2nd May 2023

COGNEX QR/Barcode scanner component (1.0.0)

# Overview

This document contains the necessary information to implement the Cognex barcode scanner in a mobile application. Once Iris Application will be integrated with this component, App can include the feature to scan the bar codes.

## A. Use case

Consider a scenario that you want to scan the barcodes by using fastest barcode scanner called Cognex, this component fulfils the requirement with the scanner.

## 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 Cognex SDK
2. It supports iOS & Android
3. Uses NFI 3.0

2. Getting Started

## A. Prerequisites:

 Before you start using the Cognex barcode scanner component, ensure the following:

* Cognex license Keys for Android and iOS.
* Android minimum SDK is 21.
* iOS minimum version is 11

## Platforms Supported

### Mobile

#### Android & iOS

### Tablets

### 1. *Android & iOS*

## Importing the app

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

##  **To import the Cognex 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#add-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.

# 3. References

## 1. Dynamic Usage:

##  You can also add **Cognex** 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 Cognex component instance \*/

var Cognex = new com.scan.cognex.CognexScanner({

 "height": "50%",

 "id": “Cognex",

 "isVisible": true,

 "left": "0dp",

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

 "isModalContainer": false,

 "skin": "slFbox",

 "top": "0dp",

 "width": "100%",

 "zIndex": 1,

 "overrides": {

 " Cognex ": {

 "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 Cognex component to a Form\*/

this.view.add(Cognex);

 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

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

1. Add Camera permissions at Project Settings/Native/Android/Permissions/Add CAMERA



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

1. Add the Camera usage description at <Project>/resources/common/infoplist\_configuration.json file like below.

{

 "NSCameraUsageDescription" : "Use Camera to Scan the Barcodes"

}

2.In the Form’s onNavigate method pass the keys for both Android and iOS like below.

onNavigate: function(params, isBackNavigation) {

 this.view.CognexScanner.androidKey = ""; // Pass the Android key here

 this.view.CognexScanner.iOSKey = ""; // Pass the iOS key here

 },

NOTE: Keys must be passed on onNavigate. Otherwise, component would not be initialized.

4.General Properties:

### A. Events

### i. startScan

|  |  |
| --- | --- |
| **Description:** | Starts the Scan |
| **Syntax**: | startScan() |
| **Parameters:** | NA |
| **Return Value:** | None |
| **Remarks:**  | It is recommended to start the scan after calling the stop scan. |
| **Example:** |  this.view.CognexScanner.startScan(); |

 i. stopScan

|  |  |
| --- | --- |
| **Description:** | Stops the running scan. |
| **Syntax**: | stopScan() |
| **Parameters:** | NA |
| **Return Value:** |  None |
| **Remarks:**  | It is recommended to invoke this method, in the result callback, so that background scans will be avoided.  |
|  **Example:** |  this.view.CognexScanner.stopScan(); |

 i. registerCallback

|  |  |
| --- | --- |
| **Description:** | Registers a callback for the Scanning results |
| **Syntax**: | registerCallback (function(result) {}) |
| **Parameters:** | Callback function |
| **Return Value:** |  None |
| **Remarks:**  | It is recommended to register this callback in the onNavigate method.  |
|  **Example:** | this.view.CognexScanner.registerCallback(function(result){}); |

 App version 1.0.0:

## Limitations

## To use cmbSDK for barcode scanning with a mobile device without an MX mobile terminal, you need to install a license key. If the license key is missing, asterisks will appear instead of scanned results.

## <https://cmbdn.cognex.com/v2.3.x/knowledge/cognex-mobile-barcode-sdk-for-android/using-cmbsdk/working-with-results>