worldpay

Intercept eCommerce with XiSecure Integration Guide

Intercept eCommerce with XiSecure Integration Guide

Contents

Contents

1 Version history

Version	Description of change
20250206-1	Added PCI v4.0 A compliance 113
20240801-1	- Updated Integration and Overview Planning 14 - Updated Display Content Based on the Selected Payment Type 61 - Changed *.paymetric.com to *.worldpay.com
20200422-1	Updated resource domain URLs throughout.
20200330-1	Removed empty section heading. This entire guide supports XiSecure token integration only. See *Intercept eComm with TMS Integration Guide* for the option to integrate one or many token types.
20200326-1	Corrected JS Plugin samples for Form Post and Ajax. Changed Intro and title to reflect this document supports integration to XiSecure only.
20190701-1	Removed xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" from all XML samples.
20190621-1	- Added 3DS 2 content and samples under the Get Access Token sections for Form Post 19, Ajax 30, Iframe 43, Hosted 66 and 3DS with Token 80. - Updated the XIPlugin URL path to the new version that supports 3DS2 for Form Post 24 and Ajax 35. - All samples were reviewed against the XSD and modified as needed.
20190403-1	In Child Element: <cardTypeIndicator> 179 section, added reference to the new unselected class that gives further control of styling the unselected card type

1 Version history

Version	Description of change
	logos.
20190215-1	Added mask-number attribute details in Custom HTML Reference section, Textbox Elements: <tbox{XXX}> 181 topic.
20181128-1	In Common Attributes: 153 section, changed is-default value from "true" to "1" to make that value the default.
20181113-1	Corrected typo in intro sentence for iFrame sequence diagram 41.
20181023-1	Correct AliPay sequence diagram 90. Correct Iframe JavaScript URL in Iframe: On-Submit Javascript Plugin 57 topic.
20181004-1	Corrected Ajax Access Token with 3DS XML 30 example. Added variable placeholders to each Access Token XML example.
20181001-1	Added 3DS with Token 79 section. Updated Ajax example in Ajax: On-Submit Javascription Plugin 35 section.
20180906-1	Formatting and wording changes.
20180824-1	Added AliPay 90 and Visa Checkout 100 sections.
20180823-1	New structure and format.
Jun-2018-3110-R3	In $XIFrame.onload 142 section, corrected description for onCardTypeChange and wording in the first paragraph.
May-2018-3110-R2	Added 150 to Error Codes 130.

1 Version history

Version	Description of change
Jan-2018-3100-R2	Corrected verbiage in Integration Type Overviews to clarify Post call is made from Client's browser. Edited Form Post workflow to include Post call in Client browser swimlane.

2 Introduction

Intercept eCommerce utilizes XML packets to submit sensitive data directly to a (SOC 1 compliant) data center where it is encrypted and securely stored by XiSecure. A token is returned for use by the merchant's application; thereby, removing sensitive data from the merchant's environment.

Authorization and settlement functionality is implemented directly through the XiPay web service.

Purpose

This document explains how to integrate to Intercept eComm for XiSecure tokens only.

If you want to integrate to the Token Management Service, which provides the flexibility of using all Worldpay token types, see the *Intercept eCommerce with Token Management Service Integration Guide*.

Audience

This document is intended for use by Web Developers and assumes a familiarity with XML and XSD.

3 Compatibility

This section includes information on browser compatibility and mobile device support.

Browser compatibility

Intercept eComm is compatible with the following browsers:

- Internet Explorer 10.0 and above
- Firefox (latest version)
- Chrome (latest version)
- Safari (latest version)

Mobile device support

Intercept eComm is compatible with both IOS and Android devices. Use the same integration steps as documented in this guide for these devices.

4 Onboarding requirements

Onboarding for the following standard raw card integration types only requires submitting the environment request via the Onboarding page:

- Form Post
- Ajax
- Iframe
- Hosted

If you are implementing 3D Secure or AliPay with Intercept eComm, you must obtain the following from Cardinal Commerce before you can submit your Intercept eComm environment requests.

- Cardinal Commerce Merchant ID
- Processor ID
- Password

If you are implementing Visa Checkout, you must obtain the following from Visa before you can submit your Intercept eComm environment requests.

- Visa API Key
- Visa Shared Secret

5 Authentication

The identity of the system making calls to the Intercept eComm platform is authenticated using the following security elements and URL endpoints.

Security element	Description
Merchant GUI	A unique ID assigned that is included in the Request packet. Each merchant has a single GUID.
Shared Key	A string generated and provided that is used to "sign" the Request Packet. Paymetric uses the same shared key to confirm the signature.
URL Endpoint	For the latest URL endpoints used to initiate the first application server call and obtain an AccessToken, see the Web Service and Resource URLs topic in Paymetric Assist.
Access Token	A string that is dynamically generated by Intercept eComm during the first application server call. The call includes the Merchant GUID and the Intercept eComm Request Packet that is signed by the Shared Key. The returned AccessToken is then used in all subsequent calls within the given session. The AccessToken is valid for 30 minutes.

Standard raw value tokenization

Form Post, Ajax, Iframe and Hosted are the Intercept eComm "standard raw card integration types" used to tokenize directly-entered information (i.e. not from some sort of wallet functionality), such as credit cards, account numbers for echeck payments, and other non-payment related sensitive data like Social Security Numbers.

These standard integration types all use the same Intercept eComm URL endpoints to trigger each of the calls.

Form Post	• POST request is made to Intercept eComm from client's browser. • Intercept eComm redirects to the URL defined by the merchant; additional fields aside from what is being tokenized can be sent in the XML packet. • The page has a high-level of page responsiveness since it is completely hosted by the merchant. • Merchant is responsible for all card validations.
Ajax	• Ajax call to Intercept eComm from the Client's browser. • User never leaves page. • The page has a high-level of page responsiveness since it is completely hosted by Merchant. • Merchant is responsible for all card validations.

Iframe	• Paymetric hosts the content of the Iframe. • Merchant can style and control what fields display; there are multiple content templates or Merchant can completely customize. • The page has a potential delay while loading the Iframe; Merchant could hide all contents of page while Iframe loads.
Hosted	• Paymetric hosts the entire page. • When initiating payment, Merchant's customer is redirected to the Paymetric URL. • Intercept eComm redirects to the URL defined by the Merchant after processing the request. • The page has a high-level of page responsiveness since it is completely hosted by Paymetric.

⚠ **Note:**

When implementing the Form Post or Ajax integration type, do NOT post or send raw card numbers in your environment.

3DS with token integration

Intercept eComm supports a separate 3DS workflow using a XiSecure Token. There is a separate URL endpoint to trigger this call.

- An Ajax call is made to Intercept eComm with the Token from the customer's browser.
- The user never leaves page.

The Ajax call in this workflow uses a different JS plugin than the Ajax call mentioned under the Standard Raw Value Tokenization section.

Alternative payment method tokenization

Intercept eComm also provides tokenization functionality for AliPay 90 and Visa Checkout 100 alternative payment methods, each with their own workflow and URL endpoint to trigger the calls.

AliPay	• Implemented as an alternative payment type with either the Form Post, Ajax or Iframe integration type (not supported with Hosted). • A POST request is made to Intercept eComm from Client's browser. • Intercept eComm redirects to Cardinal Commerce to verify enrollment then to AliPay if enrolled. • Intercept eComm redirects to URL defined by the Merchant after processing the request.

Visa Checkout	• Implemented as an alternative payment type with either the Form Post, Ajax or Iframe integration type (not supported with Hosted). • Merchant makes a call directly to Visa Checkout platform to obtain call ID. • Merchant performs a POST request to Intercept eComm from the application server. • Intercept eComm obtains full payment details from Visa Checkout and sends decrypted response packet with token in either XML or JSON as specified by the merchant.

XiSecurelink integration

- Used to trigger an email with a secure link to Intercept eComm hosted page to obtain credit card information. Can alternatively choose to copy/paste the link into a chat window.
- Paymetric hosts the entire page.
- Hosted page is launched via a link sent in an email to the specified address or from a chat window.
- Intercept eComm redirects to the URL defined by the Merchant after processing the request.

7 Form Post integration

⚠ **Note:**

When implementing Form Post, do NOT post or send raw card numbers in your environment.

Form Post integration consists of the following high-level steps:

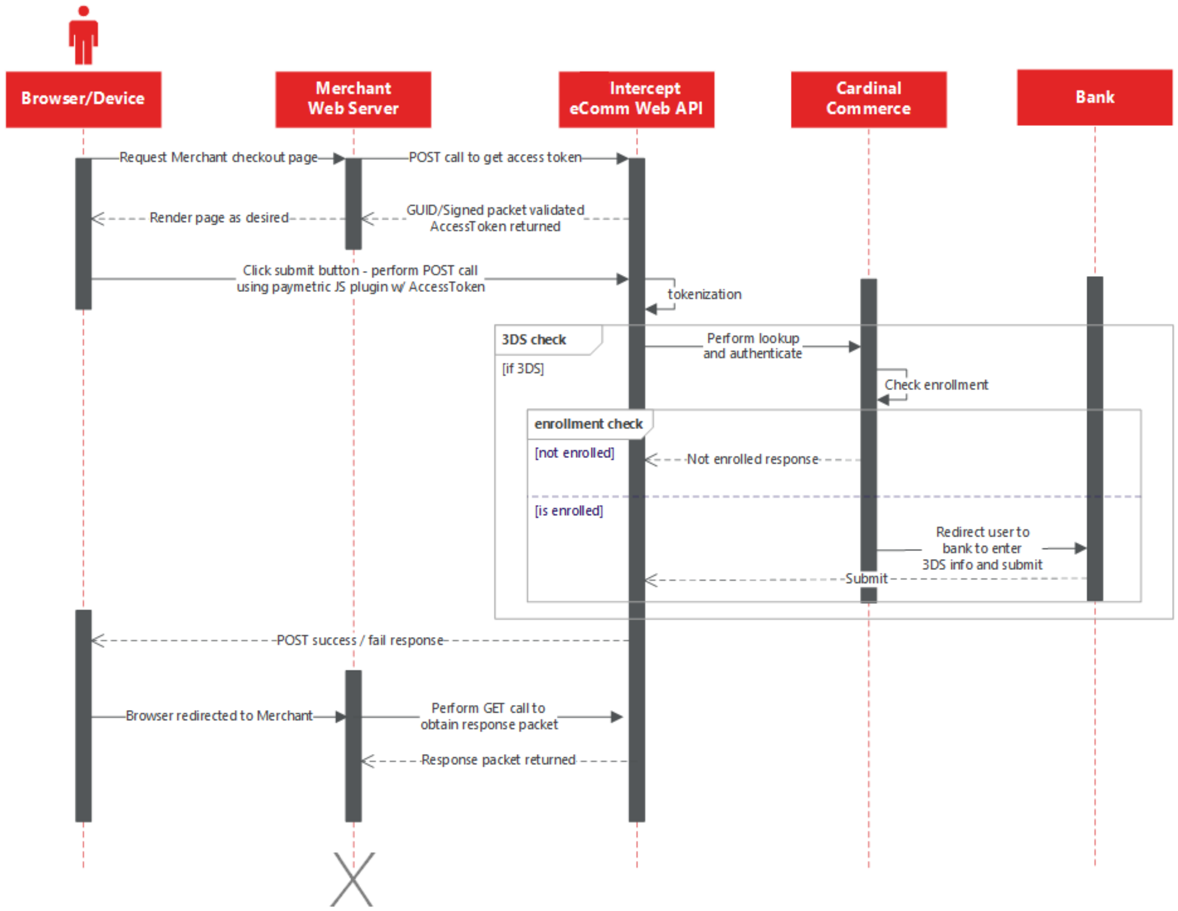
1. Initiate Intercept eComm session with a POST call to get the access token.
    - If implementing 3DS, there are extra fields that may be provided in the PacketXML. If 3DS2 there is also an attribute to indicate the 3DS version.
2. Render page as desired and upon user submission of credit card data, perform POST call using the JS plugin.
    - If implementing 3DS 2.0, Intercept eComm automatically performs the additional device data collection (DDC) used by Cardinal and the Banks.
3. Perform GET call to obtain the response packet containing the token.

The following diagram illustrates a more detailed integration flow for the the Form Post integration method. Review the subsections below for more details.

⚠ **Note:**

If implementing 3DS2, the user is only redirected to the bank to enter 3DS credentials if the DDC or merchant-provided risk information is determined to be insufficient.

7 Form Post integration


Browser/Device
Merchant Web Server
Intercept eComm Web API
Cardinal Commerce
Bank
Request Merchant checkout page
POST call to get access token
Render page as desired
GUID/Signed packet validated AccessToken returned
Click submit button - perform POST call using paymetric JS plugin w/ AccessToken
tokenization
3DS check
[if 3DS]
Perform lookup and authenticate
Check enrollment
enrollment check
[not enrolled]
Not enrolled response
[is enrolled]
Redirect user to bank to enter 3DS info and submit
Submit
POST success / fail response
Browser redirected to Merchant
Perform GET call to obtain response packet
Response packet returned

7.1 Form Post: Get AccessToken

AccessToken is required to initiate an Intercept eComm session.

AccessToken is a string that is dynamically generated by Intercept eComm during the first application server call. The call includes the Merchant GUID and the Intercept eComm Request Packet that is signed by the Shared Key.

AccessToken is valid for 30 minutes. If the AccessToken expires before the process is complete, you must restart the process and obtain a new AccessToken.

Request method

A direct server-side POST is required to retrieve the Access Token. How this POST is performed depends on the language you are using.

Content type

The content type must be "application/x-www-form-urlencoded".

Endpoints

For the latest endpoints and URLs, see the Web Service and Resource URLs topic in Paymetric Assist.

POST Call parameters

```
MerchantGuid={Your-GUID}&SessionRequestType={Int-Type-Code}
&Signature={Signature}&MerchantDevelopmentEnvironment={code language e.g.
asp.net}&Packet={PACKET_XML}
```

Parameter	Description
MerchantGuid	The Global Unique Identifier assigned.
SessionRequestType	Indicates the integration type. Use 3 for Form Post.
Signature	The Signature must be URL encoded. 1. Sign the packet XML.

Parameter	Description
	2. Encode it with HMAC-SHA-256 using the shared key. 3. Convert it to a base 64 string.
MerchantDevelopment Environment	Indicate the language in which you are coding your integration to Intercept eComm, for example, asp.net or php.
Packet XML	The Packet XML must be URL encoded. Following are the elements: • RedirectUri = the redirect URL for the Response Packet from Intercept eComm If implementing 3DS, include the following fields as well within <Cardinal3DSRequest> container: • Amount = amount of transaction • CurrencyCode = currency code for transaction • CardNumberFieldName = the Merchant's field name that will contain the card number value entered by the User • ExpirationMonthFieldName = the Merchant's field name that will contain the expiration month value entered by the User

Parameter	Description
	• ExpirationYearFieldName = the Merchant's field name that will contain the expiration year value entered by the User • OrderNumber = the Merchant's order number • AdditionalFields = optional container with additional fields for 3DS2 See samples below.

7.1.1 Form Post XML sample

Form Post XML sample

```
<?xml version="1.0"?>
<PostPacketModel>
   <RedirectUri>{http://Yourdomain.here}</RedirectUri>
</PostPacketModel>
```

7.1.2 Form Post with 3DS 1 XML sample

Form Post with 3DS 1 XML sample

```
<?xml version="1.0"?>
<PostPacketModel>
   <RedirectUri>{http://Yourdomain.here}</RedirectUri>
   <Cardinal3DSRequest>
       <Amount>{AmountValue}</Amount>
       <CurrencyCode>{CurrencyCodeValue}</CurrencyCode>
       <CardNumberFieldName>{CCNumValue}</CardNumberFieldName>
       <ExpirationMonthFieldName>{ExpMonthValue}
</ExpirationMonthFieldName>
       <ExpirationYearFieldName>{ExpYearValue}
</ExpirationYearFieldName>
       <OrderNumber>{OrderNumValue}</OrderNumber>
   </Cardinal3DSRequest>
</PostPacketModel>
```

7.1.3 Form Post with 3DS 2 XML sample

The 3DS fields that differ form the prior example and are specific to version 2 are highlighted blue in the example below. These include `threeDSVersion` attribute within the `<Cardinal3DSecure>` container and the `<AdditionalFields>` container and child elements.

The `threeDSVersion` attribute must be set to "2" or can be a specific version up to 3 digits. If there is a dot (.), then there must be a following number. E.g. "2" or "2.0" or "2.2", and so on.

Any Cardinal 3DS field can be included in `<AdditionalFields>` container. See the following URL for a complete list of the additional Cardinal fields:

https://cardinaldocs.atlassian.net/wiki/spaces/CCen/pages/905478187/Lookup+Request+Response

⚠ **Note:**

Cardinal <AdditionalFields> container cannot have duplicate field elements.

Form Post with 3DS 2 XML sample

```
<?xml version="1.0"?>
<PostPacketModel>
   <RedirectUri>{http://Yourdomain.here}</RedirectUri>
   <Cardinal3DSRequest threeDSVersion="2.2.0">
       <Amount>{AmountValue}</Amount>
       <CurrencyCode>{CurrencyCodeValue}</CurrencyCode>
       <CardNumberFieldName>{CCNumValue}</CardNumberFieldName>
       <ExpirationMonthFieldName>{ExpMonthValue}
</ExpirationMonthFieldName>
       <ExpirationYearFieldName>{ExpYearValue}
</ExpirationYearFieldName>
       <OrderNumber>{OrderNumValue}</OrderNumber>
       <AdditionalFields>
           <Field name="UserAccountAge" value="04" />
           <Field name="UserPasswwordChange" value="20190325" />
           <Field name="DeliveryTimeframe" value="02" />
           <Field name="GiftCardAmount" value="100" />
       </AdditionalFields>
   </Cardinal3DSRequest>
</PostPacketModel>
```

7.2 Form Post: On-Submit Javascript plugin

When the Customer performs the submit action by clicking the merchant-created submit button, the data will be posted to Paymetric using JavaScript functions.

Following is an example of pulling the values from the text boxes and populating the data variable.

Populating data variable example

```
var cc = document.getElementById('PaymentCreditCard').value;
var name = document.getElementById('Name').value;
var myData = $XIPlugin.createJSRequestPacket(merchantGuid, accessToken);
myData.addField($XIPlugin.createField('PaymentCreditCard', true, cc));
myData.addField($XIPlugin.createField('name', false, name));
myData.addAdditional3DSField("UserAccountAge", "04");
```

Either reference or download the following JavaScript plugin:

`https://xiecomm.paymetric.com/DIeComm/Scripts/XIPlugin/XIPlugin-1.2.0.js`

⚠ **Note:**

Existing integrations can still use XiPlugin01.1.0.js UNLESS you are adding 3DS2 support, then you must use XiPlugin-1.2.0.js version.

Include the Paymetric JavaScript. Reference the Submit Example below to handle the posting of the APM details directly to Intercept eComm.

Form Post: QA JS Submit example

```
$XIPlugin.submit({

    url: "https://cert-xiecomm.paymetric.com/DIeComm/Form",
    data: {
        Formatted JS Data– See above example myData
    },
    validation: function(a) {
        //Run custom validation
        return true; //should return true or false.

        // return true will go through with post.
        // return false will not go through with post.
    }


});
```

Form Post: QA JS Submit 3DS2 example

```
$XIPlugin.submit({

    url: "https://cert-xiecomm.paymetric.comom/DIeComm/Form",
    data: {
        Formatted JS Data– See above example myData
    },
    validation: function(a) {
        //Run custom validation
        return true; //should return true or false.

        // return true will go through with post.
        // return false will not go through with post.
    },
    threeDSVersion: "2.2.0"


});
```

7.3 Form Post: Get Response packet

Request method

A direct server-side GET is required to retrieve the Response Packet. How this GET is performed depends on the language you are using.

Content type

The content type must be "application/x-www-form-urlencoded''.

Endpoints

For the latest endpoints and URLs, see the Web Service and Resource URLs topic in Paymetric Assist.

Cert

```
https://cert-xiecomm.paymetric.com/DIeComm/ResponsePacket
```

Production

```
https://xiecomm.paymetric.com/DIeComm/ResponsePacket
```

GET Call parameters

```
MerchantGUID={MerchGUID}&Signature={Signature}
&AccessToken={AccessToken}
```

Parameter	Description
MerchantGuid	The Global Unique Identifier assigned.
Signature	Sign the Access Token; encode it with HMAC-SHA-256 using the shared key then covert it to a base 64 string.

Parameter	Description
	The Signature must be URL encoded.
AccessToken	A string that is dynamically generated by Intercept eComm during the first application server call. The call includes the Merchant GUID and the Intercept eComm Request Packet that is signed by the Shared Key.

8 Ajax integration

⚠ **Note:**

When implementing Ajax, do NOT post or send raw card numbers in your environment.

Ajax integration consists of the following high-level steps:

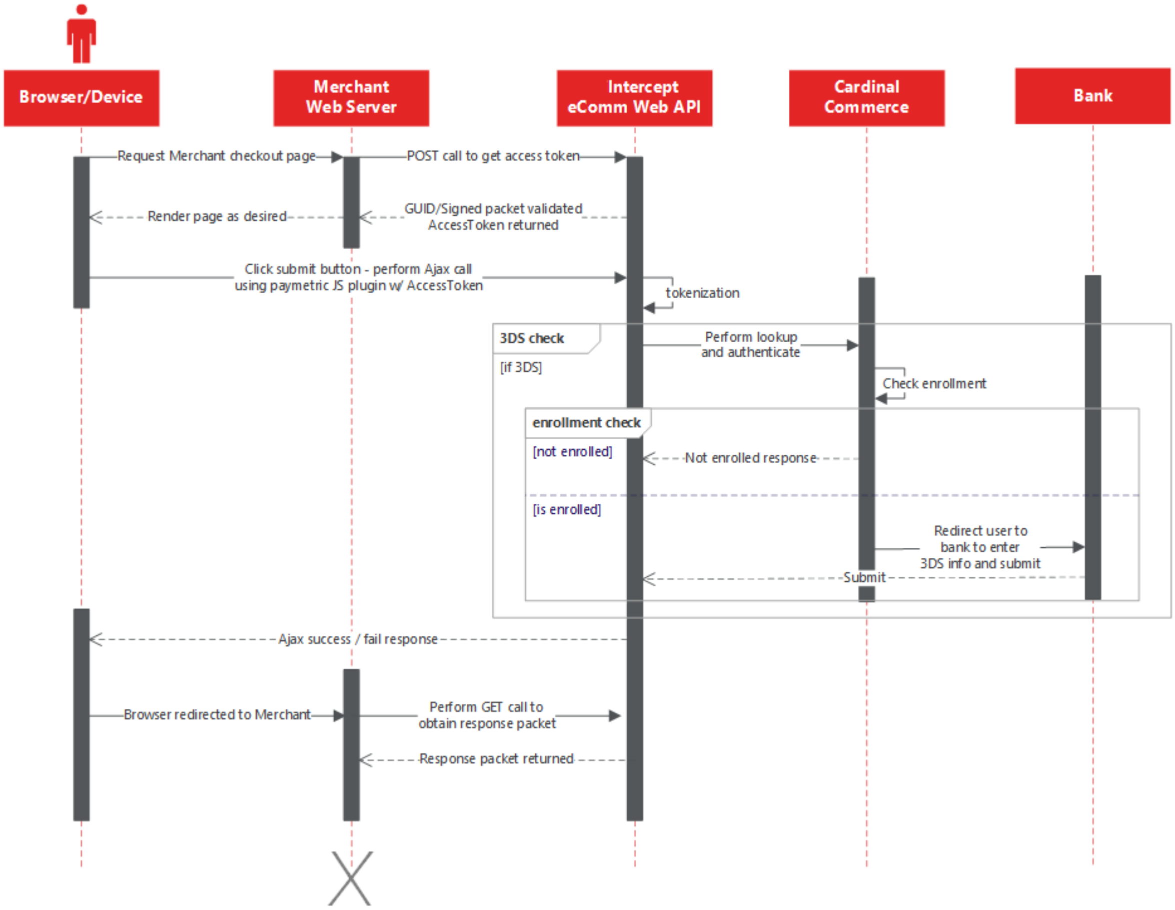
1. Initiate Intercept eComm session with a POST call to get the access token.
   - If implementing 3DS, there are extra fields that may be provided in the PacketXML. If 3DS2 there is also an attribute to indicate the 3DS version.
2. Render page as desired and upon user submission of Credit Card data, perform POST call using the Paymetric JS plugin.
   - If implementing 3DS 2.0, Intercept eComm automatically performs the additional device data collection (DDC) used by Cardinal and the Banks.
3. Perform GET call to get the response packet containing the token. If implementing 3DS, all response fields from Cardinal Commerce will be included as well.

The following sequence diagram illustrates the Ajax integration flow. Review the subsections below for more details.

⚠ **Note:**

If implementing 3DS2, the user is only redirected to the bank to enter 3DS credentials if the DDC or merchant-provided risk information is determined to be insufficient.

8 Ajax integration


Browser/Device
Merchant Web Server
Intercept eComm Web API
Cardinal Commerce
Bank
Request Merchant checkout page
POST call to get access token
GUID/Signed packet validated AccessToken returned
Render page as desired
Click submit button - perform Ajax call using paymetric JS plugin w/ AccessToken
tokenization
3DS check
[if 3DS]
Perform lookup and authenticate
Check enrollment
enrollment check
[not enrolled]
Not enrolled response
[is enrolled]
Redirect user to bank to enter 3DS info and submit
Submit
Ajax success / fail response
Browser redirected to Merchant
Perform GET call to obtain response packet
Response packet returned

8.1 Ajax: Get AccessToken

AccessToken is required to initiate an Intercept eComm session.

AccessToken is a string that is dynamically generated by Intercept eComm during the first application server call. The call includes the Merchant GUID and the Intercept eComm Request Packet that is signed by the Shared Key.

AccessToken is valid for 30 minutes. If the AccessToken expires before the process is complete, you must restart the process and obtain a new AccessToken.

Request method

A direct server-side POST is required to retrieve the Access Token. How this POST is performed depends on the language you are using.

Content type

The content type must be "application/x-www-form-urlencoded".

Endpoints

For the latest endpoints and URLs, see the Web Service and Resource URLs topic in Paymetric Assist.

Cert

```
https://cert-xiecomm.paymetric.com/DIeComm/AccessToken
```

Production

```
https://xiecomm.paymetric.com/DIeComm/AccessToken
```

POST Call parameters

```
MerchantGuid={Your-GUID}&SessionRequestType={Int-Type-Code}
&Signature={Signature}&MerchantDevelopmentEnvironment={code language e.g.
asp.net}&Packet={PACKET_XML}
```

Parameter	Description
MerchantGuid	The Global Unique Identifier assigned.

Parameter	Description
SessionRequestType	Indicates the integration type. Use 2 for Ajax.
Signature	The Signature must be URL encoded. 1. Sign the packet XML. 2. Encode it with HMAC-SHA-256 using the shared key. 3. Convert it to a base 64 string.
MerchantDevelopmentEnvironment	Indicate the language in which you are coding your integration to Intercept eComm, for example, asp.net or php.
Packet XML	For non-3DS implementations, the Packet XML parameter for the Ajax call contains no elements. **Note**: If you have a previous integration that sends HostUri, this field is not used. It will have no effect. If implementing 3DS, include the following fields as well within <Cardinal3DSRequest> container: • RedirectUri = the URI used to redirect back to the merchant after 3DS processing

Parameter	Description
	• Amount = amount of transaction • CurrencyCode = currency code for transaction • CardNumberFieldName = the Merchant's field name that will contain the card number value entered by the User • ExpirationMonthFieldName = the Merchant's field name that will contain the expiration month value entered by the User • ExpirationYearFieldName = the Merchant's field name that will contain the expiration year value entered by the User • OrderNumber = the Merchant's order number • AdditionalFields = optional container with additional fields for 3DS2

8.1.1 Ajax XML sample

Ajax XML sample

```
<?xml version="1.0"?>
<AjaxPacketModel/>
```

8.1.2 Ajax with 3DS 1 XML sample

Ajax with 3DS 1 with existing XML sample

```
<?xml version="1.0"?>
<AjaxPacketModel
    xmlns:xsd=http://www.w3.org/2001/XMLSchema
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <Cardinal3DSRequest>
       <RedirectUri>{http://YourRedirectURLHere}</RedirectUri>
       <Amount>{AmountValue}</Amount>
       <CardNumberFieldName>{CCNumValue}</CardNumberFieldName>
       <CurrencyCode>{CurrencyCodeValue}</CurrencyCode>
       <ExpirationMonth>{ExpMonthValue}</ExpirationMonth>
       <ExpirationYear>{ExpYearValue}</ExpirationYear>
       <OrderNumber>{OrderNumValue}</OrderNumber>
   </Cardinal3DSRequest>
</AjaxPacketModel>
```

8.1.3 Ajax with 3DS 2 XML sample

The 3DS fields that differ form the prior example and are specific to version 2 are highlighted blue in the example below. These include `threeDSVersion` attribute within the `<cardinal3DSecure>` container and the `<AdditionalFields>` container and child elements.

The `threeDSVersion` attribute must be set to "2" or can be a specific version up to 3 digits. If there is a dot (.), then there must be a following number. E.g. "2" or "2.0" or "2.2", and so on.

Any Cardinal 3DS field can be included in `<AdditionalFields>` container. See the following URL for a complete list of the additional Cardinal fields:

https://cardinaldocs.atlassian.net/wiki/spaces/CCen/pages/905478187/Lookup+Request+Response

⚠ **Note:**

Cardinal <AdditionalFields> container cannot have duplicate field elements.

Ajax with 3DS 2 with existing XML sample

```
<?xml version="1.0"?>
<AjaxPacketModel
    xmlns:xsd=http://www.w3.org/2001/XMLSchema
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <Cardinal3DSRequest threeDSVersion="2.2.0">
       <RedirectUri>{http://YourRedirectURLHere}</RedirectUri>
       <Amount>{AmountValue}</Amount>
       <CardNumberFieldName>{CCNumValue}</CardNumberFieldName>
       <CurrencyCode>{CurrencyCodeValue}</CurrencyCode>
       <ExpirationMonth>{ExpMonthValue}</ExpirationMonth>
       <ExpirationYear>{ExpYearValue}</ExpirationYear>
       <OrderNumber>{OrderNumValue}</OrderNumber>
       <AdditionalFields>
           <Field name="UserAccountAge" value="04" />
           <Field name="UserPasswwordChange" value="20190325" />
           <Field name="DeliveryTimeframe" value="02" />
           <Field name="GiftCardAmount" value="100" />
       </AdditionalFields>
   </Cardinal3DSRequest>
</AjaxPacketModel>
```

8.2 Ajax: On-Submit Javascript plugin

When the customer performs the submit action by clicking the merchant-created submit button, the data will be posted to Paymetric using JavaScript functions.

Following is an example of pulling the values from the text boxes and populating the data variable.

⚠ **Note:**

For 3DS2, you can add the Cardinal additional fields via the JavaScript plugin as well. The following example includes one of the 3DS2 AdditionalFields.

If you send the same additional field that you sent when getting the access token, it will be overwritten.

If you accidentally add more than one field with the same name and different values via the plugin, it will take the value of the first entry. So, for example:

- Access token field "UserAccountAge" = 01
- JavaScript field "UserAccountAge" = 02
- JavaScript field "UserAccountAge" = 03

The value used will be 02

Populating data variables example

```
var cc = document.getElementById('PaymentCreditCard').value;
var name = document.getElementById('Name').value;
var myData = $XIPlugin.createJSRequestPacket(merchantGuid, accessToken);
myData.addField($XIPlugin.createField('PaymentCreditCard', true, cc));
myData.addField($XIPlugin.createField('name', false, name));
myData.addAdditional3DSField("UserAccountAge", "04");
```

You can reference or download the following JavaScript plugin:

```
https://xiecomm.paymetric.com/DIeComm/Scripts/XIPlugin/XIPlugin-1.2.0.js
```

⚠ **Note:**

Existing integrations can still use XiPlugin01.1.0.js UNLESS you are adding 3DS2 support, then you must use XiPlugin-1.2.0.js version.

Include the Paymetric JavaScript and reference the Submit Example below to handle the posting of the credit card details directly to Intercept eComm.

Ajax: QA JS Submit example

```
$XIPlugin.ajax({

    url: '<% Intercept Url%>' + "/Ajax",
    type: "POST", //<GET/POST>
    data: {
        Formatted JS Data– See above example myData
    },
    success: function(a) {
        //function triggered on success.
    },
    error: function(a) {
        //function triggered on failure.
    },
    beforeSend: function(a) {
        //function triggered before ajax call.
    },
    complete: function(a) {
        //function triggered after ajax call.
        // After success or failure
    }
    validation: function(a) {
        //Run custom validation
        return true; //should return true or false.

        // return true will go through with ajax call.
        // return false will not go through with ajax call.
    }


});
```

Ajax: QA JS Submit 3DS2 example

```
$XIPlugin.ajax({

    url: '<% Intercept Url%>' + "/Ajax",
    type: "POST", //<GET/POST>
    data: {
        Formatted JS Data– See above example myData
    },
    success: function(a) {
        //function triggered on success.
    },
    error: function(a) {
        //function triggered on failure.
    },
    beforeSend: function(a) {
        //function triggered before ajax call.
    },
    complete: function(a) {
        //function triggered after ajax call.
        // After success or failure
    }
    validation: function(a) {
        //Run custom validation
        return true; //should return true or false.

        // return true will go through with ajax call.
        // return false will not go through with ajax call.
    },
    threeDSVersion: "2.2.0"

    intcp3DSecure: function(a) {
        //Optional function for custom 3DS challenge prompt; set to null
or omit for default prompt; default is full redirect to a Paymetric
hosted page
    }
});
```

8.3 Ajax: Get Response packet

Request method

A direct server-side GET is required to retrieve the Response Packet. How this GET is performed depends on the language you are using.

Content type

The content type must be "application/x-www-form-urlencoded''.

Endpoints

For the latest endpoints and URLs, see the Web Service and Resource URLs topic in Paymetric Assist.

Cert

```
https://cert-xiecomm.paymetric.com/DIeComm/ResponsePacket
```

Production

```
https://xiecomm.paymetric.com/DIeComm/ResponsePacket
```

GET Call parameters

```
MerchantGUID={MerchGUID}&Signature={Signature}&AccessToken={AccessToken}
```

Parameter	Description
MerchantGuid	The Global Unique Identifier assigned.
Signature	Sign the Access Token; encode it with HMAC-SHA-256 using the shared key then covert it to a base 64 string. The Signature must be URL encoded.

Parameter	Description
AccessToken	A string that is dynamically generated by Intercept eComm during the first application server call. The call includes the Merchant GUID and the Intercept eComm Request Packet that is signed by the Shared Key.

9 Iframe integration

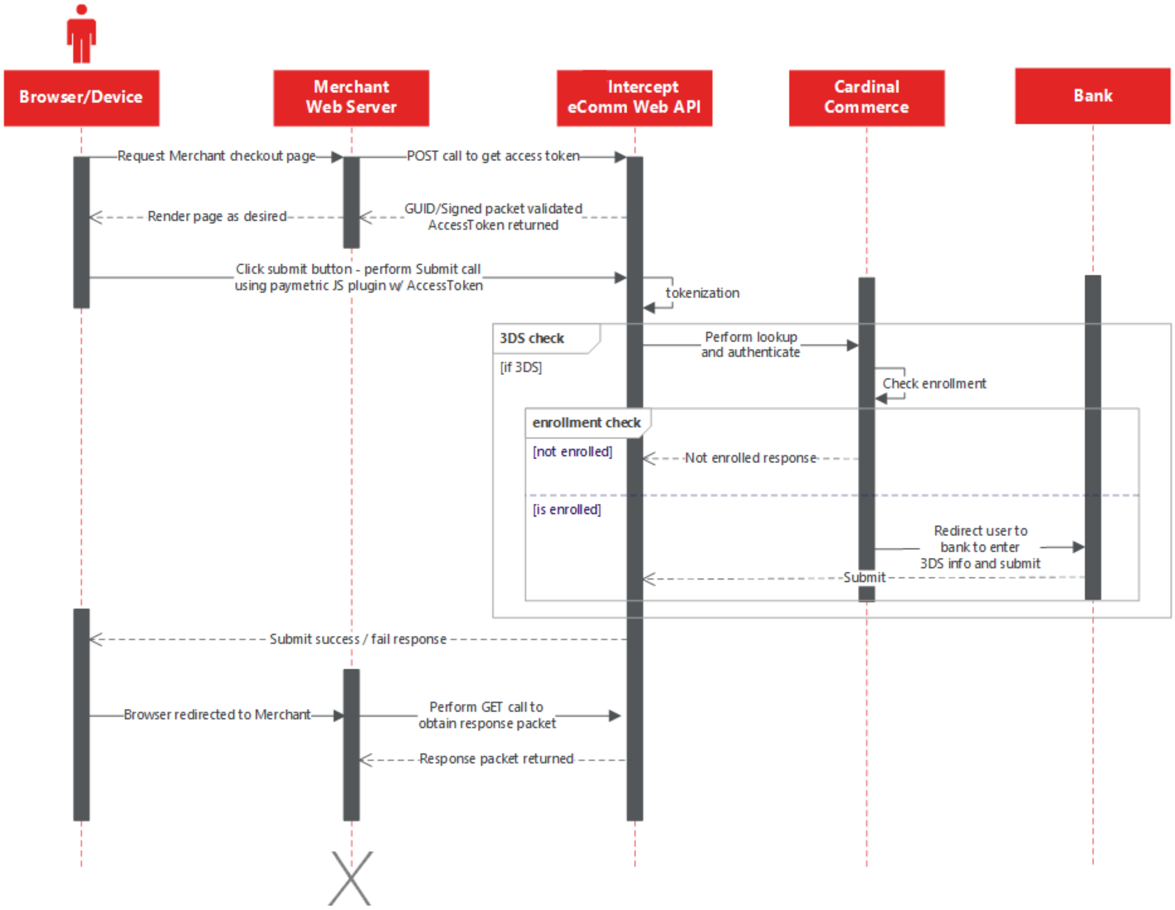
Iframe integration consists of the following high-level steps:

1. Initiate Intercept eComm session with a POST call to get the access token.
   - If implementing 3DS, there are extra fields that may be provided in the PacketXML including an attribute to indicate the 3DS version.
2. Embed Iframe in web page.
3. Render page as desired and upon user submission of Credit Card data, perform POST call using the Paymetric JS plugin.
   - If implementing 3DS 2.0, Intercept eComm automatically performs the additional device data collection (DDC) used by Cardinal and the Banks.
4. Perform GET call to get the response packet containing the token. If implementing 3DS, all response fields from Cardinal Commerce will be included as well.

The following sequence diagram illustrates the Iframe integration flow. Review the subsections below for more details.

⚠ **Note:**

If implementing 3DS2, the user is only redirected to the bank to enter 3DS credentials if the DDC or merchant-provided risk information is determined to be insufficient.


Browser/Device
Merchant Web Server
Intercept eComm Web API
Cardinal Commerce
Bank
Request Merchant checkout page
POST call to get access token
GUID/Signed packet validated AccessToken returned
Render page as desired
Click submit button - perform Submit call using paymetric JS plugin w/ AccessToken
tokenization
3DS check
[if 3DS]
Perform lookup and authenticate
Check enrollment
enrollment check
[not enrolled]
Not enrolled response
[is enrolled]
Redirect user to bank to enter 3DS info and submit
Submit
Submit success / fail response
Browser redirected to Merchant
Perform GET call to obtain response packet
Response packet returned

Templates and customizations

When using the Iframe integration type, there are templates available for the following:

- Card
- Card, CVV
- Card, CVV and Address

Alternatively, you can customize the fields that display. Refer to Custom HTML Reference 151 for complete details on the Custom Iframe option.

Whether using an Iframe template or the Custom Iframe option, you can use your own Cascading Style Sheet (CSS) to control the look and feel of your web page.

Localization

You have the ability to localize standard labels and messaging for Iframe integrations; see the Localization 126 section for more information.

9.1 Iframe: Get AccessToken

AccessToken is required to initiate an Intercept eComm session.

AccessToken is a string that is dynamically generated by Intercept eComm during the first application server call. The call includes the Merchant GUID and the Intercept eComm Request Packet that is signed by the Shared Key.

AccessToken is valid for 30 minutes. If the AccessToken expires before the process is complete, you must restart the process and obtain a new AccessToken.

Request method

A direct server-side POST is required to retrieve the Access Token. How this POST is performed depends on the language you are using.

Content type

The content type must be "application/x-www-form-urlencoded"

Endpoints

For the latest endpoints and URLs, see the Web service and resource URLs topic in Paymetric Assist.

Cert

```
https://cert-xiecomm.paymetric.com/DIeComm/AccessToken
```

Production

```
https://xiecomm.paymetric.com/DIeComm/AccessToken
```

POST Call parameters

```
MerchantGuid={Your-GUID}&SessionRequestType={Int-Type-Code}
&Signature={Signature}&MerchantDevelopmentEnvironment={code language e.g.
asp.net}&Packet={PACKET_XML}
```

Parameter	Description
MerchantGuid	The Global Unique Identifier assigned.
SessionRequestType	Indicates the integration type. Use 1 for Iframe.
Signature	Sign the Packet XML; encode it with HMAC-SHA-256 using the shared key then convert it to a base 64 string. The Signature must be URL encoded.
MerchantDevelopment Environment	Indicate the language in which you are coding your integration to Intercept eComm, for example, asp.net or php.
Packet XML	The Packet XML must be URL encoded. Following are the elements:

Parameter	Description
	• Iframe Templates o hostUri = the domain of the Iframe parent page o cssUri = the URL of the Merchant's CSS file o templateHtml = of three templates (creditcard, creditcardCVV, creditcardCVVAddress) o paymentTypes = create separate payment type for each one supported; these display in the card type/payment type dropdown • Iframe Custom: o hostUri = the domain of the Iframe parent page o cssUri = the URL of the Merchant's CSS file o merchantHTML = see Custom HTML Reference 151 for details • If implementing 3DS, include the following attributes for the <cardinal3DSecure> element for either the Iframe Template or Iframe Custom: o redirectUri = the redirect URL back to the merchant after 3DS processing o amount = amount of transaction o currencyCode = currency code for transaction

Parameter	Description
	o orderNumber = the Merchant's order number o additionalFields = optional container with additional fields for 3DS2 See samples below.

9.1.1 Iframe template XML sample

Iframe template XML sample

```
<?xml version="1.0" encoding="utf-8"?>
<merchantHtmlPacketModel>
   <iFramePacket>
       <hostUri>{http://Yourdomain.here}/</hostUri>
       <cssUri>{http://Yourcssurlhere.here}</cssUri>
   </iFramePacket>
   <templateHtml name="CreditCard">
       <paymentTypes>
           <paymentType type="american express" />
           <paymentType type="mastercard" />
           <paymentType type="maestro" />
           <paymentType type="visa" />
           <paymentType type="custom" value="HD" text="Home Depot" />
       </paymentTypes>
   </templateHtml>
</merchantHtmlPacketModel>
```

9.1.2 Iframe custom XML sample

Iframe custom XML sample

```
<merchantHtmlPacketModel
xmlns="Paymetric:Intercept:MerchantHtmlPacketModel">
   <iFramePacket>
       <hostUri>{http://Yourdomain.here}</hostUri>
       <cssUri>{http://Yourcssurlhere.here}</cssUri>
   </iFramePacket>
   <htmlFieldValues>
       <field for="cardholderName" value="John Smith" />
       <additionalField for="foo" value="My custom value" />
   </htmlFieldValues>
   <merchantHtml>
       <htmlSection class="merchant_paycontent">
           <xiProperties error-class="xiInputError" >
               <errorTooltip class="errorTooltip" show-effect="fadeIn"
show-duration="5000" hide-effect="fadeOut" hide-duration="5000" />
           </xiProperties>
           <paymentOptionRadioSection>
               <tag name="div" class="PaymentDetailHeader">Select a
payment option</tag>
               <label for="echeck" />
               <radPaymentOption for="echeck" />
               <label for="card" />
               <radPaymentOption for="card" />
           </paymentOptionRadioSection>
           <cardDropdownSection>
               <tag name="div" class="PaymentDetailHeader">Enter your
credit card information</tag>
               <tag name="div">
                   <label for="cardType" />
                   <ddlCardType id="cd">
                       <items>
                           <item for="american express" />
                           <item for="mastercard" />
                           <item for="visa" />
                           <item for="maestro" />
                           <customItem for="CA" text="Company
A"></customItem>
                       </items>
                   </ddlCardType>
               </tag>
               <tag name="div">
                   <label for="cardNumber" />
                   <tboxCardNumber tokenize="true" />
                   <validationMsg for="cardNumber" class="valmsg" />
```

Iframe custom XML sample

```
                </tag>
                <tag name="div">
                    <label for="cardholderName" />
                    <tboxCardHolderName />
                    <validationMsg for="cardholderName"
class="valmsg" />
                </tag>
                <tag name="div">
                    <label for="startMonth" />
                    <ddlStartMonth default-text="month" display-
format="MMM" class="merchant_combos" required="false" />
                    <ddlStartYear default-text="year"
class="merchant_combos" years-to-display="10" required="false" start-
date="true" />
                    <validationMsg for="startYear" class="valmsg" />
                </tag>
                <tag name="div">
                    <label for="expMonth" />
                    <ddlExpMonth default-text="month"
class="merchant_combos" required="false" />
                    <ddlExpYear default-text="year"
class="merchant_combos" years-to-display="10" required="false" exp-
date="true" />
                    <validationMsg for="expYear" class="valmsg" />
                </tag>
                <tag name="div">
                    <label for="cvv" />
                    <tboxCvv />
                    <validationMsg for="cvv" class="valmsg" />
                </tag>
            </cardDropdownSection>
            <echeckRadioSection>
                <tag name="div" class="PaymentDetailHeader">Enter your
bank information</tag>
                <tag name="div">
                    <validationMsg for="accountType" class="valmsg-ac-
type" />
                    <label for="checking" />
                    <radAccountType for="checking" />
                    <label for="savings" />
                    <radAccountType for="savings" />
                </tag>
                <tag name="div">
                    <label for="accountNumber" />
                    <tboxAccountNumber tokenize="true" error-msg-
style="imageTooltip" />
```

Iframe custom XML sample

```
                   <validationMsg for="accountNumber" error-
class="valmsg-img" />
               </tag>
               <tag name="div">
                   <label for="routingNumber" />
                   <tboxRoutingNumber error-msg-style="imageTooltip" />
                   <validationMsg for="routingNumber" error-
class="valmsg-img" />
               </tag>
               <tag name="div">
                   <label for="nameOnAccount" />
                   <tboxNameOnAccount error-msg-style="imageTooltip" />
                   <validationMsg for="nameOnAccount" error-
class="valmsg-img" />
               </tag>
               <tag name="div">
                   <label for="bankName" />
                   <tboxBankName error-msg-style="imageTooltip" />
                   <validationMsg for="bankName" error-class="valmsg-
img" />
               </tag>
           </echeckRadioSection>
       </htmlSection>
   </merchantHtml>
</merchantHtmlPacketModel>
```

9.1.3 Iframe custom with 3DS 1 XML sample

Iframe custom with 3DS 1.0 XML sample

```
<merchantHtmlPacketModel
xmlns="Paymetric:Intercept:MerchantHtmlPacketModel">
   <iFramePacket>
       <hostUri>{http://Yourdomain.here}</hostUri>
       <cssUri>{http://Yourcssurihere.here}</cssUri>
       <cardinal3DSecure amount="13.42" currencyCode="840"
orderNumber="12345" redirectUri="{http://Yourredirecturi.here}" />
   </iFramePacket>
   <merchantHtml>
```

Iframe custom with 3DS 1.0 XML sample

```
       <htmlSection class="merchant_paycontent"
xmlns="Paymetric:Intercept:MerchantHtmlPacketModel">
           <xiProperties error-class="xiInputError">
               <errorTooltip class="errorTooltip" show-effect="fadeIn"
show-duration="5000" hide-effect="fadeOut" hide-duration="5000" />
           </xiProperties>
           <paymentOptionRadioSection>
               <tag name="div" class="PaymentDetailHeader">Select a
payment option</tag>
               <label text="echeck" for="echeck" />
               <radPaymentOption for="echeck" />
               <label text="credit card" for="card" />
               <radPaymentOption for="card" />
           </paymentOptionRadioSection>
           <cardDropdownSection>
               <tag name="div" class="PaymentDetailHeader">Enter your
credit card information</tag>
               <tag name="div">
                   <label for="cardType" text="card type" />
                   <ddlCardType id="cd">
                       <items>
                           <item for="american express" />
                           <item for="mastercard" />
                           <item for="visa" />
                           <item for="maestro" />
                           <customItem for="myCard" text="My Card" />
                           <customItem for="myCard2" text="My Card
2" />
                       </items>
                   </ddlCardType>
               </tag>
               <tag name="div">
                   <label for="cardNumber" text="card number" />
                   <tboxCardNumber tokenize="true" />
                   <validationMsg for="cardNumber" class="valmsg" />
               </tag>
               <tag name="div">
                   <label for="cardholderName" text="name on card" />
                   <tboxCardHolderName />
                   <validationMsg for="cardholderName"
class="valmsg" />
               </tag>
               <tag name="div">
                   <label for="startMonth" text="start date" />
                   <ddlStartMonth default-text="month" display-
format="MMM" class="merchant_combos" required="false" />
```

Iframe custom with 3DS 1.0 XML sample

```
                   <ddlStartYear default-text="year"
class="merchant_combos" years-to-display="10" required="false" start-
date="true" />
                   <validationMsg for="startYear" class="valmsg" />
               </tag>
               <tag name="div">
                   <label for="expMonth" text="exp date" />
                   <ddlExpMonth default-text="month"
class="merchant_combos" required="false" />
                   <ddlExpYear default-text="year"
class="merchant_combos" years-to-display="10" required="false" exp-
date="true" />
                   <validationMsg for="expYear" class="valmsg" />
               </tag>
               <tag name="div">
                   <label for="cvv" text="card cvv" />
                   <tboxCvv />
                   <validationMsg for="cvv" class="valmsg" />
               </tag>
               <tag name="div">
                   <label for="cardAmount" />
                   <tboxCardAmount />
                   <validationMsg for="cardAmount" class="valmsg" />
               </tag>
           </cardDropdownSection>
           <echeckRadioSection>
               <tag name="div" class="PaymentDetailHeader">Enter your
bank information</tag>
               <tag name="div">
                   <validationMsg for="accountType" class="valmsg-ac-
type" />
                   <label for="checking" />
                   <radAccountType for="checking" />
                   <label for="savings" />
                   <radAccountType for="savings" />
               </tag>
               <tag name="div">
                   <label for="accountNumber" text="account number" />
                   <tboxAccountNumber tokenize="true" error-msg-
style="imageTooltip" />
                   <validationMsg for="accountNumber" error-
class="valmsg-img" />
               </tag>
               <tag name="div">
                   <label for="routingNumber" text="routing number" />
                   <tboxRoutingNumber error-msg-style="imageTooltip" />
```

Iframe custom with 3DS 1.0 XML sample

```
                   <validationMsg for="routingNumber" error-
class="valmsg-img" />
               </tag>
               <tag name="div">
                   <label for="nameOnAccount" text="name on account" />
                   <tboxNameOnAccount error-msg-style="imageTooltip" />
                   <validationMsg for="nameOnAccount" error-
class="valmsg-img" />
               </tag>
               <tag name="div">
                   <label for="bankName" text="bank name" />
                   <tboxBankName error-msg-style="imageTooltip" />
                   <validationMsg for="bankName" error-class="valmsg-
img" />
               </tag>
           </echeckRadioSection>
       </htmlSection>
   </merchantHtml>
</merchantHtmlPacketModel>
```

9.1.4 Iframe custom with 3DS 2 XML sample

The 3DS fields that differ form the prior example and are specific to version 2 are highlighted blue in the example below. These include `threeDSVersion` attribute within the `<cardinal3DSecure>` container and the `<additionalFields>` container and child elements.

The `threeDSVersion` attribute must be set to "2" or can be a specific version up to 3 digits. If there is a dot (.), then there must be a following number. E.g. "2" or "2.0" or "2.2", and so on.

Any Cardinal 3DS field can be included in `<additionalFields>` container. See the following URL for a complete list of the additional Cardinal fields.

https://cardinaldocs.atlassian.net/wiki/spaces/CCen/pages/905478187/Lookup+Request+Response

⚠ **Note:**

Cardinal <AdditionalFields> container cannot have duplicate field elements.

Iframe Custom with 3DS 2.0 XML sample

```
<merchantHtmlPacketModel
xmlns="Paymetric:Intercept:MerchantHtmlPacketModel">
   <iFramePacket>
       <hostUri>{http://Yourdomain.here}</hostUri>
       <cssUri>{http://Yourcssurl.here}</cssUri>
       <cardinal3DSecure amount="13.42" currencyCode="840"
orderNumber="12345" redirectUri="{http://Yourredirecturi.here}"
threeDSVersion="2.2.0">
           <additionalFields>
               <field name="UserAccountAge" value="04" />
               <field name="UserPasswwordChange" value="20190325" />
               <field name="DeliveryTimeframe" value="02" />
               <field name="GiftCardAmount" value="100" />
           </additionalFields>
       </cardinal3DSecure>
   </iFramePacket>
   <merchantHtml>
       <htmlSection class="merchant_paycontent"
xmlns="Paymetric:Intercept:MerchantHtmlPacketModel">
           <xiProperties error-class="xiInputError">
               <errorTooltip class="errorTooltip" show-effect="fadeIn"
show-duration="5000" hide-effect="fadeOut" hide-duration="5000" />
           </xiProperties>
           <paymentOptionRadioSection>
               <tag name="div" class="PaymentDetailHeader">Select a
payment option</tag>
               <label text="echeck" for="echeck" />
               <radPaymentOption for="echeck" />
               <label text="credit card" for="card" />
               <radPaymentOption for="card" />
           </paymentOptionRadioSection>
           <cardDropdownSection>
               <tag name="div" class="PaymentDetailHeader">Enter your
credit card information</tag>
               <tag name="div">
```

Iframe Custom with 3DS 2.0 XML sample

```
                   <label for="cardType" text="card type" />
                   <ddlCardType id="cd">
                       <items>
                           <item for="american express" />
                           <item for="mastercard" />
                           <item for="visa" />
                           <item for="maestro" />
                           <customItem for="myCard" text="My Card" />
                           <customItem for="myCard2" text="My Card
2" />
                       </items>
                   </ddlCardType>
               </tag>
               <tag name="div">
                   <label for="cardNumber" text="card number" />
                   <tboxCardNumber tokenize="true" />
                   <validationMsg for="cardNumber" class="valmsg" />
               </tag>
               <tag name="div">
                   <label for="cardholderName" text="name on card" />
                   <tboxCardHolderName />
                   <validationMsg for="cardholderName"
class="valmsg" />
               </tag>
               <tag name="div">
                   <label for="startMonth" text="start date" />
                   <ddlStartMonth default-text="month" display-
format="MMM" class="merchant_combos" required="false" />
                   <ddlStartYear default-text="year"
class="merchant_combos" years-to-display="10" required="false" start-
date="true" />
                   <validationMsg for="startYear" class="valmsg" />
               </tag>
               <tag name="div">
                   <label for="expMonth" text="exp date" />
                   <ddlExpMonth default-text="month"
class="merchant_combos" required="false" />
                   <ddlExpYear default-text="year"
class="merchant_combos" years-to-display="10" required="false" exp-
date="true" />
                   <validationMsg for="expYear" class="valmsg" />
               </tag>
               <tag name="div">
                   <label for="cvv" text="card cvv" />
                   <tboxCvv />
                   <validationMsg for="cvv" class="valmsg" />
```

Iframe Custom with 3DS 2.0 XML sample

```
               </tag>
               <tag name="div">
                   <label for="cardAmount" />
                   <tboxCardAmount />
                   <validationMsg for="cardAmount" class="valmsg" />
               </tag>
           </cardDropdownSection>
           <echeckRadioSection>
               <tag name="div" class="PaymentDetailHeader">Enter your
bank information</tag>
               <tag name="div">
                   <validationMsg for="accountType" class="valmsg-ac-
type" />
                   <label for="checking" />
                   <radAccountType for="checking" />
                   <label for="savings" />
                   <radAccountType for="savings" />
               </tag>
               <tag name="div">
                   <label for="accountNumber" text="account number" />
                   <tboxAccountNumber tokenize="true" error-msg-
style="imageTooltip" />
                   <validationMsg for="accountNumber" error-
class="valmsg-img" />
               </tag>
               <tag name="div">
                   <label for="routingNumber" text="routing number" />
                   <tboxRoutingNumber error-msg-style="imageTooltip" />
                   <validationMsg for="routingNumber" error-
class="valmsg-img" />
               </tag>
               <tag name="div">
                   <label for="nameOnAccount" text="name on account" />
                   <tboxNameOnAccount error-msg-style="imageTooltip" />
                   <validationMsg for="nameOnAccount" error-
class="valmsg-img" />
               </tag>
               <tag name="div">
                   <label for="bankName" text="bank name" />
                   <tboxBankName error-msg-style="imageTooltip" />
                   <validationMsg for="bankName" error-class="valmsg-
img" />
               </tag>
           </echeckRadioSection>
       </htmlSection>
   </merchantHtml>
```

Iframe Custom with 3DS 2.0 XML sample

```
</merchantHtmlPacketModel>
```

9.2 Iframe: Embed Iframe in Web page

To set the source URL of the Iframe:

```
Iframe source  = {DieCommUrl}/view/iframe/{MerchantGuid}/
{AccessTokenPacket}/{BubbleUpErrorToParentWindow(boolean Type)}
```

For example:

```
https://cert-xiecomm.paymetric.com/diecomm/View/Iframe/8c9aea21-
6063-4f30-b09d-e44706fedb05/942fb4bf-x546-4b32-942d-
76aadg3087e2/True
```

Iframe renders using fields defined in the Iframe: Get Access Token 43 request.

Iframe sizing can be handled by HTML or CSS.

Resizing of the Iframe can be handled by CSS, but if there is a need to handle resizing depending upon content that is inside Iframe (e.g. different fields for different payment methods or validation error messages), then include the following function.

Resize Iframe function sample

```
    $XIFrame.onload({
            iFrameId: {Iframe ID},
            targetUrl: {Iframe source}
            autosizewidth: true,
            onSuccess: function (e) {
                //Successfully Iframe Loaded
```

Resize Iframe function sample

```
            },
            onError: function (e) {
               //Error on Iframe Loading
            }
        });
```

9.3 Iframe: On-Submit Javascript plugin

When the customer performs the submit action by clicking the merchant-created submit button, the data will be posted to Paymetric using JavaScript functions.

You can reference or download the following JavaScript plugin:

`https://xiecomm.paymetric.com/DIeComm/Scripts/XIFrame/XIFrame-1.2.0.js`

⚠ **Note:**

Existing integrations can still use XiPlugin01.1.0.js UNLESS you are adding 3DS2 support, then you must use XiPlugin-1.2.0.js version.

Include the Paymetric JavaScript and reference the Submit Example below to handle the posting of the credit card details directly to Intercept eComm.

QA JS Submit example

```
$XIFrame.submit({
    iFrameId: {
        Iframe ID
    },
```

QA JS Submit example

```
    targetUrl: {
        Iframe source
    }
    onSuccess: function(msg) {
        //function triggered on success
    },
    onError: function(msg) {
        //function triggered on failure
    }
});
```

If validation passes and tokenization is successful, then the onSuccess function is called; otherwise, onError function is called.

QA JS Submit 3DS 2 example

```
    $XIFrame.submit({
       iFrameId: {Iframe ID},
       targetUrl: {Iframe source}
       onSuccess: function (msg)
                    {
                         //function triggered on success
                    },
       onError: function (msg)
                    {
                        //function triggered on failure
                    },
      threeDSVersion: "2.2.0"
         });
```

9.4 Iframe: Get Response packet

Request method

A direct server-side GET is required to retrieve the Response Packet. How this GET is performed depends on the language you are using.

Content type

The content type must be "application/x-www-form-urlencoded''.

Endpoints

For the latest endpoints and URLs, see the Web Service and Resource URLs topic in Paymetric Assist.

Cert

```
https://cert-xiecomm.paymetric.com/DIeComm/ResponsePacket
```

Production

```
https://xiecomm.paymetric.com/DIeComm/ResponsePacket
```

GET Call parameters

Following is the GET data:

```
MerchantGUID={MerchGUID}&Signature={Signature}
&AccessToken={AccessToken}
```

Parameter	Description
MerchantGuid	The Global Unique Identifier assigned.

Parameter	Description
Signature	Sign the Access Token; encode it with HMAC-SHA-256 using the shared key then covert it to a base 64 string. The Signature must be URL encoded.
AccessToken	A string that is dynamically generated by Intercept eComm during the first application server call. The call includes the Merchant GUID and the Intercept eComm Request Packet that is signed by the Shared Key.

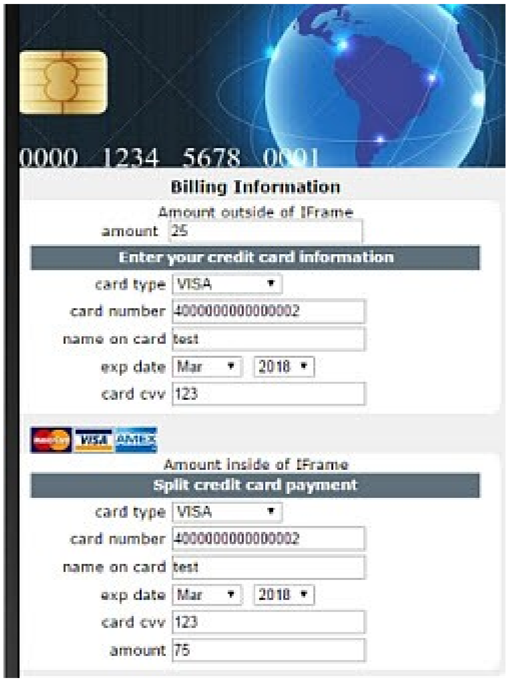
9.5 Iframe: Additional information

The section contains additional information related to Iframe integration as follows:

- Multiple IFrames 61
- Display content based on selected payment type 61
- Embed 3DS in IFrame 62

9.5.1 Multiple IFrames

You can implement multiple IFrames on your web page, for example, if you want to allow multiple payment methods.



Billing Information

Amount outside of IFrame

amount 25

Enter your credit card information

card type VISA

card number 4000000000000002

name on card test

exp date Mar 2018

card cvv 123

Amount inside of IFrame

Split credit card payment

card type VISA

card number 4000000000000002

name on card test

exp date Mar 2018

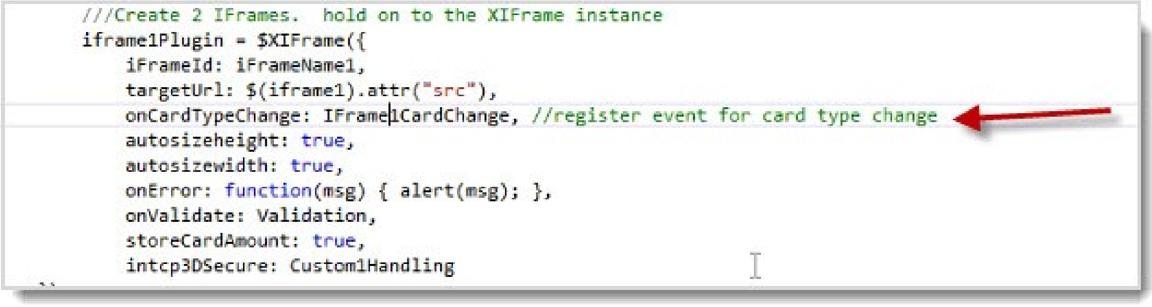
card cvv 123

amount 75

9.5.2 Display content based on selected payment type

Intercept eComm has a feature that allows a change to the Payment Type field within the Iframe to fire an event at the parent level.

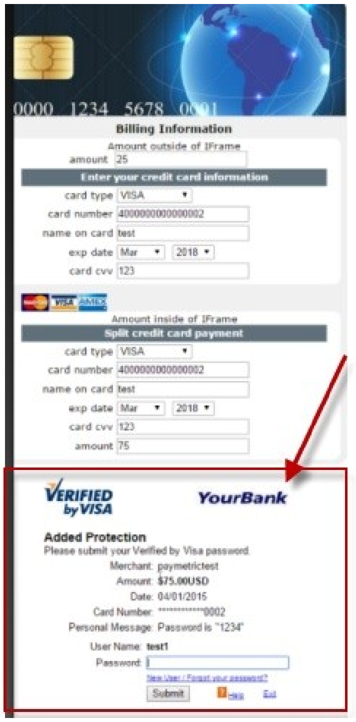
The `onCardTypeChange` element is used to trigger the on-change event.



```
///Create 2 IFrames.  hold on to the XIFrame instance
iframe1Plugin = $XIFrame({
    iFrameId: iFrameName1,
    targetUrl: $(iframe1).attr("src"),
    onCardTypeChange: IFrame1CardChange, //register event for card type change
    autosizeheight: true,
    autosizewidth: true,
    onError: function(msg) { alert(msg); },
    onValidate: Validation,
    storeCardAmount: true,
    intcp3DSecure: Custom1Handling
```

9.5.3 Embed 3DS in IFrame

In the standard workflow defined above, the 3DS call redirects to the bank's website. You can choose to embed the 3DS fields within the Iframe. The following screenshot displays an example of 3DS fields embedded within the Iframe.



Billing Information

Amount outside of IFrame

amount 25

Enter your credit card information

card type VISA

card number 4000000000000002

name on card test

exp date Mar 2018

card cvv 123

Amount inside of IFrame

Split credit card payment

card type VISA

card number 4000000000000002

name on card test

exp date Mar 2018

card cvv 123

amount 75

Added Protection

Please submit your Verified by Visa password

Merchant: paymetrictest

Amount: $75.00USD

Date: 04/01/2015

Card Number: ************0002

Personal Message: Password is "1234"

User Name: test1

Password:

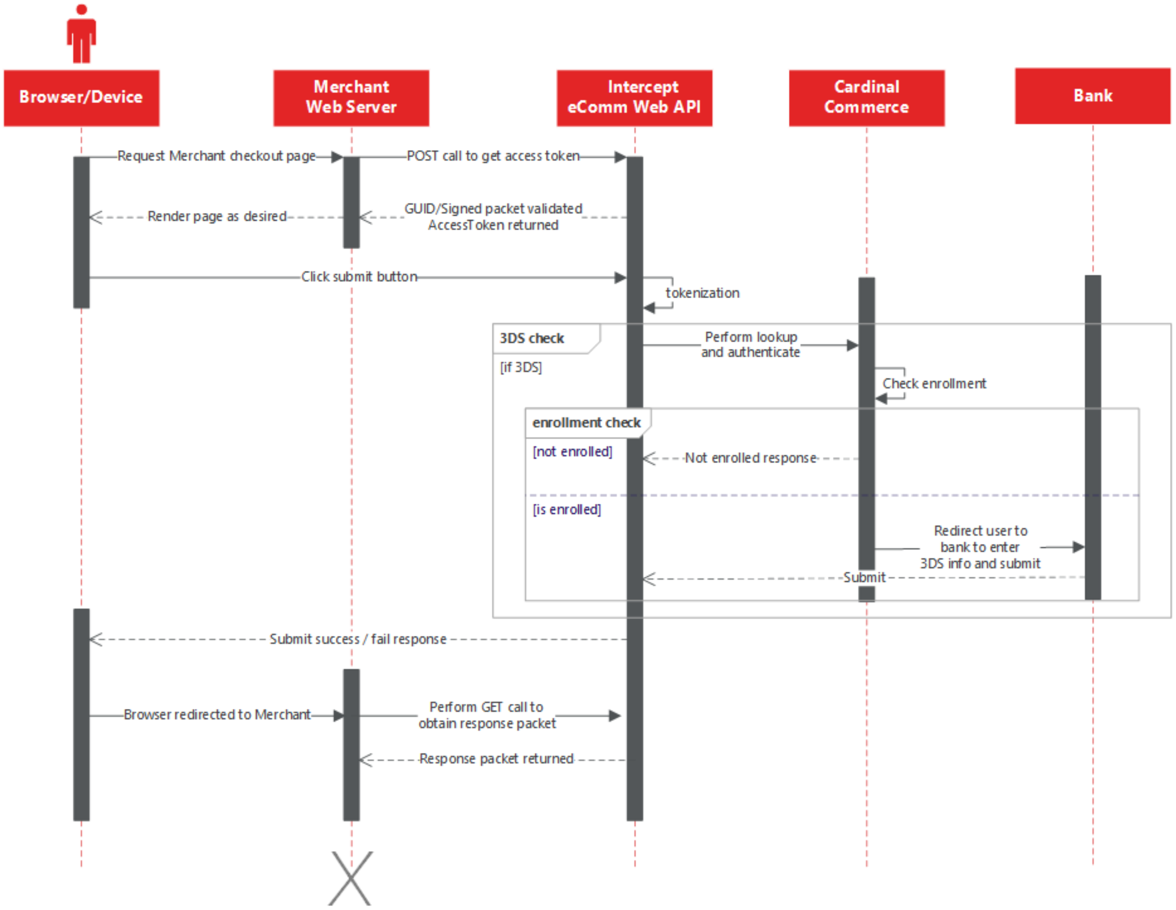
10 Hosted integration

Hosted integration consists of the following high-level steps:

1. Initiate Intercept eComm session with a POST call to get the access token upon page load then render page as desired.
   - If implementing 3DS, there are extra fields that should be provided in the PacketXML including an attribute to indicate the 3DS version.
2. Redirect to Hosted page.
   - If implementing 3DS 2.0, Intercept eComm automatically performs the additional device data collection (DDC) used by Cardinal and the Banks.
3. Perform GET call to get the response packet containing the token. If implementing 3DS, all response fields from Cardinal Commerce will be included as well.

⚠ **Note:**

If implementing 3DS2, the user is only redirected to the bank to enter 3DS credentials if the DDC or merchant-provided risk information is determined to be insufficient.


Browser/Device
Merchant Web Server
Intercept eComm Web API
Cardinal Commerce
Bank
Request Merchant checkout page
POST call to get access token
Render page as desired
GUID/Signed packet validated AccessToken returned
Click submit button
tokenization
3DS check
[if 3DS]
Perform lookup and authenticate
Check enrollment
enrollment check
[not enrolled]
Not enrolled response
[is enrolled]
Redirect user to bank to enter 3DS info and submit
Submit
Submit success / fail response
Browser redirected to Merchant
Perform GET call to obtain response packet
Response packet returned

Localization

You have the ability to localize standard labels and messaging for hosted integrations; see the Localization 126 section for more information.

10 Hosted integration

10.1 Hosted: Get AccessToken

AccessToken is required to initiate an Intercept eComm session.

AccessToken is a string that is dynamically generated by Intercept eComm during the first application server call. The call includes the Merchant GUID and the Intercept eComm Request packet that is signed by the Shared Key.

AccessToken is valid for 30 minutes. If the AccessToken expires before the process is complete, you must restart the process and obtain a new AccessToken.

Request method

A direct server-side POST is required to retrieve the Access Token. How this POST is performed depends on the language you are using.

Content type

The content type must be "application/x-www-form-urlencoded".

Endpoints

For the latest endpoints and URLs, see the Web Service and Resource URLs topic in Paymetric Assist.

Cert

```
https://cert-xiecomm.paymetric.com/DIeComm/AccessToken
```

Production

```
https://xiecomm.paymetric.com/DIeComm/AccessToken
```

POST Call parameters

Following is the Post data:

```
MerchantGuid={Your-GUID}&SessionRequestType={Int-Type-Code}
&Signature={Signature}&MerchantDevelopmentEnvironment={code language e.g.
asp.net}&Packet={PACKET_XML}
```

Parameter	Description
MerchantGuid	The Global Unique Identifier assigned.
SessionRequestType	Indicates the integration type. Use 4 for Hosted.
Signature	Sign the Packet XML; encode it with HMAC-SHA-256 using the shared key then convert it to a base 64 string. The Signature must be URL encoded.
MerchantDevelopmentEnvironment	Indicate the language in which you are coding your integration to Intercept eComm, for example, asp.net or php.
Packet XML	The Packet XML must be URL encoded. Following are the elements: • redirectUri = the redirect URL for the Response Packet from Intercept eComm • cssUri = the URL of the Merchant's CSS file • merchantHtml = see Custom HTML Reference 151 for details (and XML sample below) If implementing 3DS, include the following attributes for the <cardinal3DSescure> element for either the Iframe Template or Iframe Custom:

Parameter	Description
	• redirectUri = the redirect URL back to the merchant after 3DS processing • amount = amount of transaction • currencyCode = currency code for transaction • orderNumber = the Merchant's order number • additionalFields = optional container with additional fields for 3DS2 See samples below.
customerRedirectUri	Applicable only for the Hosted page when implementing XiXiSecurelink. Defines the Merchant redirect URL after the User clicks submit.

10.1.1 Hosted custom XML sample

Hosted custom XML sample

```
<?xml version="1.0" encoding="utf-8"?>
<merchantHtmlPacketModel
xmlns="Paymetric:Intercept:MerchantHtmlPacketModel">
   <paymentPagePacket>
       <redirectUri>{http://Yourdomain.here}</redirectUri>
       <cssUri>http://dasvm121/DIeComm
Sample/Content/PaymentStyleSheet.css</cssUri>
   </paymentPagePacket>
```

Hosted custom XML sample

```
   <merchantHtml>
       <htmlSection class="payment-content"
xmlns="Paymetric:Intercept:MerchantHtmlPacketModel">
           <tag name="div" class="merchant-logo"></tag>
           <tag name="div" class="billing-content">
               <tag name="div" class="billing-header">Billing
Information</tag>
               <tag name="div" class="billing-info">
                   <paymentOptionRadioSection>
                       <tag name="div">
                           <tag name="div">choose payment type</tag>
                       </tag>
                       <label text="echeck" for="echeck" />
                       <radPaymentOption for="echeck" />
                       <label text="credit card" for="card" />
                       <radPaymentOption for="card" />
                   </paymentOptionRadioSection>
                   <cardDropdownSection>
                       <tag name="div">Enter your credit card
information</tag>
                       <tag name="div">
                           <label for="cardType" text="select a
card" />
                           <ddlCardType id="cd">
                               <items>
                                   <item for="american express" />
                                   <item for="mastercard" />
                                   <item for="visa" />
                               </items>
                           </ddlCardType>
                       </tag>
                       <tag name="div">
                           <label for="cardNumber" text="card
number" />
                           <tboxCardNumber tokenize="true" />
                       </tag>
                       <tag name="div">
                           <label for="cardholderName" text="name on
card" />
                           <tboxCardHolderName />
                       </tag>
                       <tag name="div">
                           <label for="expMonth" text="exp date" />
                           <ddlExpMonth default-text="month"
class="merchant_combos" />
```

Hosted custom XML sample

```
                            <ddlExpYear default-text="year"
class="merchant_combos" years-to-display="10" />
                        </tag>
                        <tag name="div">
                            <label for="cvv" text="card cvv:" />
                            <tboxCvv />
                        </tag>
                    </cardDropdownSection>
                    <echeckRadioSection>
                        <tag name="div">
                            <label for="checking" />
                            <radAccountType for="checking" />
                            <label for="savings" />
                            <radAccountType for="savings" />
                        </tag>
                        <tag name="div">
                            <label for="accountNumber" text="account
number" />
                            <tboxAccountNumber tokenize="true" />
                        </tag>
                        <tag name="div">
                            <label for="routingNumber" text="routing
number" />
                            <tboxRoutingNumber />
                        </tag>
                        <tag name="div">
                            <label for="nameOnAccount" text="name on
account" />
                            <tboxNameOnAccount />
                        </tag>
                        <tag name="div">
                            <label for="bankName" text="bank name" />
                            <tboxBankName />
                        </tag>
                    </echeckRadioSection>
                </tag>
                <tag name="div" class="payment-button">
                    <cancelButton text="cancel" class="cancel-button" />
                    <submitButton text="submit" class="submit-button" />
                </tag>
            </tag>
        </htmlSection>
    </merchantHtml>
</merchantHtmlPacketModel>
```

10.1.2 Hosted custom with 3DS 1 XML sample

Hosted custom with 3DS XML sample

```
<?xml version="1.0" encoding="utf-8"?>
<merchantHtmlPacketModel
xmlns="Paymetric:Intercept:MerchantHtmlPacketModel">
   <paymentPagePacket>
       <redirectUri>{http://Yourdomain.here}</redirectUri>
       <cssUri>http://dasvm121/DIeComm
Sample/Content/PaymentStyleSheet.css</cssUri>
       <cardinal3DSecure amount="13.42" currencyCode="840"
orderNumber="12345" />
   </paymentPagePacket>
   <merchantHtml>
       <htmlSection class="payment-content"
xmlns="Paymetric:Intercept:MerchantHtmlPacketModel">
           <tag name="div" class="merchant-logo"></tag>
           <tag name="div" class="billing-content">
               <tag name="div" class="billing-header">Billing
Information</tag>
               <tag name="div" class="billing-info">
                   <paymentOptionRadioSection>
                       <tag name="div">
                           <tag name="div">choose payment type</tag>
                       </tag>
                       <label text="echeck" for="echeck" />
                       <radPaymentOption for="echeck" />
                       <label text="credit card" for="card" />
                       <radPaymentOption="" for="card" />
                   </paymentOptionRadioSection>
                   <cardDropdownSection>
                       <tag name="div">Enter your credit card
information</tag>
                       <tag name="div">
                           <label for="cardType" text="select a
card" />
                           <ddlCardType id="cd">
                               <items>
                                   <item for="american express" />
                                   <item for="mastercard" />
                                   <item for="visa" />
                               </items>
                           </ddlCardType>
                       </tag>
                       <tag name="div">
                           <label for="cardNumber" text="card
number" />
```

Hosted custom with 3DS XML sample

```
                           <tboxCardNumber tokenize="true" />
                       </tag>
                       <tag name="div">
                           <label for="cardholderName" text="name on
card" />
                           <tboxCardHolderName />
                       </tag>
                       <tag name="div">
                           <label for="expMonth" text="exp date" />
                           <ddlExpMonth default-text="month"
class="merchant_combos" />
                           <ddlExpYear default-text="year"
class="merchant_combos" years-to-display="10" />
                       </tag>
                       <tag name="div">
                           <label for="cvv" text="card cvv:" />
                           <tboxCvv />
                       </tag>
                   </cardDropdownSection>
                   <echeckRadioSection>
                       <tag name="div">
                           <label for="checking" />
                           <radAccountType for="checking" />
                           <label for="savings" />
                           <radAccountType for="savings" />
                       </tag>
                       <tag name="div">
                           <label for="accountNumber" text="account
number" />
                           <tboxAccountNumber tokenize="true" />
                       </tag>
                       <tag name="div">
                           <label for="routingNumber" text="routing
number" />
                           <tboxRoutingNumber />
                       </tag>
                       <tag name="div">
                           <label for="nameOnAccount" text="name on
account" />
                           <tboxNameOnAccount />
                       </tag>
                       <tag name="div">
                           <label for="bankName" text="bank name" />
                           <tboxBankName />
                       </tag>
                   </echeckRadioSection>
```

Hosted custom with 3DS XML sample

```
               </tag>
               <tag name="div" class="payment-button">
                   <cancelButton text="cancel" class="cancel-button" />
                   <submitButton text="submit" class="submit-button" />
               </tag>
           </tag>
       </htmlSection>
   </merchantHtml>
</merchantHtmlPacketModel>
```

10.1.3 Hosted custom with 3DS 2 XML sample

The 3DS fields that differ form the prior example and are specific to version 2 are highlighted blue in the example below. These include `threeDSVersion` attribute within the `<cardinal3DSecure>` container and the `<additionalFields>` container and child elements.

The `threeDSVersion` attribute must be set to "2" or can be a specific version up to 3 digits. If there is a dot (.), then there must be a following number. E.g. "2" or "2.0" or "2.2", and so on.

Any Cardinal 3DS field can be included in `<additionalFields>` container. See the following URL for a complete list of the additional Cardinal fields.

https://cardinaldocs.atlassian.net/wiki/spaces/CCen/pages/905478187/Lookup+Request+Response

⚠ **Note:**

Cardinal <AdditionalFields> container cannot have duplicate field elements.

Hosted custom with 3DS XML sample

```
<?xml version="1.0" encoding="utf-8"?>
<merchantHtmlPacketModel
xmlns="Paymetric:Intercept:MerchantHtmlPacketModel">
   <paymentPagePacket>
       <redirectUri>{http://Yourdomain.here}</redirectUri>
       <cssUri>http://dasvm121/DIeComm
Sample/Content/PaymentStyleSheet.css</cssUri>
       <cardinal3DSecure amount="13.42" currencyCode="840"
orderNumber="12345" threeDSVersion="2.2.0">
           <additionalFields>
               <field name="UserAccountAge" value="04" />
               <field name="UserPasswwordChange" value="20190325" />
               <field name="DeliveryTimeframe" value="02" />
               <field name="GiftCardAmount" value="100" />
           </additionalFields>
       </paymentPagePacket>
       <merchantHtml>
           <htmlSection class="payment-content"
xmlns="Paymetric:Intercept:MerchantHtmlPacketModel">
               <tag name="div" class="merchant-logo"></tag>
               <tag name="div" class="billing-content">
                   <tag name="div" class="billing-header">Billing
Information</tag>
                   <tag name="div" class="billing-info">
                       <paymentOptionRadioSection>
                           <tag name="div">
                               <tag name="div">choose payment
type</tag>
                           </tag>
                           <label text="echeck" for="echeck" />
                           <radPaymentOption for="echeck" />
                           <label text="credit card" for="card" />
                           <radPaymentOption="" for="card" />
                       </paymentOptionRadioSection>
                       <cardDropdownSection>
                           <tag name="div">Enter your credit card
information</tag>
                           <tag name="div">
                               <label for="cardType" text="select a
card" />
                               <ddlCardType id="cd">
                                   <items>
                                       <item for="american express" />
                                       <item for="mastercard" />
                                       <item for="visa" />
                                   </items>
```

Hosted custom with 3DS XML sample

```
                               </ddlCardType>
                           </tag>
                           <tag name="div">
                               <label for="cardNumber" text="card
number" />
                               <tboxCardNumber tokenize="true" />
                           </tag>
                           <tag name="div">
                               <label for="cardholderName" text="name
on card" />
                               <tboxCardHolderName />
                           </tag>
                           <tag name="div">
                               <label for="expMonth" text="exp date" />
                               <ddlExpMonth default-text="month"
class="merchant_combos" />
                               <ddlExpYear default-text="year"
class="merchant_combos" years-to-display="10" />
                           </tag>
                           <tag name="div">
                               <label for="cvv" text="card cvv:" />
                               <tboxCvv />
                           </tag>
                       </cardDropdownSection>
                       <echeckRadioSection>
                           <tag name="div">
                               <label for="checking" />
                               <radAccountType for="checking" />
                               <label for="savings" />
                               <radAccountType for="savings" />
                           </tag>
                           <tag name="div">
                               <label for="accountNumber" text="account
number" />
                               <tboxAccountNumber tokenize="true" />
                           </tag>
                           <tag name="div">
                               <label for="routingNumber" text="routing
number" />
                               <tboxRoutingNumber />
                           </tag>
                           <tag name="div">
                               <label for="nameOnAccount" text="name on
account" />
                               <tboxNameOnAccount />
                           </tag>
```

Hosted custom with 3DS XML sample

```
                           <tag name="div">
                               <label for="bankName" text="bank
name" />
                               <tboxBankName />
                           </tag>
                       </echeckRadioSection>
                   </tag>
                   <tag name="div" class="payment-button">
                       <cancelButton text="cancel" class="cancel-
button" />
                       <submitButton text="submit" class="submit-
button" />
                   </tag>
               </tag>
           </htmlSection>
       </merchantHtml>
   </merchantHtmlPacketModel>
```

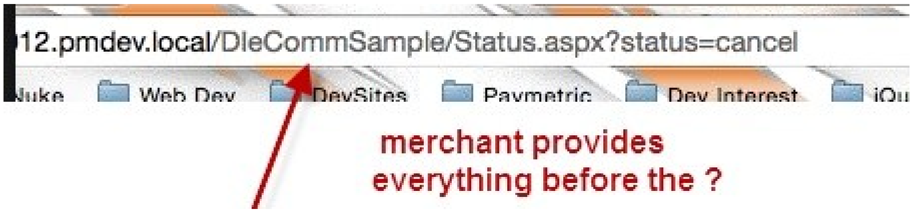
10.2 Hosted: Redirect to hosted page

After receiving the AccessToken response packet, redirect Customer to the Intercept eComm hosted page at the following URL:

QA	`https://cert-xiecomm.paymetric.com/DIeComm/paymentpage/index/{Merchant GUID}/{Access Token}/`
Production	`https://xiecomm.paymetric.com/DIeComm/paymentpage/index/{Merchant GUID}/{Access Token}/`

User enters information and submits.

If user cancels on the rendered page,



12.pmdev.local/DIeCommSample/Status.aspx?status=cancel

merchant provides everything before the ?

Figure 1: Hosted Page Cancel URL Example

10.3 Hosted: Get Response packet

Request method

A direct server-side GET is required to retrieve the Response Packet. How this GET is performed depends on the language you are using.

Content type

The content type must be "application/x-www-form-urlencoded".

Endpoints

For the latest endpoints and URLs, see the Web Service and Resource URLs topic in Paymetric Assist.

Cert

```
https://cert-xiecomm.paymetric.com/DIeComm/ResponsePacket
```

Production

```
https://xiecomm.paymetric.com/DIeComm/ResponsePacket
```

GET Call Parameters

Following is the GET data:

```
MerchantGUID=<MerchGUID>&Signature=<SIGNATURE&AccessToken=<AccessToken
>
```

Parameter	Description
MerchantGuid	The Global Unique Identifier assigned.
Signature	Sign the Access Token; encode it with HMAC-SHA-256 using the shared key then covert it to a base 64 string. The Signature must be URL encoded.
AccessToken	A string that is dynamically generated by Intercept eComm during the first application server call. The call includes the Merchant GUID and the Intercept eComm Request Packet that is signed by the Shared Key.

11 3DS call with token integration

Intercept eComm supports a workflow that allows you to submit a token for the 3DS call if, for example, you have an existing customer with a token on file.

Note that the Ajax call in this workflow does not refer to the Ajax Intercept eComm integration type for raw card numbers. This is a completely separate workflow that is implemented independently of the raw card section.

3DS with token integration consists of the following high-level steps:

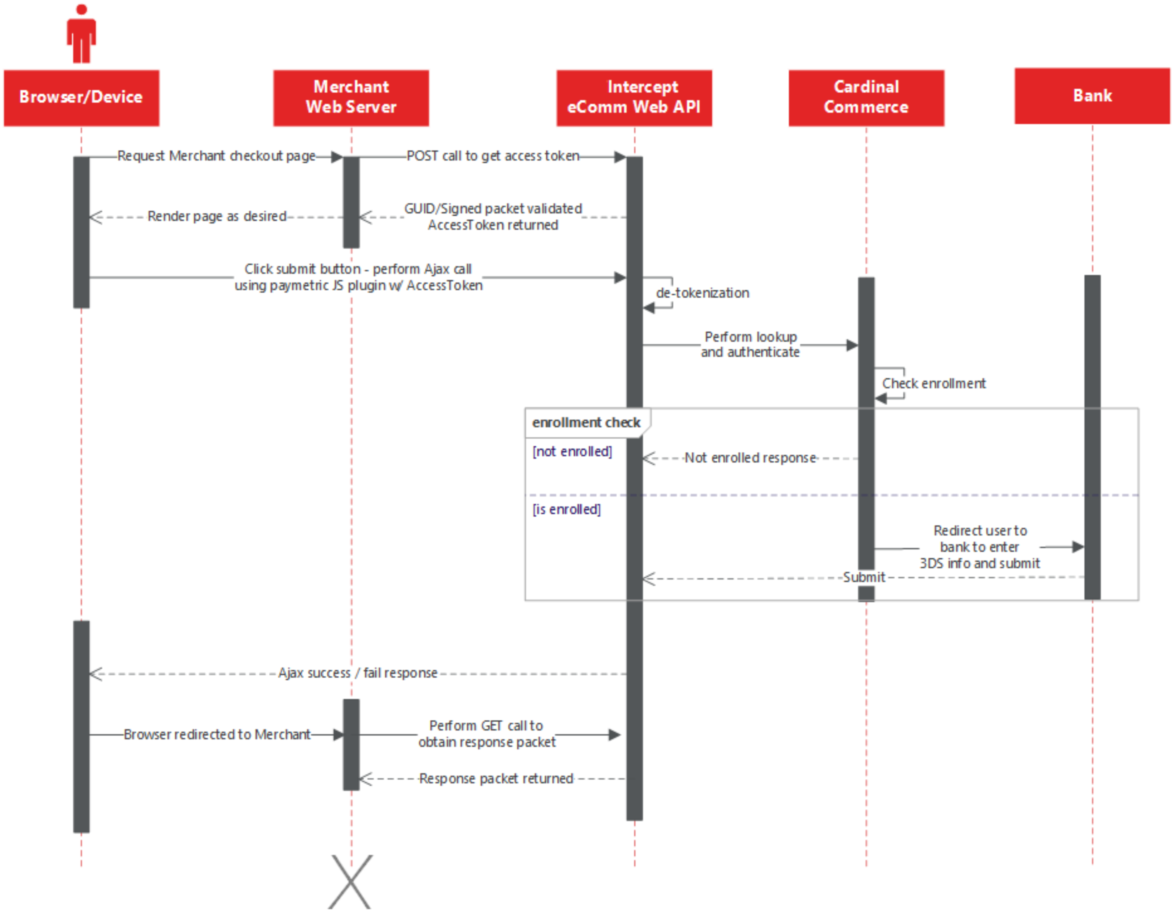
1. Initiate Intercept eComm session with a POST call to get the access token.
2. Merchant renders page as desired and upon user initiation of 3DS lookup call, perform POST call using the Paymetric JS plugin.
3. Perform GET call to get the response packet with token. All response fields from Cardinal Commerce are included in the response.

The following sequence diagram illustrates the 3DS with Token integration flow. Review the subsections below for more details.

⚠ **Note:**

> If implementing 3DS2, the user is only redirected to the bank to enter 3DS credentials if the DDC or merchant-provided risk information is determined to be insufficient.

11 3DS call with token integration


Browser/Device
Merchant Web Server
Intercept eComm Web API
Cardinal Commerce
Bank
Request Merchant checkout page
POST call to get access token
Render page as desired
GUID/Signed packet validated AccessToken returned
Click submit button - perform Ajax call using paymetric JS plugin w/ AccessToken
de-tokenization
Perform lookup and authenticate
Check enrollment
enrollment check
[not enrolled]
Not enrolled response
[is enrolled]
Redirect user to bank to enter 3DS info and submit
Submit
Ajax success / fail response
Browser redirected to Merchant
Perform GET call to obtain response packet
Response packet returned

11.1 3DS Token: Get AccessToken

AccessToken is required to initiate an Intercept eComm session.

AccessToken is a string that is dynamically generated by Intercept eComm during the first application server call. The call includes the Merchant GUID and the Intercept eComm Request Packet that is signed by the Shared Key.

AccessToken is valid for 30 minutes. If the AccessToken expires before the process is complete, you must restart the process and obtain a new AccessToken.

Request method

A direct server-side POST is required to retrieve the Access Token. How this POST is performed depends on the language you are using.

Content type

The content type must be "application/x-www-form-urlencoded".

Endpoints

For the latest endpoints and URLs, see the Web Service and Resource URLs topic in Paymetric Assist.

Cert

```
https://cert-xiecomm.paymetric.com/DIeComm/AccessToken
```

Production

```
https://xiecomm.paymetric.com/DIeComm/AccessToken
```

POST Call parameters

```
MerchantGuid={Your-GUID}&SessionRequestType={Int-Type-Code}
&Signature={Signature}&MerchantDevelopmentEnvironment={code language e.g.
asp.net}&Packet={PACKET_XML}
```

Parameter	Description
MerchantGuid	The Global Unique Identifier assigned.

Parameter	Description
SessionRequestType	Indicates the integration type. Use 2 for Ajax.
Signature	The Signature must be URL encoded. 1. Sign the packet XML. 2. Encode it with HMAC-SHA-256 using the shared key. 3. Convert it to a base 64 string.
MerchantDevelopmentEnvironment	Indicate the language in which you are coding your integration to Intercept eComm, for example, asp.net or php.
Packet XML	The Packet XML must be URL encoded. Following are the elements: • RedirectUri = the redirect URL for the Response Packet from Intercept eComm • Within the <Cardinal3DSWithToken> container: o XiSecureToken = the token representing the appropriate card number o Amount = amount of transaction o CurrencyCode = currency code for transaction o ExpirationMonth = the expiration month value entered by the User

Parameter	Description
	o ExpirationYear = the expiration year value entered by the User o OrderNumber = the Merchant's order number o AdditionalFields = optional container with additional fields for 3DS2 See samples below.

11.1.1 3DS 1 with Token XML sample

3DS with Token XML sample

```
<?xml version="1.0"?>
<Cardinal3DSWithTokenPacketModel>
   <RedirectUri>http://dasvm121/DIeComm Sample/Status.aspx</RedirectUri>
   <Cardinal3DSRequestWithToken>
       <XiSecureToken>Token</XiSecureToken>
       <Amount>1.42</Amount>
       <CurrencyCode>840</CurrencyCode>
       <ExpirationMonth>06</ExpirationMonth>
       <ExpirationYear>2018</ExpirationYear>
       <OrderNumber>12345</OrderNumber>
   </Cardinal3DSRequestWithToken>
</Cardinal3DSWithTokenPacketModel>
```

11.1.2 3DS 2 with Token XML sample

The 3DS fields that differ form the prior example and are specific to version 2 in the sample below include the `threeDSVersion` attribute within the `<Cardinal3DSRequestWithToken>` container and the `<AdditionalFields>` container and child elements.

The `threeDSVersion` attribute must be set to "2" or can be a specific version up to 3 digits. If there is a dot (.), then there must be a following number. E.g. "2" or "2.0" or "2.2", and so on.

Any Cardinal 3DS field can be included in `<AdditionalFields>` container. See the following URL for a complete list of the additional Cardinal fields.

https://cardinaldocs.atlassian.net/wiki/spaces/CCen/pages/905478187/Lookup+Request+Response

⚠ **Note:**

Cardinal <AdditionalFields> container cannot have duplicate field elements.

3DS 2 with Token XML sample

```
<?xml version="1.0"?>
<Cardinal3DSWithTokenPacketModel>
   <RedirectUri>http://dasvm121/DIeComm Sample/Status.aspx</RedirectUri>
   <Cardinal3DSRequestWithToken threeDSVersion="2.2.0">
       <XiSecureToken>Token</XiSecureToken>
       <Amount>1.42</Amount>
       <CurrencyCode>840</CurrencyCode>
       <ExpirationMonth>06</ExpirationMonth>
       <ExpirationYear>2018</ExpirationYear>
       <OrderNumber>12345</OrderNumber>
       <AdditionalFields>
           <Field name="UserAccountAge" value="04" />
           <Field name="UserPasswwordChange" value="20190325" />
           <Field name="DeliveryTimeframe" value="02" />
           <Field name="GiftCardAmount" value="100" />
       </AdditionalFields>
```

3DS 2 with Token XML sample

```
   </Cardinal3DSRequestWithToken>
</Cardinal3DSWithTokenPacketModel>
```

11.2 3DS Token: On-Submit Javascript plugin

When the customer performs the submit action by clicking the merchant-created submit button, the data will be posted to Paymetric using JavaScript functions.

Following is an example of pulling the values from the text boxes and populating the data variable.

⚠ **Note:**

For 3DS2, you can add the Cardinal additional fields via the JavaScript plugin as well. The following example includes one of the 3DS2 AdditionalFields.

If you send the same additional field that you sent when getting the access token, it will be overwritten.

If you accidentally add more than one field with the same name and different values via the plugin, it will take the value of the first entry. So, for example:

- Access token field "UserAccountAge" = 01
- JavaScript field "UserAccountAge" = 02
- JavaScript field "UserAccountAge" = 03

The value used will be 02

Populating data variables example

```
     var myData = $XIPlugin.createJSRequestPacket(merchantGuid,
accessToken);

        myData.addField($XIPlugin.createField('Amount', false,
$('#textbox').val()));
       myData.addAdditional3DSField("UserAccountAge", "04");
```

Note: Amount is optional from the browser.

You can reference or download the following JavaScript plugin:

`https://xiecomm.paymetric.com/DIeComm/Scripts/XIPlugin/XIPlugin-1.2.0.js`

Include the Paymetric JavaScript and reference the Submit Example below to handle the posting of the credit card details directly to Intercept eComm.

QA JS Submit example

```
  $XIPlugin.ajax({

      url: '<% Intercept Url%>' + "/Cardinal3DSWithToken",
      type: "POST", //<GET/POST>
      data:  {Formatted JS Data – See above example myData},
      success: function(a)
                 {
                 //function triggered on success.
                 },
      error: function(a)
                 {
                 //function triggered on failure.
                 },
```

QA JS Submit example

```
     beforeSend: function(a)
               {
               //function triggered before ajax call.
               },
     complete: function(a)
               {
               //function triggered after ajax call.
               // After success or failure
               }

     validation: function(a)
               {
               //Run custom validation
               return true; //should return true or false.

               // return true will go through with ajax call.
               // return false will not go through with ajax
        call.
               }


       });
```

QA JS Submit 3DS 2 example

```
  $XIPlugin.ajax({


     url: '<% Intercept Url%>' + "/Cardinal3DSWithToken",
     type: "POST", //<GET/POST>
     data:  {Formatted JS Data – See above example myData},
     success: function(a)
               {
               //function triggered on success.
               },
     error: function(a)
               {
               //function triggered on failure.
               },
     beforeSend: function(a)
               {
```

QA JS Submit 3DS 2 example

```
               //function triggered before ajax call.
               },
     complete: function(a)
               {
               //function triggered after ajax call.
               // After success or failure
               }

     validation: function(a)
                {
                //Run custom validation
                return true; //should return true or false.

                // return true will go through with ajax call.
                // return false will not go through with ajax
          call.
                },
     threeDSVersion: "2.2.0"



        });
```

11.3 3DS Token: Get Response packet

Request method

A direct server-side GET is required to retrieve the Response Packet. How this GET is performed depends on the language you are using.

Content type

The content type must be "application/x-www-form-urlencoded".

Endpoints

For the latest endpoints and URLs, see the Web Service and Resource URLs topic in Paymetric Assist.

Cert

```
https://cert-xiecomm.paymetric.com/DIeComm/ResponsePacket
```

Production

```
https://xiecomm.paymetric.com/DIeComm/ResponsePacket
```

GET Call parameters

```
MerchantGUID={MerchGUID}&Signature={Signature}
&AccessToken={AccessToken}
```

Parameter	Description
MerchantGuid	The Global Unique Identifier assigned.
Signature	Sign the Access Token; encode it with HMAC-SHA-256 using the shared key then covert it to a base 64 string. The Signature must be URL encoded.
AccessToken	A string that is dynamically generated by Intercept eComm during the first application server call. The call includes the Merchant GUID and the Intercept eComm Request Packet that is signed by the Shared Key.

Intercept eComm is used to integrate the AliPay alternative payment method through the Cardinal Commerce platform.

Requirements

- AliPay workflow requires Cardinal Commerce Credentials.
    - You will see AliPay listed on the Intercept eComm page in Merchant Portal if your Merchant Profile is properly configured. You choose the edit option to enter your Cardinal Credentials.
    - The same credentials are used for both AliPay and 3DS workflows. If you already have credentials entered in Merchant Portal for 3DS, you do not need to re-enter them; you will just now see AliPay listed on the Intercept eComm page as well .
- AliPay configuration also requires a Notification URL to which Intercept eComm will post the new AccessToken should an AliPay transaction result in a Pending status. See workflow below for more information.
    - For additional Onboarding information related to AliPay, search on "alipay" in the Merchant Portal help or Paymetric Assist sites.

Additional information

- The reservation and settlement of funds occurs during the Intercept eComm call through Cardinal to AliPay.
- The authorization and settlement calls to XiPay Cardinal Centinel Cartridge are for reporting purposes; a call is not made out to AliPay at this time since the funds have already been moved, unless a credit is being performed.

- Credits are not performed through Intercept eComm; they are submitted directly through XiPay via the Authorization and Settlement operations with a negative amount. Cardinal requires a reference to the original transaction being credited. The original OrderId should be submitted in the CardNumber field.

AliPay high-level integration steps

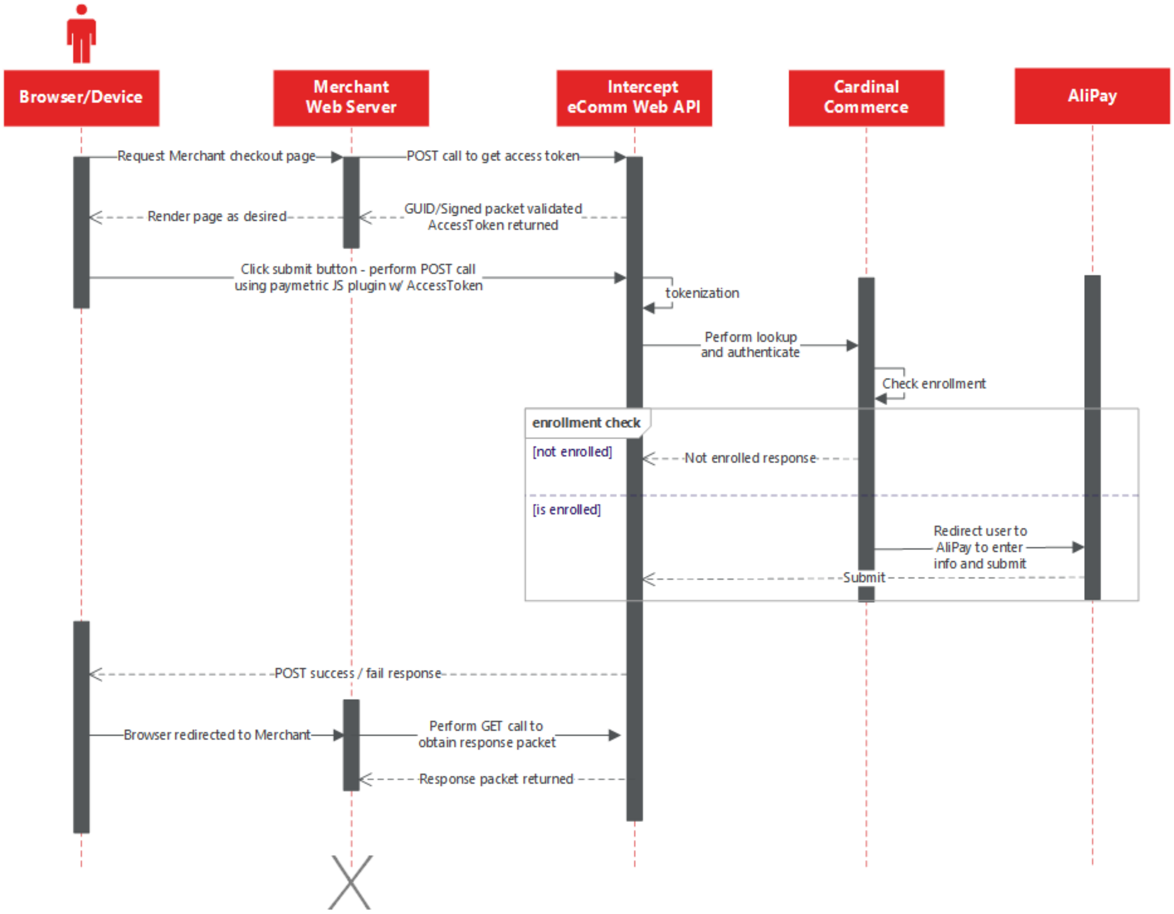
1. Initiate Intercept eComm session with a POST call to get the access token upon page load then render page as desired.

2. Upon user submit, perform POST call using the JS plugin.

3. Perform GET call to get the response packet with token. If implementing 3DS, all response fields from Cardinal Commerce will be included as well.

The following sequence diagram illustrates the Form Post integration flow. Review the subsections below for more details.

12 AliPay integration


Browser/Device
Merchant Web Server
Intercept eComm Web API
Cardinal Commerce
AliPay
Request Merchant checkout page
POST call to get access token
Render page as desired
GUID/Signed packet validated AccessToken returned
Click submit button - perform POST call using paymetric JS plugin w/ AccessToken
tokenization
Perform lookup and authenticate
Check enrollment
enrollment check
[not enrolled]
Not enrolled response
[is enrolled]
Redirect user to AliPay to enter info and submit
Submit
POST success / fail response
Browser redirected to Merchant
Perform GET call to obtain response packet
Response packet returned

Pending status

If the status from AliPay is Pending when the Response Packet is returned, the following occurs:

1. AliPay initiates a notification through CardinalCommerce when approved.
2. Intercept eComm and Cardinal perform the status request and verification.
3. Intercept eComm creates new AccessToken and posts to Notification URL defined in configuration in Merchant Portal.

4. Merchant performs a server-side call to get the Response Packet using new AccessToken.

12.1 AliPay: Get AccessToken

AccessToken is required to initiate an Intercept eComm session.

AccessToken is a string that is dynamically generated by Intercept eComm during the first application server call. The call includes the Merchant GUID and the Intercept eComm Request Packet that is signed by the Shared Key.

Request method

A direct server-side POST is required to retrieve the Access Token. How this POST is performed depends on the language you are using.

Content type

The content type must be "application/x-www-form-urlencoded".

Endpoints

For the latest endpoints and URLs, see the Web Service and Resource URLs topic in Paymetric Assist.

Cert

```
https://cert-xiecomm.paymetric.com/DIeComm/AccessToken
```

Production

```
https://xiecomm.paymetric.com/DIeComm/AccessToken
```

POST Call parameters

Following is the POST data:

```
MerchantGuid={Your-GUID}&SessionRequestType={Int-Type-Code}
&Signature={Signature}&MerchantDevelopmentEnvironment={code language e.g.
asp.net}&Packet={PACKET_XML}
```

Parameter	Description
MerchantGuid	The Global Unique Identifier assigned.
SessionRequestType	Indicates the integration type. Use 7 for AliPay.
Signature	Sign the Packet XML; encode it with HMAC-SHA-256 using the shared key then convert it to a base 64 string. The Signature should be URL encoded.
MerchantDevelopment Environment	Indicate the language in which you are coding your integration to Intercept eComm, for example, asp.net or php.
Packet XML	The Packet XML should be URL encoded. Following are the elements: Required: • RedirectUri = the redirect URL from AliPay to Merchant Optional: (see the Cardinal Centinel data mapping for more details on optional fields) • Amount = amount is optional to support partial payments (allow for client-side ability to change the amount)

Parameter	Description
	• Tax Amount • ShippingAmount • GiftCardAmount • CurrencyCode • OrderNumber • OrderDescription • MerchantData • OrderChannel • ProductCode • TransactionMode • CategoryCode • RecurringAmount • Email • BillingFirstName • Items (can have multiple instances of Items -- see AliPay XML example) • Item_Name

Parameter	Description
	• Item_Desc • Item_Price • Item_Quantity • Item_SKU See samples below.
customerRedirectUri	Applicable only for the Hosted page when implementing XiXiSecurelink. Defines the Merchant redirect URL after the User clicks submit.

AliPay XML sample

```
<?xml version="1.0"?>
<AliPayPostPacketModel>
  <RedirectUri>{http://Yourdomain.here}</RedirectUri>
  <Amount>20.00</Amount>
  <TaxAmount>1.65</TaxAmount>
  <Items>
     <Item_Desc>Blue fuzzy socks</Item_Desc>
     <Item_Name>Womens Socks</Item_Name>
     <Item_Price>2.00</Item_Price>
     <Item_Quantity>5</Item_Quantity>
     <Item_SKU>123</Item_SKU>
  </Items>
  <Items>
     <Item_Desc>Red fuzzy socks</Item_Desc>
     <Item_Name>Womens Socks</Item_Name>
     <Item_Price>2.00</Item_Price>
     <Item_Quantity>5</Item_Quantity>
     <Item_SKU>122</Item_SKU>
  </Items>
</AliPayPostPacketModel>
```

12.2 AliPay: On-Submit Javascript plugin

When Customer performs the submit action (e.g. Pay Now button), a POST call should be made to XiI-eComm using a JavaScript (JS) plugin provided by Paymetric.

You can reference or download the following JavaScript plugin:

```
https://xiecomm.paymetric.com/DIeComm/Scripts/AliPayPlugin/AliPayPlugin-
1.0.0.js
```

Include the Paymetric JavaScript and reference the Submit Example below to handle the posting of the credit card details directly to Intercept eComm.

QA JS Submit example

```
< script src = "https://cert-
xiecomm.paymetric.com/DIeComm/Form/Scripts/AliPayPlugin/AliPay-1.0.0.js"
> < /script>

…
var diecommUrl = 'https://cert-
xiecomm.paymetric.com/DIeComm/Form/AliPay';
$AliPlugin.submit({
    url: diecommUrl,
    data: myData,
    validation: function() {
        return true;
    }
});
```

The appropriate lookup and verification calls are made to check enrollment.

If enrolled, redirect to AliPay, User enters AliPay information and submits payment. Intercept eComm and CardinalCommerce make the authenticate and verification calls.

12.3 AliPay: Get Response packet

Request method

A direct server-side GET is required to retrieve the Response Packet. How this GET is performed depends on the language you are using.

Content type

The content type must be "application/x-www-form-urlencoded''.

Endpoints

For the latest endpoints and URLs, see the Web Service and Resource URLs topic in Paymetric Assist.

Cert

```
https://cert-xiecomm.paymetric.com/DIeComm/ResponsePacket
```

Production

```
https://xiecomm.paymetric.com/DIeComm/ResponsePacket
```

GET Call parameters

```
MerchantGUID=<MerchGUID>&Signature=<SIGNATURE&AccessToken=<AccessToken>
```

Parameter	Description
MerchantGuid	The Global Unique Identifier assigned.
Signature	Sign the Access Token; encode it with HMAC-SHA-256 using the shared key then covert it to a base 64 string.

Parameter	Description
	The Signature should be URL encoded.
AccessToken	A string that is dynamically generated by Intercept eComm during the first application server call. The call includes the Merchant GUID and the Intercept eComm Request Packet that is signed by the Shared Key.

Pending status

If the status from AliPay is Pending when the Response Packet is returned, the following occurs:

1. AliPay initiates a notification through CardinalCommerce when approved.
2. Intercept eComm and Cardinal perform the status request and verification.
3. Intercept eComm creates new AccessToken and posts to Notification URL defined in configuration in Merchant Portal.
4. Merchant performs a server-side call to get the Response Packet using new AccessToken.

Intercept eComm can be used to keep the raw credit number from entering the Merchant's website and environment when using the Visa Checkout alternative payment method.

Visa Checkout with Intercept highlights

- Visa Checkout can be added as an alternative payment with either the Form Post, Ajax or Iframe integration type.
- Merchants provide their customers with a Visa checkout button. When their customer clicks the button to make a payment, a unique 'callid' for the transaction is generated by Visa and given back to the Merchant. Merchant provides the unique 'callid' to allow Paymetric to interact with Visa on their behalf for the transaction.
- Intercept eComm Visa Checkout WebAPI Controller uses the incoming 'callid' sent by the merchant and makes another GetPaymentData call on behalf of the merchant requesting the FULL data packet to obtain all of the user information including the raw card number.
- Paymetric tokenizes the card number so that it never reaches the Merchant's environment nor website. The token along with all of the customer information is passed back to the merchant in the same format as the Content Media-Type in the request packet, either "application/xml" or "application/json".

13.1 Visa Checkout requirements

- Merchant GUID (global unique identifier used to authenticate to the Intercept eComm platform)

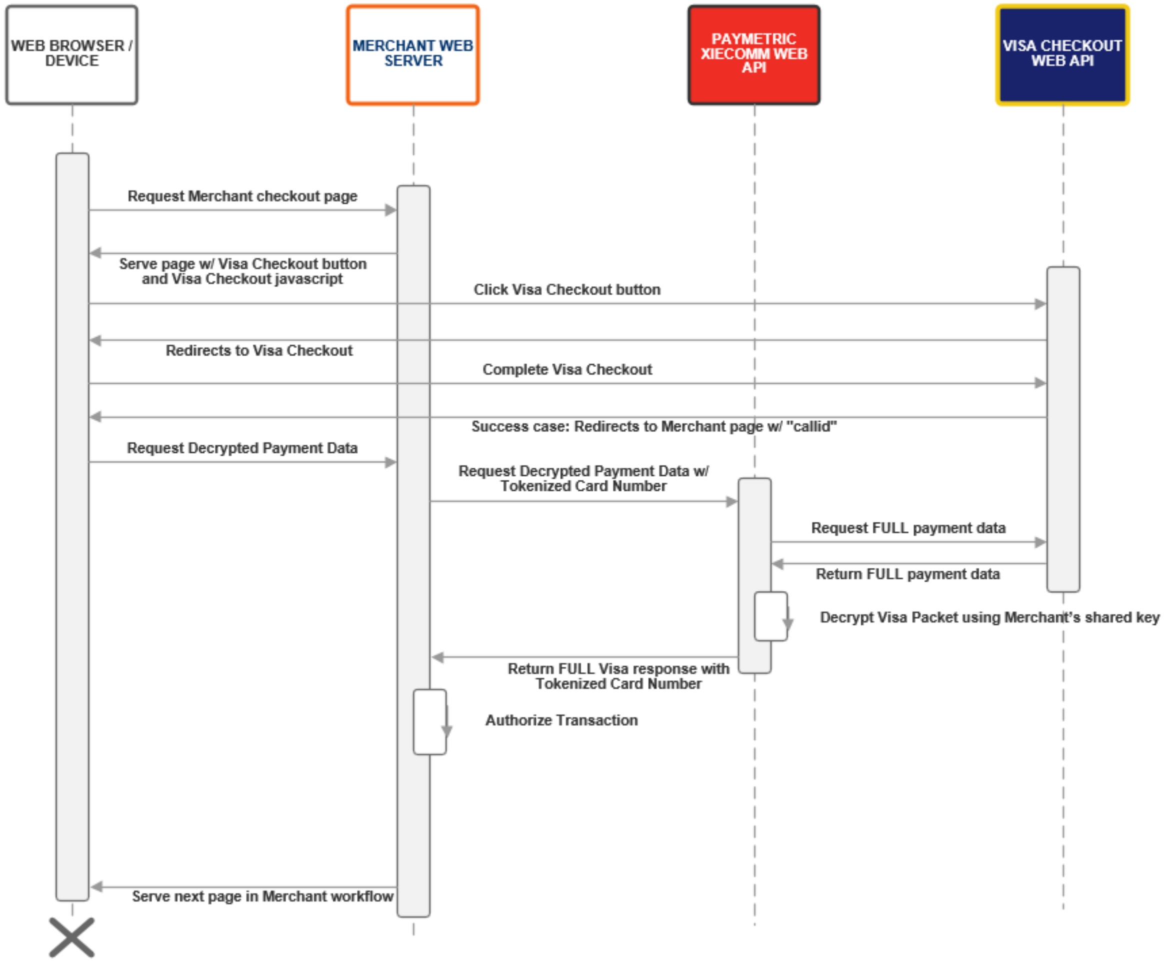
- Shared Key (a character string used for encoding the Intercept eComm Request and Response packets as well as confirming the signature of both packets)
- Visa API Key (a character string provided by Visa to authenticate to their platform)
- Visa Shared Secret (character string associated with the API Key provided by Visa)
- Visa Checkout alternative payment method must be implemented with one of the following integration types: Form Post, Ajax or Iframe

13.2 Visa Checkout integration workflow

An AccessToken is required to initiate an Intercept eComm session. The AccessToken is referenced in the Form Post, Ajax, and Iframe integration type workflows. A server-side call is made to get the Access Token.

A. Implement chosen standard integration type

Before the Visa Checkout workflow is applicable, you will implement either the Form Post, Ajax, or Iframe integration type for standard payments. After obtaining the AccessToken, the workflow will be as follows if the User chooses Visa Checkout as the payment method.


WEB BROWSER / DEVICE
MERCHANT WEB SERVER
PAYMETRIC XIECOMM WEB API
VISA CHECKOUT WEB API
Request Merchant checkout page
Serve page w/ Visa Checkout button and Visa Checkout javascript
Click Visa Checkout button
Redirects to Visa Checkout
Complete Visa Checkout
Success case: Redirects to Merchant page w/ "callid"
Request Decrypted Payment Data
Request Decrypted Payment Data w/ Tokenized Card Number
Request FULL payment data
Return FULL payment data
Decrypt Visa Packet using Merchant's shared key
Return FULL Visa response with Tokenized Card Number
Authorize Transaction
Serve next page in Merchant workflow

B. Perform Get Callid Directly from Visa

Refer to Visa's documentation for more information on implementing the Visa Checkout button and obtaining the 'callid' (https://arch.developer.visa.com/vme/merchant/documents/Visa_Checkout_JavaScript_Integration_Guide/Get_Payment_Data.html). Following are the high-level steps:

1. User accesses Merchant’s checkout page or the payment selection page

2. Merchant provides the user with the Visa Checkout button with the dataLevel query parameter set to 'Summary' ("datalevel=Summary")
3. User clicks on the Visa Checkout button and logs into his/her Visa checkout account
4. On Success, Visa sends back basic info to the Merchant in JSON which includes a ‘callid’. This is the ‘callid’ that merchant will provide to Paymetric when getting the Payment Data.

C. Perform Get Payment Data Call to Paymetric

13.3 Visa Checkout status codes

Following are the potential status codes returned in the response packet for the Visa Checkout workflow.

Code	Description
Global Intercept eComm status codes	
1	Passed
100	Internal Intercept eComm error
101	Merchant GUID invalid
102	Signature validation failed

Code	Description
104	Intercept eComm session expired
120	XiSecure error
Visa Checkout specific status codes	
701	Missing or invalid request format
702	Incorrect request data received
703	Missing or invalid signature in the request
704	Missing or invalid TimeStamp in the request
705	Missing or invalid CallId in the request
706	Missing AccessToken in the request
707	Incoming request took too long to reach
708	Error communicating with VisaCheckout service

13.4 Visa Checkout sample responses

Get Payment Data successful responses

XML successful response

```
<?xml version="1.0"?>
<XiVCGetPaymentResponse>
   <MerchantGuid>6fcc767a-890d-407a-87f5-898368fe5171</MerchantGuid>
   <Signature>DbraXtLPcKCRfwk1nAks01M8CUwvhroKg84ggZnDfqY=</Signature>
   <StatusCode>1</StatusCode>
   <TimeStamp>2016-08-25T20:46:56Z</TimeStamp>
   <VisaResponse>
       <ns2:getPaymentDataResponse
xmlns:ns4="http://www.visa.com/vme/walletservices/internal/xo"
xmlns:ns3="http://www.visa.com/vme/walletservices/internal/common"
xmlns:ns2="http://www.visa.com/vme/walletservices/external/payment"
xmlns="http://www.visa.com/vme/walletservices/external/common">
           <partialShippingAddress>
               <countryCode>US</countryCode>
               <postalCode>78704</postalCode>
           </partialShippingAddress>
           <ns2:paymentRequest>
               <currencyCode>USD</currencyCode>
               <total>21</total>
           </ns2:paymentRequest>
           <ns2:userData>
               <ns2:userFirstName>John</ns2:userFirstName>
               <ns2:userLastName>N</ns2:userLastName>
               <ns2:userFullName>John N</ns2:userFullName>
               <ns2:userName>rnigam@worldpay.com</ns2:userName>
               <ns2:userEmail>rnigam@worldpay.com</ns2:userEmail>
               <ns2:encUserId>N7KP3+fBSpiv4fthE/zxij9odHSiUdj3TwDb2wyU9
98=</ns2:encUserId>
           </ns2:userData>
           <ns2:creationTimeStamp>2016-07-
20T19:38:35.494Z</ns2:creationTimeStamp>
           <ns2:paymentInstrument>
               <nickName>PMTest Visa</nickName>
               <nameOnCard>John N</nameOnCard>
               <cardFirstName>John</cardFirstName>
               <cardLastName>N</cardLastName>
               <expirationDate>
                   <month>09</month>
                   <year>2020</year>
               </expirationDate>
```

XML successful response

```
               <paymentType>
                   <cardBrand>VISA</cardBrand>
                   <cardType>CREDIT</cardType>
               </paymentType>
               <id>pie23+5NG0/UgHyijjl16BaSSI4rltlm0pBhufM1m48=</id>
               <lastFourDigits>4242</lastFourDigits>
               <binSixDigits>424242</binSixDigits>
               <accountNumber>-E803-4242-J54PQNWFGQJ73B</accountNumber>
               <billingAddress>
                   <personName>John N</personName>
                   <firstName>John</firstName>
                   <lastName>N</lastName>
                   <line1>South Congress Avenue</line1>
                   <city>Austin</city>
                   <stateProvinceCode>TX</stateProvinceCode>
                   <postalCode>78704</postalCode>
                   <countryCode>US</countryCode>
                   <phone>5122377520</phone>
                   <default>false</default>
               </billingAddress>
               <verificationStatus>VERIFIED</verificationStatus>
               <expired>false</expired>
               <cardArts>
                   <cardArt>
                       <baseImageFileName>https://sandbox.secure.checko
ut.visa.com/VmeCardArts/wv87HR3X3jqlNXNJu_6YtLQyyO7mpu2aU6Yo3VWGKKM.png
</baseImageFileName>
                       <height>105</height>
                       <width>164</width>
                   </cardArt>
               </cardArts>
               <issuerBid>null</issuerBid>
           </ns2:paymentInstrument>
           <ns2:shippingAddress>
               <personName>John N</personName>
               <firstName>John</firstName>
               <lastName>N</lastName>
               <line1>South Congress Avenue</line1>
               <city>Austin</city>
               <stateProvinceCode>TX</stateProvinceCode>
               <postalCode>78704</postalCode>
               <countryCode>US</countryCode>
               <phone>5122377520</phone>
               <default>false</default>
               <id>na/woORRCpJUGijhyAEsFMnoU7A8nYkZDNzPMJsXYMI=</id>
```

XML successful response

```
               <verificationStatus>CONSUMER_OVERRIDE</verificationStatu
s>
           </ns2:shippingAddress>
           <ns2:riskData>
               <advice>UNAVAILABLE</advice>
               <score>0</score>
               <avsResponseCode>0</avsResponseCode>
               <cvvResponseCode>0</cvvResponseCode>
               <ageOfAccount>57</ageOfAccount>
           </ns2:riskData>
           <ns2:visaCheckoutGuest>false</ns2:visaCheckoutGuest>
       </ns2:getPaymentDataResponse>
   </VisaResponse>
</XiVCGetPaymentResponse>
```

JSON successful response

```
{
    "XiVCGetPaymentResponse": {
        "MerchantGuid": "6fcc767a-890d-407a-87f5-898368fe5171",
        "Signature": "DbraXtLPcKCRfwk1nAks01M8CUwvhroKg84ggZnDfqY=",
        "StatusCode": "1",
        "TimeStamp": "2016-08-25T20:53:02Z",
        "VisaResponse": {
            "paymentRequest": {
                "currencyCode": "USD",
                "total": "21"
            },
            "userData": {
                "userFirstName": "John",
                "userLastName": "N",
                "userFullName": "John N",
                "userName": "rnigam@worldpay.com",
                "userEmail": "rnigam@worldpay.com",
                "encUserId":
"N7KP3+fBSpiv4fthE/zxij9odHSiUdj3TwDb2wyU998="
            },
            "creationTimeStamp": 1469043515494,
            "paymentInstrument": {
                "id": "pie23+5NG0/UgHyijjl16BaSSI4rltlm0pBhufM1m48=",
                "lastFourDigits": "4242",
                "binSixDigits": "424242",
```

JSON successful response

```
                "accountNumber": "-E803-4242-J54PQNWFGQJ73B",
                "billingAddress": {
                    "personName": "John N",
                    "firstName": "John",
                    "lastName": "N",
                    "line1": "South Congress Avenue",
                    "city": "Austin",
                    "stateProvinceCode": "TX",
                    "postalCode": "78704",
                    "countryCode": "US",
                    "phone": "5122377520",
                    "default": false
                },
                "verificationStatus": "VERIFIED",
                "expired": false,
                "cardArts": {
                    "cardArt": [
                        {
                            "baseImageFileName":
"https://sandbox.secure.checkout.visa.com/VmeCardArts/wv87HR3X3jqlNXNJu
_6YtLQyyO7mpu2aU6Yo3VWGKKM.png",
                            "height": 105,
                            "width": 164
                        }
                    ]
                },
                "issuerBid": "null",
                "nickName": "PMTest Visa ",
                "nameOnCard": "John N",
                "cardFirstName": "John",
                "cardLastName": "N",
                "expirationDate": {
                    "month": "09",
                    "year": "2020"
                },
                "paymentType": {
                    "cardBrand": "VISA",
                    "cardType": "CREDIT"
                }
            },
            "shippingAddress": {
                "id": "na/woORRCpJUGijhyAEsFMnoU7A8nYkZDNzPMJsXYMI=",
                "verificationStatus": "CONSUMER_OVERRIDE",
                "personName": "John N",
                "firstName": "John",
                "lastName": "N",
```

JSON successful response

```
                "line1": "South Congress Avenue",
                "city": "Austin",
                "stateProvinceCode": "TX",
                "postalCode": "78704",
                "countryCode": "US",
                "phone": "5122377520",
                "default": false
            },
            "riskData": {
                "advice": "UNAVAILABLE",
                "score": 0,
                "avsResponseCode": "0",
                "cvvResponseCode": "0",
                "ageOfAccount": 57
            },
            "visaCheckoutGuest": false,
            "partialShippingAddress": {
                "countryCode": "US",
                "postalCode": "78704"
            }
        }
    }
}
```

GetPaymentData failure response from Visa

The following responses will occur if the call to Paymetric was successful but there is an error returned from Visa.

XML failure response from Visa

```
<?xml version="1.0"?>
<XiVCGetPaymentResponse>
   <MerchantGuid>6fcc767a-890d-407a-87f5-898368fe5171</MerchantGuid>
   <Signature>kXW7Jan19MH3r3eEuqlkYZ6zKoKNXPxH1ug5DYvtzTc=</Signature>
   <StatusCode>1</StatusCode>
   <TimeStamp>2016-08-25T20:54:36Z</TimeStamp>
   <VisaResponse>
```

XML failure response from Visa

```
       <ns2:getPaymentDataResponse
xmlns:ns4="http://www.visa.com/vme/walletservices/internal/xo"
xmlns:ns3="http://www.visa.com/vme/walletservices/internal/common"
xmlns:ns2="http://www.visa.com/vme/walletservices/external/payment"
xmlns="http://www.visa.com/vme/walletservices/external/common">
           <responseStatus>
               <status>404</status>
               <code>1010</code>
               <severity>ERROR</severity>
               <message>Expected resource is not available. Intent not
found.</message>
           </responseStatus>
       </ns2:getPaymentDataResponse>
   </VisaResponse>
</XiVCGetPaymentResponse>
```

JSON failure response from Visa

```
{
    "XiVCGetPaymentResponse": {
        "MerchantGuid": "6fcc767a-890d-407a-87f5-898368fe5171",
        "Signature": "kXW7Jan19MH3r3eEuqlkYZ6zKoKNXPxH1ug5DYvtzTc=",
        "StatusCode": "1",
        "TimeStamp": "2016-08-25T20:53:46Z",
        "VisaResponse": {
            "responseStatus": {
                "status": 404,
                "code": 1010,
                "severity": "ERROR",
                "message": "Expected resource is not available. Intent
not found."
            }
        }
    }
}
```

GetPaymentData Failure from Paymetric

The following responses will occur if there is a failure by Paymetric before it is sent Visa.

XML failure response from Paymetric

```
<?xml version="1.0"?>
<XiVCGetPaymentResponse>
   <MerchantGuid>6fcc767a-890d-407a-87f5-898368fe5171</MerchantGuid>
   <Signature />
   <StatusCode>104</StatusCode>
   <TimeStamp>2016-08-25T20:55:11Z</TimeStamp>
</XiVCGetPaymentResponse>
```

JSON failure response from Paymetric

```
{
    "XiVCGetPaymentResponse": {
        "MerchantGuid": "6fcc767a-890d-407a-87f5-898368fe5171",
        "Signature": "",
        "StatusCode": "104",
        "TimeStamp": "2016-08-25T20:55:37Z"
    }
}
```

13.5 Visa Checkout HMAC-SHA256 C# sample

Following is the basic HMAC + SHA256 sample used to generate a hash of a string input using a key:

```
public static string GetHashHmacSha256(string text, string key)
      {
          HMAC hashAlg = HMAC.Create("HMACSHA256");
          hashAlg.Key = System.Text.Encoding.Default.GetBytes(key);
          byte[] bytes = Encoding.ASCII.GetBytes(text);
          byte[] hash = hashAlg.ComputeHash(bytes);
          return Convert.ToBase64String(hash);
      }
```

You can add additional logic to handle error scenarios.

Intercept eComm supports PCI 4.0 SAQ-A workflows that still require the submission of the CVV during an authorization.

To do this, Intercept eComm has an endpoint that allows you to submit the authorization without a CVV populated. Intercept eComm then populates the CVV from the response packet into the authorization before submitting it to XiPay. This is supported for every integration type, including 3DS.

High level process

The PCI 4.0 SAQ-A workflow is detailed at a high-level below:

1. Initiate Intercept eComm session with a POST call to get the access token.
2. Merchant renders page as desired and submits according to the integration type.
3. Perform GET call to get the response packet with token.
    - The CVV will not be returned in this response if you have **PCI 4.0 SAQ-A** support enabled.
4. Create XML for authorization to XiPay.
    - This XML does not need to change, but the CVV will not be available to populate.
5. Submit authorization XML to the Intercept eComm PerformAuthorization endpoint using a POST call with merchant GUID and access token.
    - Intercept eComm will populate the CVV in the authorization XML if CVV was submitted in step 2.
6. PerformAuthorization endpoint responds with the XiPay authorization response.

- The CVV will be removed from the XiPay response if it was originally returned to comply with PCI 4.0 SAQ-A.

14.1 Form Post: Perform authorization

Request method

A direct server-side POST is required to submit the authorization. How this POST is performed depends on the language you are using.

Content type

The content type must be "application/xml" or "text/xml".

Endpoints

Cert

```
https://cert-
xiecomm.paymetric.com/DIeComm/XiPay/PerformAuthorization/<MerchantGui
d>/<AccessToken>
```

Production

```
https://xiecomm.paymetric.com/DIeComm/XiPay/PerformAuthorization/<Mer
chantGuid>/<AccessToken>
```

POST body

The body should contain the entire XiPay XML to be submitted for authorization.

Example URL

```
https://cert-xiecomm.paymetric.com/
DIeComm/XiPay/PerformAuthorization/81175969-3FFA-49C0-A246-
493A8854E3C9/188192f9-fb74-454b-aafd-7989c813584e
```

Example POST body

```
<?xml version="1.0" encoding="utf-8"?>

<SOAP-ENV:Envelope xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:wsse="http://docs
.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-
1.0.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:ns1="http:
//Paymetric/XiPaySoap30" xmlns:ns2="http://Paymetric/XiPaySoap30/messag
e/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">

    <SOAP-ENV:Body>
        <ns2:SoapOp>
            <ns2:pPacketsIn>
                <ns2:count>1</ns2:count>
                <ns2:xipayvbresult>false</ns2:xipayvbresult>
                <ns2:packets>
                    <ns2:ITransactionHeader>
                        <ns2:PacketOperation>1</ns2:PacketOperation>
                        <ns2:MerchantID>PMSalem</ns2:MerchantID>

<ns2:CardNumber>CardNumberValue</ns2:CardNumber>
                        <ns2:CardDataSource>E</ns2:CardDataSource>
                        <ns2:CardPresent>0</ns2:CardPresent>
                        <ns2:CardType>VI</ns2:CardType>
                        <ns2:CardCVV2>555</ns2:CardCVV2>
                        <ns2:Client>300</ns2:Client>

<ns2:CardExpirationDate>08/26</ns2:CardExpirationDate>

<ns2:CardHolderName>Worlpay Test</ns2:CardHolderName>
                        <ns2:Amount>123</ns2:Amount>
                        <ns2:CurrencyKey>USD</ns2:CurrencyKey>
                        <ns2:CardDataSource>E</ns2:CardDataSource>
                        <ns2:TaxLevel1>.01</ns2:TaxLevel1>
                        <ns2:TaxLevel2>.02</ns2:TaxLevel2>
                        <ns2:TaxLevel3>.03</ns2:TaxLevel3>
```

Example POST body

```
                        <ns2:TaxLevel4>.04</ns2:TaxLevel4>
                        <ns2:AdditionalInfo>X</ns2:AdditionalInfo>

<ns2:CardHolderAddress1>123 E Main St</ns2:CardHolderAddress1>

<ns2:CardHolderAddress2>Suite 101</ns2:CardHolderAddress2>
                        <ns2:CardHolderCity>Austin</ns2:CardHolderCity>

<ns2:CardHolderCountry>USA</ns2:CardHolderCountry>
                    </ns2:ITransactionHeader>
                </ns2:packets>
            </ns2:pPacketsIn>
        </ns2:SoapOp>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example response

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:u="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd">
    <s:Header>
        <o:Security s:mustUnderstand="1" xmlns:o="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
            <u:Timestamp u:Id="_0">
                <u:Created>2025-01-31T16:16:35.038Z</u:Created>
                <u:Expires>2025-01-31T16:21:35.038Z</u:Expires>
            </u:Timestamp>
        </o:Security>
    </s:Header>
    <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
        <SoapOpResponse xmlns="http://Paymetric/XiPaySoap30/message/">
            <SoapOpResult>
                <count>1</count>
                <xipayvbresult>true</xipayvbresult>
                <packets>
                    <ITransactionHeader>
                        <AVSAddress />
                        <AVSCode>N2</AVSCode>
                        <AVSZipCode />
                        <AccountingDocNumber />
                        <ActionCode />
                        <AdditionalInfo>X</AdditionalInfo>
```

Example response

```
                        <Amount>123.0000</Amount>
                        <AuthorizationCode>tst555</AuthorizationCode>
                        <AuthorizationDate>2025-01-
31T10:16:34</AuthorizationDate>
                        <AuthorizationReferenceCode />
                        <AuthorizationTime>10:16:0034</AuthorizationTim
e>
                        <AuthorizedThroughCartridge>XiPayCartPTSalem.PT
Salem.1</AuthorizedThroughCartridge>
                        <BankBatchID />
                        <BankSubBatchID />
                        <BankTransactionID />
                        <BatchID />
                        <BillingDate>1899-12-30T00:00:00</BillingDate>
                        <BillingPlanItem />
                        <CaptureDate>1899-12-30T00:00:00</CaptureDate>
                        <CaptureReferenceCode />
                        <CardDataSource>E</CardDataSource>
                        <CardExpirationDate>08/26</CardExpirationDate>
                        <CardFollowOnNumber />
                        <CardHolderAddress1>123 E Main
St</CardHolderAddress1>
                        <CardHolderAddress2>Suite
101</CardHolderAddress2>
                        <CardHolderCity>Austin</CardHolderCity>
                        <CardHolderCountry>USA</CardHolderCountry>
                        <CardHolderDistrict />
                        <CardHolderName1 />
                        <CardHolderName2 />
                        <CardHolderName>Worlpay Test</CardHolderName>
                        <CardHolderState />
                        <CardHolderZip />
                        <CardNumber>CardNumberValue</CardNumber>
                        <CardPresent>0</CardPresent>
                        <CardType>VI</CardType>
                        <CardValidFrom />
                        <ChargeAmount>0.0000</ChargeAmount>
                        <Client>300</Client>
                        <CompanyCode />
                        <CreationDate>2025-01-
31T10:16:34</CreationDate>
                        <CurrencyKey>USD</CurrencyKey>
                        <CustTXN />
                        <CustomerNumber />
                        <FiscalYear />
                        <GLAccount />
```

Example response

```
                        <LastModificationDate>2025-01-
31T10:16:35</LastModificationDate>
                        <LocationID />
                        <MerchantID>PMSalem</MerchantID>
                        <MerchantTXN />
                        <MerchantTransactionID />
                        <Message>100 Approved - Successfully
approved.</Message>
                        <ModifiedStatus>1</ModifiedStatus>
                        <OrderDate>1899-12-30T00:00:00</OrderDate>
                        <OrderID />
                        <Origin>XiPay</Origin>
                        <PONumber />
                        <PacketOperation>1</PacketOperation>
                        <Preauthorized />
                        <ReferenceCode />
                        <ReferenceLineItem />
                        <ResponseCode>100</ResponseCode>
                        <SalesDocNumber />
                        <SettlementAmount>123.0000</SettlementAmount>
                        <SettlementDate>1899-12-
30T00:00:00</SettlementDate>
                        <SettlementReferenceCode />
                        <ShippingCaptureDate>1899-12-
30T00:00:00</ShippingCaptureDate>
                        <ShippingLocationID />
                        <ShippingMethod />
                        <StatusCode>100</StatusCode>
                        <StatusTXN>Authorized</StatusTXN>
                        <TaxLevel1>0.0100</TaxLevel1>
                        <TaxLevel2>0.0200</TaxLevel2>
                        <TaxLevel3>0.0300</TaxLevel3>
                        <TaxLevel4>0.0400</TaxLevel4>
                        <TerminalID />
                        <TransactionID>22222222</TransactionID>
                        <TransactionType>Authorization</TransactionType
>
                        <VATNumberCustomer />
                        <VATNumberMerchant />
                        <XIID>1111111</XIID>
                    </ITransactionHeader>
                </packets>
            </SoapOpResult>
        </SoapOpResponse>
    </s:Body>
</s:Envelope>
```

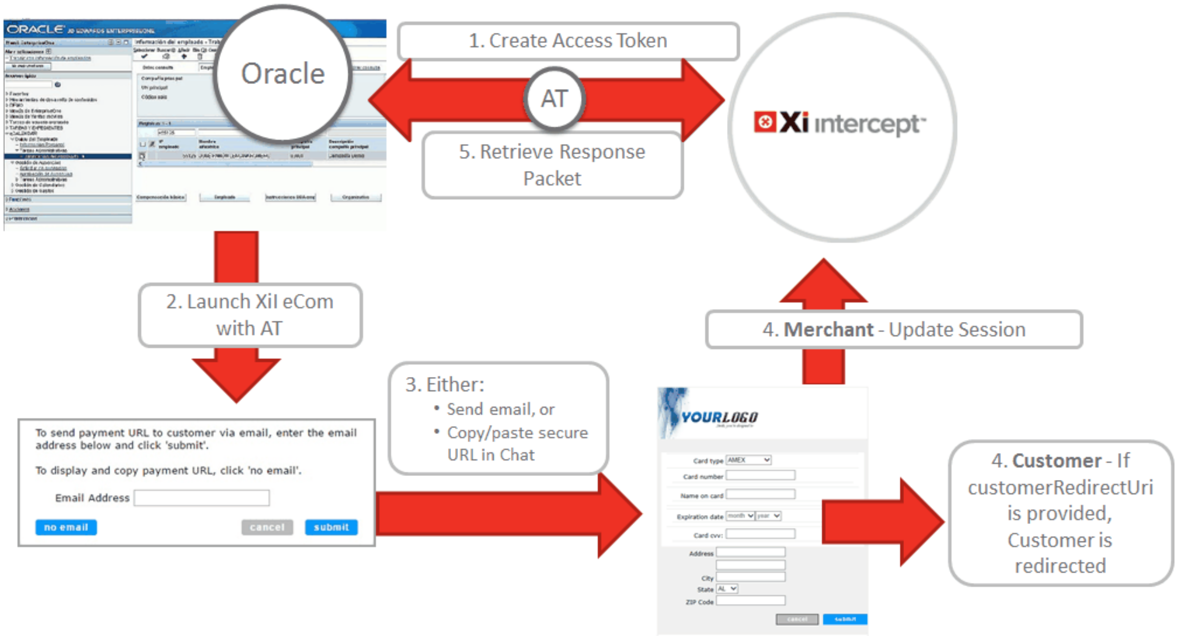
15 XiSecurelink integration

Allows the merchant to send an email that contains a secure link to the Intercept interface, and is hosted in the Paymetric SaaS environment.

The email recipient clicks the link to launch Intercept and then enters the credit card information. The credit card details and a token are returned directly to the ERP system.

The merchant can also get the secure Paymetric hosted URL, and then copy/paste it into a chat window or other secure form of communication with the customer.

For Oracle, other web stores or non-SAP ERP systems, XiSecurelink is integrated by adding the <emailPaymentOption> element in the XML packet for the get access token call that is made to Intercept eComm.


Oracle
1. Create Access Token
AT
5. Retrieve Response Packet
Xi intercept
2. Launch XiI eCom with AT
3. Either:
• Send email, or
• Copy/paste secure URL in Chat
4. Merchant - Update Session
YOURLOGO
Card type AMEX
Card number
Name on card
Expiration date month year
Card cvv:
Address
City
State AL
ZIP Code
cancel
submit
4. Customer - If customerRedirectUri is provided, Customer is redirected

To send payment URL to customer via email, enter the email address below and click 'submit'.

To display and copy payment URL, click 'no email'.

Email Address

no email cancel submit

1 A direct server-side POST is required to retrieve the Access Token. How this Post is performed depends on the language you are using.

Content type must be "application/x-www-form-urlencoded"

2 Intercept for eCommerce Hosted UI is launched in the default browser with the AccessToken and the merchant representative is presented with two options to obtain the payment information:

A. Field to enter email address for the customer

B. A button to generate the secure URL that can be copied

3 Depending upon which option is chosen in Step 2:

A. Email is sent with the link that launches Intercept hosted page. Customer enters payment details.

B. Merchant representative sends URL via chat or other method. Customer enters payment details.

4 The token is obtained.

5 Response Packet is obtained via a server-side GET call back into Oracle.

⚠ **Note:**

Other non-SAP ERP or applications/systems have the same workflow as defined for Oracle.

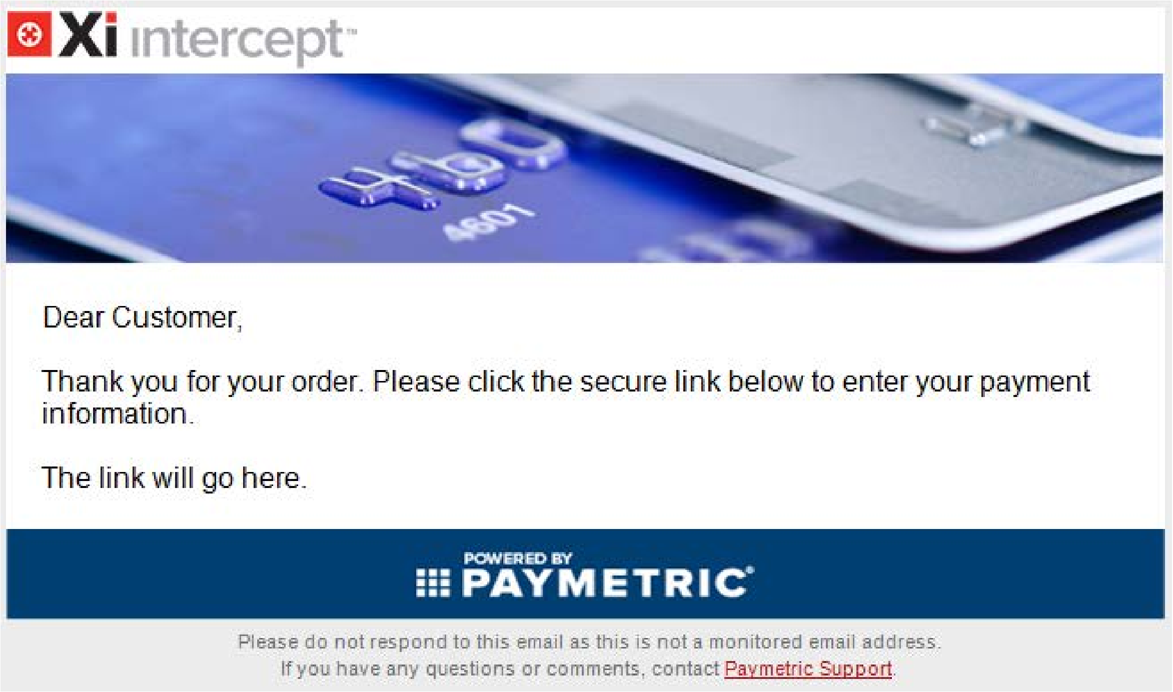
15.1 emailPaymentOption element attributes

The following table defines the available <emailPaymentOption> attributes.

Element	Description
emailTemplate	The text for the email body. Use [BR] to insert a line break as needed. The following two variables are available to add the URL link in the body. Choose which to use depending upon how much control you want over the actual link. • [PAYMENTURL] – Replaced with the appropriate QA or Prod URL, based on which system you are calling. • [PAYMENTLINK] – Replaced with the the following where PAYMENTURL is QA or PROD, based on which system you are calling. `<a href='[PAYMENTURL]'>Enter Payment Information</a>`
merchantLogoURL	Enter the URL for your custom logo. If this attribute is not provided, Intercept logo displays by default.
fromEmailAddress	Defines the from address for the email. Defaults to noreply@worldpay.com.
customerRedirectUri	The URL to which the Customer is redirected once he/she clicks submit after the entering the payment information.

15.2 Default Email text

If you do not define any of the <emailPaymentOption> attributes, a default email will be sent, if you choose the email option.



Dear Customer,

Thank you for your order. Please click the secure link below to enter your payment information.

The link will go here.

Please do not respond to this email as this is not a monitored email address.
If you have any questions or comments, contact Paymetric Support.

15.3 XML packet example with emailPaymentOption element

The endpoint addresses to call XiSecurelink

QA

https://cert-xiecomm.paymetric.com/DIeComm/paymentpage/startemailpayment/<merchantGuid>/<accessToken>

PROD

https://xiecomm.paymetric.com/DIeComm/paymentpage/startemailpayment/<merchantGuid>/<accessToken>

Following is an example of the XML packet with the `emailPaymentOption` element highlighted.

```
<?xml version="1.0" encoding="utf-8"?>
<merchantHtmlPacketModel>
    <paymentPagePacket>
        <redirectUri>http://dh-ecomm-
w2k12.pmdev.local/InterceptDemo/Status.aspx</redirectUri>
        <cssUri>http://dh-ecomm-
w2k12.pmdev.local/DIeComm/Content/PaymentStyleSheet.css</cssUri>
        <emailPaymentOption
emailTemplate="Thank you for your interest.[BR]Please click the
following link to continue.[BR][PAYMENTLINK][BR][BR]If you are
unable to click the link, copy/paste this URL into your browser:
[PAYMENTURL][BR]Regards,[BR]My Company, Inc.[BR] 1-800-555-5555"
merchantLogoUrl="http://img13.mysite.net/images/mylogo.png"
fromEmailAddress="noreply@company.com"
customerRedirectUri="https://mycompany.com" />
    </paymentPagePacket>
    <merchantHtml>
       <htmlSection class="payment-content"
xmlns="Paymetric:Intercept:MerchantHtmlPacketModel">
            <tag name="div" class="merchant-logo"/>
            <tag name="div" class="billing-content">
                <tag name="div" class="billing-header">Billing
Information</tag>
                <tag name="div" class="billing-info">
                  <paymentOptionRadioSection>
                        <tag name="div">
                            <tag name="div">choose payment
type</tag>
                        </tag>
                        <label text="echeck" for="echeck" />
                        <radPaymentOption for="echeck" />
                        <label text="credit card" for="card" />
                        <radPaymentOption for="card" />
                  </paymentOptionRadioSection>
                  <cardDropdownSection>
                        <tag name="div">Enter your credit card
information</tag>
                        <tag name="div">
                            <label for="cardType" text="select a
card" />
                            <ddlCardType id="cd">
                                <items>
                                   <item for="american express" />
                                   <item for="mastercard" />
                                   <item for="visa" />
                                   <customItem for="MSC" text="My
Super Card" />
                                </items>
                            </ddlCardType>
                        </tag>
                        <tag name="div">
                            <label for="cardNumber" text="card
number" />
                            <tboxCardNumber tokenize="true" />
                        </tag>
                        <tag name="div">
                            <label for="cardholderName" text="name
on card" />
                            <tboxCardHolderName />
                        </tag>
                        <tag name="div">
                            <label for="expMonth" text="exp
date" />
                            <ddlExpMonth default-text="month"
class="merchant_combos" />
                            <ddlExpYear default-text="year"
class="merchant_combos" years-to-display="10" />
                        </tag>
                        <tag name="div">
                            <label for="cvv" text="card cvv:" />
                            <tboxCvv />
                            <htmlCvvHelp name="div" raise-
click="1"/>
```

```
                        </tag>
                  </cardDropdownSection>
                  <echeckRadioSection>
                        <tag name="div">
                            <label for="checking" />
                            <radAccountType for="checking" />
                            <label for="savings" />
                            <radAccountType for="savings" />
                        </tag>
                        <tag name="div">
                            <label for="accountNumber"
text="account number" />
                            <tboxAccountNumber tokenize="true" />
                        </tag>
                        <tag name="div">
                            <label for="routingNumber"
text="routing number" />
                            <tboxRoutingNumber />
                        </tag>
                        <tag name="div">
                            <label for="nameOnAccount" text="name
on account" />
                            <tboxNameOnAccount />
                        </tag>
                        <tag name="div">
                            <label for="bankName" text="bank
name" />
                            <tboxBankName />
                        </tag>
                  </echeckRadioSection>
            </tag>
            <tag name="div" class="payment-button">
                  <cancelButton text="cancel" class="cancel-
button" />
                  <submitButton text="submit" class="submit-
button" />
            </tag>
        </tag>
       </htmlSection>
    </merchantHtml>
</merchantHtmlPacketModel>
```

For Iframe and Hosted integration types, there are three main options associated with localization as follows:

- If you do not specify an option, it will default to U.S. English
- Specify to use the Browser locale; if it is a language not supported by Intercept eComm, then it defaults to U.S. English
- Hard code the locale using the Intercept eComm supported languages/culture codes

The following table defines the languages currently supported by Intercept eComm along with the culture codes to use if you are hard coding the localization element.

⚠ **Note:**

When using browser localization, if the browser locale is not in the following list, then Intercept eComm labels and messaging will default to U.S. English, unless hard-coded in custom HTML text or message attributes.

Language	"cultureName" value
Czech	cs
Danish	da
German	de
Spanish (European)	es
Spanish (Latin)	es-es

Language	"cultureName" value
Finnish	fi
French (Native)	fr
French (Canadian)	fr-ca
Hungarian	hu
Italian	it
Japanese	ja
Korean	ko
Dutch	nl
Flemish	nl-be
Norwegian	no
Polish	pl
Portuguese (Portugal)	pt
Portuguese (Brazil)	pt-br
Russian	ru
Slovak	sk
Slovenian	sl

Language	"cultureName" value
Swedish	sv
Turkish	tr
Chinese (Simplified)	zh
Chinese (Traditional)	zh-tw

17 Supported card types and returned values

The following table defines the supported card types and the values returned from Intercept eComm:

Card type	Returned value
Visa	VI
MasterCard	MC
American Express	AX
Diner's	DC
Discover	DI
JCB	JC
Maestro	SW

All other card types would be considered custom cards. The values are set in the XML.

18 Error codes

Error code	Description
-100	Indicates an unexpected error occurred processing the server request
-101	Indicates the web request could not be created
-102	Indicates the attempt to retrieve the response failed
-103	Indicates no response packet was returned from the server
-104	Indicates the response packet was invalid and could not be deserialized
100	Internal Intercept eComm error
101	Merchant GUID is invalid
102	Signature validation failed
103	Request packet is invalid
104	Intercept eComm session has expired
105	Intercept eComm session error
106	Error decrypting data
107	Get AccessToken error

Error code	Description
108	Requested type hasn't been purchased
120	XiSecure error
121	XiSecure no token
122	XiSecure certificate expired UID: {0}, Expiration Date {1}
140	Error accessing database
150	The payment form has been submitted multiple times with the same access token. For example, a user clicks submit payment twice
300	Cardinal Commerce response: {message provided by Cardinal}
301	No Cardinal Commerce configuration
302	The Cardinal Commerce configuration is missing information
401	Attempt was made to load the payment page in an Iframe

19 JavaScript Plugin API reference

Section contents:

19.1 XIPlugin - Form Post and Ajax API reference

The XIPlugin JavaScript plugin is used to reach out to the Intercept eComm service by Ajax or Form POST. XIPlugin can either be referenced directly from the Intercept service or downloaded and then referenced from your server.

Reference or download the following:

`https://xiecomm.paymetric.com/DIeComm/Scripts/XIPlugin/XIPlugin-1.0.0.js`

19.1.1 Formatted JS data structure

To perform the Ajax or Form POST call to Intercept eComm, you must first create your data packet. The XIPlugin can assist in preparing this data packet.

Section contents:

- $XIPlugin.createField(fieldName, tokenize, value) 133
- XiRequestPacket - Ajax and Form Post 134

19.1.1.1 $XIPlugin.createField(fieldName, tokenize, value)

Parameters

`fieldName`
Name of the field associated to the value. This will identify the field in the return packet.

`tokenize`
The Access Token obtained during the Get Access Token call. Boolean value that specifies that this value needs to be tokenized

`value`
The value associated with the field name

Returned from the function

An instance of the XiField data field is returned to add to a data packet.

19.1.1.2 $XIPlugin.createJSRequestPacket(merchantGuid, accessToken)

$XIPlugin.createJSRequestPacket creates an empty data packet (XiRequestPacket).

Parameters

`merchantGuid`
The Merchant GUID generated when the Intercept for eCommerce Onboarding request is processed. Login to Merchant Portal and navigate to Settings | Intercept for eCommerce to obtain the GUID.

`accessToken`
The Access Token obtained during the Get Access Token call.

Returned from the function

An instance of the XiRequestPacket data packet is returned.

19.1.1.3 XiRequestPacket - Ajax and Form Post

The XiRequestPacket is the data packet sent to Ajax or Post methods of XiPlugin.

```
XiRequestPacket.addField
```

This adds an XiField object to the data packet.

19.1.2 Communication method details - Ajax and Form Post

Section contents:

19.1.2.1 $XIPlugin.ajax

```
$XIPlugin.ajax({

          url: '<% Intercept Url%>' + "/Ajax",
          type: "POST", //<GET/POST>
          data: '<% Formatted JS Data – see Packaging above %
>',
          success: function(a)
                   {
                   },
          error: function(a)
                   {
                   },
          beforeSend: function(a)
                   {
                   },
          complete: function(a)
                   {
                   },
         validation: function(a)
                   {
                      return true; //should return true or
                false.
                   }
         intcp3DSecure: function(a)
                   {
                   },

             });
```

Parameters

```
url: '<% Intercept Url%>' + "/Ajax",
```

Define Intercept eComm url and append "/Ajax". Use the appropriate QA or Production URL.

QA: https://cert-xiecomm.paymetric.com/DIeComm

Production: https://xiecomm.paymetric.com/DIeComm

```
type: "POST",
```

Call Type should be POST.

```
data: '<% Formatted JS Data – see Packaging above %>',
```

Set the Data element as defined in the Packaging 132 section above.

```
success: function(a)
```

The function that is triggered if the call is successful.

```
error: function(a)
```

The function that is triggered if the call is NOT successful.

```
beforeSend: function(a)
```

The function that is triggered before the call is sent to Intercept eComm.

```
complete: function(a)
```

The function that is triggered after the Ajax call is complete whether is successful or not .

```
validation: function(a)
      {
          return true; //should return true or false.
      }
```

The function used to perform custom validations. It returns either true false.

If True, will go through with ajax call.

If False, will not go through with ajax call.

```
intcp3DSecure: function(a)
```

Function triggered when a card is enrolled in 3DSecure for customizing Cardinal display in an Iframe or Pop up instead of redirecting to Cardinal site. If this function in not Intercepted then default behavior is invoked in which the site is redirected to Cardinal site.

Function parameters:

- args has necessary argument to post the form to //3DSecure cardinal
- UrlsAndPayloadsFor3Ds.CentinelAcsUrl – form **action**
- UrlsAndPayloadsFor3Ds.CentinelPayload – hidden //input name: **PaReq**
- UrlsAndPayloadsFor3Ds.CentinelTermUrl – hidden //input name: **TermUrl**
- e.UrlsAndPayloadsFor3Ds.PaymetricPayload – hidden //input name: **MD**

19.1.2.2 $XIPlugin.submit

```
$XIPlugin.submit({

               url: '<% Intercept Url%>' + "/Form",
               type: "POST", //<GET/POST>
               data: '<% Formatted JS Data – See Packaging above %>',
               validation: function(a)
                           {
                              return true; //should return true or
                   false.

                           }
                 });
```

Parameters

```
url: '<% Intercept Url%>' + "/Form",
```

Define Intercept eComm url and append "/Form". Use the appropriate QA or Production URL.

QA: https://cert-xiecomm.paymetric.com/DIeComm

Production: https://xiecomm.paymetric.com/DIeComm

```
type: "POST",
```

Call Type should be POST.

```
data: '<% Formatted JS Data – see Packaging above %>',
```

Set the Data element as defined in the Packaging 132 section above.

```
validation: function(a)
      {
        return true; //should return true or false.
    }
```

The function used to perform custom validations. It returns either true false.

If True, will go through with post.

If False, will not go through with post.

19.1.3 Examples

```
var cc = document.getElementById('PaymentCreditCard').value;

var name = document.getElementById('Name').value;
```

//Getting Value from TextBox with JavaScript code

```
var myData = $XIPlugin.createJSRequestPacket(merchantGuid, accessToken);
```

//Create a JavaScript object to be used in Ajax or Form Post with parameter

- MerchantGuid
- AccessToken

```
var myField = $XIPlugin.createField('creditcard', true, cc);
```

//Create a JavaScript object to be added to myData (JSRequestPacket)

- First Param : Field Name (This is up to the merchant. Any possible name)
- Second Param : true (To Tokenize Value)

- Third Param : Value from TextBox

```
myData.addField(myField);
```

//Adding myField to myData (JSRequestPacket) Object

Ajax call

```
$XIPlugin.ajax({

         url: '<% Intercept Url%>' + "/Ajax",
         type: "POST", //<GET/POST>
         data: '<% Formatted JS Data – see above %>',
         success: function(e)
                     {
                       //function triggered on success.
                     },
         error: function(e)
                     {
                       //function triggered on failure.
                     },
         beforeSend: function(e)
                     {
                       //function triggered before ajax call.
                     },
         complete: function(e)
                     {
                       //function triggered after ajax call.
                       // After success or failure
                     },
         validation: function(e)
                     {
                       //Run custom validation
                       return true; //should return true or false.

                        // return true will go through with ajax call.
                        // return false will not go through with ajax call.
                      },
                    intcp3DSecure: function(e)
                      {
//Function triggered when a card is enrolled in 3DSecure for customizing
Cardinal display in an Iframe or Pop up instead of redirecting to Cardinal
site. If this function in not Intercepted then default behavior is invoked
in which site is redirected to Cardinal site.

//e – args has necessary argument to post the form to 3DSecure cardinal.

//e.UrlsAndPayloadsFor3Ds.CentinelAcsUrlàform action
//e.UrlsAndPayloadsFor3Ds.CentinelPayloadàhidden input name: PaReq

//e.UrlsAndPayloadsFor3Ds.CentinelTermUrlàhidden input name: TermUrl

//e.UrlsAndPayloadsFor3Ds.PaymetricPayloadàhidden input name: MD

                      }

            });
```

Form Post call

```
$XIPlugin.submit({

url: '<% Intercept Url%>' + "/Form",
data: '<% Formatted JS Data – See above %>',
validation: function(a)
            {
//Run custom validation
 return true; //should return true or false.

 // return true will go through with post.
 // return false will not go through with post.
}


});
```

19.2 XIFrame - IFrame API reference

This javascript plugin is used for reaching out Intercept service by embedded Iframe in your website. XIFrame can either be referenced directly from Intercept service or downloaded from the Intercept service to your server and then referenced from your server.

Reference or download the following:

https://xiecomm.paymetric.com/DIeComm/Scripts/XIFrame/XIFrame-1.1.0.js

Section contents:

19.2.1 $XIFrame.onload

Resizing of the Iframe can be handled by a stylesheet. Should you need to resize based upon content within the Iframe, include the following javascript function on Iframe onload function.

$XIFrame.onload must be called after the Iframe renders the content from Intercept eComm. It initializes the communication from the parent page to the Iframe.

```
function Iframe_OnLoad(){
      $XIFrame.onload ({
             iFrameId: <% Iframe Name or ID %>,
      targetUrl: <%Intercept Url%>/View/Iframe/<%MerchantGuid%>/<%
 AccessToken%>
      autosizeheight: true,
      autosizewidth: true,
      storeCardAmount: true,
      onCardTypeChange: function (e){},
      onValidate: function (e){},
      onLoadSuccess: function (){},
      onError: function(msg) {},
      onInvalidHandler: function (e){},

      amount: 30.99  //Can pass amount outside of iframe.

      });

 }
```

$XIFrame.onload parameters

`iFrameID`
Name or ID of the Iframe.

`targetUrl`
Name or ID of the Iframe.

`autosizeheight`
Specifies whether to resize the height of the Iframe if the content in the Iframe increases. If this is set to false, the Iframe must be large enough to handle all content even with validation messages.

`autosizewidth`
Specifies whether to resize the width of the Iframe if the content in the Iframe increases. If this is set to false, the Iframe must be large enough to handle all content even with validation messages

`storeCardAmount`
Boolean specifying to return Amount from Iframe to parent window when $XIFrame.getCardAmount() is invoked.

`onCardTypeChange`

Callback function triggered when Card Type is changed amount.

`onValidate`
Function that should be explicitly called to perform any validation defined within the IFramePacket with (e) as the argument if iframe passes or fails validation.

`onLoadSuccess`
Callback function triggered when the Iframe is correctly rendered.

`onError`
Callback function triggered if there are any problems rendering the content of the Iframe.

`onInvalidHandler`
This function results in reports on controls failing any Intecept eComm validations with errors.

`amount`
Used to change default amount passed to 3D Secure for validation. It allows you to pass the amount outside of the Iframe.

Best practice related to Page/Iframe Onload

To ensure the Iframe loads the correct URL with a valid AccessToken, see the following functions. Note that this prevents problems that can occur when a User clicks the back button after the Submit.

This function should reside on the page frontend outside of any other functions:

```
window.onload = function () {
            var frameElement = document.getElementsByName('<%
Iframe Name or ID %>')[0];
            frameElement.contentWindow.location.href =
frameElement.src;
        };
        window.onunload = function () { }
```

This function should reside on the page backend.

```
protected override void OnInit(EventArgs e)
    {
        Response.Cache.SetCacheability(System.Web.HttpCacheability.
NoCache);
        Response.Cache.SetNoStore();
        base.OnInit(e);
    }
```

19.2.2 $XIFrame.submit

```
$XIFrame.submit({
       iFrameId: <% Iframe Name or ID %>,
       targetUrl: <%Intercept Url%>/View/Iframe/<%MerchantGuid%
>/<%AccessToken%>
       onSuccess: function (e) {},
       onError: function (e) {},
        intcp3DSecure: function(e) {},

 });
```

Parameters

`iFrameID`

Name or ID of the Iframe.

`targetUrl`

Name or ID of the Iframe.

`success: function(a)`

The function that is triggered if the call is successful.

`error: function(a)`

The function that is triggered if the call is NOT successful.

19.2.3 Multiple IFrame example

If you are using multiple IFrames, then you will need to create the same number of $XIFrame instances. The following example demonstrates using 2 IFrames to split a payment:

```
var iFrameName1 = 'IFrame1';
var iFrameName2 = 'IFrame2';
var iframe1 = document.getElementsByName(iFrameName1);
var iframe2 = document.getElementsByName(iFrameName2);
```

```
//hold on to the XIFrame instance 1

      var iframe1Plugin = $XIFrame({
           iFrameId: iFrameName1,
           targetUrl: $(iframe1).attr("src"),
           onCardTypeChange: IFrame1CardChange,
           autosizeheight: true,
           autosizewidth: true,
           onError: function(msg) { alert(msg); },
           onValidate: Validation,
           storeCardAmount: true,
           intcp3DSecure: Custom1Handling
         });

//hold on to the XIFrame instance 2
      var iframe2Plugin = $XIFrame({
           iFrameId: iFrameName2,
           targetUrl: $(iframe2).attr("src"),
           autosizeheight: true,
           autosizewidth: true,
           onError: function (msg) { alert(msg); },
           onValidate: Validation,
           storeCardAmount: true,
           intcp3DSecure: Custom2Handling
       });
```

Holding on to the instances assist in making the future call as follows:

1. `iframe1Plugin.onload();  or  iframe1Plugin.submit({onSuccess: function (e) { //trigger on iframe1 submit success }});`
2. `iframe2Plugin.onload();or  iframe2Plugin.submit({onSuccess: function (e) { //trigger on iframe1 submit success }});`

19.2.4 $XIFrame.validate

The main reason to use this function is if you are using multiple IFrames.

```
function submitform() {
      var iframe = document.getElementsByName('dieCommFrame');
      if (iframe) {
          $XIFrame.validate({
              iFrameId: 'dieCommFrame',
              targetUrl: iframe[0].getAttribute("src"),
              autosizeheight: true,
              autosizewidth: true,
              onValidate: function(e) {}

          });
}
```

Parameters

`iFrameID`
Name or ID of the Iframe.

`targetUrl`
Name or ID of the Iframe.

`autosizeheight`
Specifies whether to resize the height of the Iframe if the content in the Iframe increases. If this is set to false, the Iframe must be large enough to handle all content even with validation messages.

`autosizewidth`
Specifies whether to resize the width of the Iframe if the content in the Iframe increases. If this is set to false, the Iframe must be large enough to handle all content even with validation messages

onValidate
Function that should be explicitly called to perform any validation defined within the IFramePacket with (e) as the argument if iframe passes or fails validation.

onError
Callback function triggered if there are any problems rendering the content of the Iframe.

Example using onValidate

```
function submitform() {
            var iframe =
document.getElementsByName('dieCommFrame');
            if (iframe) {
                $XIFrame.validate({
                    iFrameId: 'dieCommFrame',
                    targetUrl: iframe[0].getAttribute("src"),
                    autosizeheight: true,
                    autosizewidth: true,
                    onValidate: function(e) {
                        if (e) {
                            $XIFrame.submit({
                                iFrameId: 'dieCommFrame',
                                targetUrl:
iframe[0].getAttribute("src"),
                                onSuccess: function(msg) {
                                    var message = JSON.parse(msg);
                                    if (message &&
message.data.HasPassed) {
                                        var accessToken =
document.getElementById('<%= AccessToken.ClientID %>');
                                        var signedToken =
document.getElementById('<%= SignedToken.ClientID %>');
                                        window.location =
"Status.aspx?id=" + accessToken.value + "&s=" + signedToken.value;
                                    } else {
```

```
alert(message.data.Message);
                                    }
                                },
                                onError: function(msg) {
                                    alert("Error function : " +
msg);
                                }
                            });
                        }
                   else {
                            alert('Iframe Validation failed');
                        }
                    },
                    onError: function(msg) {
                        alert("Error function : " + msg);
                    }

                });
      }
```

20 Custom HTML reference

Merchants can define custom html layouts for the Intercept eComm **Iframe** and **Hosted** solutions. This is done by creating XML that adheres to the schema **MerchantHtmlPacketModel.xsd**.

20.1 Element: <htmlFieldValues>

The <htmlFieldValues> element is used to define default values for textboxes.

Child elements

- field: applicable to textboxes only such as cardholder name or echeck. This is limited to the textboxes:

  o cardHolderName (credit card)

  o cardAmount (credit card)

  o bankName (echeck)

  o nameOnAccount (echeck)

- additionalField: can be used for any merchant-defined custom textbox

20.2 Element: <merchantHtml>

The XML consists of a main outer container defined by the <merchantHtml> element. Depending on the merchant's needs, this element can contain one or more of the following types of child sections:

- Payment option section – Allows the user to choose between eCheck and Credit Card payment types
- Credit card section – Provides inputs for credit card information
- eCheck section – Provides inputs for eCheck information
- Additional HTML section – Provides inputs for additional data the merchant wants collected, such as address data

Each section above defines elements for labels and inputs to capture data.

Default validations (such as required field, Luhn check for credit card numbers) are defined for the input elements. Additionally, merchants can override these defaults and/or provide custom validations for any input element.

Merchants can make use of common HTML elements such as DIVs to further control the layout of any section, label or input in the xml.

20.2.1 Common elements and attributes

These are descriptions of the common elements and attributes used throughout the schema

20.2.1.1 Common attributes

These are common attributes used throughout the custom html:

- **name**: Unless otherwise specified, this is the name of the html element to generate. Supported values are “div”, “span” and “section”. The default value is “div”.
- **class**: The CSS class for the element.

⚠ **Note:**

The class attribute should be used to style elements. Other element attributes (such as id, name, etc) are internally generated values that are subject to change in the future. The merchant’s CSS should not reference those attributes.

- **for**: Indicates a relationship between elements. For example, a “label’ has a “for” attribute indicating what input it is linked to. An input has a “for” attribute indicating what data the input is for.
- **is-default**: For radio and dropdown inputs, if “1” then this item is the default value.
- **text**: For labels, indicates the text to display. If no text is displayed, the Paymetric default is used for that label. Note this is not currently supported for default text in an input text box.
- **default-text**: For dropdown inputs, indicates the default text to display. For example, in a credit card type dropdown, if “display-text” is “select a card”, then “select a card” is the text in the dropdown when the page loads.
- **tokenize**: Indicates if the data collected in the input should be tokenized. For credit card number and eCheck account number, this must be “true”.

20.2.1.2 Common attributes for validation

In addition to the attributes above, these are attributes for input elements that control validation:

- **error-class**: The CSS class to use on the input element if validation fails

> ⚠ **Note:**
>
> The error-class attribute should be used to style elements. Other element attributes (such as id, name, etc) are internally generated values that are subject to change in the future. The merchant's CSS should not reference those attributes.

- **error-msg-style**: Supported values:
  - **text**: Error message displays as text
  - **imageTooltip**: Error message displays an image with a tooltip
  - **inputTooltip**: Error message displays as a tooltip when hovering over the input
- **digits-only**: If "true", only digits can be entered
- **digits-only-msg**: The error message displays for digits-only validation
- **luhn-check**: If "true", a luhn check is performed. Applies to card numbers only
- **luhn-check-msg**: Error message displays for luhn check validation
- **max**: If "true", maximum numeric value allowed
- **max-msg**: Error message displays for max validation
- **min**: Minimum numeric value allowed

- **min-msg**: Error message displays for min validation
- **maxlength**: Maximum text length allowed
- **maxlength-msg**: Error message displays for maxlength validation
- **minlength**: Minimum text length allowed
- **minlength-msg**: Error message displays for minlength validation
- **number**: If “true”, only a number can be entered
- **number-msg**: Error message for number validation
- **pattern**: Regular expression used to validate the input
- **pattern-msg**: Error message for pattern validation
- **range**: Numeric range for input
- **range-msg**: Error message for range validation
- **rangelength**: Range indicating the length of the input
- **rangelength-msg**: Error message for range-length validation
- **required**: If “true”, indicates input is required (cannot be blank)
- **required-msg**: Error message for required validation

20.2.1.3 Element: <tag>

Indicates an HTML tag should be written. The most common use is to use a <tag> anywhere an HTML "div" element is desired.

Attributes

- name ("div", "span", or "section")
- class

Child elements allowed

- <tag>'s can contain any child element supported by its ***parent*** section.
- <tag>'s can also contain inner text.

20.2.1.4 Element: <label>

Defines a label associated with an input.

Attributes

- name: This is the name of the html element to generate. Supported values are "div", "span" and "label". The default is "label".
- class
- text: Indicates what text is applied to the generated html element. If no text is provided, the default is used.
- for: Indicates which input is linked to the label.

Child elements allowed

- None

20.2.1.5 Element: <voidTag>

Indicates an HTML void tag should be written. A void tag is a tag that cannot contain content.

Attributes

- name: The only allowed value is “br” (generates an html <br/> element)

Child elements allowed

- None

20.2.1.6 Elements: <submitButton> and <cancelButton>

These elements are used by the Hosted solution to either submit or cancel the input and redirect to the merchant.

These button elements **cannot** be used in an iFrame solution. The merchant is required to provide their own submit/cancel framework using the JavaScript plugin.

Attributes

- text: the button text displayed
- class

Child elements allowed

- None

20.2.1.7 Element: <validationMsg>

The <validationMsg> element provides control over the layout and display of validation error messages. If no <validationMsg> is used, then the default (JQuery) behavior is used to display the message immediately following the input control.

Use this element to create a separate html element for displaying validation messages or error images.

Attributes

- name (“div”, “span”, or “section”)
- class
- for (the input the validation is for. The sections below define the valid values for this)
- error-class: the CSS class used on the validation message when a validation error exists

> ⚠ **Note:**
>
> The class and error-class attributes should be used to style elements. Other element attributes (such as id, name, etc) are internally generated values that are subject to change in the future. The merchant’s CSS should not reference those attributes.

Child elements allowed

- None

20.2.2 Main container section

20.2.2.1 Element: <htmlSection>

This is the outermost container (div) for the custom defined HTML. All other sections and elements in the XML are defined within this section.

Attributes:

- name
- class
- xmlns="Paymetric:Intercept:MerchantHtmlPacketModel"

Child elements allowed

- <xiProperties>
- <paymentOptionRadioSection>
- <paymentOptionExpandedRadioSection>
- <paymentOptionDropdownSection>
- <paymentOptionExpandedDropdownSection>
- <cardRadioSection>
- <cardDropdownSection>
- <cardNoTypeSection>
- <echeckRadioSection>
- <echeckDropdownSection>
- <cancelButton> (Hosted page only)
- <submitButton> (Hosted page only)
- <tag>
- <voidTag>
- <additionalHtmlSection>
- <textbox>
- <label>

20.2.3 XiProperties section

This section defines global properties that can be applied to all elements.

20.2.3.1 Element: <xiProperties>

Container element for global xiProperties

Attributes

- error-class: global CSS error class to use on validation messages

> ⚠ **Note:**
>
> The class and error-class attributes should be used to style elements. Other element attributes (such as id, name, etc) are internally generated values that are subject to change in the future. The merchant’s CSS should not reference those attributes.

- inline-validation: provides the ability to perform inline validation as soon as Iframe loads

Child elements allowed

- <errorTooltip>

20.2.3.1.1 Child element: <errorTooltip>

Defines effects and attributes when displaying an error tooltip

Attributes

- class
- hide-duration: String or number (in ms) to determine how long the animation will run. See http://api.jqueryui.com/hide/
- hide-effect: See http://api.jqueryui.com/category/effects/ for supported jquery vales
- show-duration: String or number (in ms) determining how long the animation will run. Refer to http://api.jqueryui.com/hide/ for detailed information.
- show-effect: Refer to http://api.jqueryui.com/category/effects/ for supported jquery values
- position-at: Defines which position on the target element to align the positioned element against. Refer to http://api.jqueryui.com/position for supported jquery values
- position-collision
- position-my: Defines which position on the element being positioned to align with the target element. Refer to http://api.jqueryui.com/position for supported jquery values

Child elements allowed

- None

20.2.4 Payment option section

Allow the web user to select the type of payment method - eCheck or Credit Card. If the merchant only supports eCheck or Credit Card (not both) then this section should be omitted and only the eCheck or Credit Card section needs to be defined in the XML.

There are four types of payment option sections as follows:

- Element: <paymentOptionRadioSection> 162
- Element: <paymentOptionDropdownSection> 163
- Element: <paymentOptionExpandedRadioSection> 165
- Element: <paymentOptionExpandedDropdownSection> 167

20.2.4.1 Element: <paymentOptionRadioSection>

Displays the payment options of “echeck” and “credit card” using radio buttons.

Attributes

- name
- class

Child elements allowed

- <label>
- <radPaymentOption>
- <tag>
- <voidTag>

20.2.4.1.1 Child element: <radPaymentOption>

Defines a radio button within the **<paymentOptionRadioSection>**

Attributes

- for (***required***) : value can be “echeck” or “card”

- class
- is-default

Child elements allowed

- None

20.2.4.1.2 Child element: <label>

Defines a label within the **<paymentOptionRadioSection>**

Attributes

- for (***required***) : value can be "echeck" or "card"
- text
- class
- name (div, span or label)

Child elements allowed

- None

20.2.4.2 Element: <paymentOptionDropdownSection>

Displays the payment options of "echeck" and "credit card" using a select dropdown.

Attributes

- name
- class

Child elements allowed

- <label>
- <ddlPaymentOption>
- <tag>
- <voidTag>

20.2.4.2.1 Child element: <ddlPaymentOption>

Defines a select dropdown within the **<paymentOptionDropdownSection>**

Attributes

- class
- default-text
- required
- required-msg
- error-class
  - NOTE: The class attribute should be used to style elements. Other element attributes (such as id, name, etc) are internally generated values that are subject to change in the future. The merchant's CSS should not reference those attributes.
- error-msg-style

Child elements allowed

- <items>: Contains a list of **<item>** elements
  - <item>
    - Attributes:
      - for (***required)***: value can be "echeck" or "card"
      - is-default
      - text
    - Child elements: None

20.2.4.2.2 Child element: <label>

Defines a label within the **<paymentOptionDropdownSection>**

Attributes

- for (***required***) : value can be "paymentOptionType"
- text
- class
- name (div, span or label)

Child elements allowed

- None

20.2.4.3 Element: <paymentOptionExpandedRadioSection>

Displays the payment options of "echeck" and ***specific*** card types using radio buttons. For example, these payment options could be defined: "echeck", "mastercard" and "visa".

Attributes

- name
- class

Child elements allowed

- <label>
- <radPaymentOption>
- <tag>
- <voidTag>

20.2.4.3.1 Child element: <radPaymentOption>

Defines a radio button within the **<paymentOptionExpandedRadioSection>**

Attributes

- for (***required***) : value can be “echeck”, “american express”, “discover”, “jcb”, “maestro”, “mastercard” or “visa” or "diners"
- is-default
- required
- required-msg

Child elements allowed

- None

20.2.4.3.2 Child element: <label>

Defines a label within the **<paymentOptionExpandedRadioSection>**

Attributes

- for (***required***) : value can be “echeck”, “american express”, “discover”, “jcb”, “maestro”, “mastercard” or “visa” or
- text
- class
- name (div, span or label)

Child elements allowed

- None

20.2.4.4 Element: <paymentOptionExpandedDropdownSection>

Displays the payment options of “echeck” and ***specific*** card types using a select dropdown. For example, these payment options could be defined: “echeck”, “mastercard” and “visa”.

Attributes

- name
- class

Child elements allowed

- <label>
- <ddlPaymentOption>
- <tag>
- <voidTag>

20.2.4.4.1 Child element: <ddlPaymentOption>

Defines a dropdown within the **<paymentOptionExpandedDropdownSection>**

Attributes

- class
- default-text
- required
- required-msg
- error-class
    - **Note**: The class attribute should be used to style elements. Other element attributes (such as id, name, etc) are internally generated Paymetric values that are subject to change in the future. The merchant’s CSS should not reference those attributes.
- error-msg-style

Child elements allowed

- <items>: Contains a list of **<item>** elements
  - <item>
    - Attributes:
      - for: value can be "echeck", "american express", "discover", "jcb", "maestro", "mastercard" or "visa" or "diners"
      - is-default
      - text
    - Child elements: None

20.2.4.4.2 Child Element: <label>

Defines a label within the **<paymentOptionExpandedDropdownSection>**

Attributes

- for (***required***) : value can be "echeck", "american express", "discover", "jcb", "maestro", "mastercard" or "visa" or "diners"
- text
- class
- name (div, span or label)

Child elements allowed

- None

20.2.5 Credit card sections

The credit card section is used to allow the web user to input credit card information.

There are three types of credit card sections. Each of these sections contains common credit card elements defined in the next section.

20.2.5.1 Default card validations

By default, the following client-side validations will be performed against credit card data. Any or all of these can be overridden or disabled in the XML.

Card number checks

- Luhn check
- Amex: Card number matches regex “^(34|37)([0-9]{13})$“
- Discover: Card number matches regex “(^(64|65|66)([0-9]{12}|[0-9]{14})$)|(^(622|624|625|626|628)([0-9]{11}|[0-9]{13})$)|(^(6011)([0-9]{10}|[0-9]{12})$)$ “
- JCB: Card number matches regex “(^(353|355|356|357|358)([0-9]{13})$)|(^(3528|3529)([0-9]{12})$)“
- MasterCard: Card number matches regex “^(36|51|52|53|54|55)([0-9]{12}|[0-9]{14})$“
- Visa: Card number matches regex “^(4)([0-9]{12}|[0-9]{15})$“

Expiration and start dates

Validation to verify the card expiration and start dates are valid is enabled by default.

20.2.5.2 Element: <cardRadioSection>

Displays the credit card section with credit card types rendered using radio buttons.

Attributes

- name
- class

Child elements allowed

- <radCardType>
- <radCustomCardType>
- <label>
- <customLabel>
- <tag>
- <voidTag>
- <validationMsg> - "for" attribute can be any supported input ("cardType", "cardNumber", "cardholderName", "startDate", "startMonth", "startYear", "expDate", "expMonth", "expYear", "cvv"), or the merchant can provide a custom value.
- ***See Common Credit Card Elements for descriptions of the following:***
    - <tboxCardNumber>
    - <tboxCardHolderName>
    - <tboxStartMonth>
    - <tboxStartYear>
    - <ddlStartMonth>
    - <ddlStartYear>
    - <tboxExpMonth>
    - <tboxExpYear>
    - <ddlExpMonth>
    - <ddlExpYear>
    - <tboxCvv>

20.2.5.2.1 Child element: <radCardType>

Defines a radio button for a supported card type within the **<cardRadioSection>**

The <radCardType> must be declared as an empty element. <radCardType for=”visa”/> is valid. <radCardType for=”visa”></radCardType> is not at this time.

Attributes

- for (***required***) : value can be “american express”, “discover”, “jcb”, “maestro”, “mastercard” or “visa” or "diners"
- class
- is-default
- error-class
- error-msg-style
- required
- required-msg

Child elements allowed

- <validateFor> - ***See Common Credit Card Elements***

20.2.5.2.2 Child element: <radCustomCardType>

Defines a radio button for a custom (merchant-defined) card type within the **<cardRadioSection>**

Attributes

- for (***required***) : merchant defines the value
- class
- is-default
- error-class
- error-msg-style
- required
- required-msg
- show-exp-date: If “true” or not supplied, expiration date fields are displayed. If “false”, expiration date fields are hidden.
- show-start-date: If “true”, start date fields are displayed. If “false” or missing, start date fields are hidden.

- show-cvv: If “true” or not supplied, cvv fields are displayed. If “false”, cvv fields are hidden.
- show- cardholder-name: If “true” or not supplied, cardholder name fields are displayed. If “false”, cardholder name fields are hidden.

Child elements allowed

- <validateFor> - ***See Common Credit Card Elements***

20.2.5.2.3 Child element: <label>

Defines a label within the **<cardRadioSection>**

Attributes

- for (***required***) : value can be “american express”, “discover”, “jcb”, “maestro”, “mastercard”, “visa”, "diners", “cardType”, “cardNumber”, “cardholderName”, “startDate”, “startMonth”, “startYear”, “expDate”, “expMonth”, “expYear” or “cvv”
- text
- class
- name (div, span or label)

Child elements allowed

- None

20.2.5.2.4 Child element: <customLabel>

Defines a label to be used with a custom card type within the **<cardRadioSection>**

Attributes

- for: Merchant defined value. This should match the “for” used in the <customItem> defined in <radCustomCardType>
- text (***required)***
- class
- name (div, span or label)

Child elements allowed

- None

20.2.5.3 Element: <cardDropdownSection>

Displays the credit card section with credit card types rendered using a select dropdown.

Attributes

- name
- class

Child elements allowed

- <label>
- <ddlCardType>
- <tag>
- <voidTag>
- <validationMsg> - “for” attribute can be any supported input (“cardType”, “cardNumber”, “cardholderName”, “startDate”, “startMonth”, “startYear”, “expDate”, “expMonth”, “expYear”, “cvv”), or the merchant can provide a custom value.
- ***See Common Credit Card Elements for descriptions of the following:***
  - <tboxCardNumber>
  - <tboxCardHolderName>
  - <tboxStartMonth>
  - <tboxStartYear>
  - <ddlStartMonth>
  - <ddlStartYear>
  - <tboxExpMonth>

  - <tboxExpYear>
  - <ddlExpMonth>
  - <ddlExpYear>
  - <tboxCvv>

20.2.5.3.1 Child element: <ddlCardType>

Defines a dropdown for card type within the **<cardDropdownSection>**

Attributes

- class
- default-text
- required
- required-msg
- error-class
- error-msg-style

Child elements allowed

- <items>: Contains a list of **<item>** elements
  - <item>
    - Attributes:
      - for (***required)***: value can be “american express”, “discover”, “jcb”, “maestro”, “mastercard” or “visa” or "diners"
      - is-default
      - text
    - Child elements:
      - <validateFor>: ***See Common Credit Card Elements***
  - <customItem> - Merchant’s can use this to define custom card types
    - Attributes:
      - for (***required)***: Merchant supplied value (code) for the custom card
      - is-default
      - text

    - show-exp-date: If “true” or not supplied, expiration date fields are displayed. If “false”, expiration date fields are hidden.
    - show-start-date: If “true”, start date fields are displayed. If “false” or missing, start date fields are hidden.
    - show-cvv: If “true” or not supplied, cvv fields are displayed. If “false”, cvv fields are hidden.
    - show- cardholder-name: If “true” or not supplied, cardholder name fields are displayed. If “false”, cardholder name fields are hidden.
  - Child elements:
    - <validateFor>: ***See Common Credit Card Elements***

20.2.5.3.2 Child element: <label>

Defines a label within the **<cardRadioSection>**

Attributes

- for (***required***) : value can be “cardType”, “cardNumber”, “cardholderName”, “startDate”, “startMonth”, “startYear”, “expDate”, “expMonth”, “expYear” or “cvv”
- text
- class
- name (div, span or label)

Child elements allowed

- None

20.2.5.3.3 Child element: <customLabel>

Defines a label for custom card types within the **<cardRadioSection>**

Attributes

- for: Merchant defined value. This should match the "for" defined in the <customItem> in <ddlCardType>
- text (***required)***
- class
- name (div, span or label)

Child elements allowed

- None

20.2.5.4 Element: <cardNoTypeSection>

Displays the credit card section without a control for the card type. Merchants will use this in conjunction with the **<paymentOptionExpandedRadio>** or **<paymentOptionExpandedDropdown>** section which defines the card type outside the Card Section.

Attributes

- name
- class

Child elements allowed

- <label>
- <tag>
- <voidTag>
- <validationMsg> - "for" attribute can be any supported input ("cardType", "cardNumber", "cardholderName", "startDate", "startMonth", "startYear", "expDate", "expMonth", "expYear", "cvv"), or the merchant can provide a custom value.
- ***See Common Credit Card Elements for descriptions of the following:***

- <tboxCardNumber>
- <tboxCardHolderName>
- <tboxStartMonth>
- <tboxStartYear>
- <ddlStartMonth>
- <ddlStartYear>
- <tboxExpMonth>
- <tboxExpYear>
- <ddlExpMonth>
- <ddlExpYear>
- <tboxCvv>

20.2.5.4.1 Child element: <label>

Defines a label within the **<cardNoTypeSection>**

Attributes

- for (***required***) : value can be “cardType”, “cardNumber”, “cardholderName”, “startDate”, “startMonth”, “startYear”, “expDate”, “expMonth”, “expYear” or “cvv”
- text
- class
- name (div, span or label)

Child elements allowed

- None

20.2.5.5 Element: <cardIndicatorSection>

Displays the credit card section with credit card types rendered using a dropdown.

Attributes

- name
- class

Child elements allowed

- <label>
- <cardTypeIndicator>
- <tag>
- <voidTag>
- <validationMsg> - "for" attribute can be any supported input ("cardType", "cardNumber", "cardholderName", "startDate", "startMonth", "startYear", "expDate", "expMonth", "expYear", "cvv"), or the merchant can provide a custom value.
- ***See Common Credit Card Elements for descriptions of the following:***
    - <tboxCardNumber>
    - <tboxCardHolderName>
    - <tboxStartMonth>
    - <tboxStartYear>
    - <ddlStartMonth>
    - <ddlStartYear>
    - <tboxExpMonth>
    - <tboxExpYear>
    - <ddlExpMonth>
    - <ddlExpYear>
    - <tboxCvv>

20.2.5.5.1 Child element: <cardTypeIndicator>

Defines a dropdown for card type within the **<cardTypeIndicator>**

Attribute

- class

Child elements allowed

- <items>: Contains a list of **<item>** elements
    - <item>
        - Attributes:
            - for (***required)***: value can be “american express”, “discover”, “jcb”, “maestro”, “mastercard” or “visa” or "diners"
            - class (***required***): represents the class in the CSS file containing the unselected/selected image; for the selected image, the CSS should contain a class with the same name + " .selected"; for unselected image, the CSS should contain a class with the same name + " .unselected"

```
EXAMPLE in CSS file:
  .mcIndicator {
     background-image: url("../Images/mc.png");
     width: 33px;
     height: 26px;
     vertical-align: middle;
     margin-left: 40px;
     opacity: .3;
  }
  .mcIndicator.selected {
     opacity: 1;
  }
```

            - bin-range-expression: represents the card number BIN range regular expression that selects the appropriate card type image
    - <customItem> - Merchant’s can use this to define custom card types
        - Attributes:

- for (***required)***: Merchant supplied value (code) for the custom card
- class (***required)***: represents the class in the CSS file containing the unselected image; for the selected image, the CSS should contain a second class with the same name + " .selected"
- bin-range-expression (***required)***: represents the card number BIN range regular expression that selects the appropriate card type image
- show-exp-date: If “true” or not supplied, expiration date fields are displayed. If “false”, expiration date fields are hidden.
- show-start-date: If “true”, start date fields are displayed. If “false” or missing, start date fields are hidden.
- show-cvv: If “true” or not supplied, cvv fields are displayed. If “false”, cvv fields are hidden.
- show- cardholder-name: If “true” or not supplied, cardholder name fields are displayed. If “false”, cardholder name fields are hidden.

20.2.5.5.2 Card type indicator validations

If bin-range-expression is not included, the following values are used by default to determine the card type:

- Amex - starts with 34 or 37
- Visa - starts with 4
- MC - starts with 222, 2720,36, or 51 thru 55
- Discover - starts with 64-66, 622, 624-626,628, 6011
- JCB - starts with 353-358, 3528, 3529
- Diners - starts with 30+[0-5]+[0-9], 3095+[0-9], 36+[0-9], or 3+[8-9]+[0-9]

- Maestro - we will not provide a default match. Merchant must send in they're regex match pattern in the 'bin-range-expression' field

20.2.6 Common credit card elements

Below is a list of common inputs used by all 3 types of Credit Card section elements.

- <tboxCardNumber>
- <tboxCardHolderName>
- <tboxStartMonth>
- <tboxStartYear>
- <ddlStartMonth>
- <ddlStartYear>
- <tboxExpMonth>
- <tboxExpYear>
- <ddlExpMonth>
- <ddlExpYear>
- <tboxCvv>

20.2.6.1 Textbox elements: <tbox{XXX}>

<tboxCardNumber>, <tboxCardHolderName>, <tboxStartMonth>, <tboxStartYear>, <tboxExpMonth>, <tboxExpYear>, <tboxCvv>

Attributes (All of these elements support these common attributes)

- class
- error-class
- error-msg-style
- text

- tokenize (***required for <tboxCardNumber>***)
- is-numeric: prevents any character other than numeric; mobile device displays numeric keyboard
- html-required: required if Merchant wants to use floating labels
- placeholder: provides the default text for textbox
- any of the ***Validation Attributes***, with the exception that only <tboxCardNumber> supports “luhn-check” and "luhn-check-msg")

Child elements allowed

- None

<tboxCardNumber> only

In addition to the above attributes, tboxCardNumber has the following attribute as well:

- mask-number
    - Displays the credit card number that is entered by the user with spaces based on one of the following:
        - <cardTypeIndicator> 179 - based off BIN range
        - <cardDropdownSection> 173 - based off selected card type in dropdown
        - <cardRadioSection> 170 - based off selected card type radio button
    - Spacing pattern for Amex is 4 6 5, for example:
        - 34XX XXXXXX XXXXX
    - Spacing pattern for all other card types are in groups of 4, for example:
        - XXXX XXXX XXXX
        - XXXX XXXX XXXX X

    - XXXX XXXX XXXX XXX
    - XXXX XXXX XXXX XXXX
  - Spacing pattern is controlled by Intercept for eCommerce; users are not allowed to enter spaces.
  - This attribute is ignored if Intercept for eCommerce is configured to accept token entry. You can see this setting in Merchant Portal. Go to Settings > Services > Intercept for eCommerce.

20.2.6.2 Elements: < ddlStartMonth>, <ddlStartYear>

Defines dropdowns for card start month and year. These dropdowns only display if **Maestro** is selected.

Attributes

- class
- default-text
- display-format: valid values for <ddlStartMonth> are "M","MM","MMM" or "MMMM" (e.g. '9', '09', 'Sep' or 'September'); valid values for <ddlStartYear> are "yy" or "yyyy" (e.g. '14' or '2014') The defaults are "MMM" and "yyyy"
- error-class
- error-msg-style
- required
- required-msg
- start-date: if "true", then a single validation is performed against both the month and year. Use this to simplify validation by having a single message for the web user rather than separate messages for month and date. If this is used, set the ***required*** attributes on <ddlStartMonth> and <ddlStartYear> to ***"false"***
- start-date-msg: the message to display if the selected month and year are not a valid start date

- years-to-display (***<ddlStartYear> only)***: The number of years to display in the dropdown. The default is 10.

Child elements allowed: For <ddlStartMonth> only

  - <months>: this is an ***optional*** element that contains child elements <jan>, <feb>, <mar>, <apr>, <may>, <jun>, <jul>, <aug>, <sep>, <oct>, <nov> and <dec>. Each of these child elements can have a single attribute **text** to define the text to use for that month if the default text generated is not sufficient.

20.2.6.3 Elements: < ddlExpMonth>, <ddlExpYear>

Defines dropdowns for card expiration month and year.

Attributes

- class
- default-text
- display-format: valid values for <ddlExpMonth> are "M","MM","MMM" or "MMMM" (e.g. '9', '09', 'Sep' or 'September'); valid values for <ddlExpYear> are "yy" or "yyyy" (e.g. '14' or '2014') The defaults are "MMM" and "yyyy"
- error-class
- error-msg-style
- required
- required-msg
- exp-date: if "true", then a single validation is performed against both the month and year. Use this to simplify validation by having a single message for the web user rather than separate messages for month and date. If this is used, set the ***required*** attributes on <ddlExpMonth> and <ddlExpYear> to ***"false"***
- exp-date-msg: the message to display if the selected month and year are not a valid expiration date
- years-to-display (***<ddlExpYear> only)***: The number of years to display in the dropdown. The default is 10.

Child elements allowed: For <ddlExpMonth> only

  - o <months>: this is an ***optional*** element that contains child elements <jan>, <feb>, <mar>, <apr>, <may>, <jun>, <jul>, <aug>, <sep>, <oct>, <nov> and <dec>. Each of these child elements can have a single attribute **text** to define the text to use for that month if the default text generated is not sufficient.

20.2.6.4 Element: <validationFor>

Only used within a credit card section. This element is used to define a cross-control validation dependency between the selected card type and another credit card input field. For example, a “regex” validation dependency can be created between a specific card type and the card number textbox such that if that card type is selected, then card number must follow the regex pattern defined.

Use this element to create a separate html element for displaying validation messages or error images.

Attributes

- None

Child elements allowed

- <validateCardNumber>: Sets up a link between the card type and card number. Any of the ***Validation Attributes*** can be used on this element to add or remove validation.
- <validate>: Sets up a link between the card type and another input besides card number. The valid attributes for this element are:
  - o for (***required)***: The input this validation applies to. Valid values are “cardNumber”, “cardholderName”, “startDate”, “startMonth”, “startYear”, “expDate”, “expMonth”, “expYear” or “cvv”
  - o Any of the ***Validation Attributes*** defined above, except for luhn-check and luhn-check-msg.

20.2.7 eCheck sections

The eCheck section is used to allow the web user to input electronic check information.

There are two types of eCheck sections. Each of these sections contains common eCheck elements defined in the next section.

20.2.7.1 Element: <echeckRadioSection>

Displays the eCheck section with eCheck options rendered using radio buttons.

Attributes

- name
- class

Child elements allowed

- <radAccountType>
- <label>
- <tag>
- <voidTag>
- <validationMsg> - "for" attribute can be any supported input ("accountType", "accountNumber", "routingNumber", "nameOnAccount", or "bankName")
- ***See Common eCheck Elements for descriptions of the following:***
  - <tboxAccountNumber>
  - <tboxBankName>
  - <tboxNameOnAccount>
  - <tboxRoutingNumber>

20.2.7.1.1 Child element: < radAccountType >

Defines a radio button for supported eCheck options within the **<echeckRadioSection>**

Attributes

- for (***required***) : value can be "checking" or "savings"
- class
- is-default
- error-class
- error-msg-style
- required
- required-msg

Child elements allowed

- None

20.2.7.1.2 Child element: <label>

Defines a label within the **< echeckRadioSection >**

Attributes

- for (***required***) : value can be "checking", "savings", "accountNumber", "routingNumber", "nameOnAccount", or "bankName"
- text
- class
- name (div, span or label)

Child elements allowed

- None

20.2.7.2 Element: <echeckDropdownSection>

Displays the eCheck section with eCheck options rendered using a select dropdown.

Attributes

- name
- class

Child elements allowed

- <ddlAccountType>
- <label>
- <tag>
- <voidTag>
- <validationMsg> - "for" attribute can be any supported input ("accountType", "accountNumber", "routingNumber", "nameOnAccount", or "bankName")
- ***See Common eCheck Elements for descriptions of the following:***
  - <tboxAccountNumber>
  - <tboxBankName>
  - <tboxNameOnAccount>
  - <tboxRoutingNumber>

20.2.7.2.1 Child element: < ddlAccountType >

Defines a dropdown for supported eCheck options within the **< echeckDropdownSection >**

Attributes

- class
- default-text
- error-class
- error-msg-style
- required

- required-msg

Child elements allowed

- <items>: Contains a list of **<item>** elements
  - <item>
    - Attributes:
      - for (***required)***: value can be “checking” or “savings”
      - is-default
      - text
    - Child elements:
      - None

20.2.7.2.2 Child element: <label>

Defines a label within the **< echeckDropdownSection >**

Attributes

- for (***required****)* : value can be “accountType”, “accountNumber”, “routingNumber”, “nameOnAccount”, or “bankName”
- text
- class
- name (div, span or label)

Child elements allowed

- None

20.2.8 Common eCheck elements

Below is a list of common inputs used by both eCheck section elements.

- <tboxAccountNumber>
- <tboxBankName>
- <tboxNameOnAccount>
- <tboxRoutingNumber>

20.2.8.1 Textbox elements: < tbox{XXX} >

<tboxAccountNumber>, <tboxBankName >, < tboxNameOnAccount >, < tboxRoutingNumber >

Attributes: All of these elements support these common attributes

- class
- error-class
- error-msg-style
- text
- tokenize (***required for <tboxAccountNumber>***)
- is-numeric: prevents any character other than numeric; mobile device displays numeric keyboard
- html-required: required if Merchant wants to use floating labels
- placeholder: provides the default text for textbox
- any of the ***Validation Attributes***, except for “luhn-check” and "luhn-check-msg"

Child elements allowed

- None

20.2.9 Additional HTML section

This section is used to contain any additional data input that the merchant wants to collect. For example, to collect address information add this section to the XML.

20.2.9.1 Element: <additionalHtmlSectionElements>

Attributes

- class
- name

Child elements allowed

- <tag>
- <label>
- <textbox>
- <ddlUsStateDropdown>
- <validationMsg>

20.2.9.1.1 Child element: <label>

Defines a label within the **<additionalHtmlSectionElements>**

Attributes

- for: Merchant defined value
- text
- class
- name (div, span or label)

Child elements allowed

- None

20.2.9.1.2 Child element: <textBox>

Defines an input textbox within the **<additionalHtmlSectionElements >**

Attributes

- class
- text
- name (***required***): Merchant defined value. This HTML input generated will have a “name” attribute with this value.
- is-numeric: prevents any character other than numeric; mobile device displays numeric keyboard
- html-required: **required** if Merchant wants to use floating labels
- placeholder: provides the default text for textbox
- any of the ***Validation Attributes***, except for “luhn-check” and "luhn-check-msg"

Child elements allowed

- None

20.2.9.1.3 Child element: <ddlUsStateDropdown>

Defines a dropdown of US states within the **<additionalHtmlSectionElements >**

Attributes

- class
- default-text
- error-class
- error-msg-style
- required
- required-msg

Child elements allowed

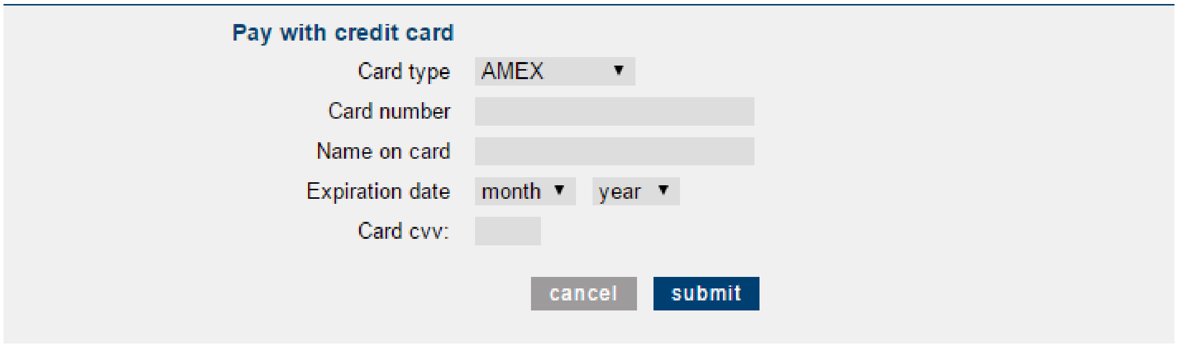
- None

20.2.10 Sample merchant XMLs

Below are samples that show various ways to configure the Merchant Custom XML and the HTML output generated for the web user.

These samples use Paymetric logos and copyright information. Merchant logos and information can be supplied thru the XML and CSS.

20.2.10.1 Sample 1: Credit card (dropdown)



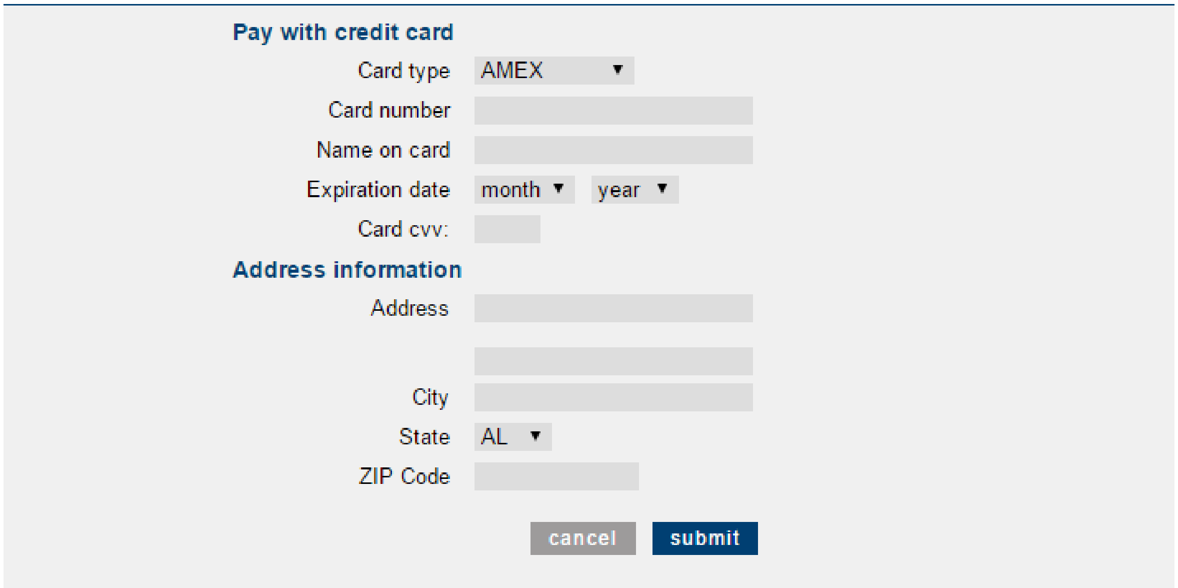
Pay with credit card

Card type	AMEX ▼
Card number	
Name on card	
Expiration date	month ▼ year ▼
Card cvv:	

```
<htmlSection class="payment-content"
xmlns="Paymetric:Intercept:MerchantHtmlPacketModel">
  <tag>
    <tag name="div" class="header-top"/>
    <tag name="div" class="merchant-logo"/>
    <tag name="div" class="main-content">
      <tag name="div" class="billing-content">
        <tag name="div" class="billing-header"></tag>
        <tag name="div" class="billing-info">
          <cardDropdownSection>
            <tag name="div">
              <label for="cardType" text="Card type"/>
              <ddlCardType>
                <items>
                  <item for="american express"/>
                  <item for="mastercard"/>
                  <item for="visa"/>
                  <item for="maestro"/>
                </items>
              </ddlCardType>
            </tag>
            <tag name="div">
              <label for="cardNumber" text="Card number"/>
              <tboxCardNumber tokenize="true" class="xi-long-text"/>
              <validationMsg for="cardNumber" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="cardholderName" text="Name on card"/>
              <tboxCardHolderName class="xi-long-text"/>
              <validationMsg for="cardholderName" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="startDate"/>
              <ddlStartMonth display-format="MMM" class="date_combos"
required="false"/>
              <ddlStartYear class="date_combos" years-to-display="10"
required="false" start-date="true"/>
              <validationMsg for="startYear" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="expDate" text="Expiration date"/>
              <ddlExpMonth default-text="month" class="date_combos"
required="false"/>
              <ddlExpYear default-text="year" class="date_combos" years-
to-display="10" required="false" exp-date="true"/>
              <validationMsg for="expYear" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="cvv" text="Card cvv:"/>
              <tboxCvv class="xi-short-text"/>
              <validationMsg for="cvv" class="valmsg"/>
            </tag>
          </cardDropdownSection>
        </tag>
        <tag name="div" class="payment-button">
          <cancelButton text="cancel" class="cancel-button"/>
          <submitButton text="submit" class="submit-button"/>
        </tag>
      </tag>
    </tag>
    <tag name="div" class="copyright-footer">
      <tag name="div" class="footer-spacing"></tag>
      <tag name="div" class="top-bottom-frame"></tag>
      <tag name="div" class="middle-bottom-frame">
        <tag name="div" class="middle-bottom-frame-content">
          Copyright © 2015. worldpay.com. All Rights Reserved. Various
trademarks held by their respective owners. Legal Notice.
        </tag>
      </tag>
    </tag>
  </tag>
```

```
</htmlSection>
```

20.2.10.2 Sample 2: Credit card (dropdown) with additional HTML section for capturing address



Pay with credit card

Card type: AMEX

Card number

Name on card

Expiration date: month, year

Card cvv:

Address information

Address

City

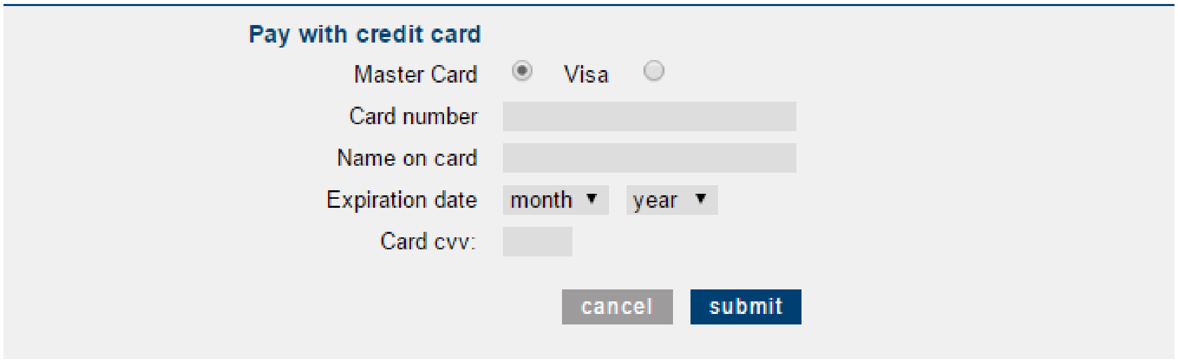
State: AL

ZIP Code

```
<htmlSection class="payment-content"
xmlns="Paymetric:Intercept:MerchantHtmlPacketModel">
  <tag>
    <tag name="div" class="header-top"/>
    <tag name="div" class="merchant-logo"/>
    <tag name="div" class="main-content">
      <tag name="div" class="billing-content">
        <tag name="div" class="billing-header"></tag>
        <tag name="div" class="billing-info">
          <cardDropdownSection>
            <tag name="div">
              <label for="cardType" text="Card type"/>
              <ddlCardType>
                <items>
                  <item for="american express"/>
                  <item for="mastercard"/>
                  <item for="visa"/>
                  <item for="maestro"/>
                </items>
              </ddlCardType>
            </tag>
            <tag name="div">
              <label for="cardNumber" text="Card number"/>
              <tboxCardNumber tokenize="true" class="xi-long-text"/>
              <validationMsg for="cardNumber" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="cardholderName" text="Name on card"/>
              <tboxCardHolderName class="xi-long-text"/>
              <validationMsg for="cardholderName" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="startDate"/>
              <ddlStartMonth display-format="MMM" class="date_combos"
required="false"/>
              <ddlStartYear class="date_combos" years-to-display="10"
required="false" start-date="true"/>
              <validationMsg for="startYear" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="expDate" text="Expiration date"/>
              <ddlExpMonth default-text="month" class="date_combos"
required="false"/>
              <ddlExpYear default-text="year" class="date_combos" years-
to-display="10" required="false" exp-date="true"/>
```

```
              <validationMsg for="expYear" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="cvv" text="Card cvv:"/>
              <tboxCvv class="xi-short-text"/>
              <validationMsg for="cvv" class="valmsg"/>
            </tag>
          </cardDropdownSection>
        </tag>
        <tag name="div" class="address-header"></tag>
        <tag name="div" class="address-info">
          <additionalHtmlSection class="address-content">
            <tag name="div">
              <label for="address1"/>
              <textBox name="address1" class="xi-long-text" required-msg="Please enter an address"/>
            </tag>
            <tag name="div">
              <label for="address2"/>
              <textBox name="address2" class="xi-long-text" required="false"/>
              <validationMsg for="address1" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="city"/>
              <textBox name="city" class="xi-long-text" required-msg="Please enter the city"/>
              <validationMsg for="city" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="state"/>
              <ddlUsStateDropdown/>
              <validationMsg for="state" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="postalCode"/>
              <textBox name="postalCode" class="xi-medium-text"
pattern="^\d{5}(-\d{4})?$" required-msg="Please enter the postal code"
pattern-msg="Please enter a valid postal code"/>
              <validationMsg for="postalCode" class="valmsg"/>
            </tag>
          </additionalHtmlSection>
        </tag>
        <tag name="div" class="payment-button">
          <cancelButton text="cancel" class="cancel-button"/>
          <submitButton text="submit" class="submit-button"/>
        </tag>
      </tag>
    </tag>
    <tag name="div" class="copyright-footer">
      <tag name="div" class="footer-spacing"></tag>
      <tag name="div" class="top-bottom-frame"></tag>
      <tag name="div" class="middle-bottom-frame">
        <tag name="div" class="middle-bottom-frame-content">
          Copyright © 2015. worldpay.com. All Rights Reserved. Various
trademarks held by their respective owners. Legal Notice.
        </tag>
      </tag>
    </tag>
  </tag>
</htmlSection>
```

20.2.10.3 Sample 3: Credit card (radio)



Pay with credit card

Master Card ● Visa ○

Card number

Name on card

Expiration date month ▾ year ▾

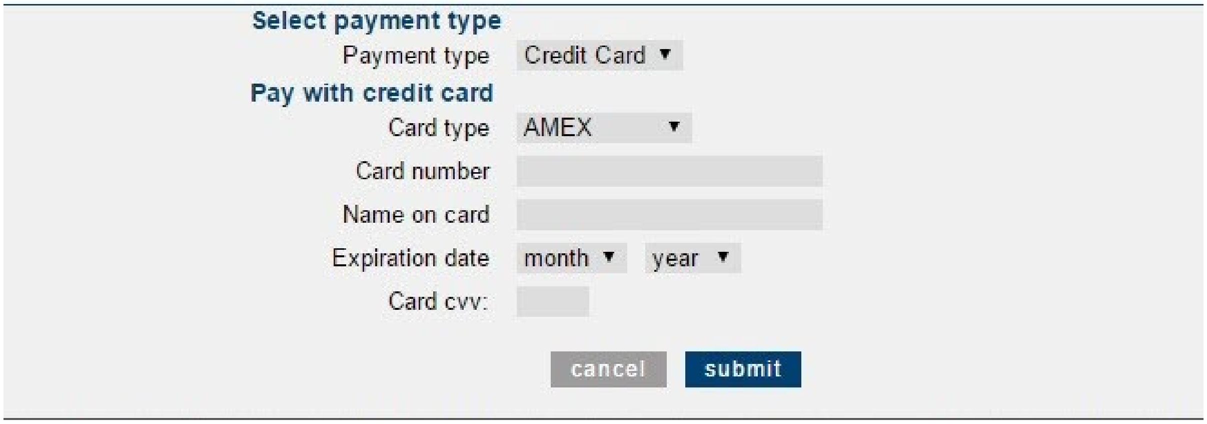
Card cvv:

```
<htmlSection class="payment-content"
xmlns="Paymetric:Intercept:MerchantHtmlPacketModel">
  <tag>
    <tag name="div" class="header-top"/>
    <tag name="div" class="merchant-logo"/>
    <tag name="div" class="main-content">
      <tag name="div" class="billing-content">
        <tag name="div" class="billing-header"></tag>
        <tag name="div" class="billing-info">
          <cardRadioSection>
            <tag name="div">
              <label for="mastercard"/>
              <radCardType for="mastercard"/>
              <label for="visa" text="Visa"/>
              <radCardType for="visa"/>
            </tag>
            <tag name="div">
              <label for="cardNumber" text="Card number"/>
              <tboxCardNumber tokenize="true" class="xi-long-text"/>
              <validationMsg for="cardNumber" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="cardholderName" text="Name on card"/>
              <tboxCardHolderName class="xi-long-text"/>
              <validationMsg for="cardholderName" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="startDate"/>
              <ddlStartMonth display-format="MMM" class="date_combos"
required="false"/>
              <ddlStartYear class="date_combos" years-to-display="10"
required="false" start-date="true"/>
              <validationMsg for="startYear" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="expDate" text="Expiration date"/>
              <ddlExpMonth default-text="month" class="date_combos"
required="false"/>
              <ddlExpYear default-text="year" class="date_combos" years-
to-display="10" required="false" exp-date="true"/>
              <validationMsg for="expYear" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="cvv" text="Card cvv:"/>
              <tboxCvv class="xi-short-text"/>
              <validationMsg for="cvv" class="valmsg"/>
            </tag>
          </cardRadioSection>
        </tag>
        <tag name="div" class="payment-button">
          <cancelButton text="cancel" class="cancel-button"/>
          <submitButton text="submit" class="submit-button"/>
        </tag>
```

```
          </tag>
        </tag>
        <tag name="div" class="copyright-footer">
          <tag name="div" class="footer-spacing"></tag>
          <tag name="div" class="top-bottom-frame"></tag>
          <tag name="div" class="middle-bottom-frame">
            <tag name="div" class="middle-bottom-frame-content">
              Copyright © 2015. worldpay.com. All Rights Reserved. Various
trademarks held by their respective owners. Legal Notice.
            </tag>
          </tag>
        </tag>
      </tag>
    </htmlSection>
```

20.2.10.4 Sample 4: Credit card (dropdown) and eCheck (dropdown) with payment option (dropdown)

Credit card view



Select payment type

Payment type: Credit Card

Pay with credit card

Card type: AMEX

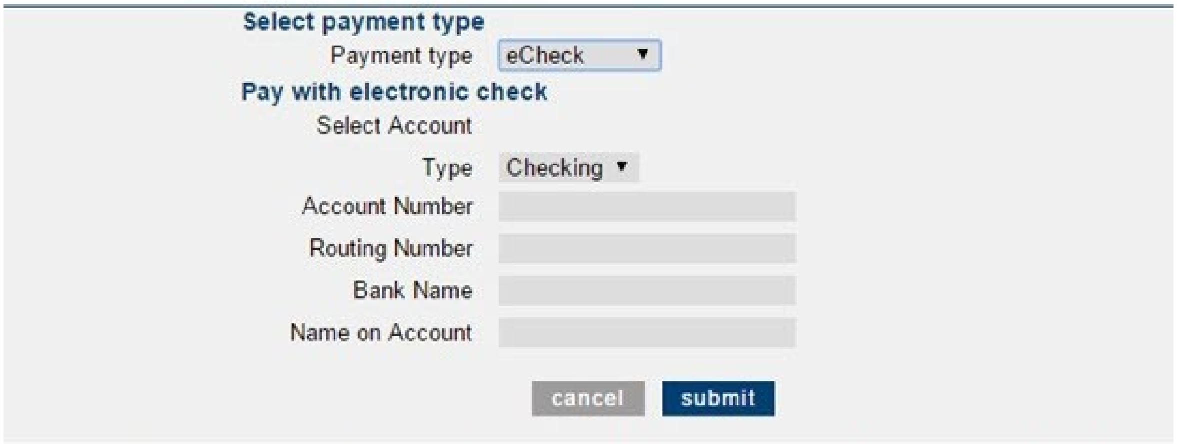
Card number

Name on card

Expiration date: month, year

Card cvv:

eCheck view



Select payment type

Payment type: eCheck

Pay with electronic check

Select Account

Type: Checking

Account Number

Routing Number

Bank Name

Name on Account

```
<htmlSection class="payment-content"
xmlns="Paymetric:Intercept:MerchantHtmlPacketModel">
  <tag>
    <tag name="div" class="header-top"/>
```

```
<tag name="div" class="merchant-logo"/>
<tag name="div" class="main-content">
  <paymentOptionDropdownSection>
    <tag name="div" class="billing-content">
      <tag name="div" class="payoption-header"></tag>
      <tag name="div" class="payoption-info">
        <tag>
          <label for="paymentOptionType" text="Payment type"/>
          <ddlPaymentOption>
            <items>
              <item for="card"/>
              <item for="echeck"/>
            </items>
          </ddlPaymentOption>
        </tag>
      </tag>
    </tag>
  </paymentOptionDropdownSection>
  <cardDropdownSection>
    <tag name="div" class="billing-content">
      <tag name="div" class="billing-header"></tag>
      <tag name="div" class="billing-info">
        <tag name="div">
          <label for="cardType" text="Card type"/>
          <ddlCardType>
            <items>
              <item for="american express"/>
              <item for="mastercard"/>
              <item for="visa"/>
              <item for="maestro"/>
            </items>
          </ddlCardType>
```

```
            </tag>
            <tag name="div">
              <label for="cardNumber" text="Card number"/>
              <tboxCardNumber tokenize="true" class="xi-long-text"/>
              <validationMsg for="cardNumber" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="cardholderName" text="Name on card"/>
              <tboxCardHolderName class="xi-long-text"/>
              <validationMsg for="cardholderName" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="startDate"/>
              <ddlStartMonth display-format="MMM" class="date_combos"
required="false"/>
              <ddlStartYear class="date_combos" years-to-display="10"
required="false" start-date="true"/>
              <validationMsg for="startYear" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="expDate" text="Expiration date"/>
              <ddlExpMonth default-text="month" class="date_combos"
required="false"/>
              <ddlExpYear default-text="year" class="date_combos" years-
to-display="10" required="false" exp-date="true"/>
              <validationMsg for="expYear" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="cvv" text="Card cvv:"/>
              <tboxCvv class="xi-short-text"/>
              <validationMsg for="cvv" class="valmsg"/>
            </tag>
          </tag>
```

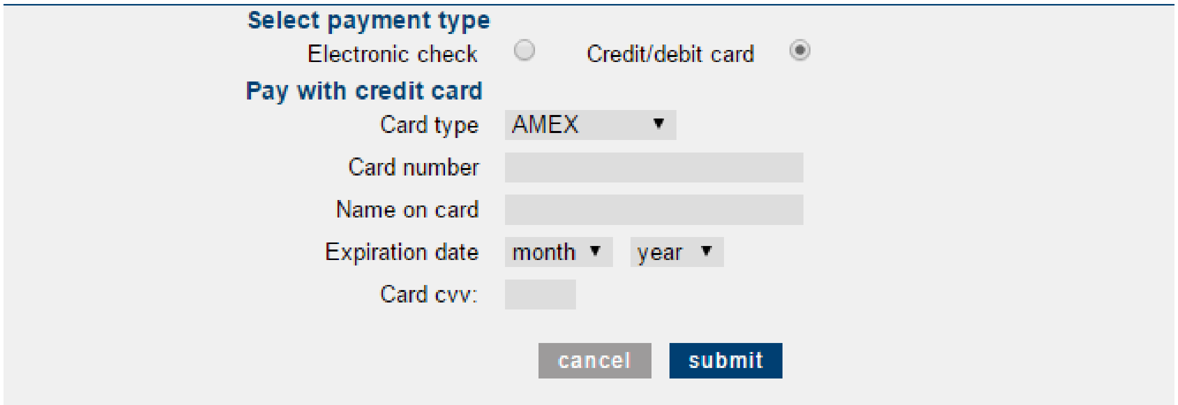
```
  </tag>

</cardDropdownSection>
<echeckDropdownSection>
  <tag name="div" class="billing-content">
    <tag name="div" class="echeck-header"></tag>
    <tag name="div" class="billing-info">
      <tag name="div">
        <label for="accountType"/>
        <ddlAccountType>
          <items>
            <item for="checking"/>
            <item for="savings"/>
          </items>
        </ddlAccountType>
      </tag>
      <tag name="div">
        <label for="accountNumber"/>
        <tboxAccountNumber tokenize="true"/>
      </tag>
      <tag name="div">
        <label for="routingNumber"/>
        <tboxRoutingNumber/>
      </tag>
      <tag name="div">
        <label for="bankName"/>
        <tboxBankName/>
      </tag>
      <tag name="div">
        <label for="nameOnAccount"/>
        <tboxNameOnAccount/>
      </tag>
```

```
          </tag>
        </tag>
      </echeckDropdownSection>
      <tag name="div" class="payment-button">
        <cancelButton text="cancel" class="cancel-button"/>
        <submitButton text="submit" class="submit-button"/>
      </tag>
    </tag>
    <tag name="div" class="copyright-footer">
      <tag name="div" class="footer-spacing"></tag>
      <tag name="div" class="top-bottom-frame"></tag>
      <tag name="div" class="middle-bottom-frame">
        <tag name="div" class="middle-bottom-frame-content">
          Copyright © 2014. worldpay.com. All Rights Reserved. Various
trademarks held by their respective owners. Legal Notice.
        </tag>
      </tag>
    </tag>
  </tag>
</htmlSection>
```

20.2.10.5 Sample 5: Credit card (dropdown) and eCheck (radio) with payment option (radio)

Credit card view



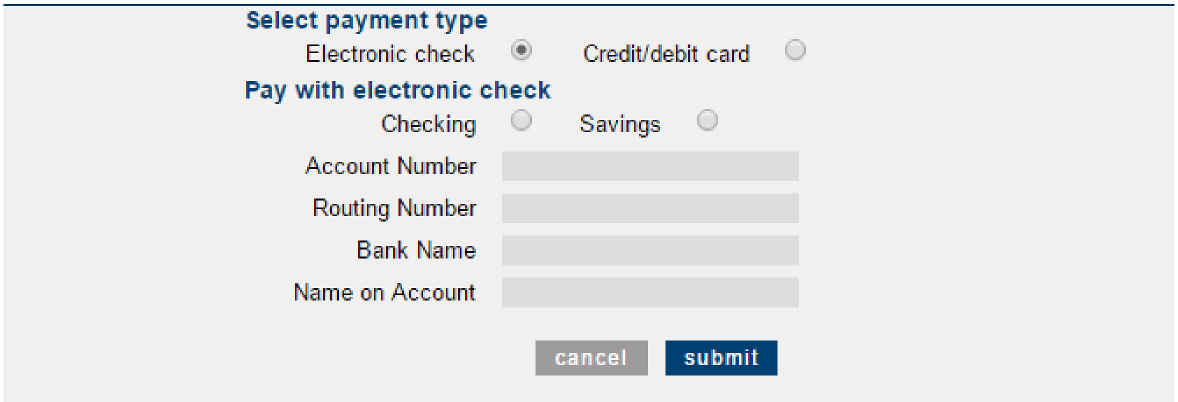
Select payment type

Electronic check ○ Credit/debit card ●

Pay with credit card

Card type	AMEX ▼
Card number	
Name on card	
Expiration date	month ▼ year ▼
Card cvv:	

eCheck view



Select payment type

Electronic check ● Credit/debit card ○

Pay with electronic check

Checking ○	Savings ○
Account Number	
Routing Number	
Bank Name	
Name on Account	

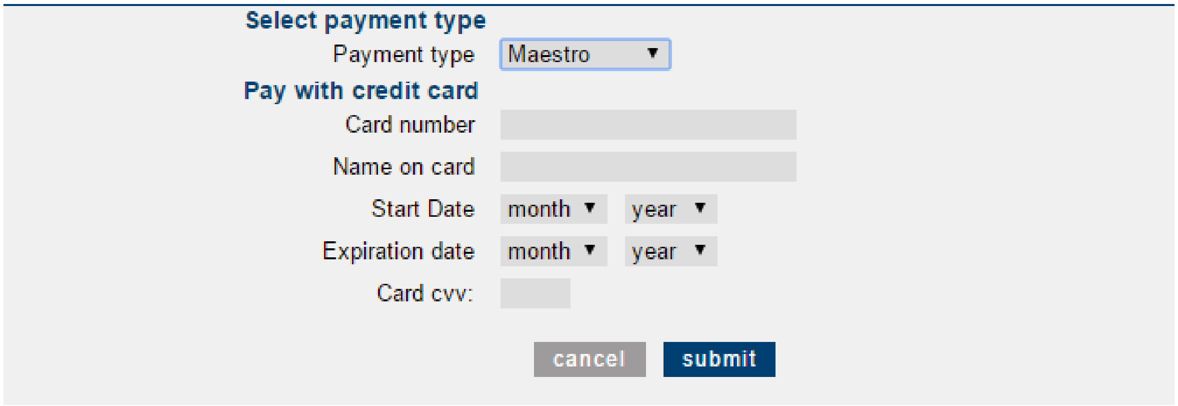
```
<htmlSection class="payment-content"
xmlns="Paymetric:Intercept:MerchantHtmlPacketModel">
  <tag>
    <tag name="div" class="header-top"/>
    <tag name="div" class="merchant-logo"/>
    <tag name="div" class="main-content">
      <paymentOptionRadioSection>
        <tag name="div" class="billing-content">
          <tag name="div" class="payoption-header"></tag>
          <tag name="div" class="payoption-info">
            <tag>
              <label for="echeck" text="Electronic check"/>
              <radPaymentOption for="echeck"/>
              <label for="card" text="Credit/debit card"/>
              <radPaymentOption for="card"/>
            </tag>
          </tag>
        </tag>
      </paymentOptionRadioSection>
      <cardDropdownSection>
        <tag name="div" class="billing-content">
          <tag name="div" class="billing-header"></tag>
          <tag name="div" class="billing-info">
            <tag name="div">
              <label for="cardType" text="Card type"/>
              <ddlCardType>
                <items>
                  <item for="american express"/>
                  <item for="mastercard"/>
                  <item for="visa"/>
                  <item for="maestro"/>
                </items>
              </ddlCardType>
            </tag>
            <tag name="div">
              <label for="cardNumber" text="Card number"/>
              <tboxCardNumber tokenize="true" class="xi-long-text"/>
              <validationMsg for="cardNumber" class="valmsg"/>
            </tag>
```

```
            <tag name="div">
              <label for="cardholderName" text="Name on card"/>
              <tboxCardHolderName class="xi-long-text"/>
              <validationMsg for="cardholderName" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="startDate"/>
              <ddlStartMonth display-format="MMM" class="date_combos"
required="false"/>
              <ddlStartYear class="date_combos" years-to-display="10"
required="false" start-date="true"/>
              <validationMsg for="startYear" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="expDate" text="Expiration date"/>
              <ddlExpMonth default-text="month" class="date_combos"
required="false"/>
              <ddlExpYear default-text="year" class="date_combos" years-
to-display="10" required="false" exp-date="true"/>
              <validationMsg for="expYear" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="cvv" text="Card cvv:"/>
              <tboxCvv class="xi-short-text"/>
              <validationMsg for="cvv" class="valmsg"/>
            </tag>
          </tag>
        </tag>
      </cardDropdownSection>
      <echeckRadioSection>
        <tag name="div" class="billing-content">
          <tag name="div" class="echeck-header"></tag>
          <tag name="div" class="billing-info">
            <tag name="div">
              <label for="checking"/>
              <radAccountType for="checking"/>
              <label for="savings" class="ec-label"/>
              <radAccountType for="savings"/>
            </tag>
            <tag name="div">
              <label for="accountNumber"/>
              <tboxAccountNumber tokenize="true"/>
            </tag>
            <tag name="div">
              <label for="routingNumber"/>
              <tboxRoutingNumber/>
            </tag>
            <tag name="div">
              <label for="bankName"/>
              <tboxBankName/>
            </tag>
            <tag name="div">
              <label for="nameOnAccount"/>
              <tboxNameOnAccount/>
            </tag>
          </tag>
        </tag>
      </echeckRadioSection>
      <tag name="div" class="payment-button">
        <cancelButton text="cancel" class="cancel-button"/>
        <submitButton text="submit" class="submit-button"/>
      </tag>
    </tag>
    <tag name="div" class="copyright-footer">
      <tag name="div" class="footer-spacing"></tag>
```

```
      <tag name="div" class="top-bottom-frame"></tag>
      <tag name="div" class="middle-bottom-frame">
        <tag name="div" class="middle-bottom-frame-content">
          Copyright © 2015. worldpay.com. All Rights Reserved. Various
trademarks held by their respective owners. Legal Notice.
        </tag>
      </tag>
    </tag>
  </tag>
</htmlSection>
```

20.2.10.6 Sample 6: Credit card (NoType) and eCheck (radio) with payment option (expanded dropdown)

Credit card view



Select payment type

Payment type: Maestro

Pay with credit card

Card number:

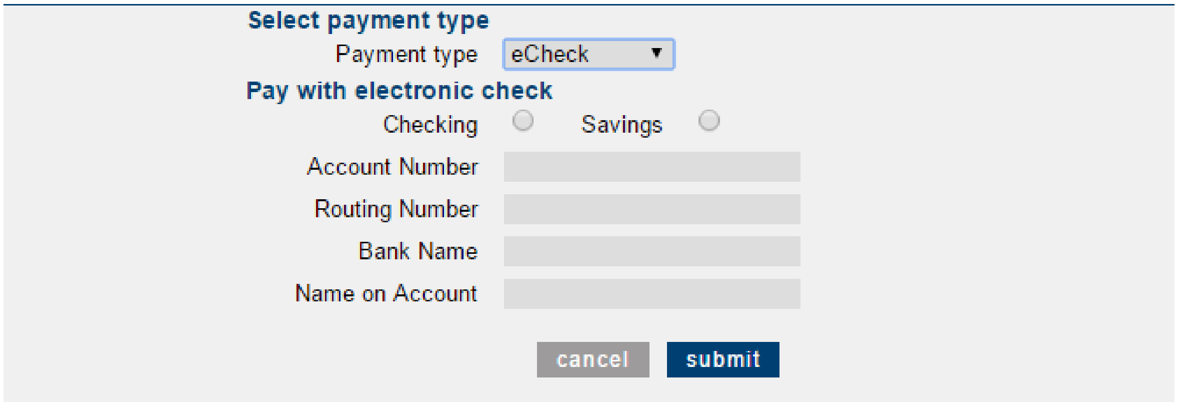
Name on card:

Start Date: month / year

Expiration date: month / year

Card cvv:

eCheck view



Select payment type

Payment type eCheck ▼

Pay with electronic check

Checking ○ Savings ○

Account Number

Routing Number

Bank Name

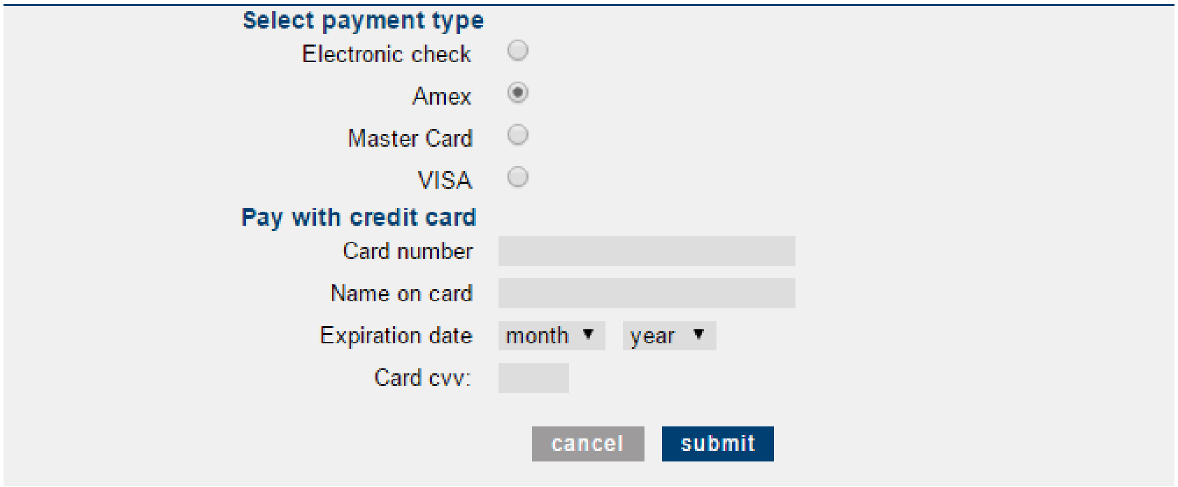
Name on Account

```
<htmlSection class="payment-content"
xmlns="Paymetric:Intercept:MerchantHtmlPacketModel">
  <tag>
    <tag name="div" class="header-top"/>
    <tag name="div" class="merchant-logo"/>
    <tag name="div" class="main-content">
      <paymentOptionExpandedDropdownSection>
        <tag name="div" class="billing-content">
          <tag name="div" class="payoption-header"></tag>
          <tag name="div" class="payoption-info">
            <tag>
              <label for="paymentOptionType" text="Payment type"/>
              <ddlPaymentOption>
                <items>
                  <item for="echeck"/>
                  <item for="american express"/>
                  <item for="discover"/>
                  <item for="jcb"/>
                  <item for="maestro"/>
                  <item for="mastercard"/>
                  <item for="visa"/>
                </items>
              </ddlPaymentOption>
            </tag>
          </tag>
        </tag>
      </paymentOptionExpandedDropdownSection>
      <cardNoTypeSection>
        <tag name="div" class="billing-content">
          <tag name="div" class="billing-header"></tag>
          <tag name="div" class="billing-info">
            <tag name="div">
              <label for="cardNumber" text="Card number"/>
              <tboxCardNumber tokenize="true" class="xi-long-text"/>
              <validationMsg for="cardNumber" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="cardholderName" text="Name on card"/>
              <tboxCardHolderName class="xi-long-text"/>
              <validationMsg for="cardholderName" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="startDate"/>
              <ddlStartMonth display-format="MMM" class="date_combos"
required="false" default-text="month"/>
              <ddlStartYear class="date_combos" years-to-display="10"
required="false" start-date="true" default-text="year"/>
              <validationMsg for="startYear" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="expDate" text="Expiration date"/>
              <ddlExpMonth default-text="month" class="date_combos"
required="false"/>
```

```
              <ddlExpYear default-text="year" class="date_combos" years-to-display="10" required="false" exp-date="true"/>
              <validationMsg for="expYear" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="cvv" text="Card cvv:"/>
              <tboxCvv class="xi-short-text"/>
              <validationMsg for="cvv" class="valmsg"/>
            </tag>
          </tag>
        </tag>
      </cardNoTypeSection>
      <echeckRadioSection>
        <tag name="div" class="billing-content">
          <tag name="div" class="echeck-header"></tag>
          <tag name="div" class="billing-info">
            <tag name="div">
              <label for="checking"/>
              <radAccountType for="checking"/>
              <label for="savings" class="ec-label"/>
              <radAccountType for="savings"/>
            </tag>
            <tag name="div">
              <label for="accountNumber"/>
              <tboxAccountNumber tokenize="true"/>
            </tag>
            <tag name="div">
              <label for="routingNumber"/>
              <tboxRoutingNumber/>
            </tag>
            <tag name="div">
              <label for="bankName"/>
              <tboxBankName/>
            </tag>
            <tag name="div">
              <label for="nameOnAccount"/>
              <tboxNameOnAccount/>
            </tag>
          </tag>
        </tag>
      </echeckRadioSection>
      <tag name="div" class="payment-button">
        <cancelButton text="cancel" class="cancel-button"/>
        <submitButton text="submit" class="submit-button"/>
      </tag>
    </tag>
    <tag name="div" class="copyright-footer">
      <tag name="div" class="footer-spacing"></tag>
      <tag name="div" class="top-bottom-frame"></tag>
      <tag name="div" class="middle-bottom-frame">
        <tag name="div" class="middle-bottom-frame-content">
          Copyright © 2015. worldpay.com. All Rights Reserved. Various
trademarks held by their respective owners. Legal Notice.
        </tag>
      </tag>
    </tag>
  </tag>
</htmlSection>
```

20.2.10.7 Sample 7: Credit card (NoType) and eCheck (radio) with payment option (expanded radio)

Credit card view



Select payment type

Electronic check ○

Amex ●

Master Card ○

VISA ○

Pay with credit card

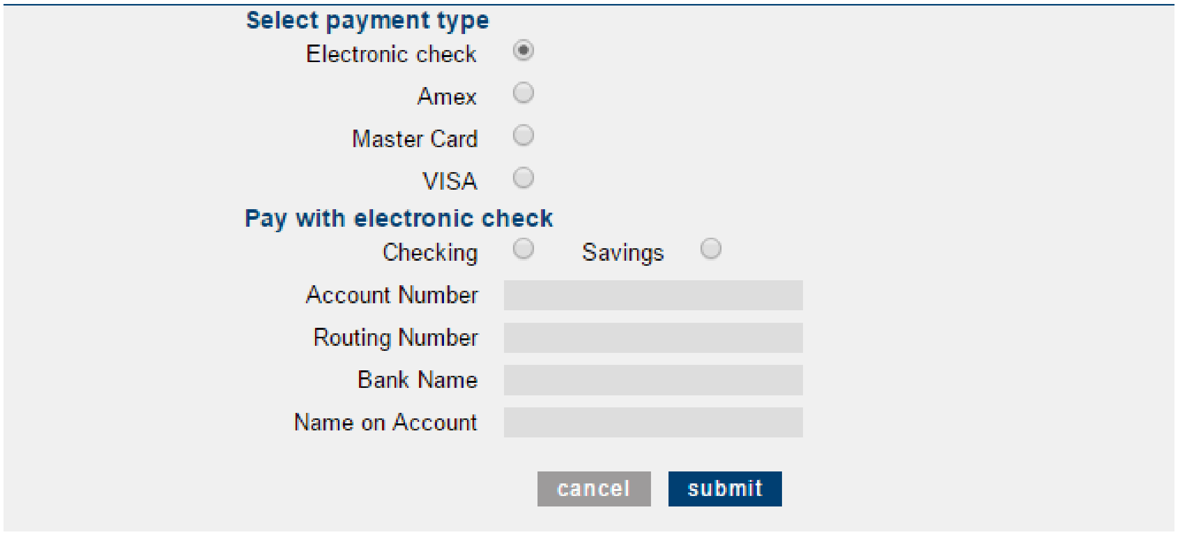
Card number

Name on card

Expiration date month ▼ year ▼

Card cvv:

eCheck view



Select payment type

Electronic check ●

Amex ○

Master Card ○

VISA ○

Pay with electronic check

Checking ○ Savings ○

Account Number

Routing Number

Bank Name

Name on Account

```
<htmlSection class="payment-content"
xmlns="Paymetric:Intercept:MerchantHtmlPacketModel">
```

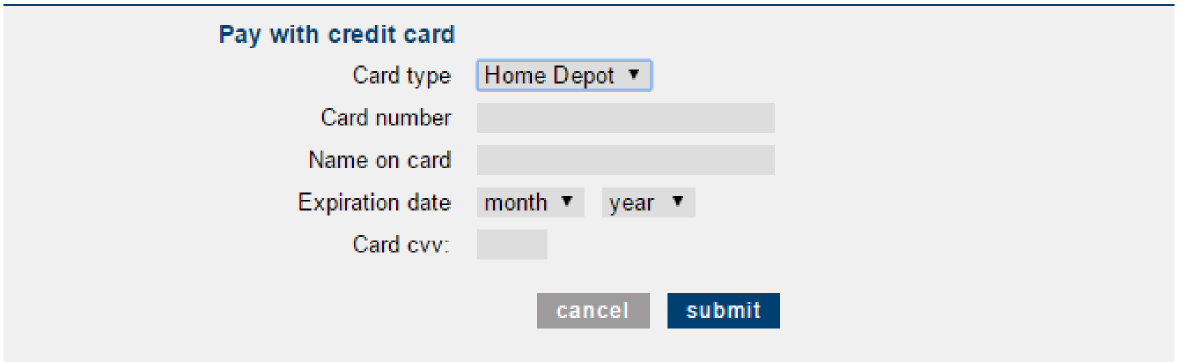
```
<tag>
  <tag name="div" class="header-top"/>
  <tag name="div" class="merchant-logo"/>
  <tag name="div" class="main-content">
    <paymentOptionExpandedRadioSection>
      <tag name="div" class="billing-content">
        <tag name="div" class="payoption-header"></tag>
        <tag name="div" class="payoption-info">
          <tag>
            <label for="echeck" text="Electronic check"/>
            <radPaymentOption for="echeck"/>
          </tag>
          <tag>
            <label for="american express" text="Amex"/>
            <radPaymentOption for="american express"/>
          </tag>
          <tag>
            <label for="mastercard"/>
            <radPaymentOption for="mastercard"/>
          </tag>
          <tag>
            <label for="visa"/>
            <radPaymentOption for="visa"/>
          </tag>
        </tag>
      </tag>
    </paymentOptionExpandedRadioSection>
    <cardNoTypeSection>
      <tag name="div" class="billing-content">
        <tag name="div" class="billing-header"></tag>
        <tag name="div" class="billing-info">
          <tag name="div">
```

```
              <label for="cardNumber" text="Card number"/>
              <tboxCardNumber tokenize="true" class="xi-long-text"/>
              <validationMsg for="cardNumber" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="cardholderName" text="Name on card"/>
              <tboxCardHolderName class="xi-long-text"/>
              <validationMsg for="cardholderName" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="startDate"/>
              <ddlStartMonth display-format="MMM" class="date_combos"
required="false"/>
              <ddlStartYear class="date_combos" years-to-display="10"
required="false" start-date="true"/>
              <validationMsg for="startYear" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="expDate" text="Expiration date"/>
              <ddlExpMonth default-text="month" class="date_combos"
required="false"/>
              <ddlExpYear default-text="year" class="date_combos" years-
to-display="10" required="false" exp-date="true"/>
              <validationMsg for="expYear" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="cvv" text="Card cvv:"/>
              <tboxCvv class="xi-short-text"/>
              <validationMsg for="cvv" class="valmsg"/>
            </tag>
          </tag>
        </tag>
      </cardNoTypeSection>
```

```
<echeckRadioSection>
  <tag name="div" class="billing-content">
    <tag name="div" class="echeck-header"></tag>
    <tag name="div" class="billing-info">
      <tag name="div">
        <label for="checking"/>
        <radAccountType for="checking"/>
        <label for="savings" class="ec-label"/>
        <radAccountType for="savings"/>
      </tag>
      <tag name="div">
        <label for="accountNumber"/>
        <tboxAccountNumber tokenize="true"/>
      </tag>
      <tag name="div">
        <label for="routingNumber"/>
        <tboxRoutingNumber/>
      </tag>
      <tag name="div">
        <label for="bankName"/>
        <tboxBankName/>
      </tag>
      <tag name="div">
        <label for="nameOnAccount"/>
        <tboxNameOnAccount/>
      </tag>
    </tag>
  </tag>
</echeckRadioSection>
<tag name="div" class="payment-button">
  <cancelButton text="cancel" class="cancel-button"/>
  <submitButton text="submit" class="submit-button"/>
```

```
        </tag>
      </tag>
      <tag name="div" class="copyright-footer">
        <tag name="div" class="footer-spacing"></tag>
        <tag name="div" class="top-bottom-frame"></tag>
        <tag name="div" class="middle-bottom-frame">
          <tag name="div" class="middle-bottom-frame-content">
            Copyright © 2015. worldpay.com. All Rights Reserved. Various
trademarks held by their respective owners. Legal Notice.
          </tag>
        </tag>
      </tag>
    </tag>
</htmlSection>
```

20.2.10.8 Sample 8: Credit card (dropdown) with merchant custom card type (company A)



Pay with credit card

Card type: Home Depot

Card number

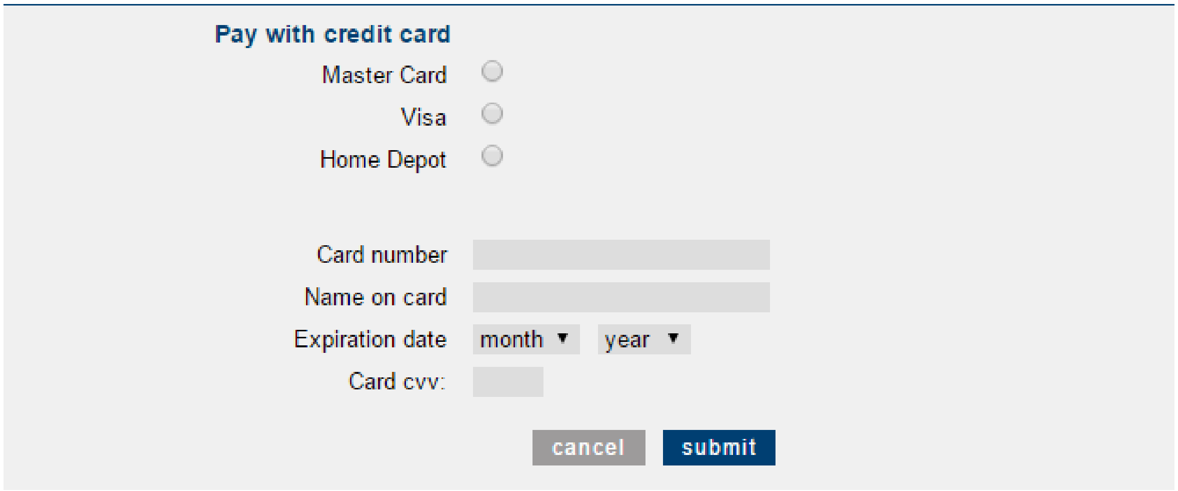
Name on card

Expiration date: month, year

Card cvv:

```
<htmlSection class="payment-content"  xmlns="Paymetric:Intercept:MerchantHtmlPacketModel">
  <tag>
    <tag name="div" class="header-top"/>
    <tag name="div" class="merchant-logo"/>
    <tag name="div" class="main-content">
      <tag name="div" class="billing-content">
        <tag name="div" class="billing-header"></tag>
        <tag name="div" class="billing-info">
          <cardDropdownSection>
            <tag name="div">
              <label for="cardType" text="Card type"/>
              <ddlCardType>
                <items>
                  <item for="american express"/>
                  <item for="mastercard"/>
                  <item for="visa"/>
                  <item for="maestro"/>
                  <customItem for="HD" text="Company A"/>
                </items>
              </ddlCardType>
            </tag>
            <tag name="div">
              <label for="cardNumber" text="Card number"/>
              <tboxCardNumber tokenize="true" class="xi-long-text"/>
              <validationMsg for="cardNumber" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="cardholderName" text="Name on card"/>
              <tboxCardHolderName class="xi-long-text"/>
              <validationMsg for="cardholderName" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="startDate"/>
              <ddlStartMonth display-format="MMM" class="date_combos" required="false"/>
              <ddlStartYear class="date_combos" years-to-display="10" required="false" start-
date="true"/>
              <validationMsg for="startYear" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="expDate" text="Expiration date"/>
              <ddlExpMonth default-text="month" class="date_combos" required="false"/>
              <ddlExpYear default-text="year" class="date_combos" years-to-display="10"
required="false" exp-date="true"/>
              <validationMsg for="expYear" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="cvv" text="Card cvv:"/>
              <tboxCvv class="xi-short-text"/>
              <validationMsg for="cvv" class="valmsg"/>
            </tag>
          </cardDropdownSection>
        </tag>
        <tag name="div" class="payment-button">
          <cancelButton text="cancel" class="cancel-button"/>
          <submitButton text="submit" class="submit-button"/>
        </tag>
      </tag>
    </tag>
    <tag name="div" class="copyright-footer">
      <tag name="div" class="footer-spacing"></tag>
      <tag name="div" class="top-bottom-frame"></tag>
      <tag name="div" class="middle-bottom-frame">
        <tag name="div" class="middle-bottom-frame-content">
          Copyright © 2015. worldpay.com. All Rights Reserved. Various trademarks held by their
respective owners. Legal Notice.
        </tag>
      </tag>
    </tag>
  </tag>
</htmlSection>
```

20.2.10.9 Sample 9: Credit card (radio) with merchant custom card type (company A)



Pay with credit card

Master Card ○

Visa ○

Home Depot ○

Card number

Name on card

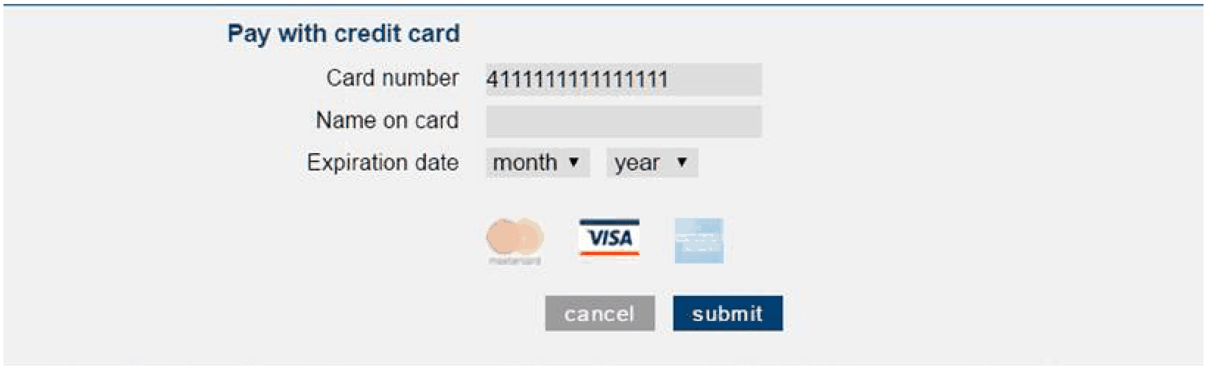
Expiration date month ▼ year ▼

Card cvv:

```
<htmlSection class="payment-content"
xmlns="Paymetric:Intercept:MerchantHtmlPacketModel">
  <tag>
    <tag name="div" class="header-top"/>
    <tag name="div" class="merchant-logo"/>
    <tag name="div" class="main-content">
      <tag name="div" class="billing-content">
        <tag name="div" class="billing-header"></tag>
        <tag name="div" class="billing-info">
          <cardRadioSection>
            <tag name="div">
              <label for="mastercard"/>
              <radCardType for="mastercard"/>
            </tag>
            <tag name="div">
              <label for="visa" text="Visa"/>
              <radCardType for="visa"/>
            </tag>
            <tag name="div">
```

```
              <customLabel for="HD" text="Company A"/>
              <radCustomCardType for="HD"/>
            </tag>
            <voidTag name="br"/>
            <tag name="div">
              <label for="cardNumber" text="Card number"/>
              <tboxCardNumber tokenize="true" class="xi-long-text"/>
              <validationMsg for="cardNumber" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="cardholderName" text="Name on card"/>
              <tboxCardHolderName class="xi-long-text"/>
              <validationMsg for="cardholderName" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="startDate"/>
              <ddlStartMonth display-format="MMM" class="date_combos"
required="false"/>
              <ddlStartYear class="date_combos" years-to-display="10"
required="false" start-date="true"/>
              <validationMsg for="startYear" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="expDate" text="Expiration date"/>
              <ddlExpMonth default-text="month" class="date_combos"
required="false"/>
              <ddlExpYear default-text="year" class="date_combos" years-
to-display="10" required="false" exp-date="true"/>
              <validationMsg for="expYear" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="cvv" text="Card cvv:"/>
              <tboxCvv class="xi-short-text"/>
              <validationMsg for="cvv" class="valmsg"/>
            </tag>
          </cardRadioSection>
        </tag>
        <tag name="div" class="payment-button">
          <cancelButton text="cancel" class="cancel-button"/>
          <submitButton text="submit" class="submit-button"/>
        </tag>
      </tag>
    </tag>
    <tag name="div" class="copyright-footer">
      <tag name="div" class="footer-spacing"></tag>
      <tag name="div" class="top-bottom-frame"></tag>
      <tag name="div" class="middle-bottom-frame">
        <tag name="div" class="middle-bottom-frame-content">
          Copyright © 2015. worldpay.com. All Rights Reserved. Various
trademarks held by their respective owners. Legal Notice.
        </tag>
      </tag>
    </tag>
  </tag>
</htmlSection>
```

20.2.10.10 Sample 10: Credit card with card indicator



Pay with credit card

Card number 4111111111111111

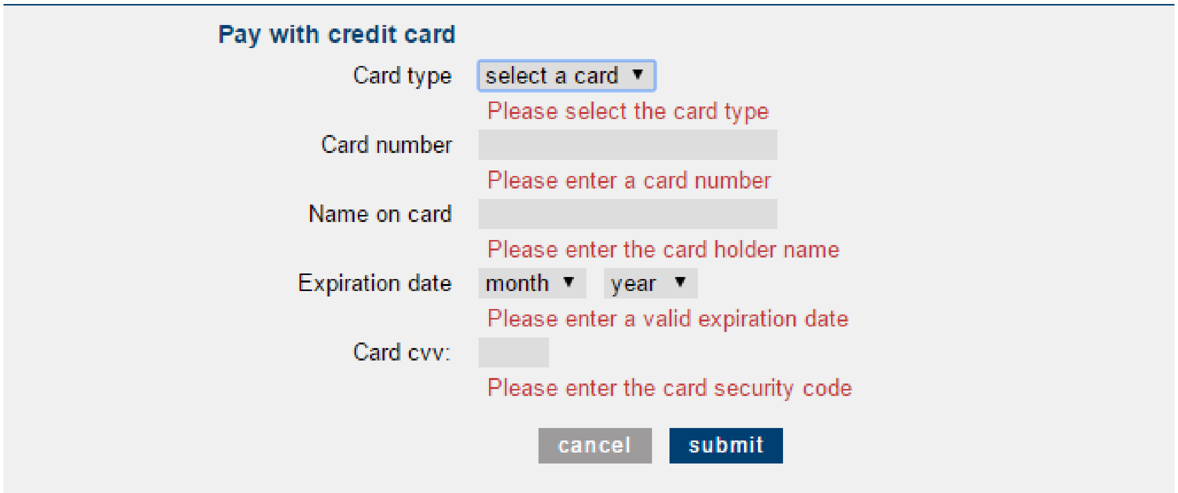
Name on card

Expiration date month ▾ year ▾

```
<htmlSection class="payment-content"
xmlns="Paymetric:Intercept:MerchantHtmlPacketModel">
  <tag>
    <tag name="div" class="header-top"/>
    <tag name="div" class="merchant-logo"/>
    <tag name="div" class="main-content">
      <tag name="div" class="billing-content">
        <tag name="div" class="billing-header"/>
        <tag name="div" class="billing-info">
          <cardIndicatorSection>
            <tag name="div">
              <label for="cardNumber" text="Card number"/>
              <tboxCardNumber tokenize="true" class="xi-long-text"/>
              <validationMsg for="cardNumber" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="cardholderName" text="Name on card"/>
              <tboxCardHolderName class="xi-long-text"/>
              <validationMsg for="cardholderName" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="expDate" text="Expiration date"/>
              <ddlExpMonth default-text="month" class="date_combos"
required="false"/>
              <ddlExpYear default-text="year" class="date_combos" years-to-
display="10" required="false" exp-date="true"/>
              <validationMsg for="expYear" class="valmsg"/>
            </tag>
            <tag name="div">
```

```
              <cardTypeIndicator class="cardIndicator">
                <items>
                  <item for="mastercard" class="mcIndicator" bin-range-
expression="^5" ></item>
                  <item for="visa" class="viIndicator"></item>
                  <item for="american express" class="axIndicator"></item>
                </items>
              </cardTypeIndicator>
            </tag>
          </cardIndicatorSection>
        </tag>
        <tag name="div" class="payment-button">
          <cancelButton text="cancel" class="cancel-button"/>
          <submitButton text="submit" class="submit-button"/>
        </tag>
      </tag>
    </tag>
    <tag name="div" class="copyright-footer">
      <tag name="div" class="footer-spacing"/>
      <tag name="div" class="top-bottom-frame"/>
      <tag name="div" class="middle-bottom-frame">
        <tag name="div" class="middle-bottom-frame-content">Copyright ©
2017. worldpay.com. All Rights Reserved. Various trademarks held by their
respective owners. Legal Notice.</tag>
      </tag>
    </tag>
  </tag>
</htmlSection>
```

20.2.10.11 Sample 11: Input validations: text



Pay with credit card

Card type select a card ▼

Please select the card type

Card number

Please enter a card number

Name on card

Please enter the card holder name

Expiration date month ▼ year ▼

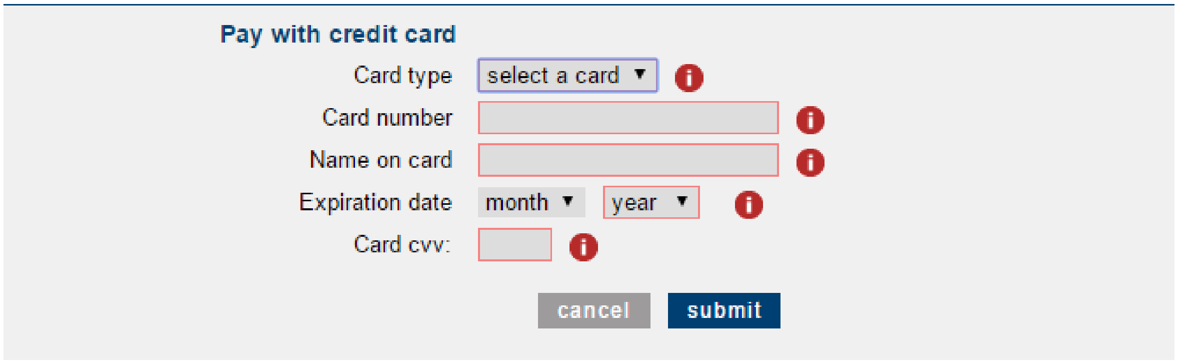
Please enter a valid expiration date

Card cvv:

Please enter the card security code

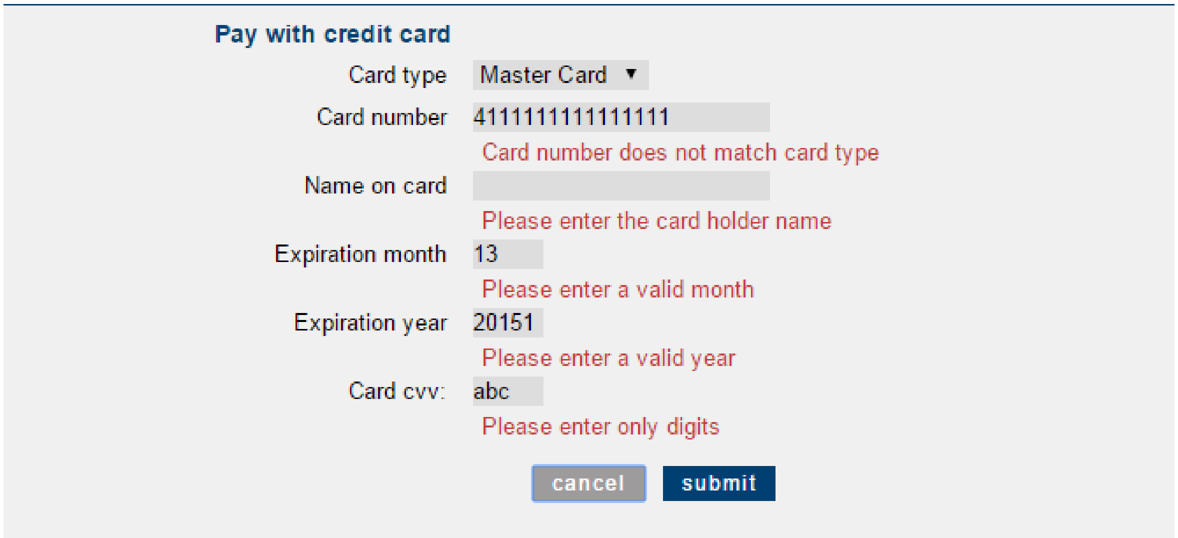
```
<htmlSection class="payment-content"
xmlns="Paymetric:Intercept:MerchantHtmlPacketModel">
  <tag>
    <tag name="div" class="header-top"/>
    <tag name="div" class="merchant-logo"/>
    <tag name="div" class="main-content">
      <tag name="div" class="billing-content">
        <tag name="div" class="billing-header"></tag>
        <tag name="div" class="billing-info">
          <cardDropdownSection>
            <tag name="div">
              <label for="cardType" text="Card type"/>
              <ddlCardType default-text="select a card">
                <items>
                  <item for="american express"/>
                  <item for="mastercard"/>
                  <item for="visa"/>
                  <item for="maestro"/>
                </items>
              </ddlCardType>
              <validationMsg for="cardType" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="cardNumber" text="Card number"/>
              <tboxCardNumber tokenize="true" class="xi-long-text"/>
              <validationMsg for="cardNumber" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="cardholderName" text="Name on card"/>
              <tboxCardHolderName class="xi-long-text"/>
              <validationMsg for="cardholderName" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="startDate"/>
              <ddlStartMonth display-format="MMM" class="date_combos"
required="false"/>
              <ddlStartYear class="date_combos" years-to-display="10"
required="false" start-date="true"/>
              <validationMsg for="startYear" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="expDate" text="Expiration date"/>
              <ddlExpMonth default-text="month" class="date_combos"
required="false"/>
              <ddlExpYear default-text="year" class="date_combos" years-
to-display="10" required="false" exp-date="true"/>
              <validationMsg for="expYear" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="cvv" text="Card cvv:"/>
              <tboxCvv class="xi-short-text"/>
              <validationMsg for="cvv" class="valmsg"/>
            </tag>
          </cardDropdownSection>
        </tag>
        <tag name="div" class="payment-button">
          <cancelButton text="cancel" class="cancel-button"/>
          <submitButton text="submit" class="submit-button"/>
        </tag>
      </tag>
    </tag>
    <tag name="div" class="copyright-footer">
      <tag name="div" class="footer-spacing"></tag>
      <tag name="div" class="top-bottom-frame"></tag>
      <tag name="div" class="middle-bottom-frame">
        <tag name="div" class="middle-bottom-frame-content">
          Copyright © 2015. worldpay.com. All Rights Reserved. Various
trademarks held by their respective owners. Legal Notice.
        </tag>
      </tag>
    </tag>
  </tag>
</htmlSection>
```

20.2.10.12 Sample 12: Input validations: image with Tooltip



```
<htmlSection class="payment-content"
xmlns="Paymetric:Intercept:MerchantHtmlPacketModel">
  <xiProperties error-class="input-error" inline-validation="true">
    <errorTooltip class="errorTooltip" show-effect="fadeIn" show-duration="5000" hide-effect="fadeOut" hide-duration="5000" />
  </xiProperties>
  <tag>
    <tag name="div" class="header-top"/>
    <tag name="div" class="merchant-logo"/>
    <tag name="div" class="main-content">
      <tag name="div" class="billing-content">
        <tag name="div" class="billing-header"></tag>
        <tag name="div" class="billing-info">
          <cardDropdownSection>
            <tag name="div">
              <label for="cardType" text="Card type"/>
              <ddlCardType default-text="select a card" error-msg-style="imageTooltip">
                <items>
                  <item for="american express"/>
                  <item for="mastercard"/>
                  <item for="visa"/>
                  <item for="maestro"/>
                </items>
              </ddlCardType>
              <validationMsg for="cardType" error-class="valmsg-img"/>
            </tag>
            <tag name="div">
              <label for="cardNumber" text="Card number"/>
              <tboxCardNumber tokenize="true" class="xi-long-text" error-msg-style="imageTooltip"/>
              <validationMsg for="cardNumber" error-class="valmsg-img"/>
            </tag>
            <tag name="div">
              <label for="cardholderName" text="Name on card"/>
              <tboxCardHolderName class="xi-long-text" error-msg-style="imageTooltip"/>
              <validationMsg for="cardholderName" error-class="valmsg-img"/>
            </tag>
            <tag name="div">
              <label for="startDate"/>
              <ddlStartMonth display-format="MMM" class="date_combos" required="false"/>
              <ddlStartYear class="date_combos" years-to-display="10" required="false" start-date="true" error-msg-style="imageTooltip"/>
              <validationMsg for="startYear" error-class="valmsg-img"/>
            </tag>
            <tag name="div">
              <label for="expDate" text="Expiration date"/>
              <ddlExpMonth default-text="month" class="date_combos" required="false"/>
              <ddlExpYear default-text="year" class="date_combos" years-to-display="10" required="false" exp-date="true" error-msg-style="imageTooltip"/>
              <validationMsg for="expYear" error-class="valmsg-img"/>
            </tag>
            <tag name="div">
              <label for="cvv" text="Card cvv:"/>
              <tboxCvv class="xi-short-text" error-msg-
style="imageTooltip"/>
              <validationMsg for="cvv" error-class="valmsg-img"/>
            </tag>
          </cardDropdownSection>
        </tag>
        <tag name="div" class="payment-button">
          <cancelButton text="cancel" class="cancel-button"/>
          <submitButton text="submit" class="submit-button"/>
        </tag>
      </tag>
    </tag>
    <tag name="div" class="copyright-footer">
      <tag name="div" class="footer-spacing"></tag>
      <tag name="div" class="top-bottom-frame"></tag>
      <tag name="div" class="middle-bottom-frame">
        <tag name="div" class="middle-bottom-frame-content">
          Copyright © 2015. worldpay.com. All Rights Reserved. Various
trademarks held by their respective owners. Legal Notice.
        </tag>
      </tag>
    </tag>
  </tag>
</htmlSection>
```

20.2.10.13 Sample 13: Custom validations



Pay with credit card

Card type: Master Card ▼

Card number: 4111111111111111

Card number does not match card type

Name on card:

Please enter the card holder name

Expiration month: 13

Please enter a valid month

Expiration year: 20151

Please enter a valid year

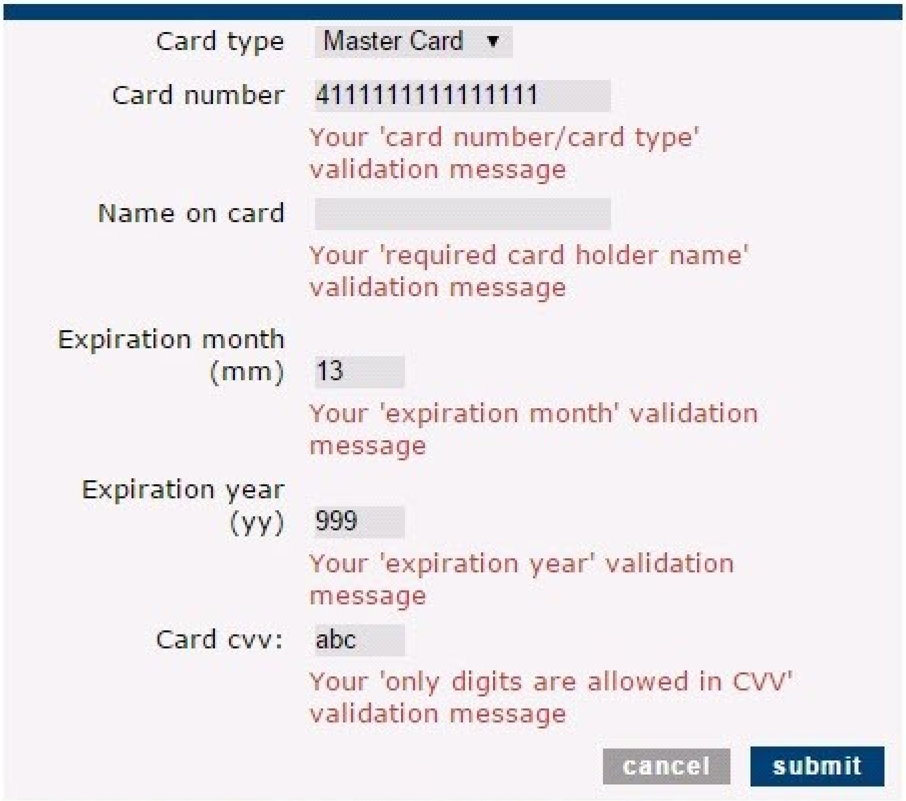
Card cvv: abc

Please enter only digits

```
<htmlSection class="payment-content"
xmlns="Paymetric:Intercept:MerchantHtmlPacketModel">
  <tag>
    <tag name="div" class="header-top"/>
    <tag name="div" class="merchant-logo"/>
    <tag name="div" class="main-content">
      <tag name="div" class="billing-content">
        <tag name="div" class="billing-header"></tag>
        <tag name="div" class="billing-info">
          <cardDropdownSection>
            <tag name="div">
              <label for="cardType" text="Card type"/>
              <ddlCardType default-text="select a card">
                <items>
                  <item for="american express"/>
                  <item for="mastercard"/>
                  <item for="visa"/>
                  <item for="maestro"/>
                </items>
              </ddlCardType>
              <validationMsg for="cardType" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="cardNumber" text="Card number"/>
              <tboxCardNumber tokenize="true" class="xi-long-text" luhn-check="false"/>
              <validationMsg for="cardNumber" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="cardholderName" text="Name on card"/>
              <tboxCardHolderName class="xi-long-text"/>
              <validationMsg for="cardholderName" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="startDate"/>
              <ddlStartMonth display-format="MMM" class="date_combos" required="false"/>
              <ddlStartYear class="date_combos" years-to-display="10" required="false" start-date="true"/>
              <validationMsg for="startYear" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="expMonth" text="Expiration month"/>
              <tboxExpMonth class="xi-short-text"/>
              <validationMsg for="expMonth" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="expYear" text="Expiration year"/>
              <tboxExpYear pattern="[0-9]{4}" pattern-msg="Please enter a valid year" class="xi-short-text"/>
```

```
       <validationMsg for="expYear" class="valmsg"/>
      </tag>
      <tag name="div">
       <label for="cvv" text="Card cvv:"/>
       <tboxCvv class="xi-short-text" minlength="3" maxlength="3"
minlength-msg="CVV must be 3 digits" maxlength-msg="CVV must be 3 digits"
 digits-only="true"/>
       <validationMsg for="cvv" class="valmsg"/>
      </tag>
     </cardDropdownSection>
    </tag>
    <tag name="div" class="payment-button">
     <cancelButton text="cancel" class="cancel-button"/>
     <submitButton text="submit" class="submit-button"/>
    </tag>
   </tag>
  </tag>
  <tag name="div" class="copyright-footer">
   <tag name="div" class="footer-spacing"></tag>
   <tag name="div" class="top-bottom-frame"></tag>
   <tag name="div" class="middle-bottom-frame">
    <tag name="div" class="middle-bottom-frame-content">
     Copyright © 2015. worldpay.com. All Rights Reserved. Various
trademarks held by their respective owners. Legal Notice.
    </tag>
   </tag>
  </tag>
 </tag>
</htmlSection>
```

20.2.10.14 Sample 14: Custom messages



Card type	Master Card ▼
Card number	4111111111111111 Your 'card number/card type' validation message
Name on card	 Your 'required card holder name' validation message
Expiration month (mm)	13 Your 'expiration month' validation message
Expiration year (yy)	999 Your 'expiration year' validation message
Card cvv:	abc Your 'only digits are allowed in CVV' validation message

```
<htmlSection class="payment-content"
xmlns="Paymetric:Intercept:MerchantHtmlPacketModel">
  <tag>
    <tag name="div" class="header-top"/>
    <tag name="div" class="merchant-logo"/>
    <tag name="div" class="main-content">
      <tag name="div" class="billing-content">
        <tag name="div" class="billing-header"></tag>
        <tag name="div" class="billing-info">
          <cardDropdownSection>
            <tag name="div">
              <label for="cardType" text="Card type"/>
              <ddlCardType>
                <items>
                  <item for="american express"/>
                  <item for="mastercard"/>
                  <item for="visa"/>
                  <item for="maestro"/>
                </items>
              </ddlCardType>
              <validationMsg for="cardType" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="cardNumber" text="Card number"/>
              <tboxCardNumber tokenize="true" class="xi-long-text" luhn-
check-msg="Your 'luhn-check' validation message" pattern-msg="Your 'card
number/card type' validation message"/>
              <validationMsg for="cardNumber" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="cardholderName" text="Name on card"/>
              <tboxCardHolderName class="xi-long-text" required-msg="Your
'required card holder name' validation message"/>
              <validationMsg for="cardholderName" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="startDate"/>
              <ddlStartMonth display-format="MMM" class="date_combos"
required="false"/>
              <ddlStartYear class="date_combos" years-to-display="10"
required="false" start-date="true"/>
              <validationMsg for="startYear" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="expMonth" text="Expiration month (mm)"/>
              <tboxExpMonth class="xi-short-text" range-msg="Your
'expiration month' validation message"/>
              <validationMsg for="expMonth" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="expYear" text="Expiration year (yy)"/>
              <tboxExpYear range="0,99" range-msg="Your 'expiration year'
validation message" class="xi-short-text" exp-date="false"/>
              <validationMsg for="expYear" class="valmsg"/>
            </tag>
            <tag name="div">
              <label for="cvv" text="Card cvv:"/>
              <tboxCvv class="xi-short-text" minlength="3" maxlength="3"
minlength-msg="Your 'CVV must be 3 digits' validation message" maxlength-
msg="Your 'CVV must be 3 digits' validation message"
```

```
                       digits-only="true" digits-only-msg="Your 'only
digits are allowed in CVV' validation message"/>
              <validationMsg for="cvv" class="valmsg"/>
            </tag>
          </cardDropdownSection>
        </tag>
        <tag name="div" class="payment-button">
          <cancelButton text="cancel" class="cancel-button"/>
          <submitButton text="submit" class="submit-button"/>
        </tag>
      </tag>
    </tag>
    <tag name="div" class="copyright-footer">
      <tag name="div" class="footer-spacing"></tag>
      <tag name="div" class="top-bottom-frame"></tag>
      <tag name="div" class="middle-bottom-frame">
        <tag name="div" class="middle-bottom-frame-content">
          Copyright © 2016. worldpay.com. All Rights Reserved. Various
trademarks held by their respective owners. Legal Notice.
        </tag>
      </tag>
    </tag>
  </tag>
</htmlSection>
```