisted by cardholder N: 3D Secure requestor is not whitelisted by cardholder 	Validate	String (1)

Table 19 Reply Fields (Continued)

Field Name	Description	Returned By	Type & Length
payerAuthValidateReply_whiteListStatusSource	<p>This field is populated by the system setting Whitelist Status.</p> <p>Possible Values:</p> <ul style="list-style-type: none"> ■ 01: 3D Secure Server ■ 02: Directory server ■ 03: ACS 	Validate	Integer (2)
payerAuthValidateReply_xid	<p>Transaction identifier generated by CyberSource for validation checks. Use this value, which is in Base64, to match the PAREq with the PAREs. CyberSource forwards the XID with the card authorization service to these payment processors:</p> <ul style="list-style-type: none"> ■ Barclays ■ Streamline when the commerce indicator is <code>spa</code> 	Validate	String (255)
purchaseTotals_currency	Currency used for the order. Use the standard ISO codes located in the Support Center .	Enroll Validate	String (255)
reasonCode	Numeric value corresponding to the result of the overall request. See Appendix B, "Reason Codes," on page 183 for a list of possible values.	Enroll Validate	Integer (5)
requestID	Identifier for the request.	Enroll Validate	String (255)
requestToken	<p>Identifier for the request generated by CyberSource.</p> <p>Request token data created by CyberSource for each reply. The field is an encoded string that contains no confidential information such as an account or card verification number. The string can contain a maximum of 256 characters.</p>	Enroll Validate	String (256)

Reason Codes

The following table lists the reason codes that are returned with the reply. 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.

Table 20 Reason Codes

Reason Code	Description
100	Successful transaction.
101	The request is missing one or more required fields. Possible action: See the reply 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 reply 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 CyberSource 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.

Request and Reply Examples

This appendix contains examples for the check enrollment service and the validate authentication service. All examples are in name-value pair format.



Important

These examples contain only the fields essential to the demonstration. Do not prepare your implementation according to the fields shown in these examples. They are provided for your information only.

Standard Integration Examples

The following is an example of a request for the check enrollment service:

Check Enrollment Request Example

Example 25 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=xxxxxxxxxxxxxxxxxx  
card_cardType=001  
payerAuthEnrollService_authenticationTransactionID=F18d1UW9VwTyawKTdex0
```


Check Enrollment Reply Example

Example 26 Transaction Reply 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=F18d1UW9VwTyawKTdex0
payerAuthEnrollReply_cavv=Y2FyZGluYWxjb21tZXJjZWFlOGg=
payerAuthEnrollReply_commerceIndicator=vbv
payerAuthEnrollReply_eci=5
payerAuthEnrollReply_eciRaw=05
payerAuthEnrollReply_paresStatus=Y
payerAuthEnrollReply_reasonCode=100
payerAuthEnrollReply_specificationVersion=2.0.1
payerAuthEnrollReply_veresEnrolled=Y
```

Hybrid Integration Examples

Check Enrollment Request Example

Example 27 Check Enrollment

```
payerAuthEnrollService_run=true
merchantID=patest
merchantReferenceCode=23AEE8CB6B62EE2AF07
item_0_unitPrice=19.99
purchaseTotals_currency=USD
card_expirationMonth=01
card_expirationYear=2020
card_accountNumber=xxxxxxxxxxxxxxxxxx
card_cardType=001
...
<Other 2.0 optional fields>
referenceID=CybsTester-778d0f67
```

Check Enrollment Reply Example

Example 28 Transaction Reply for Mastercard with Identity Check

```
requestID=0340290070000167905080
merchantReferenceCode=23AEE8CB6B62EE2AF07
purchaseTotals_currency=USD
decision=REJECT
reasonCode=475
payerAuthEnrollReply_reasonCode=475
payerAuthEnrollReply_acsURL=https://www.example.com
payerAuthEnrollReply_authenticationTransactionID=x0Jpbq2uIT7o0tQqwd60
payerAuthEnrollReply_paReq=value in base64:
eJxVUuFyggjAMfhXPw9TkV9g6...
payerAuthEnrollReply_specificationVersion=2.0.1
payerAuthEnrollReply_veresEnrolled=Y
request_
token=AhjzbwSTHCfKtXsaE6e1EQJP+BFNcZtIHTiD9au3tjijj5Uar+AuAAAAkAY5
```

Validate Authentication Request Example

Example 29 Validate Authentication Request

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

Validate Authentication Reply Example

Example 30 Transaction Reply for Mastercard with Identity Check

```
requestID=0340290070000167905080
merchantReferenceCode=23AEE8CB6B62EE2AF07
purchaseTotals_currency=USD
decision=ACCEPT
reasonCode=100
payerAuthValidateReply_reasonCode=100
payerAuthValidateReply_authenticationResult=0
payerAuthValidateReply_authenticationStatusMessage=Success
payerAuthValidateReply_cavv=Y2FyZGluYWxjb21tZXJjZWFlOGg=
payerAuthValidateReply_commerceIndicator=vbv
payerAuthValidateReply_eci=5
payerAuthValidateReply_eciRaw=05
payerAuthValidateReply_paresStatus=Y
payerAuthValidateReply_reasonCode=100
payerAuthValidateReply_specificationVersion=2.0.1
request_
token=AhjzbwSTHCfKtXsaE6e1EQJP+BFNcZtIHTiD9au3tjijj5Uar+AuAAAAkAY5
```

Web Site Modification Reference

This appendix contains information about modifying your web site to integrate CyberSource Payer Authentication services into your checkout process. It also provides links to payment card company web sites where you can download the appropriate logos.

Web Site Modification Checklist

1 Modify web page buttons:

- Order submission button: disable the final “buy” button until the customer completes all payment and authentication requirements.
- Browser back button: account 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.

2 Add appropriate logos:

- Make sure you have downloaded 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 ["3D Secure Services Logos," page 191](#).

3 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 or to sign up for the authentication program specific to their card. For examples of messages you can use, see ["Informational Message Examples," page 192](#).

3D Secure Services Logos

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

Table 21 3D Secure Services Logos Download Location

3D Secure Service	Download Location
Visa Secure	https://usa.visa.com/run-your-business/small-business-tools/payment-technology/visa-secure.html This web site 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 web site contains information about Identity Check, links to logos for download, and information about integrating the Identity Check information into your web site 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 web site contains information about SafeKey and links to logos for download.
JCB J/Secure	http://partner.jcbcard.com/security/jsecure/logo.html This web site 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 web site contains information about Discover ProtectBuy and links to logos for download.

Informational Message Examples

Add a brief message adjacent to your final buy button on your checkout page to inform customers that they may be prompted to provide their authentication password or to enroll in the authentication program for their card.

The following examples may be used, but consult your specific card authentication program to make sure you conform to their messaging requirements.

Example 31

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 will be 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 32

Your card may be eligible for enrollment or is enrolled in the 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 for your password before you complete your purchase.

Payer Authentication Transaction Details in the Business Center

This appendix describes how to search the Business Center for details of transactions that are screened by CyberSource Payer Authentication. Transaction data is stored for 12 months so that you can send it to payment card companies if necessary.

Searching for Payer Authentication Details

Payer authentication data that is returned in API reply fields can be searched by using Transaction Search in the Business Center.

With other services, green means that the service request was successful, red means that it failed, and black means that the service request was not run. However, you need to interpret the result of the enrollment check differently:

- If the card is enrolled, the application result is shown in red, which indicates that you need to authenticate the user before you can request card authorization.
- If the card is not enrolled, the application result is shown in green because you do not need to authenticate the user. You can authorize the card immediately.

Enrolled Card

When a card is enrolled, the process consists of two steps: after you check for enrollment, you need to authenticate the customer.

Enrollment Check

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

- Request Information section: The enrollment check service is shown in red because the card is enrolled. You receive the corresponding reply information. If the card authorization service had been requested at the same time, it would not have been run and would appear in black.

You can obtain additional information about related orders by clicking the links on the right (Event Search and Details).

- Order Information section: When authentication is required, save the XID for use later. You do not receive an ECI or AAV_CAVV because the authentication is not complete.

You need to authenticate the user by requesting the validation service.

Events Related to Payer Authentication

When the XID value is available, you also have the option to search for other parts of the transaction with the By Payer Authentication History under Similar Searches link.

You can use the link to find the details page that shows the associated card validation and authorization results. On the results page:

- The most recent event is the successful authentication. If you click the request ID, the authentication details page opens. If the event also returned an XID value, the By Payer Authentication History link is present. If you click it, you return to the results page.
- The older event is the enrollment check.

If the card authorization service had been requested at the same time as payer authentication, authorization would not have run with the enrollment check but with the validate authentication request. On the results page:

- The most recent event is the combined successful customer authentication and card authorization. If you click the request ID, the details page opens.
- The older event is the enrollment check in red because the card is enrolled.

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 the following sections:

- Request Information section: The validation service succeeded. You receive the reason code 100, and the corresponding reply message. The necessary payer authentication information was passed to the card authorization service, which was 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 enrollment check service result is shown 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 the following sections:

- Request Information section: the service is shown 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:
 - The ECI value is 1: authentication is not required because the customer's Mastercard card is not enrolled.
 - 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 is not enrolled.

Payer Authentication Search

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

- Search options:
 - Use the XID as search parameter to find both parts of a transaction processed with an enrolled card. When using the XID as search option, make sure to keep the = sign at the end of the string.
 - 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 XID and the customer's account number (PAN). Use the XID to find all parts of the transaction.
- Payer authentication details: all transaction details are discussed under ["Searching for Payer Authentication Details," page 193](#).

Storing Payer Authentication Data

Payment card companies allow a certain number of days between the payer authentication and the authorization requests. If you settle transactions older than the pre-determined number of days, payment card companies may require that you send them the AAV, CAVV, or the XID if a chargeback occurs. The requirements depend on the card type and the region. For more information, see 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.

You may be required to show the values that you receive in the PAREs, the proof XML, and the PAREq fields as proof of enrollment checking for any payer authentication transaction that you present again because of a chargeback. Your account provider may require that you provide all data in human-readable format, so make sure that you can decode the PAREq and PAREs. For enrollment reply examples, see [Appendix C, "Request and Reply Examples," on page 184](#). The replies are similar for all card types.

Payment card companies have implemented the [3D Secure](#) protocol in different ways throughout the world. CyberSource recommends that you contact your merchant account provider to find out what is required. For more information on decrypting and providing the PAREs contact your account manager.

Payer Authentication Reports

This chapter describes the reports that you can download from the Business Center. All reports on the production servers are retained for 16 months but the transaction history is only kept in the database for six months. All reports on the test servers are deleted after 60 days. Only transactions that were processed are reported. Those that resulted in system error or time-out are not. For more information about API replies and their meanings, see [Appendix A, "API Fields," on page 142](#).

**Note**

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 your transactions are screened by payer authentication: successful, attempted, and incomplete authentication. The cards reported are Visa, Mastercard, Maestro, American Express, JCB, Diners Club, and Discover. 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:

- Step 1** In the left navigation panel, click the **Reports** icon.
- Step 2** Under Transaction Reports, click **Payer Auth Summary**. The Payer Auth Summary Report page appears.
- Step 3** In the search toolbar, select **Date Range** you want to include in the report. Account level users must select a merchant as well.
- Step 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 from the period prior to or following the selected period, click **Previous** or **Next** below the search toolbar.

Matching the Report to the Transaction Search Results

The figure below shows the search results that contain the transactions that appear in the above report. For more information on search results, see "[Searching for Payer Authentication Details](#)," page 193.

Figure 3 Payer Authentication Report Details

Mar 30 2006					
ubcvp1_2	1437540121000167904064	PATRICK MCMAHON	1.00 USD	Credit Card Authorization Payer Authentication Validation	
Mar 30 2006 03:42:16 PM	1143754012100	null@cybersource.com	0771		
ubcvp1_2	1437543646410167904065	P MAN	101.00 USD	Credit Card Authorization Payer Authentication Validation	
Mar 30 2006 03:41:17 PM	1143754364636	null@cybersource.com	0771		
ubcvp1_2	1437538846880167904064	PATRICK MCMAHON	16.00 USD	Credit Card Authorization Payer Authentication Validation	
Mar 30 2006 03:40:09 PM	1143753884687	null@cybersource.com	0771		

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, you receive a summary of your payer authentication validation results displayed as total amount and number of transactions.

Table 22 Payer Authentication Report Interpretation

Card Type	Interpretation	Protected?	Reported	
			Commerce Indicator	ECI
Visa, American Express, and JCB	No authentication	No	Internet	7
	Recorded attempt to authenticate	Yes	VbV, Aesk, or JS Attempted	6
	Successful authentication	Yes	VbV, JS, or Aesk	5
Mastercard and Maestro	No authentication	No	Internet ²	7 ¹
	Recorded attempt to authenticate	Yes	SPA	1
	Successful authentication	Yes	SPA	2
Diners Club and Discover	No authentication	No	Internet	7
	Recorded attempt to authenticate	Yes	PB or DIPB Attempted	6
	Successful authentication	Yes	PB or DIPB	5

1 Although the report heading is 7, you receive a collection indicator value of 1, or the reply field is empty.

2 Although the report heading is Internet, you receive `spa_failure` in the commerce indicator reply field.

Transactions are divided into two groups: those for which you are protected and those for which you are not:

- For Visa, American Express, JCB, Diners Club, and Discover: 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 because this report shows the transactions that were validated by payer authentication, not necessarily the transactions that were authorized. There are more likely to be reconciliation discrepancies if you process your authorizations outside of CyberSource.

Example 33 Payer Authentication Reports Compared to Payment Reports

For 10,000 orders, you may receive the following results:

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

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 Detail Report

Refer to the *Business Center Reporting User Guide* for instructions to download the report and additional report information.

Report Elements

<Report>

The <Report> element is the root element of the report.

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

Table 23 Child Elements of <Report>

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

Example <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>
```

Table 24 Child Elements of <PayerAuthDetail>

Element Name	Description	Type & Length
<RequestID>	Unique identifier generated by CyberSource for the transaction. This field corresponds to the requestID API field.	Numeric (26)
<MerchantID>	CyberSource merchant ID used for the transaction.	String (30)
<RequestDate>	Date on which the transaction was processed.	DateTime (25)
<TransactionType>	CyberSource service requested in SCMP format. This field can contain one of the following values: <ul style="list-style-type: none"> ■ ics_auth: Card authorization service ■ ics_pa_enroll: Payer Authentication Enrollment Check ■ ics_pa_validate: Payer Authentication Validation 	String (20)
<ProofXML>	Data that includes the date and time of the enrollment check and the VEReq and VERes elements. This field corresponds to the payerAuthEnrollReply_proofXML API field.	String (1024)
<VEReq>	Verify Enrollment Request (VEReq) sent by the merchant's server to the directory server and by the directory server to the ACS to determine whether authentication is available for the customer's card number. For a list of child elements, see " <VEReq> ," page 204.	
<VERes>	Verify Enrollment Response (VERes) sent by the directory server. For a list of child elements, see " <VERes> ," page 205.	
<PAREq>	Payer Authentication Request message that you send to the ACS through the payment card company. Corresponds to the payerAuthEnrollReply_paReq API field. For a list of child elements, see " <PAREq> ," page 206.	
<PAREs>	Payer Authentication Response message sent by the ACS. For a list of child elements, see " <PAREs> ," page 207.	
<AuthInfo>	Address and card verification data. For a list of child elements, see " <AuthInfo> ," page 209.	

Example <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>

```

Table 25 Child Elements of <ProofXML>

Element Name	Description	Type & Length
<Date>	Date when the proof XML is generated. Note 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)

Table 25 Child Elements of <ProofXML> (Continued)

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 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: <ul style="list-style-type: none"> ■ Y: Authentication available. ■ N: Cardholder not participating. ■ U: Unable to authenticate regardless of the reason. 	String (1)

Example <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>44444444</MerID>
  <Password />
  <Enrolled>Y</Enrolled>
</ProofXML>

```

<VEReq>

The <VEReq> element contains the enrollment check request data.

```

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

```

Table 26 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)

Example <VEReq> Element

```

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

```

<VERes>

The <VERes> element contains the enrollment check reply data.

```

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

```

Table 27 Child Elements of <VERes>

Element Name	Description	Type & Length
<Enrolled>	Result of the enrollment check. This field can contain one of these values: <ul style="list-style-type: none"> ■ 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)

Example <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>
```

Table 28 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 web site.	String
<XID>	Unique transaction identifier generated by CyberSource 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 reply. 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. Note 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 the following fields: ■ ccAuthReply_amount (see <i>Credit Card Services Using the Simple Order API</i> [PDF HTML]) or purchaseTotals_grandTotalAmount from external data.	Amount (15)
<AcctID>	Masked string used by the ACS.	String (28)
<Expiry>	Expiration month and year of the customer's card.	Number (4)

Example <PAREq> Element

```

<PAREq>
  <AcqBIN>123456</AcqBIN>
  <MerID>444444</MerID>
  <Name>example</Name>
  <Country>US</Country>
  <URL>http://www.example.com</URL>
  <XID>fr2VCDrbEdyC37MOPfIzMwAHBwE=</XID>
  <Date>2020-02-09T08:00:34-08:00</Date>
  <PurchaseAmount>1.00 USD</PurchaseAmount>
  <AcctID>NDAXMjAwMTAxMTAwMDc3MQ==</AcctID>
  <Expiry>2309</Expiry>
</PAREq>

```

<PAREs>

The <PAREs> element contains the payer authentication reply message.

```

<PAREs>
  (AcqBIN)
  (MerID)
  (XID)
  (Date)
  (PurchaseAmount)
  (PAN)
  (AuthDate)
  (Status)
  (CAVV)
  (ECI)
</PAREs>

```

Table 29 Child Elements of <PAREs>

Element Name	Description	Type & Length
<AcqBIN>	First six digits of the acquiring bank's identification number.	Numeric (6)
<MerID>	Identifier provided by your acquirer; used to log in to the ACS URL.	String (24)
<XID>	XID value returned in the customer authentication reply. This element corresponds to the payerAuthEnrollReply_xid and payerAuthValidateReply_xid API fields.	String (28)
<Date>	Date and time of request. Note 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)

Table 29 Child Elements of <PAREs> (Continued)

Element Name	Description	Type & Length
<PurchaseAmount>	Authorization amount and currency for the transaction. This element corresponds to the totals of the offer lines or from the following fields: ■ ccAuthReply_amount (see <i>Credit Card Services Using the Simple Order API</i> [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. Note 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. ■ U: Authenticate not completed regardless of the reason. ■ A: Proof of authentication attempt was generated.	String (1)
<CAVV>	CAVV (Visa, American Express, JCB, Diners Club, and Discover cards = * below) or AAV (Mastercard, and Maestro cards = ** below) returned in the customer authentication reply. This element corresponds to the payerAuthValidateReply_cavv (*) and payerAuthValidateReply_ucafAuthenticationData (**) API fields.	String (50)
<ECI>	Electronic commerce indicator returned in the customer authentication reply. This element corresponds to the payerAuthValidateReply_eci (*) and payerAuthValidateReply_ucafCollectionIndicator (**) API fields.	Numeric (1)

Example <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>AAAAAAAAAAAAAAAAAAAAAAAAAAAA=</CAVV>
  <ECI>5</ECI>
</PAREs>

```

<AuthInfo>

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

```

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

```

Table 30 Child Elements of <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 <i>Credit Card Services Using the Simple Order API</i> (PDF HTML).	String (1)
<CVVResult>	Optional results of the card verification number test. See ccAuthReply_cvvCode or afsService_cvCode (if from external data) in <i>Credit Card Services Using the Simple Order API</i> (PDF HTML).	String (1)

Example <AuthInfo> Element

```

<AuthInfo>
  <AVSResult>Y</AVSResult>
  <CVVResult/>
</AuthInfo>

```

Examples

These examples show a complete transaction: the failed enrollment check (enrolled card) and the subsequent successful authentication.

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="1895549430000167904548"
      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="1895549900000167904548"
      TransactionType="ics_pa_validate"
      XID="2YNaNGDBEdydJ6WI6aFJWAAHBwE="
      RequestDate="2020-02-09T08:00:02-08:00"
      <PRes>
        <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>AAAAAAAAAAAAAAAAAAAAAAAAAAAA=</CAVV>
        <ECI>5</ECI>
      </PRes>
    </PayerAuthDetail>
  </PayerAuthDetails>
</Report>

```

Rules-Based Payer Authentication

Rules-based payer authentication enables you to specify rules that define how transactions are authenticated by a [3D Secure](#) card authentication program. For example, you can decide to turn off active authentication for transactions that would otherwise require customer interaction to avoid degrading the customer experience. However, you may decide to authenticate customers from card-issuing banks that use risk-based authentication because the authentication is performed without customer interaction.

To enable your account for rules-based payer authentication, contact your CyberSource sales representative.

**Note**

Depending on the card type and country, active mandates supersede rules-based payer authentication and revert to traditional 3D Secure.

Available Rules

By default, when payer authentication is enabled on your account, authentication is attempted on all transactions.

For transaction types that are not bypassed, you may be required to complete authentication.

You can enable one or more of the following authentication transaction types. Any transaction types that are set to bypass authentication return the reason code 100. If you receive reason code 475 from the enrollment check, you must complete validation even if no customer participation is needed.

Table 31 Rules-Based Payer Authentication Types

Authentication Type	Description	Test Case Example
Active Authentication	Customer is prompted to authenticate.	Test Case 1: Visa Secure Card Enrolled: Successful Authentication
Attempts Processing	Customer is prompted to enroll in a 3D Secure card authentication program. This transaction type provides full 3D Secure benefits.	Test Case 3: Visa Secure Card Enrolled: Attempts Processing
Non-Participating Bank	Card-issuing bank does not participate in a 3D Secure program. When enrollment is checked, this transaction type provides full 3D Secure benefits, including fraud chargeback liability shift for customer “I didn’t do it” transactions and interchange reduction of 5-59 basis points.	Test Case 9: Visa Secure Card Not Enrolled
Passive Authentication	Customer is not prompted to authenticate. This transaction type provides full 3D Secure benefits when passive authentication is completed.	Test Case 12: Visa Secure Enrollment RIBA_PASS
Risk-Based Bank	Card-issuing bank uses risk-based authentication. The likely outcome is that the customer is not challenged to enter credentials. Most authentications proceed without customer interaction. This transaction type provides full 3D Secure benefits.	Test Case 14: Visa Secure Enrollment RIBA

API Replies



Note

By default, API replies that are specifically associated with rules-based payer authentication are turned off. Contact CyberSource Customer Support to enable these API replies when rules are triggered.

Bypassed Authentication Transactions

When card authentication is bypassed as a result of your rules-based payer authentication configuration, you can receive the following value for enrollment checks:

- **payerAuthEnrollReply_veresEnrolled** = B (indicates that authentication was bypassed)

Risk-Based Bank Transactions

When a transaction involves a card-issuing bank that supports risk-based authentication, you may receive the following authentication path replies, depending on whether the card-issuing bank deems the transaction risky:

■ `payerAuthEnrollReply_authenticationPath`

- `= RIBA`

The card-issuing bank supports risk-based authentication, but whether the cardholder is likely to be challenged cannot be determined.

- `= RIBA_PASS`

The card-issuing bank supports risk-based authentication, and it is likely that the cardholder will not be challenged to provide credentials; also known as *silent authentication*.

Implementing Hybrid or Standard Payer Authentication

This appendix summarizes the process of integrating Payer Authentication services into your existing business processes with Hybrid or Standard integration. CyberSource Payer Authentication services use CardinalCommerce JavaScript to leverage the authentication. The JavaScript Documentation links in this section navigate to the Cardinal site.



Note

The Cardinal Cruise Direct Connection API is the recommended integration method. For more information, see [Chapter 2, Implementing Cardinal Cruise Direct Connection API Payer Authentication](#).

Hybrid Payer Authentication



Important

If you are using tokenization, you must use the Hybrid integration method.

Implementation Overview

Notify your CyberSource account representative that you want to implement payer authentication (3D Secure). Give them the CyberSource merchant ID that you will use for testing. For more information, see ["Required Merchant Information," page 21](#).

Implementation tasks include:

- Add the JavaScript code to your checkout page
- For each purchase request
 - Build the authentication request
 - Call the **payerAuthEnrollService**: Payer Authentication Enrollment Check service
 - Invoke the authentication
 - Handle declines

- Call the following services:
 - **payerAuthValidateService**: Payer Authentication Validation (only for Hybrid integration)
 - **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 [Chapter 5, "Testing Payer Authentication Services,"](#) on page 63.
- Ensure that your account is configured for production.

Process Flow for Hybrid 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 You request the Enrollment Check service, passing in transaction details and the **payerAuthEnrollService_referenceID** request field.
- 6 If the issuing bank does not require authentication, you receive the following information in the Enrollment Check reply:
 - E-commerce indicator
 - CAVV (all card types except Mastercard)
 - AAV (Mastercard only)
 - Transaction ID
 - 3D Secure version
 - Directory server transaction ID
- 7 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.
- 8 The JavaScript displays the authentication window, and 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 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, 3D 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," page 224](#).

Before You Begin

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.

Credentials/API Keys

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

Create the JSON Web Token (JWT)

The Cardinal Cruise Direct Connection API integration uses JWTs as the method of authentication.



Note

For security reasons, all JWT creation must be done on the server side.

When creating the JWT, use your company API Key as the JWT secret. You can use any JWT library that supports JSON Web Signature (JWS). For further information about JWTs, see <https://jwt.io/>.

JWT Claims

Table 32 lists the standard claims that can be used in a JWT claim set.

Table 32 JWT Claims

Claim Name		Description
Required	Note Each claim key is case sensitive.	
	jti	JWT ID - unique identifier for the JWT. This field should change each time a JWT is generated.
	iat	Issued at - the epoch time in seconds beginning when the JWT is issued. This value indicates how long a JWT has existed and can be used to determine if it is expired.
	iss	Issuer - identifier of who is issuing the JWT. Contains the API key identifier or name.
	OrgUnitId	The merchant SSO Org Unit Id.
	Payload	The JSON data object being sent. This object is usually an order object.
Optional	ReferenceId	Merchant-supplied identifier that can be used to match up data collected from the Cardinal Cruise Direct Connection API and enrollment check service.
	ObjectifyPayload	Boolean flag that indicates how the API should consume the payload claim. If set to true, the payload claim is an object. If set to false, the payload claim is a stringified object. Some JWT libraries do not support passing objects as claims; this allows those who only allow strings to use their libraries without customization.
	exp	Expiration - the numeric epoch time in which the JWT should be considered expired. This value is ignored if it is more than 4 hours.

JWT Examples

[Example 34](#) shows the JSON content of a basic JWT payload that passes an object within the payload claim.

Example 34 Raw JWT

```
{
  "jti": "a5a59bfb-ac06-4c5f-be5c-351b64ae608e",
  "iat": 1448997865,
  "iss": "56560a358b946e0c8452365ds",
  "OrgUnitId": "565607c18b946e058463ds8r",
  "Payload": {
    "OrderDetails": {
      "OrderNumber": "0e5c5bf2-ea64-42e8-9ee1-71fff6522e15",
      "Amount": "1500",
      "CurrencyCode": "840"
    }
  },
  "ObjectifyPayload": true,
  "ReferenceId": "c88b20c0-5047-11e6-8c35-8789b865ff15",
  "exp": 1449001465,
}
```

[Example 35](#) shows the JSON content of a basic JWT payload that passes a string within the payload claim.

Example 35 Stringified JWT

```
{
  "jti": "29311a10-5048-11e6-8c35-8789b865ff15",
  "iat": 1448997875,
  "iss": "56560a358b946e0c8452365ds",
  "OrgUnitId": "565607c18b946e058463ds8r",
  "Payload": "{\"OrderDetails\":{\"OrderNumber\":\"19ec6910-5048-11e6-8c35-8789b865ff15\",\"Amount\":\"1500\",\"CurrencyCode\":\"840\"}}",
  "ObjectifyPayload": false,
  "ReferenceId": "074fda80-5048-11e6-8c35-8789b865ff15",
  "exp": 1449001465,
}
```

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 allows the card-issuing bank's ACS provider to collect additional device data; it can help speed up the authentication process by collecting this data before the checkout page launches. This step occurs prior to authentication and must occur before the *Cardinal.start* event (Standard integration) or Check Enrollment service request (Hybrid integration). For further information, see the [JavaScript Documentation](#).

Implementing Hybrid Payer Authentication

Requesting the Check Enrollment Service (Hybrid)

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_mobilePhone](#)
- [payerAuthEnrollService_referenceID](#)
- [payerAuthEnrollService_run](#)
- [purchaseTotals_currency](#)
- [purchaseTotals_grandTotalAmount](#)

**Note**

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 144](#).

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 in order 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 Reply Fields) in the authorization service request (Card Authorization Request Fields).

[Table 33](#) lists these fields.

Table 33 Enrollment Check and Reply Fields

Identifier	Enrollment Check Reply Field	Card Authorization Request Field
E-commerce indicator	payerAuthEnrollReply_commerceIndicator	ccAuthService_commerceIndicator
Collection indicator (Mastercard only)	payerAuthEnrollReply_ucafCollectionIndicator	ucaf_collectionIndicator
Result of the enrollment check for Asia, Middle East, and Africa Gateway	payerAuthEnrollReply_veresEnrolled	ccAuthService_veresEnrolled
3D Secure version	payerAuthEnrollReply_specificationVersion	ccAuthService_paSpecificationVersion

Table 33 Enrollment Check and Reply Fields (Continued)

Identifier	Enrollment Check Reply Field	Card Authorization Request Field
Directory server transaction ID Note Not required for 3D Secure 1.0.	payerAuthEnrollReply_ directoryServerTransactionID	ccAuthService_ directoryServerTransactionID

Interpreting the Reply

The replies are similar for all card types. See [Appendix C, "Request and Reply Examples," on page 184](#) for examples of enrollment replies.

- *Enrolled Cards*

You receive reason code 475 if the customer's card is enrolled in a payer authentication program. When you receive this reply, you can proceed to validate authentication.

- *Cards Not Enrolled*

You receive reason code 100 in the following cases:

- When the account number is not enrolled in a payer authentication program. The other services in your request are processed normally.
- When payer authentication is not supported by the card type.

When you receive this reply, you can proceed to card authorization.

Authenticating Enrolled Cards

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** reply field in the *Cardinal.continue* function in order to proceed with the authentication session as shown in [Example 36](#).

Example 36 Cardinal.continue

```
Cardinal.continue('cca',

{

    "AcsUrl":"https://testcustomer34.cardinalcommerce.com/
merchantacsfrontend/pareq.jsp?vaa=b&gold=AAAAAAAA...AAAAAA",

    "Payload":"eNpVUk1zgjAQvedXME7PJEFBVdKt1CECeDkVCk2PcfcnNjv8Kr+7tx4n1bGO
cz/se6GluMENPTPeeIz1G37WGEUth7YnpO21TfTvF3wDCBqspQ=="

},

{

    "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.

Receiving the Authentication Results

Next, *payments.validated* launches, and returns the authentication results and response JWT along with the **processor transaction ID** as shown in [Example 37](#).

Example 37 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 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 reply that you receive contains the validation result.

CyberSource recommends that you request both payer authentication and card authorization services at the same time. When you do so, CyberSource automatically sends the correct information to your payment processor; CyberSource converts the values of these fields 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 Reply 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, and Discover include the CAVV (cardholder authentication verification value).
- For Mastercard, include the UCAF (universal cardholder authentication field) and the collection indicator.

Table 34 lists these fields.

Table 34 Validation Check and Reply Fields

Identifier	Validation Check Reply 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
3D Secure version	payerAuthValidateReply_specificationVersion	ccAuthService_paSpecificationVersion
Directory server transaction ID	payerAuthValidateReply_directoryServerTransactionID	ccAuthService_directoryServerTransactionID
Note Not required for 3D Secure 1.0.		

Interpreting the Reply



Important

If the authentication fails, Visa, American Express, JCB, Diners Club, and Discover require that you do 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 replies are similar for all card types:

- **Success:**

You receive the 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 CyberSource 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 replies 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 nonenrolled 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.
```

Standard Payer Authentication

Implementation Overview

Notify your CyberSource account representative that you want to implement payer authentication (3D Secure). Give them the CyberSource merchant ID that you will use for testing. For more information, see "[Required Merchant Information](#)," [page 21](#).

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 [Chapter 5, "Testing Payer Authentication Services,"](#) on page 63.
- 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 web site, 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 232.](#)

Before You Begin

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.

Credentials/API Keys

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

Create the JSON Web Token (JWT)

The Cardinal Cruise Direct Connection API integration uses JWTs as the method of authentication.



Note

For security reasons, all JWT creation must be done on the server side.

When creating the JWT, use your company API Key as the JWT secret. You can use any JWT library that supports JSON Web Signature (JWS). For further information about JWTs, see <https://jwt.io/>.

JWT Claims

[Table 32](#) lists the standard claims that can be used in a JWT claim set.

Table 35 JWT Claims

Claim Name	Description
Required	Note Each claim key is case sensitive.
jti	JWT ID - unique identifier for the JWT. This field should change each time a JWT is generated.
iat	Issued at - the epoch time in seconds beginning when the JWT is issued. This value indicates how long a JWT has existed and can be used to determine if it is expired.

Table 35 JWT Claims (Continued)

Claim Name		Description
	iss	Issuer - identifier of who is issuing the JWT. Contains the API key identifier or name.
	OrgUnitId	The merchant SSO Org Unit Id.
	Payload	The JSON data object being sent. This object is usually an order object.
Optional	ReferenceId	Merchant-supplied identifier that can be used to match up data collected from the Cardinal Cruise Direct Connection API and enrollment check service.
	ObjectifyPayload	Boolean flag that indicates how the API should consume the payload claim. If set to true, the payload claim is an object. If set to false, the payload claim is a stringified object. Some JWT libraries do not support passing objects as claims; this allows those who only allow strings to use their libraries without customization.
	exp	Expiration - the numeric epoch time in which the JWT should be considered expired. This value is ignored if it is more than 4 hours.

JWT Examples

Example 34 shows the JSON content of a basic JWT payload that passes an object within the payload claim.

Example 38 Raw JWT

```
{
  "jti": "a5a59bfb-ac06-4c5f-be5c-351b64ae608e",
  "iat": 1448997865,
  "iss": "56560a358b946e0c8452365ds",
  "OrgUnitId": "565607c18b946e058463ds8r",
  "Payload": {
    "OrderDetails": {
      "OrderNumber": "0e5c5bf2-ea64-42e8-9ee1-71fff6522e15",
      "Amount": "1500",
      "CurrencyCode": "840"
    }
  },
  "ObjectifyPayload": true,
  "ReferenceId": "c88b20c0-5047-11e6-8c35-8789b865ff15",
  "exp": 1449001465,
}
```

[Example 35](#) shows the JSON content of a basic JWT payload that passes a string within the payload claim.

Example 39 Stringified JWT

```
{
  "jti": "29311a10-5048-11e6-8c35-8789b865ff15",
  "iat": 1448997875,
  "iss": "56560a358b946e0c8452365ds",
  "OrgUnitId": "565607c18b946e058463ds8r",
  "Payload": "{\"OrderDetails\":{\"OrderNumber\":\"19ec6910-5048-11e6-8c35-8789b865ff15\",\"Amount\":\"1500\",\"CurrencyCode\":\"840\"}}",
  "ObjectifyPayload" false
  "ReferenceId": "074fda80-5048-11e6-8c35-8789b865ff15"
  "exp":1449001465,
}
```

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 allows the card-issuing bank's ACS provider to collect additional device data; it can help speed up the authentication process by collecting this data before the checkout page launches. This step occurs prior to authentication and must occur before the *Cardinal.start* event (Standard integration) or Check Enrollment service request (Hybrid integration). For further information, see the [JavaScript Documentation](#).

Implementing Standard Payer Authentication

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 [Example 40](#). The second argument of data is a Request Order Object. You can construct this object ahead of time or pass it directly as shown.

Example 40 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>
    }
  }
  ...
});
```

Payments.validated returns the authentication results and response JWT along with the **processor transaction ID** as shown in [Example 41](#).

Example 41 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"
  }
}
```

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 nonenrolled 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.

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_authenticationTransactionID** 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_mobilePhone](#)
- [payerAuthEnrollService_referenceID](#)
- [payerAuthEnrollService_run](#)
- [purchaseTotals_currency](#)
- [purchaseTotals_grandTotalAmount](#)

**Note**

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 144](#).

CyberSource recommends that you request both payer authentication and card authorization services at the same time. When you do so, CyberSource automatically sends the correct information to your payment processor; CyberSource converts the values of these fields 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 Reply 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, and Discover include the CAVV (cardholder authentication verification value).

- For Mastercard, include the UCAF (universal cardholder authentication field) and the collection indicator.

Table 36 lists these fields.

Table 36 Enrollment Check and Reply Fields

Identifier	Enrollment Check Reply 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	payerAuthEnrollReply_directoryServerTransactionID	ccAuthService_directoryServerTransactionID
Note Not required for 3D Secure 1.0.		

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.

Glossary

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Numerics

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

A

AAV Account Authentication Value. Unique 32-character transaction token for a 3D 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 A 6-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 reply to the request to check enrollment. This is where you must send the [PAREq](#) so that the customer can be authenticated.

ADS Activation During Shopping. The card issuer’s ability to ask the cardholder to enroll in the card authentication service when the merchant posts to the [ACS URL](#).

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

A (Continued)

American Express	A globally issued card type that starts with 3 and which is identified as card type 003 by CyberSource. These cards participate in a card authentication service (SafeKey) provided by 3D Secure .
API	Application Programming Interface. A specification that can be 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.

B

Base64	Standard encoding method for data transfer over the Internet.
BIN	Bank Identification Number. The 6-digit number at the beginning of the card that identifies the card issuer.

C

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 reply passed back when the PAREs status is a Y or an A. If your processor is ATOS, this must be passed in the authorization, if available.
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.

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

D

Diners Club	A globally issued card type that starts with a 3 or a 5. CyberSource identifies Diners Club cards with a card type of 005. These cards participate in a card authentication service (ProtectBuy) provided by 3D 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. CyberSource identifies Discover cards with a card type of 004. These cards participate in a card authentication service (ProtectBuy) provided by 3D Secure .

E

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

H

HTTP	Hypertext Transfer Protocol. An application protocol used for data transfer on the Internet.
HTTP POST request	POST is one of the request methods supported by the HTTP protocol. The POST request method is used when the client needs to send data to the server as part of the request, such as when uploading a file or submitting a completed form.
HTTPS	Hypertext Transfer Protocol combined with SSL/TLS (Secure Sockets Layer/Transport Layer Security) to provide secure encryption of data transferred over the Internet.

I

Identity Check	Trademarked name for Mastercard's card authentication service.
-----------------------	----------------------------------------------------------------

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

I (Continued)

issuer The bank that issued a credit card.

J

J/Secure The [3D Secure](#) program of [JCB](#).

JCB Japan Credit Bureau. A globally issued card type that starts with a 3. CyberSource identifies JCB cards with a card type of 007. These cards participate in a card authentication service ([J/Secure](#)) provided by [3D Secure](#).

M

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. CyberSource identifies Maestro cards with the 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 SecureCode card authentication program.

Mastercard A globally issued card that includes credit and debit cards. These cards start with a 5. CyberSource identifies these cards as card type 002 for both credit and debit cards. These cards participate in a card authentication service ([Identity Check](#)) provided by [3D Secure](#).

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](#).

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

P

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. Pass this data into CyberSource 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	CyberSource field that 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.

R

request ID	A 22- or 23-digit number that uniquely identifies each transaction sent to CyberSource. 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 degree of risk that they perceive the transaction to have. 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.

S

SafeKey	Trademarked name for the American Express card authentication service.
----------------	------------------------------------------------------------------------

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

S (Continued)

SCMP API	CyberSource's legacy name-value pair API that has been superseded by the Simple Order API .
Simple Order API	CyberSource's current API, which provides three ways to access CyberSource services: name-value pair (NVP), XML, and SOAP.
Solo	A debit card type that was owned by Maestro. It was permanently discontinued March 31, 2011.
Switch	See Maestro .

T

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

U

UCAF	Universal Cardholder Authentication Field. A Base64-encoded string sent back with Mastercard Identity Check -enrolled cards that specifically identifies 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.

V

validate	CyberSource service for decoding and decrypting the PAREs to determine success. The <i>validate</i> 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.

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z**V (Continued)**

Visa A globally issued card that includes credit and debit cards. These cards start with a 4. CyberSource identifies these cards as card type 001 for both credit and debit cards. These cards participate in a card authentication service ([Visa Secure](#)) provided by [3D Secure](#).

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

X

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.