2nd May 2023

Scandit Barcode scanner component (1.0.o)

# Overview

This document contains the necessary information to implement the Scandit 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 Scandit, 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 Scandit SDK
2. It supports only iOS & Android
3. Uses NFI 2.0

2. Getting Started

## A. Prerequisites:

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

* Scandit license Keys for Android and iOS. License keys can get from https://www.scandit.com
* Android minimum SDK is 21.
* iOS minimum version is 9
* iOS NFI zips.
* Volt MX Iris

## Platforms Supported

### Mobile

#### Android & 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 Scandit component, do the following:**

## 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

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

Graphical user interface, text, application, Teams

Description automatically generated

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 **Scandit** 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.



A screenshot of a computer

Description automatically generated with medium confidence

/\* Creating Scandit component instance \*/

var Scandit = new com.scanner.scandit.ScanditComponent({

"height": "50%",

"id": “Scandit",

"isVisible": true,

"left": "0dp",

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

"isModalContainer": false,

"skin": "slFbox",

"top": "0dp",

"width": "100%",

"zIndex": 1,

"overrides": {

" Scandit ": {

"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 Scandit component to a Form\*/

this.view.add(Scandit);

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 (Android)

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

1. Add the below dependency at Settings/Native/Android Mobile/Tablet/Gradle Entries/build.gradle entries to Suffix,

dependencies {

implementation 'com.scandit.datacapture:barcode:6.7.0'

implementation 'com.scandit.datacapture:core:6.7.0'

}

Graphical user interface, text, application, email

Description automatically generated

1. Add Camera permissions at Project Settings/Native/Android/Permissoins/Add CAMERA  
     
     
   Graphical user interface, text, application, email

   Description automatically generated
2. Request Runtime camera permissions before accessing the Scandit as documented following: <https://docs.kony.com/konylibrary/visualizer/viz_api_dev_guide/content/kony.application_functions_runtimepermissionsapi.htm#requestPermission>

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.

key: “”,

onNavigate: function(params, isBackNavigation) {

this.context = context;

if(kony.os.deviceInfo().name === "android") {

this.view.ScanditComponent.androidKey = this.key; // Pass Android key here

}

else {

this.view.ScanditComponent.iOSKey = this.key; // Pass iOS key here

}

},

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

4.General Properties:

## API

i. startScan

|  |  |
| --- | --- |
| **Description:** | Starts the barcode scanning. |
| **Syntax**: | startScan() |
| **Parameters:** | None |
| **Return Value:** | None |
| **Remarks:** | Invoke this method in the Form’s postShow method to get it worked. |
| **Example:** | this.view.Scandit.startScan(); |

### 

### ii. stopScan

|  |  |
| --- | --- |
| **Description:** | Stops the running scanning. |
| **Syntax**: | stopScan() |
| **Parameters:** | None |
| **Return Value:** | None |
| **Remarks:** | To stop the continues scan, invoke this method in the result callback. |
| **Example:** | this.view.Scandit.stopScan(); |

### B. Events

### i.registerResultCallback

|  |  |
| --- | --- |
| **Description:** | Registers the callback where scanned data is receiving. |
| **Syntax**: | registerResultCallback(this.callback) |
| **Parameters:** | Function - callback |
| **Return Value:** | None |
| **Remarks:** | It is recommended to register this callback in the form’s post show method. |
| **Example:** | onDoneAction(scannedCode, symbol) {  // Write the logic here  },  this.view.Scandit.registerResultCallback(this.onDoneAction); |

App version 1.0.0:

## Limitations

B. Known Issue