evopayments.com

3-D Secure 2.0

External Authentication Integration Guide

©2018 EVO Payments, Inc.

Table of Contents

Version History 2

Overview 3

- Data Fields 3

Workflow 4

- Authentication Request 4
- Authorization 5
- Authentication ECI Values by Card Brand 6
- Out of Scope Transactions 6
- Exemptions 7
  - Submitting an Exempted Transaction 9
- Follow-On Transactions 11
- Card on File for Non-Payment Transactions 11
  - Adding Card on File without Processing Payment 11
  - Repeat Card on File Transactions 11
- Card on File for Payment Transactions 12
  - Adding a Card on File as Part of a Single Payment 12
  - Repeat Card on File Transactions 12
  - Tokenization 14
  - Best Practices 14

Sandbox Trigger Values 15

- SCA Challenge Soft Decline Triggers 15
  - Low Value Exemption soft decline 15
  - Low Risk Exemption soft decline 15
  - Recurring Exemption soft decline 15
  - Delegated SCA Exemption soft decline 15
  - Whitelisted Exemption soft decline 16
  - Secure Corporate Exemption soft decline 16
  - Authentication Outage Exemption soft decline 16
  - SCA Exclusion Soft Decline Triggers 16
  - Merchant Initiated Repeat Card On File soft decline 16
- 3DS V1 Decommisioning 17

Version History

Version	Date	Description of Changes
V1	25 May 2020	> Initial version
V2	N/A	> Removed September 2019 mandate (out of date)
V3	16 June 2020	> .35R2 Updates for Exemptions and Out of Scope
V4	01 July 2020	> Added clarification on submitting an Exempted Transaction Request
V5	08 July 2020	> Workflow update for Follow-On Transactions
V6	09 July 2020	> Updated authenticationmethod value from frictionlessflow to frictionless > Updated timestamp for AuthenticationTimestamp > Removed ThreeDSMerchantData from the Exemption Transaction Request
V7	14 August 2020	> Card on File for Non-Payment Transactions section added > Addition of ProtocolVersion field
V8	21 August 2020	> Card on File for Payment Transactions section added
V9	06 October 2020	> Clarification on TokenResult from Authorization Response added
V10	15 September 2022	> AMEX Support , Trigger Values and 3DS V1 deprecation information added.

Overview

3-D Secure is a protocol developed to make online payments more secure through password authentication and cardholder verification. The new European Card Scheme mandates are the next wave of 3-D Secure that will bring additional eCommerce security to EMV. These updates will be supported for eService and TRON processing.

The contents of this document outline the process for an integrator using a third party for the Authentication piece of 3-D Secure 2.0 processing. This document assumes Authentication is complete at the time of this integration.

Data Fields

New values have been added to the information sent to Snap* on the Authorization message after External Authentication is complete.

The following new fields are located under BankcardTransaction/BankcardTenderData/EcommerceSecurityData/and are detailed below:

Required	Parameter	Data Type	Description
Conditional	AuthenticationECI	String	Payment System-specific value provided by the Access Control Server or Directory Server to indicate the results of the attempt to authenticate the Cardholder. The Electronic Commerce Indicator is required if TransactionStatus is SuccessfullyAuthenticated or UnableToAuthenticate.
Conditional	AuthenticationMethod	Enum	Mechanism used by the Cardholder to authenticate. > *NotSet* > *Frictionless* > *ChallengeFlow* > *AVSOnly* > *Other* The Authentication Method is required if TransactionStatus is SuccessfullyAuthenticated or UnableToAuthenticate.
Conditional	AuthenticationTimestamp	DateTime	Date and time in UTC of the cardholder authentication. The Authentication Timestamp is required if TransactionStatus is SuccessfullyAuthenticated or UnableToAuthenticate.
Conditional	AuthenticationValue	String	Payment System-specific value provided by the ACS or DS using an algorithm defined by Payment
			System. It is used to provide proof of authentication. This value is required if TransactionStatus is SuccessfullyAuthenticated or UnableToAuthenticate.

Required	DSTransactionID	String	Identifier assigned by the Directory Server to identify a single transaction.
Conditional	ACSTransactionID	String	Identifier assigned by the Access Control Server to identify a single transaction. This identifier is required if it is a Card On File transaction initiated by the Merchant, (i.e. BankcardTransactionData/CardOnFileInfo/InitiatedBy is Merchant).
Required	TransactionStatus	Enum	This value defines the authentication status for validation purposes. It is required for processing. Accepted enumerations: > *SuccessfullyAuthenticated* > *NotAuthenticated* > *UnableToAuthenticate* > *AttemptsProcessingPerformed* > *ChallengeRequired* > *DecoupledAuthenticationRequired* > *AuthenticationRejected InformationalOnly*

The following new fields are located under BankcardTransactionData/ThreeDSData and are detailed below:

Required	Parameter	Data Type	Description
Required	ProtocolVersion	Enum	This will be the Protocol Version Number the transaction was processed as. Could be changed from request due to Fallback. Accepted enumerations: > *NotSet* > *v1_0* > *v2_1_0* > *v2_2_0*

Workflow

This workflow outlines the process for an integrator using Third-party Authentication for the Authentication piece of 3-D Secure 2.0 processing.

Authentication Request

The integrator follows their outlined procedure for handling the Authentication workflow for 3-D Secure 2.0 processing, including the Cardholder Challenges, if required.

Authorization

Once the ISV has received all the necessary information from their authentication results, they will submit their transaction to the Snap* Platform, including the additional values listed above, for authorization. The Is3DSecure field should be set to False as this is an indicator for using the Snap* Authentication flow. ProtocolVersion should be set to the Protocol Version Number the transaction should be processed as.

Request

Note the addition of the ProtocolVersion field and the new fields in EcommerceSecurityData to the transaction request (bolded below):

```
<?xml version="1.0" encoding="utf-8"?>
<cwtxn:Transaction xmlns:cwtms="http://schemas.evosnap.com/CWS/v2.0/DataServices/TMS"
xmlns:cwsi="http://schemas.evosnap.com/CWS/v2.0/ServiceInformation"
xmlns:cwbcppro="http://schemas.evosnap.com/CWS/v2.0/Transactions/Bankcard/Pro"
xmlns:cweck="http://schemas.evosnap.com/CWS/v2.0/Transactions/ElectronicChecking"
xmlns:cwsva="http://schemas.evosnap.com/CWS/v2.0/Transactions/StoredValue"
xmlns:cwbcp="http://schemas.evosnap.com/CWS/v2.0/Transactions/Bankcard"
xmlns:cwenc="http://schemas.evosnap.com/CWS/v2.0/Transactions/Encryption"
p8:type="cwbcppro:BankcardTransactionPro" xmlns:p8="http://www.w3.org/2001/XMLSchemainstance"
xmlns:cwtxn="http://schemas.evosnap.com/CWS/v2.0/Transactions">
       <cwtxn:ExtensionData/>
       <cwtxn:IsOffline>false</cwtxn:IsOffline>
       <cwbcp:BankcardTenderData p8:type="cwbcppro:BankcardTenderDataPro">
              <cwtxn:ExtensionData/>
              <cwbcp:CardData>
                     <cwbcp:ExtensionData/>
                     <cwbcp:CardType>MasterCard</cwbcp:CardType>
                     <cwbcp:CardholderName>Mark Cruz Jr.</cwbcp:CardholderName>
                     <cwbcp:PAN>530780XXXXXX5130</cwbcp:PAN>
                     <cwbcp:Expire>0624</cwbcp:Expire>
              </cwbcp:CardData>
              <cwbcp:EcommerceSecurityData>
                     <cwbcp:AuthenticationMethod>Frictionless</cwbcp:AuthenticationMethod>
                     <cwbcp:AuthenticationTimestamp>2020-05-
06T21:12:25.047Z</cwbcp:AuthenticationTimestamp>
                     <cwbcp:AuthenticationECI>02</cwbcp:AuthenticationECI>

       <cwbcp:AuthenticationValue>MTIzNDU2Nzg5MDA5ODc2NTQzMjE=</cwbcp:AuthenticationValue>
                     <cwbcp:DSTransactionId>4556e481-1591-4c28-946b-
7eeb2ca1d1d6</cwbcp:DSTransactionId>

       <cwbcp:TransactionStatus>SuccessfullyAuthenticated</cwbcp:TransactionStatus>
              </cwbcp:EcommerceSecurityData>
       </cwbcp:BankcardTenderData>
       <cwbcp:BankcardTransactionData p8:type="cwbcppro:BankcardTransactionDataPro">
              <cwtxn:ExtensionData/>
              <cwtxn:Amount>1011.00</cwtxn:Amount>
              <cwtxn:CurrencyCode>USD</cwtxn:CurrencyCode>
               <cwtxn:TransactionDateTime>2020-0814T19:59:39.006325</cwtxn:TransactionDateTime>
               <cwtxn:InternalTransactionId>1000</cwtxn:InternalTransactionId>
 <cwtxn:InternalTxnDateTime>2020-08-
14T20:07:25.435065+00:00</cwtxn:InternalTxnDateTime>
              <cwtxn:InternalSeqNum>200</cwtxn:InternalSeqNum>
              <cwbcp:CashBackAmount>0.00</cwbcp:CashBackAmount>
```

```
            <cwbcp:EmployeeId>1234</cwbcp:EmployeeId>
            <cwbcp:EntryMode>Keyed</cwbcp:EntryMode>
            <cwbcp:TipAmount>0.00</cwbcp:TipAmount>
            <cwbcp:ThreeDSData>
                  <cwbcp:ProtocolVersion>v2_1_0</cwbcp:ProtocolVersion>
            </cwbcp:ThreeDSData>
      </cwbcp:BankcardTransactionData>
</cwtxn:Transaction>
```

Response

```
<cwbcp:TokenResult>05</cwbcp:TokenResult>
```

Authentication ECI Values by Card Brand

Electronic Commerce Indicator (ECI) is a value returned by Directory Servers (namely Visa, MasterCard, JCB, and American Express) indicating the outcome of authentication attempted on transactions enforced by 3DS.

Possible value returned by Visa, American Express, and JCB and its interpretation:

- ECI 05: 3DS authentication was successful; transactions are secured by 3DS.
- ECI 06: authentication was attempted but was not or could not be completed; possible reasons being either the card or its Issuing Bank has yet to participate in 3DS.
- ECI 07: 3DS authentication is either failed or could not be attempted; possible reasons being both card and Issuing Bank are not secured by 3DS, technical errors, or improper configuration.

Possible value returned by MasterCard and its interpretation:

- ECI 02: 3DS authentication is successful; both card and Issuing Bank are secured by 3DS.
- ECI 01: 3DS authentication was attempted but was not or could not be completed; possible reasons being either the card or its Issuing Bank has yet to participate in 3DS, or cardholder ran out of time to authorize.
- ECI 00: 3DS authentication is either failed or could not be attempted; possible reasons being both card and Issuing Bank are not secured by 3DS, technical errors, or improper configuration.

Out of Scope Transactions

3-D Secure 2.0 has presented options for Merchants to ensure the highest rate of Frictionless transaction processing. They have provided two categories of transactions where a Challenge is unlikely to be required.

First, the benefit of Out of Scope transactions is offered. The Out of Scope identifier represents transactions where the Cardholder is not available for Authentication. Because of this, there is little benefit to performing 3-D Secure Authentication on these transactions and they are considered out of scope for Authentication mandates.

Snap* Platform will bypass 3-D Secure Authentication and submit the transaction for Authorization if a transaction is identified by the Merchant Application as an Out of Scope transaction. The Authorization will identify the transaction as Out of Scope to achieve highest possibility of approval. There are four transaction types that qualify as Out of Scope:

1. **Merchant Initiated Transactions**, which is existing Snap* functionality, are defined as Out of Scope of
   Authentication since the Merchant is initiating the payment on behalf of the Cardholder. The initial MIT transaction where the Cardholder is setting up the recurrence will require Authentication. This is the transaction where CardOnFile is First. ThreeRIIndicator will now be an additional required field for 3-D Secure 2.0 MIT transactions. Snap* will identify MIT Out of Scope transactions as any payment where:
   - BankcardTransactionData/CardOnFileInfo/InitiatedBy is Merchant and
   - BankcardTransactionData/CardOnFileInfo/CardOnFile is Repeat and
   - BankcardTransactionData/ThreeDSData/ThreeRIIndicator is NotSet.

2. **MOTO transactions**, which is existing Snap* functionality, are currently defined on the Merchant Profile. Snap* will identify MOTO transactions as any Merchant that is set up as a MOTO merchant.

3. **Inter-Regional transactions** are defined as transactions where the Issuer or Acquirer are not based in Europe are also considered exempt from SCA. Therefore, European businesses will be able to accept payments from non-European shoppers without problem. Snap* will identify Inter-Regional Out of Scope transactions as any payment where:
   - BankcardTransactionData/ThreeDSData/IsInterRegionalTransaction is true.

4. **Anonymous Prepaid Transactions** are defined as transactions where the card is not tied to a bank account or an individual, but rather to a sum of money, which can originate from cash. For these transactions, the Cardholder is unknown to the Issuer. Snap will identify Anonymous Prepaid out of scope transactions as any payment where:
   - BankcardTransactionData/ThreeDSData/IsAnonymousPrepaidTransaction is true.

If the Issuer rejects the Out of Scope transaction, a Decline response will be sent to the Merchant Application. Out of Scope transactions are only available for 3-D Secure 2.0 transactions but are available for both MasterCard and Visa.

Exemptions

Second, Snap* Platform will support Exemption identifiers for the External Authentication workflow. Exemptions from the Challenge exist for low risk transactions and enable a greater percentage of Frictionless flow transactions. If a transaction qualifies as an Exemption, the Cardholder is available and known, but a request for no Challenge Authentication is made. There are six types of Exemptions that are defined below:

1. Whitelisted Merchants - (Visa, MasterCard, Maestro, JCB, American Express)

The Issuer keeps a database of whitelisted Merchants for each Cardholder. If a Merchant is whitelisted, Authentication will not be required. Snap* will return a WhitelistStatus indicating if the merchant is whitelisted. Snap* will identify Whitelist exempted transactions as any payment where:

- BankcardTransactionData/ThreeDSData/ExemptionInfo/IsWhitelisted is true.

2. Secure Corporate Payments (B2B) Transactions - (Visa, MasterCard, Maestro, JCB, American Express)

For Secure Corporate (B2B) Transactions, Merchant Applications can indicate to the Issuer that the payment is being initiated using a secure process or protocol, such as a physical card used within a secure corporate procurement system or process. Snap will identify Secure Corporate Payment exempted transactions as any payment where:

- BankcardTransactionData/ThreeDSData/ExemptionInfo/IsSecureCorporate is true.

3. Low Value – (Visa, MasterCard, Maestro, JCB, American Express)

Any transaction under 30 Euros is exempt from 3-D Secure Authentication. After the fifth consecutive Low Value exempted transaction, Authentication will again be required. Additionally, if the cumulative transaction amount with Low Value exemption exceeds 100 Euros, Authentication will again be required. The Exemption should be used as the last resort. ○ BankcardTransactionData/ThreeDSData/ExemptionInfo/IsLowValue is true.

4. Low Risk - (Visa, MasterCard, Maestro, JCB, American Express)

The initial release will not include any ability for Snap* to assess risk on behalf of the Merchant. However, the Merchant Application may request the Low Risk exemption based on any risk assessment they have done outside of the Snap* platform. Snap* will identify a Low Risk exempted transactions as any payment where:

- BankcardTransactionData/ThreeDSData/ExemptionInfo/IsLowRisk is true.

5. Recurring/Installment Payments - (Visa, MasterCard, Maestro, JCB)

The majority of Recurring Payments are eligible for Out of Scope processing as MIT transactions. However, Mastercard does allow the Recurring Payment Exemption to be set as a request for Exemption. Snap* will identify a Recurring or Installment exempted transaction as any payment where:

- BankcardTransactionData/ThreeDSData/ExemptionInfo/IsRecurring is true.

6. Delegated SCA - (Visa, MasterCard, Maestro, JCB)

Delegated SCA is where the transaction is authenticated by a third-party Authenticator who is certified to the individual card brands. Issuers and Acquirers are then able to delegate Authentication to these third-party Authenticators.

Delegated Authenticators authenticate the Cardholder with two-factor Authentication. Authenticator categories include:

- Device Authenticators (usually biometrics on mobile or PC device)

- Wallet Authenticators (applications often take advantage of device authenticators)
- Merchant Authenticators (Merchant Applications that meet SCA requirements as part of normal processing)

Since many existing applications have been using these Authenticators since their creation, the Delegated SCA Exemption is meant to eliminate the need for SCA to be performed twice (leading to poor customer experience). For new applications, Delegated Authentication offers Merchant Applications the ability to take full control of the Challenge flow leading to better customer experience.

If the Merchant Application takes advantage of Delegated Authentication, they can identify the Delegated SCA Exemption by setting:

> BankcardTransactionData/ExemptionInfo/IsDelegatedSCA is true

7. **Authentication Outage – (MasterCard, Maestro, JCB, American Express)**

American Express, MasterCard and Maestro supports the new exemption AuthenticationOutage. This exemption can be claimed under the following circumstances:

- The Access Control Server (ACS) is unavailable.
- The Merchant cannot access the SafeKey service due to a 'Merchant problem'.
- The Merchant cannot access the SafeKey service due to a 'major outage'.

```
"ExemptionInfo": {
        "ExemptionControl": "AuthorizationFlow",
        "IsAuthenticationOutage": true
```

Exemptions are offered on American Express, MasterCard, Maestro and Visa for both 2.1 and 2.2 protocols. To correctly identify the Authorization attempt with a 3-D Secure 2.0 Exemption, the Merchant Application must set the following field for the External Authentication workflow:

> BankcardTransactionData/ThreeDSData/ExemptionInfo/ExemptionControlParam to AuthorizationFlow

Submitting an Exempted Transaction

When submitting the initial request for a 3-D Secure 2.0 Exempted transaction, the Merchant Application is still required to populate all other required 3-D Secure 2.0 proof of Authentication fields. This is because the Authorization attempt is required to contain the Exemption fields that were present in the original Authentication. Since Snap* was not used for the Authentication initially, they will need to be present for the Authorization request.

Sending directly on Authorization request has the benefit of eliminating the latency associated with 3-D Secure processing. However, if the Merchant Application does not have a high degree of confidence that the Exemption applies, there may be higher risk of decline as the Authorization does not include the rich 3-D

Secure Authentication data set. Additionally, sending directly on the Authorization request using the Snap* External Authentication workflow may lead to longer transactional latency if the Exemption is not accepted.

Submitting an Exempted Transaction Request

To start an Exempted Transaction request, the Merchant Application will send an Authorize or AuthorizeAndCapture transaction with the Exemption set but without EcommerceSecurityData populated. If the Issuer rejects the Exemption, Snap* will return a decline response to the Merchant Application indicating the ReasonForNotHonoringExemption returned from the Issuer. The Merchant must then Authenticate the transaction outside of Snap* processing, after which the Merchant Application sends a new Authorize or AuthorizeAndCapture transaction with EcommerceSecurrityData set.

Below is an example of a MasterCard Exemption Request transaction for 2.2:

```
{
"$type": "AuthorizeTransaction, http://schemas.evosnap.com/CWS/v2.0/Transactions/Rest",
"Transaction": {
 "$type": "BankcardTransactionPro,
  http://schemas.evosnap.com/CWS/v2.0/Transactions/Bankcard/Pro",
 "TenderData": {
       "CardData": {
   "CardType": "Visa",
   "CardholderName": "Johnny Bravo",
   "PAN": "4500000000000000",
   "Expire": "1225"
       },
            "CardSecurityData": null,
            "EcommerceSecurityData": {
                "AuthenticationMethod": "Frictionless",
   "AuthenticationTimestamp": "2020-05-06T21:12:25.047Z",
   "AuthenticationECI": "02",
   "AuthenticationValue": "MTIzNDU2Nzg5MDA5ODc2NTQzMjE=",
   "ACSTransactionId": "37E03E40-CDD6-4B57-9E29-A552BAF08A1B",
   "DSTransactionId": "0AAAB12F-FB04-4D0C-A06D-3E3D45566D5A",
   "TransactionStatus": "SuccessfullyAuthenticated",
   "ServerTransactionId": "1E57CCB8-BBC2-4E91-B200-164FC6328803"
            }
        },
 "TransactionData": {
       "Amount": 1.00,
       "CurrencyCode": "USD",
       "Is3DSecure": false,
       "ThreeDSData": {
"ProtocolVersion": "v2_2_0",
                "ExemptionInfo": {
                    "ExemptionControl": "AuthorizationFlow",
                    "IsSecureCorporate": "true"
                  }
            }
 }
},
"ApplicationProfileId": "797441",
"MerchantProfileId": "Bonnie TestClient .35" }
```

Note that Out of Scope transactions and Exemptions are supported the same way for both Browser and Application-Based workflows.

Follow-On Transactions

No changes are needed from the Merchant Application in order to process follow-on transactions – the Merchant Application sends in an Authorize request as detailed above, and then submits a follow on Capture or Undo request as normal.

Card on File for Non-Payment Transactions

There are three possible options to manage a card on file without processing a payment:

1. A card can be added to the account
2. A card on file can be updated on the account.
3. Account data can be verified by Merchant Applications before processing a recurring payment.

Note that the Merchant Application is not sending the required and conditional 3-D Secure 2.0 fields because no Challenge is being performed.

Adding Card on File without Processing Payment

Merchants first Authenticate outside of Snap*. After Authentication, Merchant Applications must indicate a card is being added by setting BankcardTransactionData/CardOnFileInfo/CardOnFile to First. Merchant Applications will call Verify with BankcardTransactionData/Is3DSecure as False and BankcardTransactionData/CardOnFileInfo/InitiatedBy as Merchant or Cardholder.

Note that the required EcommerceSecurityData fields must be set. For a full list of those fields, see the Snap* Documentation Portal.

Repeat Card on File Transactions

Updating Existing Card on File without Processing Payment

Merchants first Authenticate outside of Snap*. After Authentication, Merchant Applications indicate a card is being updated by setting BankcardTransactionData/CardOnFileInfo/CardOnFile to Repeat. Merchant Applications will call Verify with BankcardTransactionData/Is3DSecure as False and BankcardTransactionData/CardOnFileInfo/InitiatedBy as Cardholder. Required EcommerceSecurityData fields must be set.

Repeat Card on File transactions require a reference to the First Card on File transaction. Whether the Merchant Application is using Snap* tokenization or Third Party Tokenization, the Repeat Card on File transaction will require the appropriate reference field to be set. See the Tokenization section below for more details.

Merchant Verification

Merchant applications may wish to verify account data prior to processing recurring payments.

Merchant applications will call Verify with the required EcommerceSecurityData fields set, BankcardTransactionData/Is3DSecure as False, BankcardTransactionData/CardOnFileInfo/InitiatedBy as Merchant, and BankcardTransactionData/CardOnFileInfo/CardOnFile as Repeat.

Repeat Card on File transactions require a reference to the First Card on File transaction. Whether the Merchant Application is using Snap* tokenization or Third Party Tokenization, the Repeat Card on File transaction will require the appropriate reference field to be set. See the Tokenization section below for more details.

Card on File for Payment Transactions

There are several possible options to manage a card on file while processing a payment:

1. A card on file can be added as part of a single payment.
2. A card on file can be added as part of the first recurring payment.
3. Merchants can initiate a transaction for a recurring payment.
4. A cardholder can initiate a transaction for a recurring payment.
5. A cardholder can update the card on file while processing a payment.

Adding a Card on File as Part of a Single Payment

Merchants first Authenticate outside of Snap*. After Authentication, Merchant Applications must indicate a card is being added by setting BankcardTransactionData/CardOnFileInfo/CardOnFile to First. Merchant Applications will call Authorize or AuthorizeAndCapture with BankcardTransactionData/Is3DSecure as False, BankcardTransactionData/CardOnFileInfo/InitiatedBy as Cardholder, BankcardTransactionData/ThreeDSData/ProtocolVersion as 2.2, and with an Amount greater than zero.

Note that the required EcommerceSecurityData fields must be set. For a full list of those fields, see the Snap* Documentation Portal.

Repeat Card on File Transactions

Updating Card on File as Part of the First Recurring Payment

This process follows a similar workflow as adding a card on file for a single payment, with one exception. Merchants first Authenticate outside of Snap*. After Authentication, Merchant Applications must indicate a card is being added by setting BankcardTransactionData/CardOnFileInfo/CardOnFile to First. Merchant Applications will call Authorize or AuthorizeAndCapture with BankcardTransactionData/Is3DSecure as False, BankcardTransactionData/CardOnFileInfo/InitiatedBy as Cardholder, BankcardTransactionData/ThreeDSData/ProtocolVersion as 2.1 or 2.2, and with an Amount greater than zero. Merchant Applications must also set BankcardTransactionPro/BankcardInterchangeData/BillPayment as Recurring to indicate this is a recurring transaction. Required EcommerceSecurityData fields must also be set.

Repeat Card on File transactions require a reference to the First Card on File transaction. Whether the Merchant Application is using Snap* tokenization or Third Party Tokenization, the Repeat Card on File transaction will require the appropriate reference field to be set. See the Tokenization section below for more details.

Merchant Initiated Transaction

In this workflow, the Merchant application processes a recurring transaction and wants liability to shift to the Issuer. This workflow is supported only for protocol version 2.2. Merchants first Authenticate outside of Snap*. After Authentication, Merchant Applications must indicate a card is being added by setting BankcardTransactionData/CardOnFileInfo/CardOnFile to Repeat. Merchant Applications will call Authorize or AuthorizeAndCapture with BankcardTransactionData/Is3DSecure as False, BankcardTransactionData/CardOnFileInfo/InitiatedBy as Cardholder, BankcardTransactionData/ThreeDSData/ProtocolVersion as 2.2, and with an Amount greater than zero. Merchant Applications must also set BankcardTransactionPro/BankcardInterchangeData/BillPayment as Recurring to indicate this is a recurring transaction. Required EcommerceSecurityData fields must also be set.

Merchant Applications must also set BankcardTransactionData/CardOnFileInfo/InitiatedBy as Merchant, BankcardTransactionData/ThreeDSData/PaymentTokenIndicator as True if the Merchant is using their own tokenization service, BankcardTransactionData/ThreeDSData/ThreeRIIndicator as Recurring, BankcardTransactionPro/BankcardInterchangeData/RecurringExpirationDate as the recurring payment expiration date, and BankcardTransactionPro/BankcardInterchangeData/RecurringFrequency as the interval at which the recurring payment is processed.

Repeat Card on File transactions require a reference to the First Card on File transaction. Whether the Merchant Application is using Snap* tokenization or Third Party Tokenization, the Repeat Card on File transaction will require the appropriate reference field to be set. See the Tokenization section below for more details.

Cardholder Initiated Transactions

For Cardholder Initiated transactions, the Merchant Application is responsible for setting BankcardTransactionData/ThreeDSData/AccountInfo, which is optional but suggested. This additional information allows the ACS to make risk based decisions with no direct interaction with cardholder for tokenized transaction.

Cardholder Initiated Transactions using Snap* Token

Merchants first Authenticate outside of Snap*. After Authentication, Merchant Applications must indicate a card is being added by setting BankcardTransactionData/CardOnFileInfo/CardOnFile to Repeat. Merchant Applications using a Snap* token will call Authorize or AuthorizeAndCapture with BankcardTransactionData/CardOnFileInfo/InitiatedBy as Cardholder, TenderData/PaymentAccountDataToken as the Snap Token, and an Amount greater than zero. Required EcommerceSecurityData fields must also be set.

As before, Repeat Card on File transactions require a reference to the First Card on File transaction. Whether the Merchant Application is using Snap* tokenization or Third Party Tokenization, the Repeat Card on File transaction will require the appropriate reference field to be set. See the Tokenization section below for more details.

Cardholder Initiated Transactions using Third Party Token

Again, Merchants first Authenticate outside of Snap*. After Authentication, Merchant Applications must indicate a card is being added by setting BankcardTransactionData/CardOnFileInfo/CardOnFile to Repeat. Merchant Applications using a third party token will call Authorize or AuthorizeAndCapture with

BankcardTransactionData/Is3DSecure as False,
BankcardTransactionData/CardOnFileInfo/InitiatedBy as Cardholder,
BankcardTenderData/TokenInformation as the third party token information, and an Amount greater than zero. Required EcommerceSecurityData fields must also be set.

As before, Repeat Card on File transactions require a reference to the First Card on File transaction. Whether the Merchant Application is using Snap* tokenization or Third Party Tokenization, the Repeat Card on File transaction will require the appropriate reference field to be set. See the Tokenization section below for more details.

Cardholder Updates Card on File as Part of Processing Payment

Again, Merchants first Authenticate outside of Snap*. After Authentication, Merchant Applications will call Authorize or AuthorizeAndCapture with BankcardTransactionData/Is3DSecure as False,
BankcardTransactionData/CardOnFileInfo/InitiatedBy as Cardholder,
BankcardTransactionData/ThreeDSData/ProtocolVersion as 2.1 or 2.2,
BankcardTransactionData/CardOnFileInfo/CardOnFile as Repeat, and an Amount greater than zero.
Required EcommerceSecurityData fields must also be set.
As before, Repeat Card on File transactions require a reference to the First Card on File transaction. Whether the Merchant Application is using Snap* tokenization or Third Party Tokenization, the Repeat Card on File transaction will require the appropriate reference field to be set. See the Tokenization section below for more details.

Tokenization

Merchants Using Snap* Tokenization

For Merchant Applications using Snap* tokenization, Repeat Card on File transactions must be tokenized transactions with TenderData/PaymentAccountDataToken set to the PaymentAccountDataToken returned on the First Card on File transaction response.

Merchants Using Third Party Tokenization

The Merchant Application receives the reference ID on their First Card on File transaction response as TransmissionNumber. On a Repeat Card on File transaction, the Merchant Application must submit CardOnFileInfo/OriginalTransactionId as the TransmissionNumber from the First Card on File transaction. Field length for TransmissionNumber is expected to be 20 characters.

Best Practices

It is highly recommended that Merchant Applications not set AuthenticationIndicator to values other than MaintainCard and VerifyCardholder. This is to avoid setting it incorrectly and the transaction being rejected based on validation. Snap Platform will default AuthenticationIndicator for all use cases outside of MaintainCard and VerifyCardholder.

For merchants that have purchased tokenization (i.e. a Snap PaymentAccountDataToken is returned on the response), all transactions with card data will be defaulted to CardOnFile First. If the Merchant Application specifically sets AuthenticationIndicator to Payment, a validation error will occur as CardonFile First transactions must have AuthenticationIndicator set to AddCard. If the Merchant Application does not specifically set the AuthenticationIndicator, Snap Platform will default the correct value and no validation error will occur.

Note that Card on File transactions are supported the same way for both Browser and Application-Based workflows.

Sandbox Trigger Values

**** Must have the Snap* serviceid set to route to the Snap* sandbox

SCA Challenge Soft Decline Triggers

- All of the triggers listed in this section will return BankcardTransactionResponse/StatusCode = “65”, BankcardTransactionResponse/StatusMessage = “SCA Required” and BankcardTransactionResponse/ThreeDSInformation/SCARequired = "true".
- BankcardTransaction/TenderData/CardType can be Visa, Electron, MasterCard, Maestro or AmericanExpress.
- BankcardMerchantData/IndustryType must = “Ecommerce”.
- A transaction that includes ThreeDSData/InternalSCARequired = “true” will bypass all triggers and return an approved response. This field indicates that SCA has already been performed and the workflow is on the second pass into sandbox.

Generic soft decline for non-authenticated ecommerce transaction with no exclusions or exemptions
This must be a normal ecommerce transaction with no 3DS, no exemptions, no exclusions and no CardOnFile data present.

- BankcardTransaction/TransactionData/Amount = “452.00”

Low Value Exemption soft decline

- BankcardTransaction/TransactionData/ThreeDSData/ExemptionInfo/IsLowValue = “true” and BankcardTransaction/TransactionData/Amount = “1012.00” or “1014.00”

Low Risk Exemption soft decline

- BankcardTransaction/TransactionData/ThreeDSData/ExemptionInfo/IsLowRisk = “true” and BankcardTransaction/TransactionData/Amount = “1016.00” or “1018.00”

Recurring Exemption soft decline

- BankcardTransaction/TransactionData/ThreeDSData/ExemptionInfo/IsRecurring = “true” and BankcardTransaction/TransactionData/Amount = “1020.00” or “1022.00”

Delegated SCA Exemption soft decline

- BankcardTransaction/TransactionData/ThreeDSData/ExemptionInfo/IsDelegatedSCA = “true” and BankcardTransaction/TransactionData/Amount = “1024.00” or “1026.00”

Whitelisted Exemption soft decline

- BankcardTransaction/TransactionData/ThreeDSData/ExemptionInfo/IsWhitelisted = “true” and BankcardTransaction/TransactionData/Amount = “1004.00” or “1006.00”

Secure Corporate Exemption soft decline

- BankcardTransaction/TransactionData/ThreeDSData/ExemptionInfo/IsSecureCorporate = “true” and BankcardTransaction/TransactionData/Amount = “1008.00” or “1010.00”

Authentication Outage Exemption soft decline

- BankcardTransaction/TransactionData/ThreeDSData/ExemptionInfo/IsAuthenticationOutage = “true” and BankcardTransaction/TransactionData/Amount = “1011.00” or “1013.00”

SCA Exclusion Soft Decline Triggers

There are 4 total exclusion triggers but the one for MOTO transactions behave differently than the Ecommerce transactions.

- For MOTO, BankcardTransactionResponse/StatusCode = “65” and the BankcardTransactionResponse/StatusMessage will be different for Visa, AmericanExpress and MasterCard. Visa/Electron and AmericanExpress transactions will return “Reserved for future Postilion use” and MasterCard/Maestro transactions will return “Exceeds withdrawal frequency”. BankcardTransactionResponse/ThreeDSInformation/SCARequired will **NOT** be set.

- The 3 Ecommerce triggers will all return BankcardTransactionResponse/StatusCode = “65”, BankcardTransactionResponse/StatusMessage = “SCA Required” and BankcardTransactionResponse/ThreeDSInformation/SCARequired = "true".

- BankcardTransaction/TenderData/CardType can be Visa, Electron, MasterCard, Maestro or AmericanExpress.

- BankcardMerchantData/IndustryType must = “Ecommerce” or “MOTO”.

Merchant Initiated Repeat Card On File soft decline

- BankcardMerchantData/IndustryType = “Ecommerce” and BankcardTransaction/TransactionData/CardOnFileInfo/InitiatedBy = “Merchant” and BankcardTransaction/TransactionData/CardOnFileInfo/CardOnFile = “Repeat” and BankcardTransaction/TransactionData/Amount = “700.00”.

IsInterRegionalTransaction soft decline

- BankcardMerchantData/IndustryType = “Ecommerce” and BankcardTransaction/TransactionData/ThreeDSData/IsInterRegionalTransaction = “true” and BankcardTransaction/TransactionData/Amount = “720.00”.

IsAnonymousPrepaidTransaction soft decline

- BankcardMerchantData/IndustryType = “Ecommerce” and BankcardTransaction/TransactionData/ThreeDSData/ IsAnonymousPrepaidTransaction = “true” and BankcardTransaction/TransactionData/Amount = “730.00”.

MOTO hard decline

- BankcardMerchantData/IndustryType = “MOTO” and BankcardTransaction/TransactionData/Amount = “710.00”.

3DS V1 Decommisioning

Beginning in October of 2022, the 3 major card brands will no longer support 3DS Version 1. While the dates for each card brand are not the same, EVO will no longer be supporting 3DS V1 starting October 14th, 2022.

De-commissioning dates by card brand:

- American Express: 10/14/2022
- Discover: 10/14/2022
- Visa: 10/15/2022
- MasterCard: 10/18/2022
- Maestro: 10/18/2022
- JCB: 10/18/2022