# **Card on File Mandate FAQs**

Card on File (COF) mandates uses specific terminologies, and they can be confusing. To help with this, we have created a field matrix. This matrix includes the API field names applicable to Authorize.Net as well as short descriptions of the fields intended use. We will refer to these naming conventions throughout the FAQs. This matrix also exists in our [User Guide](https://support.authorize.net/s/article/Authorize-Net-Mandate-Compliance-Overview) with expanded details, if needed. For an overview and understanding of what your business will need to do to become compliant with COF mandates, we would suggest that you first review the User Guide and [Developer Center](https://developer.authorize.net/) prior to using this FAQ.

**Card on File (COF) Field Matrix**

| **COF Fields** | **Field Values** | **Description** | **Notes** |
| --- | --- | --- | --- |
| processingOptions |  |  | **isFirst**\* indicates the initial authorization  |
|  | isFirstRecurringPayment | isFirstRecurring would be used to indicate a recurring subscription payment | **Establishment of Relationship Auth**- should **NOT** include recurringBilling field set to true |
|  | isFirstSubsequentAuth | isSubsequentAuth would be used to indicate non-recurring initial payment | **Establishment of Relationship Auth** |
|  | isSubsequentAuth | isSubsequentAuth would be used for **MIT** charges | **Submitted with a reason unless UCOF** |
|  | isStoredCredentials | isStoredCredentials would be used for **CIT** (Customer Initiated Transactions)  | originalNetworkTransId/originalAuthAmount are **NOT** Required with CIT transactions |
| subsequentAuthInformation |  |  |  |
|  | originalNetworkTransId | The NetworkTransId returned with the Establishment of Relationship authorization | Currently returned with authorizations on our FDCN and Vantiv connections |
|  | reason | * resubmission
* delayedCharge
* reauthorization
* noShow
 | CIT transactions will not use a reason, nor will MIT UCOF or non-first recurring (recurringBilling) transactions |
|  | originalAuthAmount | Amount authorized with the Establishment of Relationship transaction | Only required for Discover cards |
| recurringBilling | true | Uses the existing [transactionSettings recurringBilling](https://developer.authorize.net/api/reference/features/payment_transactions.html#Transaction_Settings) field to identify non-isFirstRecurringPayment COF transactions.MUST be submitted with the subsequentAuthInformation fields:* originalNetworkTransId
* originalAuthAmount
 | Used for all recurring transactions other than the first recurring payment |

**Q: I’ve been storing my customer card data prior to COF mandate compliance availability. Do I need to update my existing stored payment credentials for compliance with the COF mandate?**

**A:** Yes. If you have historically been storing your customer payment data, you likely have at least some stored credentials that do not have an originalNetworkTransId or originalAuthAmount associated.

Once your business has completed the steps to meet COF compliance, any stored credentials which do not have an associated Network Transaction ID ([NetworkTransId](#NetworkTransId)) should process the next authorization as an [isFirst\*](#isFirst) transaction. It is important that the [isFirst\*](#isFirst) transaction is both authorized and settled (i.e. funded to the merchant). This should only be done after you have obtained consent/agreement from your customer for the storage of their payment credentials, if you have not previously done so.

**Q: What will happen if I don’t include any COF fields in transaction requests via API? Will I be compliant**?

A: No, you will not be compliant. Regardless of how you store your customer payment credentials, you will need to send in the [required COF fields for the type of transaction you are processing](#COFFieldMatrix).

**Q: I use Customer Information Manager (CIM) but I already have NetworkTransIds for my customers. Can I use them instead of submitting isFirst\* COF transactions?**

**A:** Yes. Even if you use Customer Information Manager (CIM) to store your customer payment data, merchants are able to include any value for the [originalNetworkTransId](#originalNetworkTransId) and/or [originalAuthAmount](#originalAuthAmount) fields in a [createTransactionRequest via API](https://developer.authorize.net/api.html). Authorize.Net will, in turn, send these values to your processor, issuer and acquirer with authorization and settlement.

Please note that Authorize.Net does not perform validations of the COF field values submitted via API. When including these values, it is the merchant responsibility to ensure that the correct values are included.

**Q: What happens if the** [**isFirst\***](#isFirst) **COF transaction is authorized by the customer’s bank but fails to settle successfully (i.e. merchant is not funded) or I void it before settlement?**

A: The card networks (i.e. Visa, MasterCard, Discover) stipulate that the [NetworkTransId](#NetworkTransId) returned with authorization may only be used in subsequent COF transactions as the [originalNetworkTransId](#originalNetworkTransId) value if the transaction it was assigned to is successfully authorized and settled. In the case that the merchant voids the isFirst\* transaction prior to settlement, or if the transaction fails to settle and the merchant is not funded for the authorization, a new [isFirst\*](#isFirst) transaction will be necessary to obtain a valid NetworkTransId value to be used in subsequent COF transactions.

**Q: Can I include multiple COF transaction types with a createTransactionRequest?**

No, you should not include conflicting COF fields and values. While our API will allow you to do so, since we do not perform validation of the fields and values submitted into our system, if you include conflicting field values we will attempt to identify the transaction as a COF transaction but we cannot guarantee that it will be identified properly and, at times, it may be processed as a non-COF transaction as a result. Please see our [User Guide](https://support.authorize.net/s/article/Authorize-Net-Mandate-Compliance-Overview) and [Developer Center](https://developer.authorize.net/api.html) for more information on the appropriate field requirements.

**Q: What will I need to do to remain in compliance if I have an in progress, recurring payment agreement with my customer and they update their payment card number on file, meaning payments have already authorized and settled prior to the card number update?**

A: COF compliance for recurring transaction types mandates that the [isFirst\*](#isFirst) transaction be tied to the card number directly, as the card network takes the position that the new card requires a new agreement between the merchant and cardholder, thereby nullifying the previous agreement. What this means to merchants is that any payment processed after customer(s) have updated their payment card numbers for recurring agreements ***must*** be submitted as an [isFirstRecurringPayment](#isFirstRecurringPayment) transaction, regardless of whether the customer has been previously and successfully charged with another card number. The new isFirst\* transaction NetworkTransId returned in the authorization response becomes the new originalNetworkTransId value for all future [recurringBilling](#recurringBilling) payments for your customer unless, and until, they cancel or update the agreement with the merchant.

**Q: What will happen if I store the payment data within my Authorize.Net account and my customer’s address changes?**

A: When customers have address updates there is no impact to COF functionality. The [isFirst\*](#isFirst) transaction processed for that customer will remain to be considered the “first” COF transaction for all future transactions.

**Q: Are transactions processed from the Virtual Terminal compliant with the COF Mandate?**

A: No. The Authorize.Net Virtual Terminal will not support the use of COF fields. Transactions which are processed using stored credentials should be processed via the [Customer Information Manager](https://support.authorize.net/s/article/How-to-Charge-a-Payment-Method-Stored-in-Customer-Information-Manager-CIM) or via [API](https://developer.authorize.net/) with the required COF fields.