2nd May 2023

AVBarcode Scanner (1.0.o)

# Overview

The AV Barcode/QR Code Scanner component can scan and decode barcodes. Under the hood, the component contains NFIs that use the AV Barcode API.

## Use case

 1. An event organizer app, where a user needs to continuously scan barcodes on passes and allow entry to the attendees.

 2. A calendar app that adds events to a user's calendar using the QR Code of the event

 B .Percentage of re-use:

Approximate 90% of reuse. It sets an expectation of how much can be used out of the box, and how much needs to be customized for a specific app.

 C. Features

1. The component utilizes AV Barcode SDK for decoding barcodes and QR codes.
2. It supports the following Barcode types

iOS:( <https://developer.apple.com/documentation/avfoundation/avmetadatamachinereadablecodeobject/machine-readable_object_types> )

1. Uses NFI 3.0

2. Getting Started

## A. Prerequisites

 Before you start using the AVBarcode /QR code Scanner component, ensure the following:

 • [HCL Foundry](https://manage.hclvoltmx.com/)

 • Volt MX Iris

## Platforms Supported

### Mobile

#### iOS

### Tablets

## Importing the app

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

##  **To import the Appple AV Barcode/QR code Scanner component, do the following:**

## Open your app project in Volt MX Iris.

 2. In the Project Explorer, click the **Templates** tab.

 

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



4.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%22%20%5Cl%20%22add-a-component-to-a-form)

## Building and previewing the app

After performing all the above steps, you can build your app and run it on your device. For more information, you can refer to the [Building and Viewing an Application](https://opensource.hcltechsw.com/volt-mx-docs/docs/documentation/Iris/iris_user_guide/Content/Cloud_Build_in_VoltMX_Iris.html#cloud) section of the Volt MX User Guide.

 You can then run your app to see the Barcode/QR code Scanner work in real time.

# References

## Dynamic Usage

 You can also add**AV Barcode/QR code Scanner** component dynamically. To do so,

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



/\* Creating AV Barcode/QR code Scanner component instance \*/

var AVBCScanner = new com.technohub.AVBarCodeScanner({

 "height": "50%",

 "id": “AVBCScanner",

 "isVisible": true,

 "left": "0dp",

 "masterType": constants.MASTER\_TYPE\_USERWIDGET,

 "isModalContainer": false,

 "skin": "slFbox",

 "top": "0dp",

 "width": "100%",

 "zIndex": 1,

 "overrides": {

 "AVBCScanner": {

 "right": "viz.val\_cleared",

 "bottom": "viz.val\_cleared",

 "minWidth": "viz.val\_cleared",

 "minHeight": "viz.val\_cleared",

 "maxWidth": "viz.val\_cleared",

 "maxHeight": "viz.val\_cleared",

 "centerX": "viz.val\_cleared",

 "centerY": "viz.val\_cleared"

 }

 }

 }, {

 "overrides": {}

 }, {

 "overrides": {}

});

 /\*Adding the AV Barcode/QR code Scanner component to a Form\*/

this.view.add(AVBCScanner);

 In the code snippet, you can edit the properties of the component as per your requirement. For more information, see Setting Properties.

2. Save the file

Configuring Native Settings (iOS)

To configure the native settings for iOS, follow these steps:

1. From the **Project** explorer, go to **Assets** and expand **Media**.
2. Right-click **Common**, and then select **Resource Location**. Volt MX Iris opens the common resources folder in a file explorer.

3. Open the **infoplist\_configuration.json** file with a text or code editor.
4. At the end of the file, type the following code. You can change the description based on your preference.

"NSCameraUsageDescription" : "Your Description"

 

5. Save the file.

Configuring Deployment Target

1. From the left navigation menu, click **Project Settings**.
2. In the **Project Settings** window, go to **Native** → **iPhone/iPad**.
3. Under **Target Versions**, from the **iOS Version** list, select **10.0** or higher.


General Properties

## API

###  i.startSession

|  |  |
| --- | --- |
|  **Description:** | Starts the scanning. |
|   Syntax: | startSession () |
|  **Parameters:** | None |
|  **Return Value:** |  None |
|  **Remarks:**  | Call this API to start the scanning.  |
|  **Example:** | this.view.AVBarCodeScanner.startSession(); |

###  ii.stopSession

|  |  |
| --- | --- |
|  **Description:** | Stops the scanning. |
|   Syntax: | stopSession () |
|  **Parameters:** | None |
|  **Return Value:** |  None |
|  **Remarks:**  | Call this API after successful scan to avoid the continues scan.And call this API when app moves to background.  |
|  **Example:** | this.view.AVBarCodeScanner.stopSession(); |

###  iii. restartSession

|  |  |
| --- | --- |
|  **Description:** | Restarts the stopped scanning. |
|   Syntax: | restartSession () |
|  **Parameters:** | None |
|  **Return Value:** |  None |
|  **Remarks:**  | Call this API restart the scan after the stopSession() invocation.And In case session stopped on the background. call this API when app moves to foreground to resume the scanning.  |
|  **Example:** | this.view.AVBarCodeScanner.restartSession (); |

### iv.enableFlash

|  |  |
| --- | --- |
|  **Description: Turn on/off the flash light** | Turns on/off the device Flash light. |
|   Syntax: enableFlash(true/false) | enableFlash () |
|  **Parameters: Boolean, send true or false**  | None |
|  **Return Value: None** |  None |
| **Remarks:** Call this API restart the turn on/off the flash light. | Call this API to turn on the flashlight |
|  **Example:** this.view.AVBarCodeScanner.enableFlash(true); | this.view.MLKitBCScanner.enableFlash(true/false); |
| v. zoomIn

|  |  |
| --- | --- |
|  **Description:** |  Increase zoom by 1 step. |
|   Syntax: | zoomIn() |
|  **Parameters:** | None |
|  **Return Value:** |  None |
| **Remarks:** | Call this API to Zoom In |
|  **Example:** | this.view. AVBarCodeScanner.zoomIn(); |

 |  |
| vi. zoomOut

|  |  |
| --- | --- |
|  **Description:** |  decrease zoom by 1 step. |
|   Syntax: | zoomOut() |
|  **Parameters:** | None |
|  **Return Value:** |  None |
| **Remarks:** | Call this API to Zoom Out |
|  **Example:** | this.view. AVBarCodeScanner.zoomOut(); |

 |  |
|  |  |
| vii. isFlashSupports

|  |  |
| --- | --- |
|  **Description:** |  Returns status about the Flash light support in the device. |
|   Syntax: | isFlashSupports() |
|  **Parameters:** | None |
|  **Return Value:** | Boolean: True/False |
| **Remarks:** | Call this API to get the Flash light support status |
|  **Example:** | this.view. AVBarCodeScanner.isFlashSuuports(); |

 |  |
|  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| viii. setupRectOfInterest

|  |  |
| --- | --- |
|  **Description:** |  Adds the Rectangle/Square image on top of the Camera Screen. |
|   Syntax: | setupCustomRectOfInterest () |
|  **Parameters:** | None |
|  **Return Value:** | None |
| **Remarks:** | Call this API to add Rectangle/Square image on top of the Camera. And it should be Called every time when session restarts.  |
|  **Example:** | this.view. AVBarCodeScanner. setupRectOfInterest (); |

 |  |

### C. Events

### 1. retrieveData

|  |  |
| --- | --- |
| **Description:** | Invoked after the component scans and decodes a barcode or a QR code. |
|   Syntax: | retrieveData |
| **Parameters**:  |  data[JSON] : The data that is decoded from the barcode or QR code. |
|  **Remarks:** | If you want the camera to scan codes continuously, do not call the stopSession() in retrieveData. |
|  **Example:** | onNavigate: function(context) { this.view. AVBarCodeScanner. retrieveData (this.callback); }, callback: function(data) {  alert(JSON.stringify(data)); }  this.view. AVBarCodeScanner.stopSession(); }, |

 App version 1.0.0:

## Limitations

B. Known Issue