Date :  26-Jan-22

Apple Pay (2.0.2)

# Overview

Apple Pay is a mobile payment and digital wallet service that facilitates fast and easy payments for users and also eliminates the need to manually enter payment and shipping information. The Apple Pay component integrates the payment feature with Voltmx. Under the hood, the component uses the [Apple Pay Passkit API](https://developer.apple.com/documentation/passkit).

You can configure the currency, the country, and the merchant information. You can also select the payment networks and payment methods that you want to support in your app. You can use the component is scenarios such as: A retail app for purchasing products, in which you want to provide an option to the user to make payments by using the Apple Pay services.

## Use case

### Apple Pay can be after a booking request to complete the payment for that booking.

### Apple Pay can be also used when support service is called and user should pay for that support service.

## Percentage of re-use:

85%-90% (Data can be customizable and skins are customized and also can be changed manually).

# Getting Started

## Prerequisites

Before you start using the component, ensure you have the following:

* [HCL Foundry](https://manage.hclvoltmx.com/)
* Volt MX Iris
* A registered Merchant ID that is associated with your Apple Developer account. For more information, refer to [Setting Up Apply Pay Requirements](https://developer.apple.com/documentation/passkit/apple_pay/setting_up_apple_pay) in the Apple Pay Documentation.
* iOS: iOS 11.0, or later
* Active Internet Connection

 Note: Make sure that you configure the following properties before you build your app:

[ISO 4217 Currency Code](https://docs.kony.com/marketplace/V8Marketplace/Content/Marketplace/applepay.htm#currencyCode)

[Merchant Identifier](https://docs.kony.com/marketplace/V8Marketplace/Content/Marketplace/applepay.htm#merchantIdentifier)

[ISO 3316 Country Code](https://docs.kony.com/marketplace/V8Marketplace/Content/Marketplace/applepay.htm#countryCode)

[Payment Summary Items](https://docs.kony.com/marketplace/V8Marketplace/Content/Marketplace/applepay.htm#paymentSummaryItems)

[Merchant Capabilities](https://docs.kony.com/marketplace/V8Marketplace/Content/Marketplace/applepay.htm#merchantCapabilities): It is mandatory to support 3-D Secure Protocol

## Platforms Supported

#### Iphone Mobile

#### Ipad

## Importing the app

 You can import the Forge components only into the apps that are of the Reference Architecture type.

**To import the Apple Pay component, do the following:**

1. Open your app project in Volt MX Iris.
2. In the Project Explorer, click the **Templates** tab.

Graphical user interface, text, application

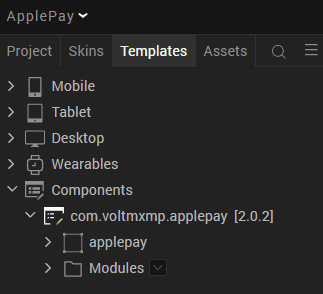
Description automatically generated

1. Right-click **Components**, and then select **Import Component**. The **Import Component** dialog box appears.

Graphical user interface, text, application, Teams

Description automatically generated

1. Click **Browse** to navigate to the location of the component, select the component, and then click **Import**. The component and its associated widgets and modules are added to your project.



Once you have imported a component to your project, you can easily add the component to a form. For more information, refer [Add a Component to a Form](https://opensource.hcltechsw.com/volt-mx-docs/docs/documentation/Iris/iris_user_guide/Content/C_UsingComponents.html#add-a-component-to-a-form).

After adding a component to a form, you can configure the component the way you want it using the **Look**, **Skin**, and **Action** tabs on the **Properties** pane. Configuring the properties on the **Properties** pane is similar to configuring the properties of any widget in Volt MX Iris.

You can also see that a new tab, **Component**, is added on the **Properties** pane. The **Component** tab contains assorted properties relevant to the component that allow you to customize the component as required. The properties on the **Component** tab are categorized as **General**, **Axis Titles**, **Grid** and **Title** properties. The **General** properties are the default properties of individual widgets in the component.

# References

## Dynamic Usage

You can also add Apple Pay component dynamically. To do so:

1. In the **Project Explorer**, on the **Projects** tab, click **Controllers** section to access the respective **formController**. Create a method and implement the code snippet similar to the sample code mentioned below.

/\* Creating the component s object \*/

var applepay = new com.voltmxmp.applepay(

{

"id":"applepayDynamic",

"top":"40%",

"width":"100%",

"height":"20%",

"left":"32%"

},{}, {});

/\* Setting the component's properties \*/

applepay.currencyCode = "USD";

applepay.merchantIdentifier = "merchant.com.jackinthebox.jibpush";

applepay.merchantCapabilities =

{

"data":

[

{"capability":"3DSecureprotocol"},

{"capability" : "EMV protocol"}

]

};

applepay.countryCode = "US";

applepay.paymentSummaryItems =

{

"data":

[

{

"amount": "5.20",

"label": "Order Amount"

},

{

"amount": "10.00",

"label": "Shipping Cost"

},

{

"amount": "-1.00",

"label": "discount"

}

]

};

applepay.shippingMethods =

{

"data":

[

{

"amount": "10.00",

"detail": "guarantees that the order will by delivered by the next Day",

"identifier": "oneDayDelivery",

"name": "one Day Delivery"

},

{

"amount": "5.00",

"detail": "guarantees that the order will by delivered by the next two Day",

"identifier": "twoDayDeliver",

"name": "two day Delviery"

},

{

"amount": "0.00",

"detail": "This shipping Method usually take 1 week for Delivery",

"identifier": "NormalDeliver",

"name": "Normal Delivery"

}

]

};

applepay.supportedNetworks =

{

"data":

[

{"paymentNetwork": "American Express"},

{"paymentNetwork": "Cartes Bancaires"},

{"paymentNetwork": "China Union Pay"},

{"paymentNetwork": "Discover"},

{"paymentNetwork": "iD"},

{"paymentNetwork": "Interac"},

{"paymentNetwork": "JCB"},

{"paymentNetwork": "MasterCard"},

{"paymentNetwork": "Store credit and debit cards"},

{"paymentNetwork": "QUICPay"},

{"paymentNetwork": "Suica"},

{"paymentNetwork": "Visa"}

]

};

applepay.supportedCountries =

{

"data":

[

{"country": "US"}

]

};

applepay.isbillingAddressRequired = true;

applepay.isShippingAddressRequired = true;

applepay.isPhoneNumberRequired = true;

applepay.isEmailAddressRequired = true;

applepay.isContactNameRequired = true;

applepay.isContactPhoneticNameRequired = true;

applepay.type="buy";

applepay.style="black";

/\* Adding the component to a form \*/

this.view.add(applepay);

In the code snippet, you can edit the properties of the component as per your requirement.

1. Save the file.
2. Deploying your Application

Configuring your Environment

You need to create a Merchant Identifier for the application. For more information, refer to the [Configuring your Environment](https://developer.apple.com/library/content/ApplePay_Guide/Configuration.html) section in the Apple Pay Programming Guide.

During the binary or IPA creation process, after extracting the .kar file and creating the XCode project from the Plugin, you need to turn on the Apple Pay option. You can find this option in XCode's KRelease section, under the Capabilities Tab.

## Properties

The properties provided on the **Component** tab allow you to customize the UI elements in the Apple Pay component. You can set the properties directly on the **Component** tab or by writing a JavaScript. This section provides information on how to set the properties by writing JavaScript.

[ISO 4217 Currency Code](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Specifies the currency that Apple Pay interprets while providing the summary items. |
| **Syntax:** | currencyCode |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Default Value:** | USD |
| **Remarks:** | * This field is mandatory. * Make sure that you use the three-letter **ISO 4217** currency code. |
| **Example:** | this.view.componentID.currencyCode = "USD"; |
|  |  |

[Merchant Identifier](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Specifies the Merchant ID that is registered to the application bundle identifier.  For more information, you can refer to [Apple Pay Requirements](https://developer.apple.com/documentation/passkit/apple_pay/setting_up_apple_pay_requirements) in the Apple Pay Documentation. |
| **Syntax:** | merchantIdentifier |
| **Type:** | String |
| **Read/Write:** | Read + Write |
| **Remarks:** | * This field is mandatory. * Make sure that this value matches one of the identifiers specified by the com.apple.developer.in-app-payments key in the app's entitlements. |
| **Example:** | this.view.componentID.merchantIdentifier = "merchant.com.voltmxmp.test"; |

[ISO 3316 Country Code](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Specifies the country in which the payment is processed. |
| **Syntax:** | countryCode |
| **Type:** | * Data Grid * JSON |
| **Read/Write:** | Read + Write |
| **Default Value:** | US |
| **Remarks:** | * This field is mandatory. * You need to set the two-letter **ISO 3316** currency code. |
| **Example:** | this.view.componentID.countryCode = "US"; |

[Payment Summary Items](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Specifies the items that summarize the amount of the payment. Summary items include the **order total**, **shipping cost**, **taxes**, and the **grand total**. |
| **Syntax:** | paymentSummaryItems |
| **Type:** | * Data Grid * JSON |
| **Read/Write:** | Read + Write |
| **Remarks:** | * This field is mandatory. * Every item in the grid contains the following keys:   + label: The name of the summary item   + amount: The value of the summary item * You can enter the amount for an item as negative, but make sure that the total sum of all amounts is greater than zero. * The component automatically calculates the grand total and displays it to the user. |
| **Example:** | this.view.componentID.paymentSummaryItems =  {  "data":  [  {  "amount": "125.23",  "label": "Order Amount"  },  {  "amount": "-5.00",  "label": "Shipping Cost"  },  {  "amount": "10.00",  "label": "discount"  }  ]  }; |

[Shipping Methods](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Specifies the shipping method that delivers the physical goods. |
| **Syntax:** | shippingMethods |
| **Type:** | * Data Grid * JSON |
| **Read/Write:** | Read + Write |
| **Remarks:** | * Every item in the grid contains the following keys:   + amount: The cost of the shipping method.   + detail: The description of the shipping method.   + name The summary of the shipping method.   + identifier: The unique identifier of the shipping method |
| **Example:** | this.view.componentID.shippingMethods =  {  "data":  [  {  "amount": "10.00",  "detail": "Guarantees that the order will by delivered by the next Day",  "identifier": "oneDayDelivery",  "name": "One Day Delivery"  },  {  "amount": "5.00",  "detail": "Guarantees that the order will by delivered in the next two days",  "identifier": "twoDayDelivery",  "name": "Two day Delivery"  },  {  "amount": "0.00",  "detail": "This shipping Method usually take 1 week for Delivery",  "identifier": "normalDelivery",  "name": "Normal Delivery"  }  ]  }; |

[Supported Networks](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Specifies the payment cards that you want to support for the transactions in your app. |
| **Syntax:** | supportedNetworks |
| **Type:** | * Data Grid * JSON |
| **Read/Write:** | Read + Write |
| **Values:** | * American Express * Cartes Bancaires * China Union Pay * Discover * iD * Interac * JCB * MasterCard * Store credit and debit cards * QUICPay * Suica * Visa |
| **Remarks:** | * By default, the component enables all the supported networks. * **Store credit and debit cards** refers to the **privateLabel** payment network. For more information, you can refer to [PKPaymentNetwork](https://developer.apple.com/documentation/passkit/pkpaymentnetwork" \t "_blank) in the Apple Pay documentation. |
| **Example:** | this.view.componentID.supportedNetworks =  {  "data":  [  {"paymentNetwork": "American Express"},  {"paymentNetwork": "MasterCard"},  {"paymentNetwork": "Visa"}  ]  }; |

[Billing Address Required](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Specifies whether the user needs to add a billing address to complete payment. |
| **Syntax:** | isbillingAddressRequired |
| **Type:** | Boolean |
| **Read/Write:** | Read + Write |
| **Default Value:** | False |
| **Example:** | this.view.componentID.isbillingAddressRequired = true; |

[Shipping Address Required](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Specifies whether the user needs to add a shipping address to complete payment. |
| **Syntax:** | isShippingAddressRequired |
| **Type:** | Boolean |
| **Read/Write:** | Read + Write |
| **Default Value:** | False |
| **Example:** | this.view.componentID.isShippingAddressRequired = true; |

[Phone Number Required](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Specifies whether the user needs to add a phone number to complete the shipping process. |
| **Syntax:** | isPhoneNumberRequired |
| **Type:** | Boolean |
| **Read/Write:** | Read + Write |
| **Default Value:** | False |
| **Example:** | this.view.componentID.isPhoneNumberRequired = true; |

[Email Address Required](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Specifies whether the user needs to add an email address to complete the shipping process. |
| **Syntax:** | isEmailAddressRequired |
| **Type:** | Boolean |
| **Read/Write:** | Read + Write |
| **Default Value:** | False |
| **Example:** | this.view.componentID.isEmailAddressRequired = true; |

[Contact Name Required](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Specifies whether the user needs to provide a contact name to complete the shipping process. |
| **Syntax:** | isContactNameRequired |
| **Type:** | Boolean |
| **Read/Write:** | Read + Write |
| **Default Value:** | False |
| **Remarks:** | If you set the [Shipping Address Required](https://docs.kony.com/marketplace/V8Marketplace/Content/Marketplace/applepay.htm#shipping) property to **true**, then the component does not display this field. |
| **Example:** | this.view.componentID.isContactNameRequired = true; |

[Contact Phonetic Name Required](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Specifies whether the user needs to provide a contact's phonetic name to complete the shipping process. |
| **Syntax:** | isContactPhoneticNameRequired |
| **Type:** | Boolean |
| **Read/Write:** | Read + Write |
| **Default Value:** | False |
| **Remarks:** | If you set the [Shipping Address Required](https://docs.kony.com/marketplace/V8Marketplace/Content/Marketplace/applepay.htm#shipping) property to **true**, then the component does not display this field. |
| **Example:** | this.view.componentID.isContactPhoneticNameRequired = true; |

[Merchant Capabilities](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Specifies the processing protocols that you want to support for the payments in your app. |
| **Syntax:** | merchantCapabilities |
| **Type:** | * Data Grid * JSON |
| **Read/Write:** | Read + Write |
| **Values:** | * 3DSecureProtocol * EMV protocol * credit cards * debit cards |
| **Remarks:** | * By default, the component enables all the merchant capabilities. * It is mandatory to support the 3-D Secure Protocol. |
| **Example:** | this.view.componentID.merchantCapabilities =  {  "data":  [  {"capability": "3DSecureprotocol"},  {"capability": "EMV protocol"},  {"capability": "credit cards"},  {"capability": "debit cards"}  ]  }; |

[Supported Countries](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Limits the payment options to cards that are issued in the specified countries.  For example, debit cards may accept transactions only in the country that issues the card. |
| **Syntax:** | supportedCountries |
| **Type:** | * Data Grid * JSON |
| **Read/Write:** | Read + Write |
| **Default Value:** | {  "data":  [  {"country": "US"}  ]  }; |
| **Remarks:** | You need to provide the list of supported countries as **ISO 3166** Codes. |
| **Example:** | this.view.componentID.supportedCountries =  {  "data":  [  {"country": "US"}  ]  }; |

#### **Apple Pay Button**

The following properties affect the Apple Pay button. The button is used either to trigger payments through Apple Pay or to prompt the user to set up a card.

[Type](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Specifies the type of button you want to render to start the Apple Pay process. For more information, you can refer to [PKPaymentButtonType](https://developer.apple.com/documentation/passkit/pkpaymentbuttontype?language=objc" \t "_blank) in the Apple Pay Documentation. |
| **Syntax:** | Type |
| **Type:** | * String * List Selector |
| **Read/Write:** | Read + Write |
| **Values:** | * addmoney * book * buy * checkout * contribute * donate * order * paywith * plain * reload * rent * setup * subscribe * support * tip |
| **Default Value:** | buy |
| **Example:** | this.view.componentID.type = "plain"; |

[Style](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Specifies the style that you want to apply on the Apple Pay button. For more information, you can refer to the [Apple Pay Documentation](https://developer.apple.com/documentation/passkit/pkpaymentbuttonstyle?language=objc). |
| **Syntax:** | style |
| **Type:** | * String * List Selector |
| **Read/Write:** | Read + Write |
| **Values:** | * Automatic * white * whiteoutline * black |
| **Default Value:** | automatic |
| **Example:** | this.view.componentID.style = "black"; |

## Events

[onAuthorizationCompletion](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Invoked when the user validates the details and authenticates the transaction by Touch ID, Face ID, or by entering the passcode. |
| **Parameters:** | *paymentInformation:* This Parameter provides the encrypted payment credentials, billing and shipping information. Payment Information contains the following keys.  [Token:](javascript:void(0);)  This contains the user's encrypted payment credentials. For more information, refer to [PKPaymentToken](https://developer.apple.com/documentation/passkit/pkpaymenttoken" \t "_blank) in the Apple Pay Documentation.  **Data Type**[PKPayment](https://developer.apple.com/documentation/passkit/pkpayment)  **Syntax**token paymentInformation.token  [Billing Information:](javascript:void(0);)  This contains the user's Billing Address and their name.  **Data Type***JSON*  **Syntax**billingInformation paymentInformation.billingInformation  **Sample JSON**  {  "postalAddress":  {  "street": "10th Avenue",  "subAdministrativeArea": "",  "subLocality": "",  "city": "Bristol",  "state": "Washington",  "postalCode": "90210",  "country": "United States",  "ISOCountryCode": "US"  },  "name":  {  "givenName": "FirstName",  "namePrefix": "",  "middleName": "",  "familyName": "LastName",  "nameSuffix": "",  "nickname": "",  "phoneticRepresentation": {}  }  }  [Shipping Information:](javascript:void(0);)  This contains the user's Shipping Address and other related information such as their name, phone number, and email address.  **Data Type***JSON*  **Syntax**shippingInformation paymentInformation.shippingInformation  **Sample JSON**  {  "emailAddress": "test.account@gmail.com",  "phoneNumber": "9876543210",  "postalAddress":  {  "street": "10th Avuenue",  "subAdministrativeArea": "",  "subLocality": "",  "city": "Bristol",  "state": "Washingtonn",  "postalCode": "90210",  "country": "United States",  "ISOCountryCode": "US"  },  "name":  {  "givenName": "FirstName",  "namePrefix": "",  "middleName": "",  "familyName": "LastName",  "nameSuffix": "",  "nickname": "",  "phoneticRepresentation": {}  }  }  [shippingMethod:](javascript:void(0);)  This contains information about the shipping method selected by the user.  **Data Type***JSON*  **Syntax**shippingMethod paymentInformation.shippingMethod  **Sample JSON**  {  "detail": "Guarantees that the order will be delivered by the next day.",  "identifier": "oneDayDelivery"  } |
| **Available Methods:** | *completionCallback* This callback is a parameter to the **onAuthorizationCompletion** event, along with the **paymentInformation**. You need to call this method when the user completes the transaction with the token generated by Apple Pay. You should pass the payment status and any available errors with this call.You need to pass the following parameters to this method.  [paymentStatus:](javascript:void(0);)  This specifies whether the transaction is successful, or it failed.  **Data Type***Boolean*  **Remarks**This value is **true** if the transaction is successful, and **false** if the transaction fails.  [errors:](javascript:void(0);)  This is an array of the errors caused due to user inputs.  **Error:** It is a JSON containing the following keys.  **Type**The types of errors can be   * shippingAddress * billingAddress * contactField * Custom   **fieldName**This value represents the exact key that caused the error. For example, **street** in **billingAddress**, or **email** in **shippingAddress**, and so on.**Message**This is a custom description that you can display to the user when an error occurs.  [Sample Code:](javascript:void(0);)  /\*  Component Consumer logic  The specified code is a sample implementation  \*/  this.view.componentID.onAuthorizationCompletion = function(paymentInformation, completionCallback){  var NSJSONSerialization = objc.import("NSJSONSerialization");  payData = NSJSONSerialization.JSONObjectWithDataOptionsError(paymentInformation.token.paymentData, NSJSONReadingAllowFragments, null);  var summaryItems = this.paymentSummaryItems;  summaryItems.data[1].amount = "10.00";  /\*send the payData in json format to your backend with base 64 encode  and your backend will deal with payment data and response it.  Refer to https://developer.apple.com/library/archive/documentation/PassKit/Reference/PaymentTokenJSON/PaymentTokenJSON.html\*/  completionCallback();  }; |

[onPaymentMethodSelection](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Invoked when the user switches the payment method, that is, switches any card. |
| **Parameters:** | *paymentMethod:* This Parameter gives information about the selected payment method.The value of the payment method type is as follows:   * **PaymentMethodTypeUnknown** value is **0** * **PaymentMethodTypeDebit** value is **1** * **PaymentMethodTypeCredit** value is **2** * **PaymentMethodTypePrepaid** value is **3** * **PaymentMethodTypeStore** value is **4**   **Data Type***JSON*  **Sample JSON**  {  "type": 1  } |
| **Available Methods:** | *completionCallback* This callback is a parameter to the **onPaymentMethodSelection** event, along with the **paymentMethod**. You need to call this method when the user completes changing the payment method. You may also need to change the payment summary items.  **Data Type***Method*  **Parameters***Updated Payment Summary [Array of JSON]* *The object containing all the updated payment summary items.*  **Remarks**If the user doesn't want to make any changes, you can send the existing payment summary items as a parameter.  [Sample Code:](javascript:void(0);)  /\*  Component Consumer logic  The specified code is a sample implementation  \*/  this.view.componentID.onPaymentMethodSelection = function(selectedPaymentMethod, completionCallback){  var summaryItems = this.paymentSummaryItems;  switch (selectedPaymentMethod.type) {  case 0:  summaryItems.data[2].amount = "-0.50";  break;  case 1:  summaryItems.data[2].amount = "-2.00";  break;  case 2:  summaryItems.data[2].amount = "-2.00";  break;  case 3:  summaryItems.data[2].amount = "-2.00";  break;  case 4:  summaryItems.data[2].amount = "-2.00";  break;  }  completionCallback(summaryItems);  }; |

[onShippingAddressSelection](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Invoked when the user confirms the shipping address. |
| **Parameters:** | *shippingAddress:* This Parameter gives information about the selected shipping address.  **Data Type***JSON*  **Sample JSON**  {  "city": "Bristol",  "state": "Washington",  "postalCode": "90210",  "country": "United States",  "ISOCountryCode": "US"  } |
| **Available Methods:** | *completionCallback* This callback is a parameter to the **onShippingAddressSelection** event, along with the **shippingAddress**. You need to call this method when the user completes changing the shipping address. You may also need to change the payment summary items and shipping methods according to the selected shipping address ID. You can also create an error and send it to the completion callback if you don't want to support the selected shipping address.  **Data Type***Method*  **Parameters**   * **Error:** This is a JSON object that contains an error message that shows up if the merchant doesn't support the shipping address. * { * "message": "Shipping to the current address is unavailable"   }   * **summaryItems:** This is a JSON containing the updated summary items. * **shippingMethods:** This is a JSON containing the updated shipping methods.   [Sample Code:](javascript:void(0);)  /\*  Component Consumer logic  The specified code is a sample implementation  \*/  this.view.componentID.onShippingAddressSelection = function(selectedShippingAddress,completionCallback){  var summaryItems = this.paymentSummaryItems;  var shippingMethods = this.shippingMethods;  var error = null;  if(selectedShippingAddress.country !== "United States")  {  error =  {  "message": "shipping cannot be possible to this address"  };  }  completionCallback(error, summaryItems, shippingMethods);  }; |

[onShippingMethodSelection](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Invoked when the user confirms the shipping method. |
| **Parameters:** | *shippingMethod:* This Parameter gives information about the selected shipping method.  **Data Type***JSON*  **Sample JSON**  {  "detail": "guarantees that the order will by delivered by the next Day",  "identifier": "oneDayDelivery"  } |
| **Available Methods:** | *completionCallback* This callback is a parameter to the **onShippingMethodSelection** event, along with the **shippingMethod**. You need to call this method when the user completes changing the shipping method. You may also need to change the payment summary items according to the selected shipping method.  **Data Type***Method*  **Parameters***Updated Payment Summary [Array of JSON]* *The object containing all the updated payment summary items.*  **Remarks**If the user doesn't want to make any changes, you can send the existing payment summary items as a parameter.  [Sample Code:](javascript:void(0);)  /\*  Component Consumer logic  The specified code is a sample implementation  \*/  this.view.componentID.onShippingMethodSelection = function(selectedShippingMethod, completionCallback){  var summaryItems = this.paymentSummaryItems;  switch (selectedShippingMethod.identifier) {  case "oneDayDelivery":  summaryItems.data[1].amount = "10.00";  break;  case "twoDayDeliver":  summaryItems.data[1].amount = "5.00";  break;  case "NormalDeliver":  summaryItems.data[1].amount = "0.00";  break;  }  completionCallback(summaryItems);  }; |

[onApplePayDismiss](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Invoked when the user dismisses the apple pay view. The component uses this event to update any other state. |
| **Parameters:** | None |

.

## API’s

[startPayment](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Starts the payment procedure. |
| **Syntax:** | startPayment() |
| **Parameters:** | None |
| **Return Value:** | None |
| **Example:** | this.view.componentID.startPayment(); |

[isApplePaySupported](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Confirms whether Apple Pay is supported in the current region of the device. |
| **Syntax:** | isApplePaySupported() |
| **Parameters:** | None |
| **Return Value:** | *applePaySupport [Boolean]* : Specifies whether Apple Pay is supported in the current region of the device. |
| **Example:** | var applePaySupport = this.view.componentID.isApplePaySupported();  if(applePaySupport === true)  {  alert("Apple Pay is supported.");  }  else  {  alert("Apple Pay is not supported.");  } |

[availableNetworks](javascript:void(0);)

|  |  |
| --- | --- |
| **Description:** | Returns the payment methods (networks) that are supported by Apple Pay. |
| **Syntax:** | availableNetworks() |
| **Parameters:** | None |
| **Return Value:** | *AvailableNetworks [JSON]* : Specifies whether Apple Pay is supported in the current region of the device. |
| **Example:** | var AvailableNetworks = this.view.componentID.availableNetworks();  alert("Available Networks are: " + JSON.stringify(AvailableNetworks)); |
| **Sample Return JSON:** | {  "data":  [  {"paymentNetwork": "americanexpress"},  {"paymentNetwork": "cartesbancaires"},  {"paymentNetwork": "chinaunionpay"},  {"paymentNetwork": "discover"},  {"paymentNetwork": "id"},  {"paymentNetwork": "interac"},  {"paymentNetwork": "jcb"},  {"paymentNetwork": "mastercard"},  {"paymentNetwork": "storecreditanddebitcards"},  {"paymentNetwork": "quicpay"},  {"paymentNetwork": "suica"},  {"paymentNetwork": "visa"},  ]  } |

.

# Revision History

App version 2.0.2:

## Known Issues

None

## Limitations

If user change any selection in the billing page, for example shipping method, and then complete the payment or cancel the billing page, next time user open the billing page again he will be always presented with default choice in the shipping method field. This is because Apple does not provide a way to keep the choice after billing page dismissed.