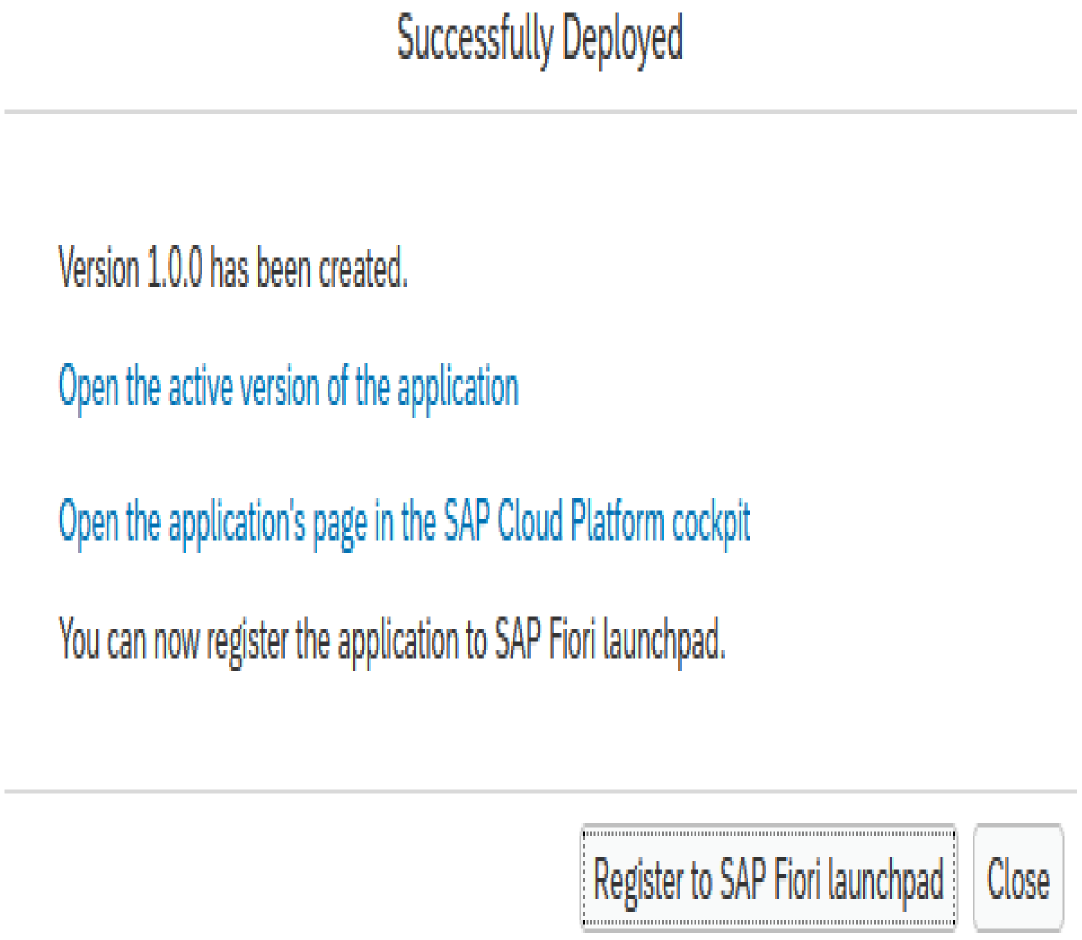
InTouch Machine Edition (v2.2.x)

Introduction

In this wiki pages we will explain how to use our cmbSDK trough cordova plugin as custom widget in ITME project

Getting Started

Open ITME Studio and create new project



Successfully Deployed

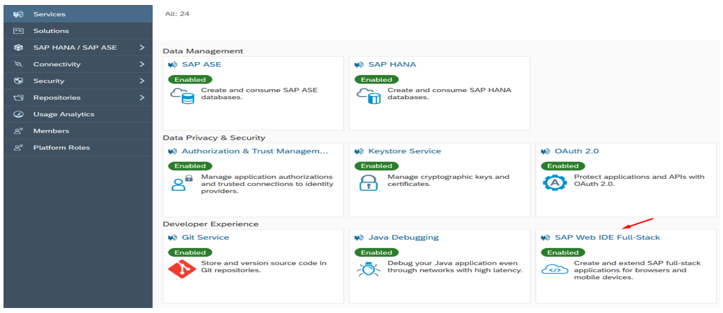
Version 1.0.0 has been created.

Open the active version of the application

Open the application's page in the SAP Cloud Platform cockpit

You can now register the application to SAP Fiori launchpad.

Set your project resolution



Data Management

SAP ASE — Enabled — Create and consume SAP ASE databases.

SAP HANA — Enabled — Create and consume SAP HANA databases.

Data Privacy & Security

Authorization & Trust Managem... — Enabled — Manage application authorizations and trusted connections to identity providers.

Keystore Service — Enabled — Manage cryptographic keys and certificates.

OAuth 2.0 — Enabled — Protect applications and APIs with OAuth 2.0.

Developer Experience

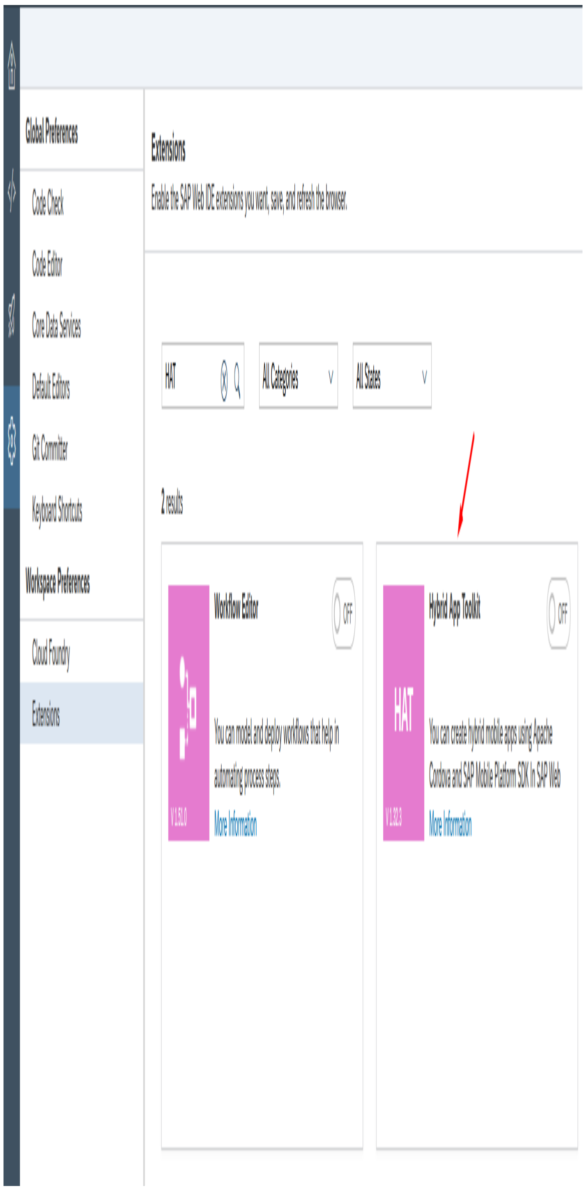
Git Service — Enabled — Store and version source code in Git repositories.

Java Debugging — Enabled — Debug your Java application even through networks with high latency.

SAP Web IDE Full-Stack — Enabled — Create and extend SAP full-stack applications for browsers and mobile devices.

Project Tags

Go in Global section of Project Explorer, open Project Tags -> Datasheet View and insert project tags that we will use. We will explain every tag later in this section



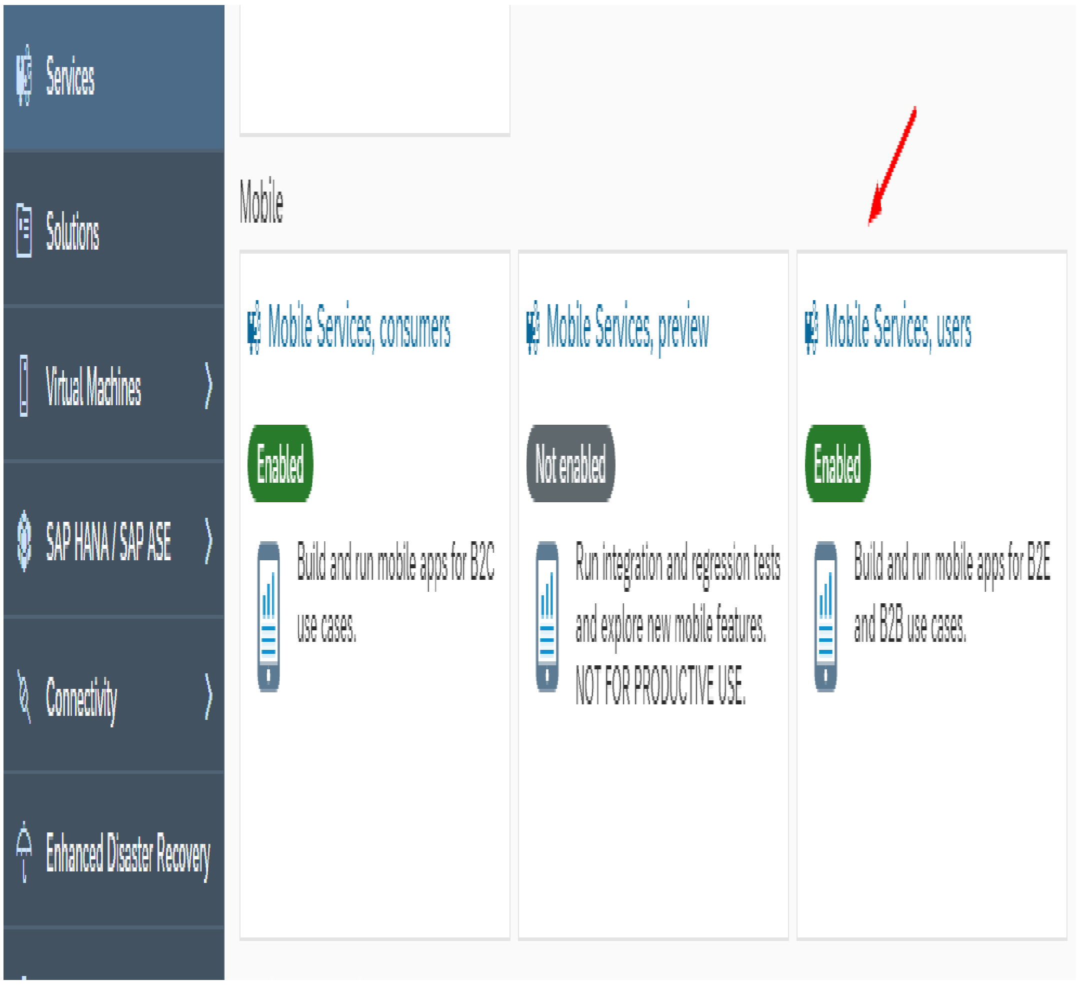
Extensions

Enable the SAP Web IDE extensions you want, save, and refresh the browser.

2 results

Workflow Editor — OFF — You can model and deploy workflows that help in automating process steps. — V 1.61.0 — More Information

Hybrid App Toolkit — OFF — You can create hybrid mobile apps using Apache Cordova and SAP Mobile Platform SDK In SAP Web — V 1.32.3 — More Information


Services
Solutions
Virtual Machines
SAP HANA / SAP ASE
Connectivity
Enhanced Disaster Recovery
Mobile
Mobile Services, consumers
Enabled
Build and run mobile apps for B2C use cases.
Mobile Services, preview
Not enabled
Run integration and regression tests and explore new mobile features. NOT FOR PRODUCTIVE USE.
Mobile Services, users
Enabled
Build and run mobile apps for B2E and B2B use cases.

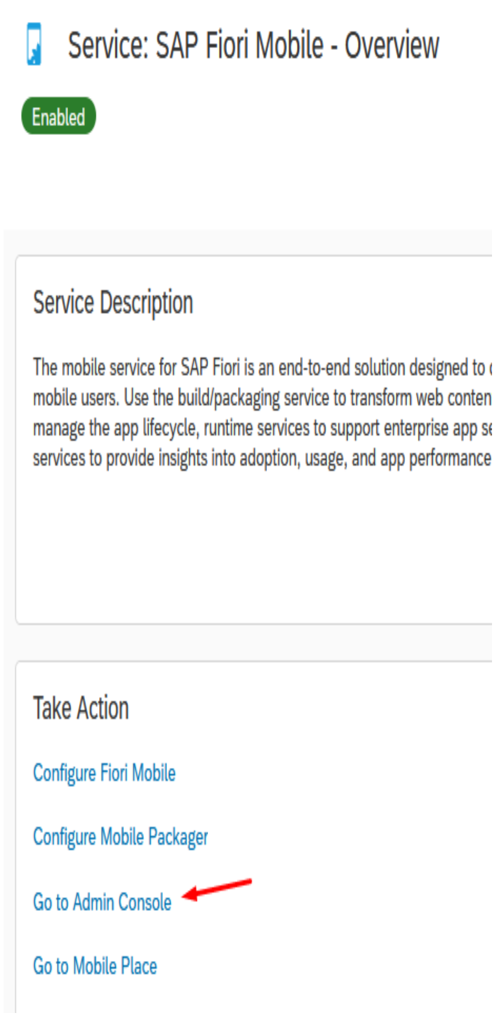
All tags are local and change to the tag value affects only the station on which the change was made.

Barcode Widget

How this widget is working ? There is trigger properties that call api methods from cordova plugin. There is output properties where we return results from api methods and there is events like callback functions that are called when api method is executed. On this link you can read about every property and here about events.

Project screens

Go back in Graphics section, right click on Screens category and click Insert



Service: SAP Fiori Mobile - Overview

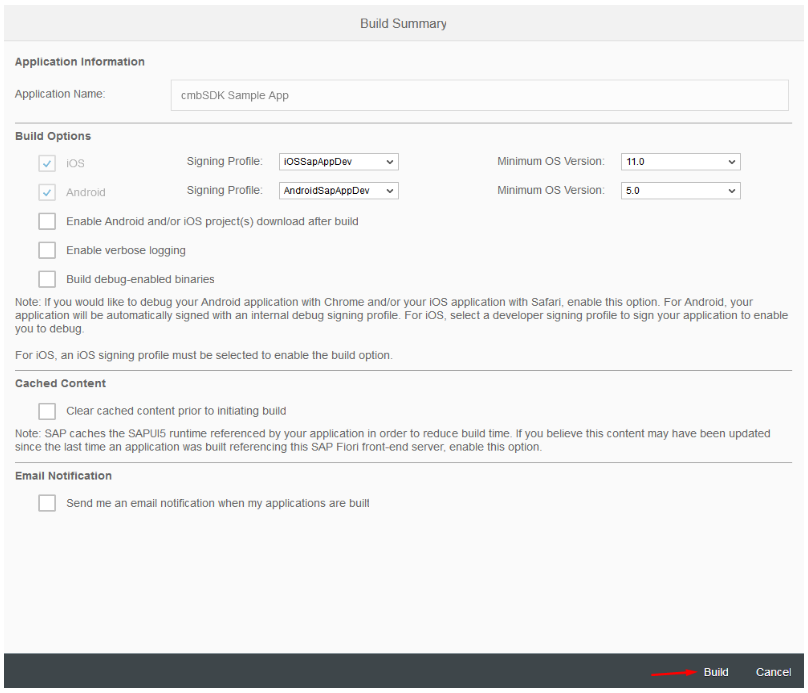
Enabled

Service Description

The mobile service for SAP Fiori is an end-to-end solution designed to
mobile users. Use the build/packaging service to transform web conten
manage the app lifecycle, runtime services to support enterprise app se
services to provide insights into adoption, usage, and app performance

Take Action

Set screen attributes and press OK.



Build Summary

Application Information

Application Name: cmbSDK Sample App

Build Options

- ✓ iOS — Signing Profile: iOSSapAppDev — Minimum OS Version: 11.0
- ✓ Android — Signing Profile: AndroidSapAppDev — Minimum OS Version: 5.0
- ☐ Enable Android and/or iOS project(s) download after build
- ☐ Enable verbose logging
- ☐ Build debug-enabled binaries

Note: If you would like to debug your Android application with Chrome and/or your iOS application with Safari, enable this option. For Android, your application will be automatically signed with an internal debug signing profile. For iOS, select a developer signing profile to sign your application to enable you to debug.

For iOS, an iOS signing profile must be selected to enable the build option.

Cached Content

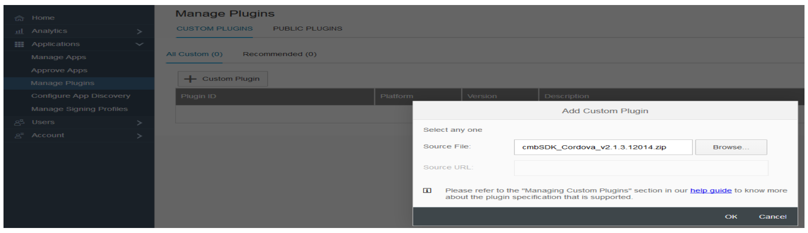
- ☐ Clear cached content prior to initiating build

Note: SAP caches the SAPUI5 runtime referenced by your application in order to reduce build time. If you believe this content may have been updated since the last time an application was built referencing this SAP Fiori front-end server, enable this option.

Email Notification

- ☐ Send me an email notification when my applications are built

Design your header screen and name it as header.



Manage Plugins

CUSTOM PLUGINS | PUBLIC PLUGINS

All Custom (0) | Recommended (0)

Custom Plugin

Plugin ID	Platform	Version	Description

Add Custom Plugin

Select any one

Source File: cmbSDK_Cordova_v2.1.3.12014.zip Browse...

Source URL:

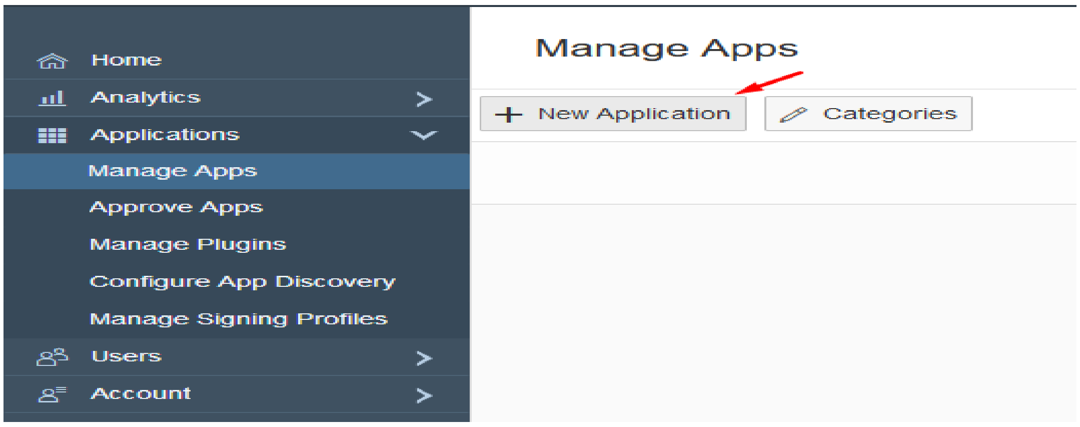
Please refer to the "Managing Custom Plugins" section in our help guide to know more about the plugin specification that is supported.

OK Cancel

Here we have header text, label where we will show reader device battery level and smart message object that show current reader device connection status.

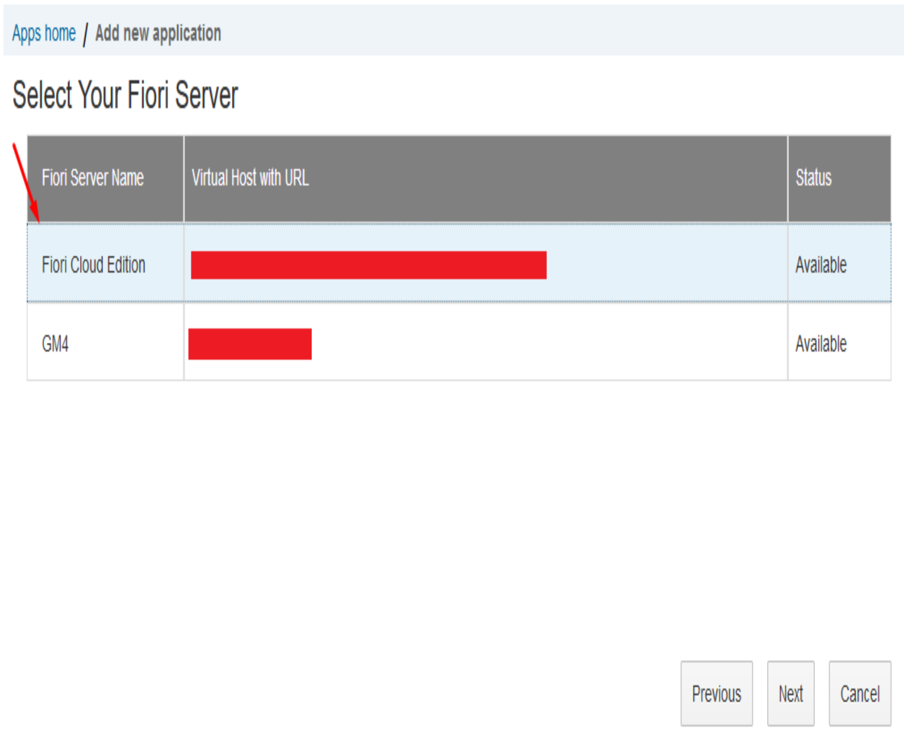
In header screen we will add our barcodeReader custom widget.

Click on Custom Widget icon



Manage Apps

Click import and find your barcodeReader.cwp file. barcodeReader should be shown as available widget. Select it and click OK.



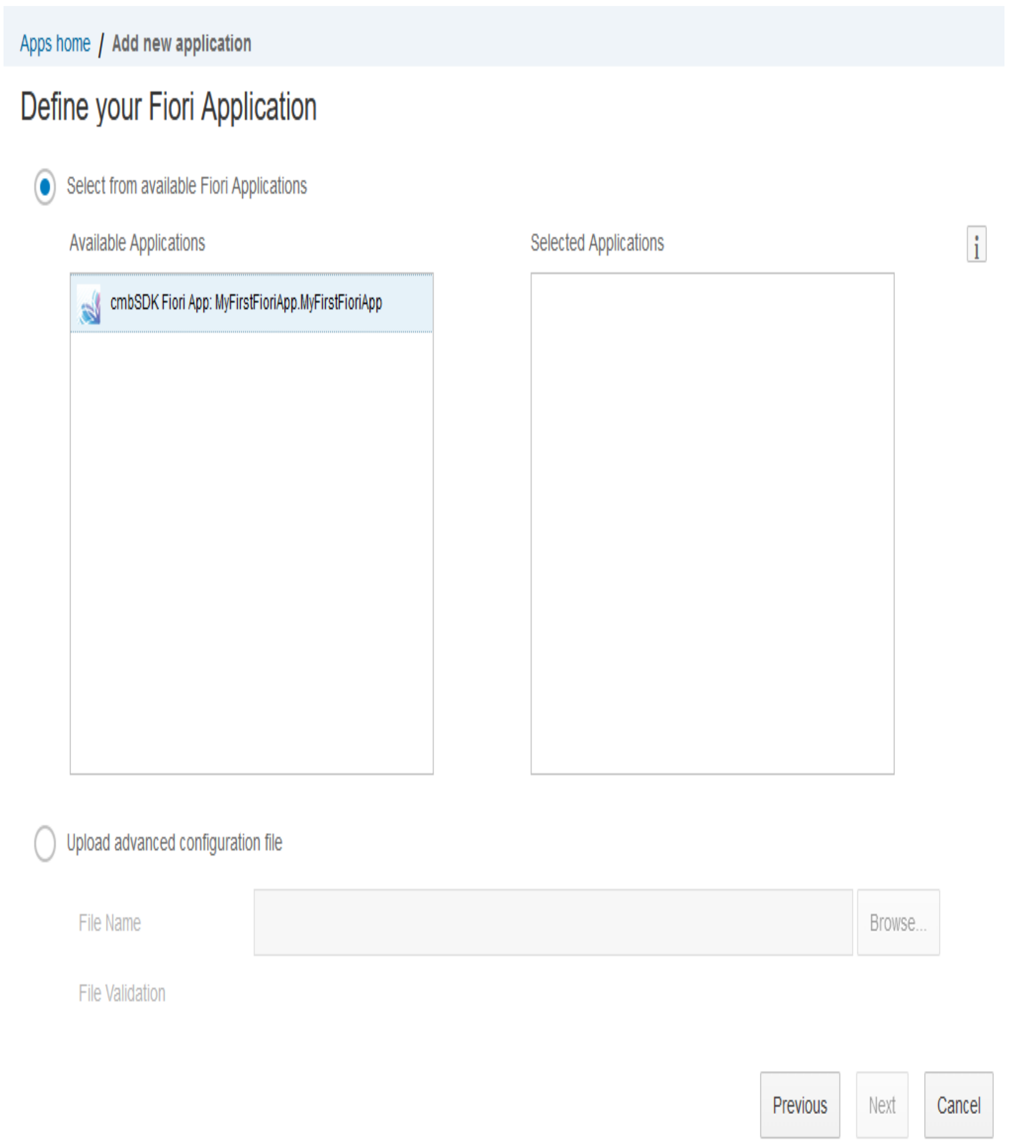
Apps home / Add new application

Select Your Fiori Server

Fiori Server Name	Virtual Host with URL	Status
Fiori Cloud Edition	[redacted]	Available
GM4	[redacted]	Available

barcodeReader custom widget is added in header screen

Open barcodeReader properties and go in widget members



Apps home / Add new application

Define your Fiori Application

(●) Select from available Fiori Applications

Available Applications

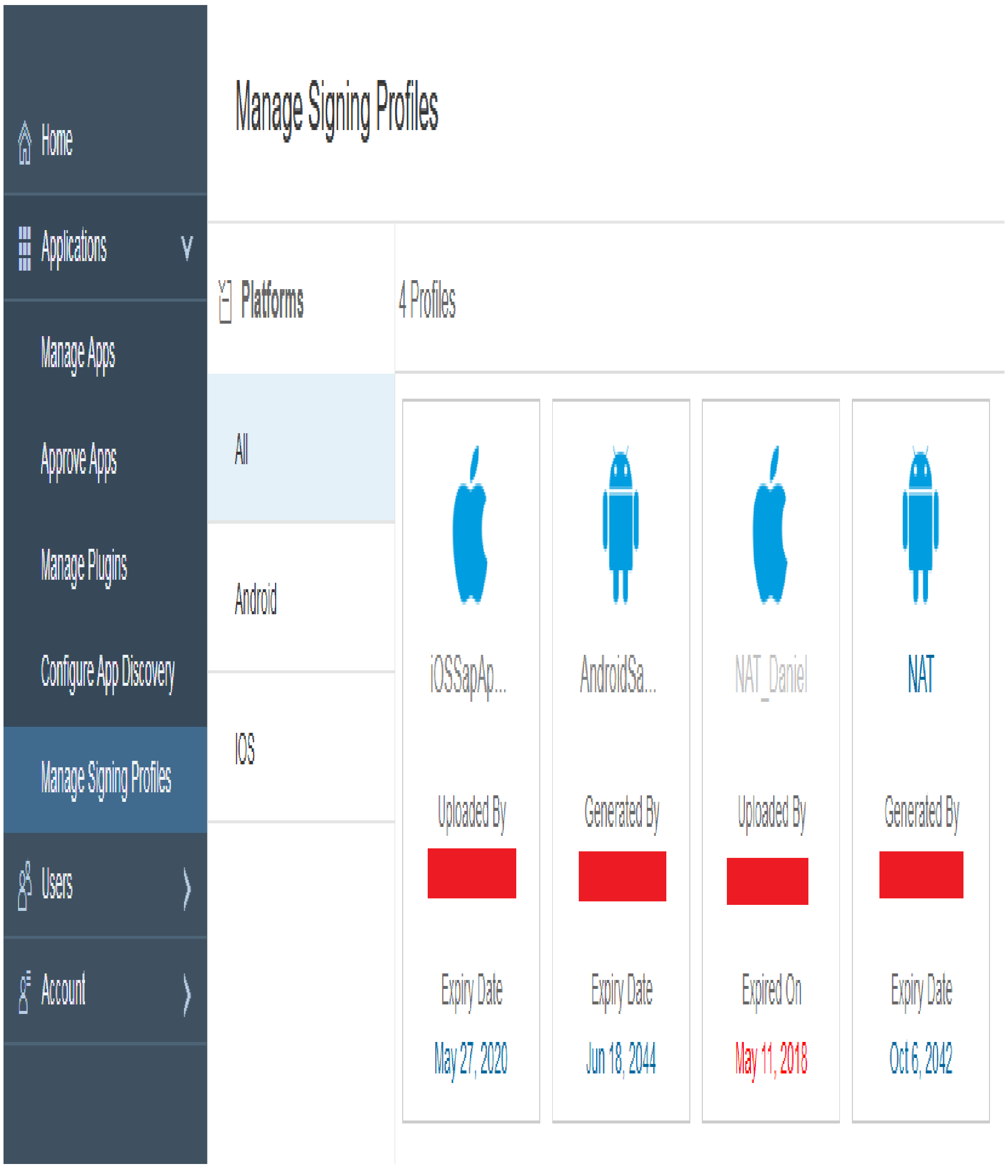
cmbSDK Fiori App: MyFirstFioriApp.MyFirstFioriApp

Selected Applications

( ) Upload advanced configuration file

File Name

File Validation



Manage Signing Profiles

4 Profiles

iOSSapAp...
Uploaded By
[redacted]
Expiry Date
May 27, 2020

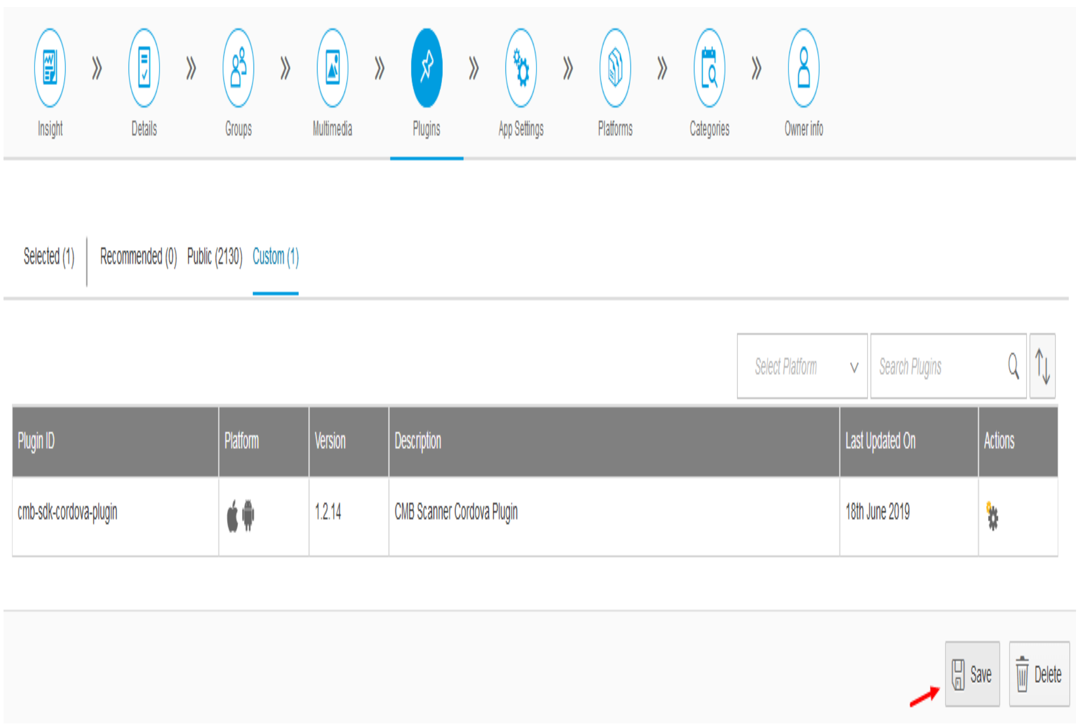
AndroidSa...
Generated By
[redacted]
Expiry Date
Jun 18, 2044

NAT_Daniel
Uploaded By
[redacted]
Expired On
May 11, 2018

NAT
Generated By
[redacted]
Expiry Date
Oct 6, 2042

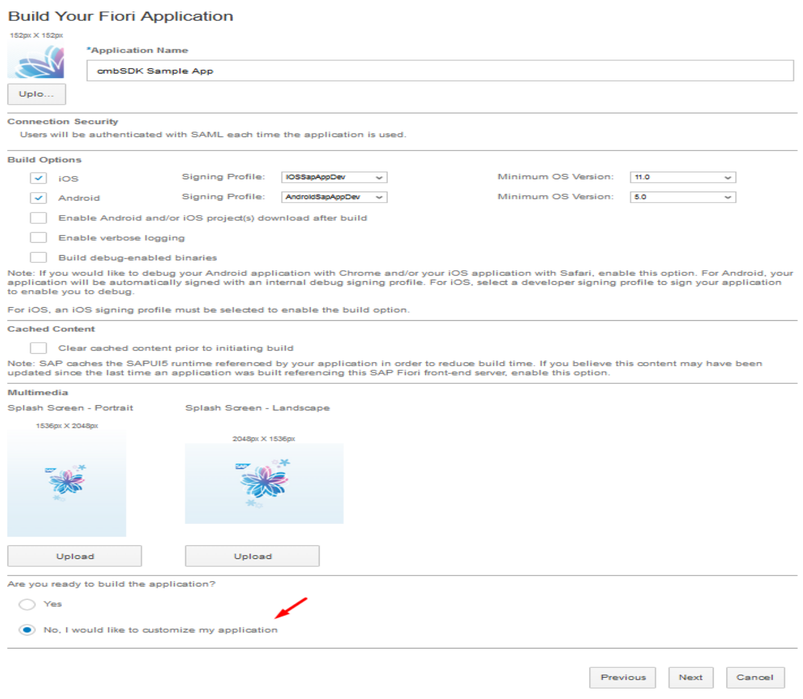
Map properties from custom widget with project tags and set action Set+Get and set script that you want to execute in callback events.

Then open header screen script and call setPreviewContainerPositionAndSizeEventTrigger and loadScanner methods when header screen is opened. Also open another screen (main) while this one is opening.



Plugin ID	Platform	Version	Description	Last Updated On	Actions
cmb-sdk-cordova-plugin		1.2.14	CMB Scanner Cordova Plugin	18th June 2019	

Now create another screen and name it main.



Build Your Fiori Application

152px X 152px

*Application Name: cmbSDK Sample App

Uplo...

Connection Security

Users will be authenticated with SAML each time the application is used.

Build Options

✓ iOS	Signing Profile: IOSSapAppDev	Minimum OS Version: 11.0
✓ Android	Signing Profile: AndroidSapAppDev	Minimum OS Version: 5.0

☐ Enable Android and/or iOS project(s) download after build

☐ Enable verbose logging

☐ Build debug-enabled binaries

Note: If you would like to debug your Android application with Chrome and/or your iOS application with Safari, enable this option. For Android, your application will be automatically signed with an internal debug signing profile. For iOS, select a developer signing profile to sign your application to enable you to debug.

For iOS, an iOS signing profile must be selected to enable the build option.

Cached Content

☐ Clear cached content prior to initiating build

Note: SAP caches the SAPUI5 runtime referenced by your application in order to reduce build time. If you believe this content may have been updated since the last time an application was built referencing this SAP Fiori front-end server, enable this option.

Multimedia

Splash Screen - Portrait: 1536px X 2048px — Upload

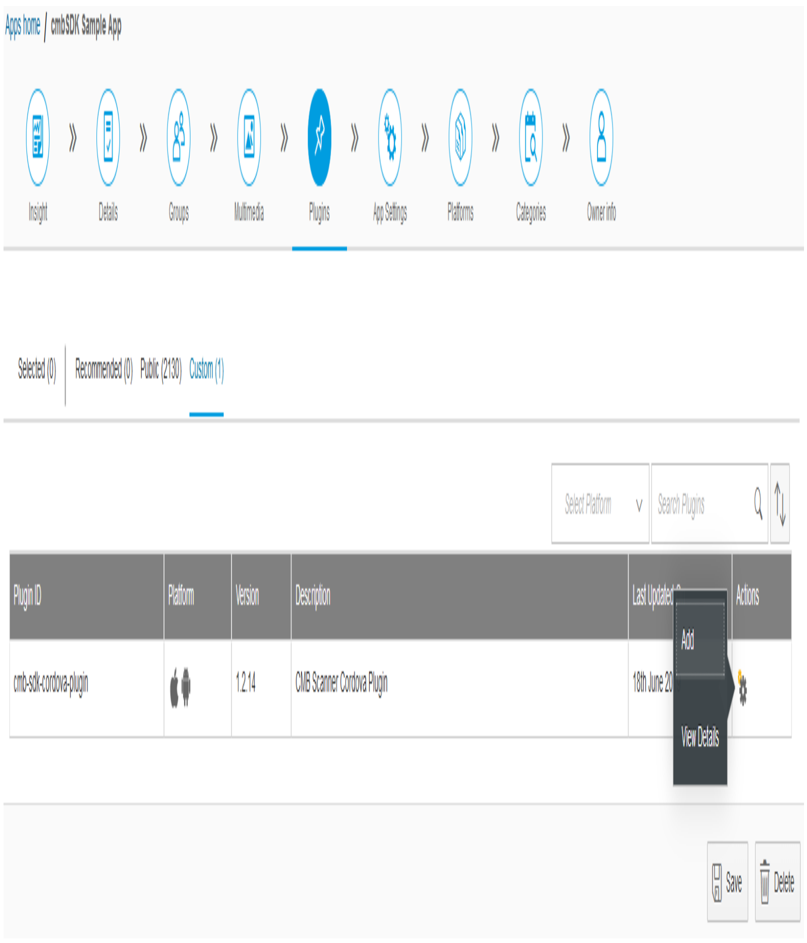
Splash Screen - Landscape: 2048px X 1536px — Upload

Are you ready to build the application?

Yes [✓ No, I would like to customize my application]

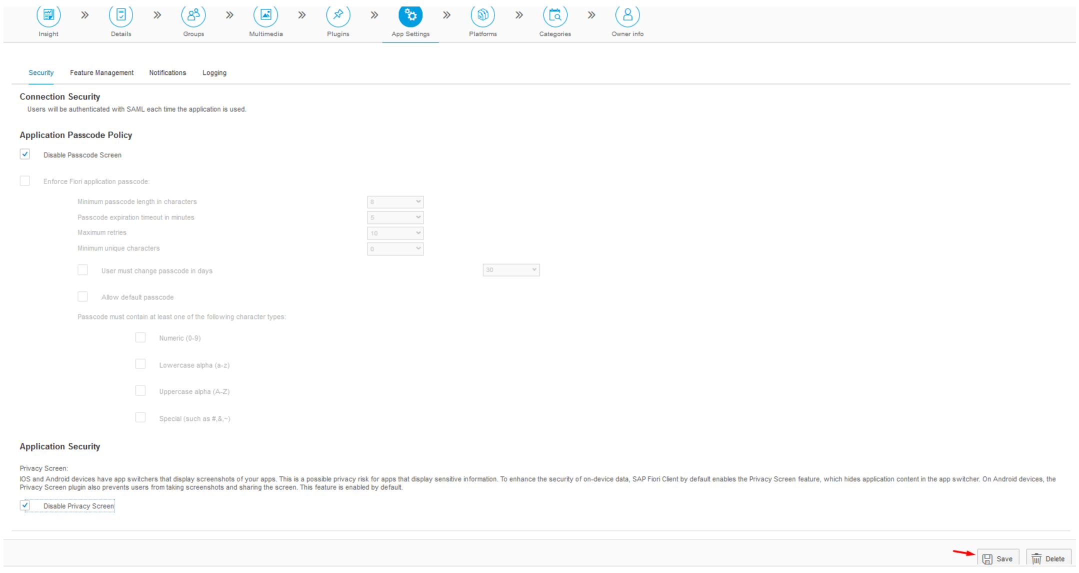
Previous Next Cancel

In this screen we have rectangle where we display scanned results, button to start/stop scanning and button to clear displayed results.



Plugin ID	Platform	Version	Description	Last Updated	Actions
cmb-sdk-cordova-plugin		1.2.14	CMB Scanner Cordova Plugin	18th June 20	

Later right click on Screen Group category and create StartUp screen group.



Connection Security

Users will be authenticated with SAML each time the application is used.

Application Passcode Policy

- [✓] Disable Passcode Screen
- [ ] Enforce Fiori application passcode:

Minimum passcode length in characters	8
Passcode expiration timeout in minutes	5
Maximum retries	10
Minimum unique characters	0

- [ ] User must change passcode in days 30
- [ ] Allow default passcode

Passcode must contain at least one of the following character types:

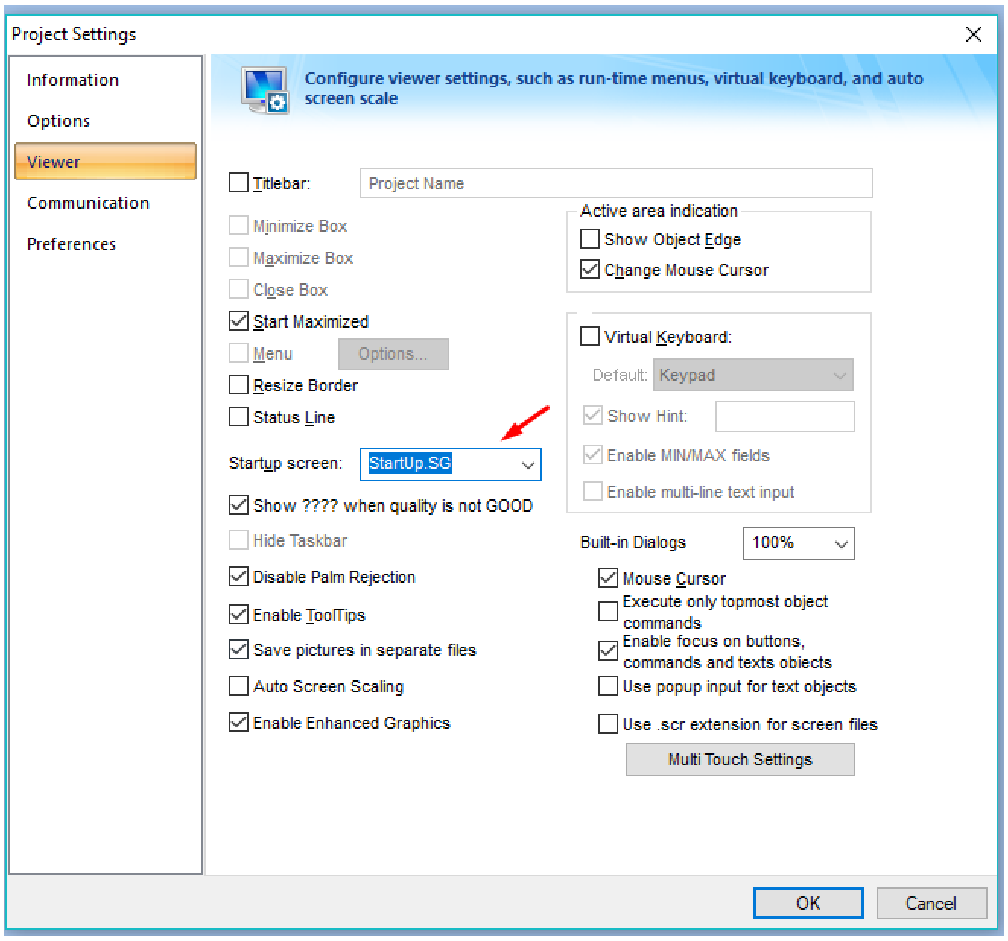
- [ ] Numeric (0-9)
- [ ] Lowercase alpha (a-z)
- [ ] Uppercase alpha (A-Z)
- [ ] Special (such as #,&,~)

Application Security

Privacy Screen:
IOS and Android devices have app switchers that display screenshots of your apps. This is a possible privacy risk for apps that display sensitive information. To enhance the security of on-device data, SAP Fiori Client by default enables the Privacy Screen feature, which hides application content in the app switcher. On Android devices, the Privacy Screen plugin also prevents users from taking screenshots and sharing the screen. This feature is enabled by default.

- [✓] Disable Privacy Screen

Open Project properties and set this group as Startup screen.



Project Settings

- Information
- Options
- Viewer
- Communication
- Preferences

Configure viewer settings, such as run-time menus, virtual keyboard, and auto screen scale

[ ] Titlebar: Project Name

[ ] Minimize Box
[ ] Maximize Box
[ ] Close Box
[✓] Start Maximized
[ ] Menu — Options...
[ ] Resize Border
[ ] Status Line

Startup screen: StartUp.SG

[✓] Show ???? when quality is not GOOD
[ ] Hide Taskbar
[✓] Disable Palm Rejection
[✓] Enable ToolTips
[✓] Save pictures in separate files
[ ] Auto Screen Scaling
[✓] Enable Enhanced Graphics

Active area indication
[ ] Show Object Edge
[✓] Change Mouse Cursor

[ ] Virtual Keyboard:
Default: Keypad
[✓] Show Hint:
[✓] Enable MIN/MAX fields
[ ] Enable multi-line text input

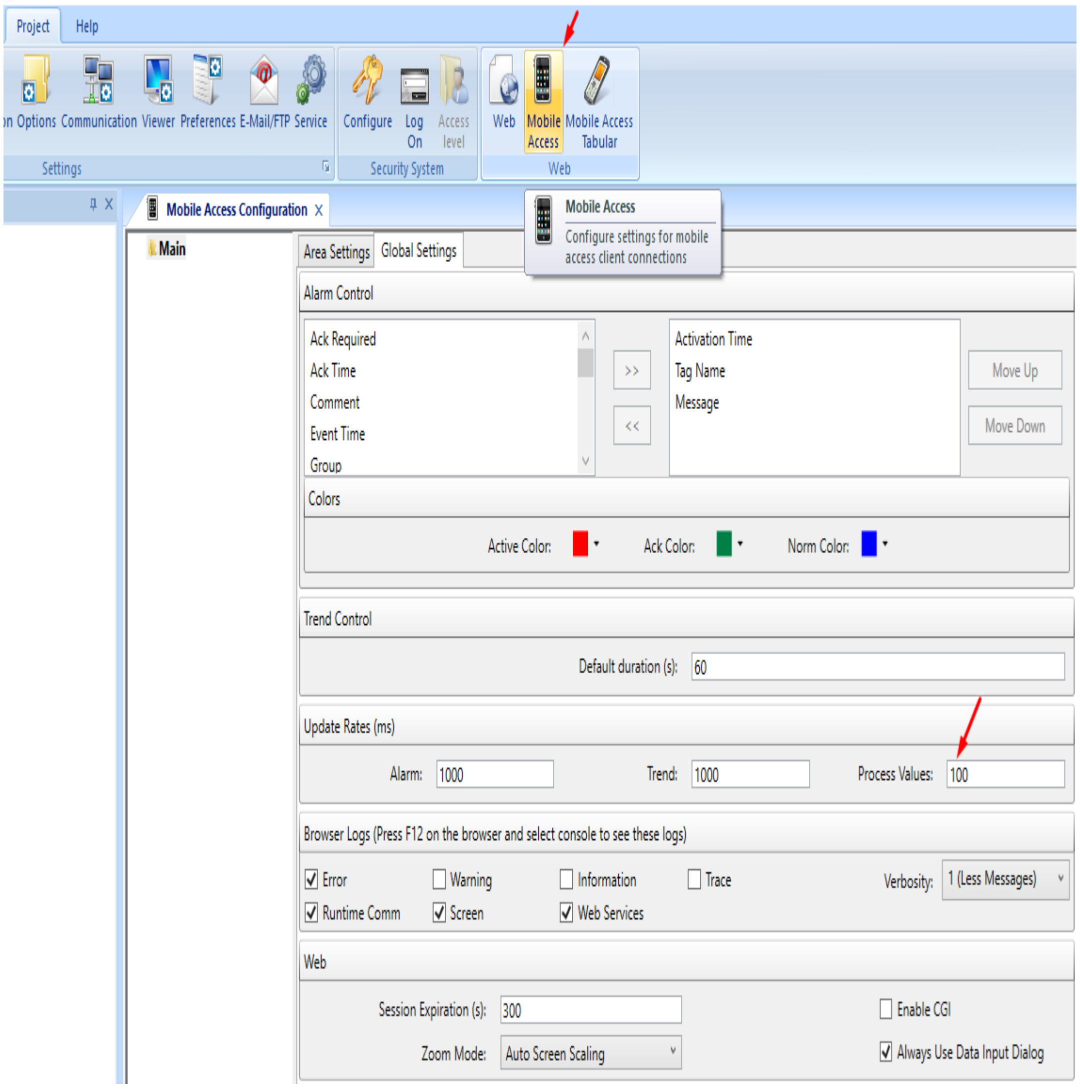
Built-in Dialogs: 100%

[✓] Mouse Cursor
[ ] Execute only topmost object commands
[✓] Enable focus on buttons, commands and texts objects
[ ] Use popup input for text objects
[ ] Use .scr extension for screen files

Multi Touch Settings

OK Cancel

Because we want to deploy this application as HTML web interface and interact from mobile devices we need to open Mobile Access menu at least ones and save changes on close. Also, for better communication with custom widget, set Process Values at minimum value which is 100.



Alarm Control

Ack Required		Activation Time
Ack Time		Tag Name
Comment		Message
Event Time		
Group		

Colors

Active Color: 🟥 Ack Color: 🟩 Norm Color: 🟦

Trend Control

Default duration (s): 60

Update Rates (ms)

Alarm: 1000 Trend: 1000 Process Values: 100

Browser Logs (Press F12 on the browser and select console to see these logs)

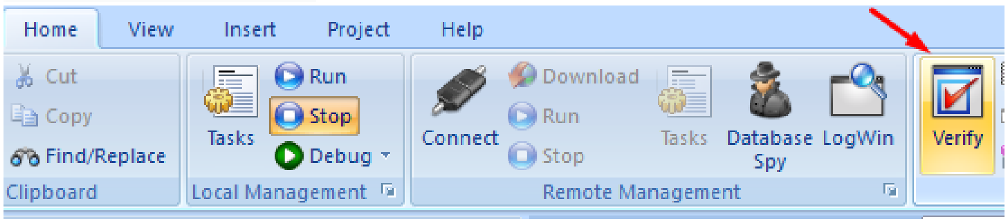
☑ Error	☐ Warning	☐ Information	☐ Trace	Verbosity: 1 (Less Messages)
☑ Runtime Comm	☑ Screen	☑ Web Services		

Web

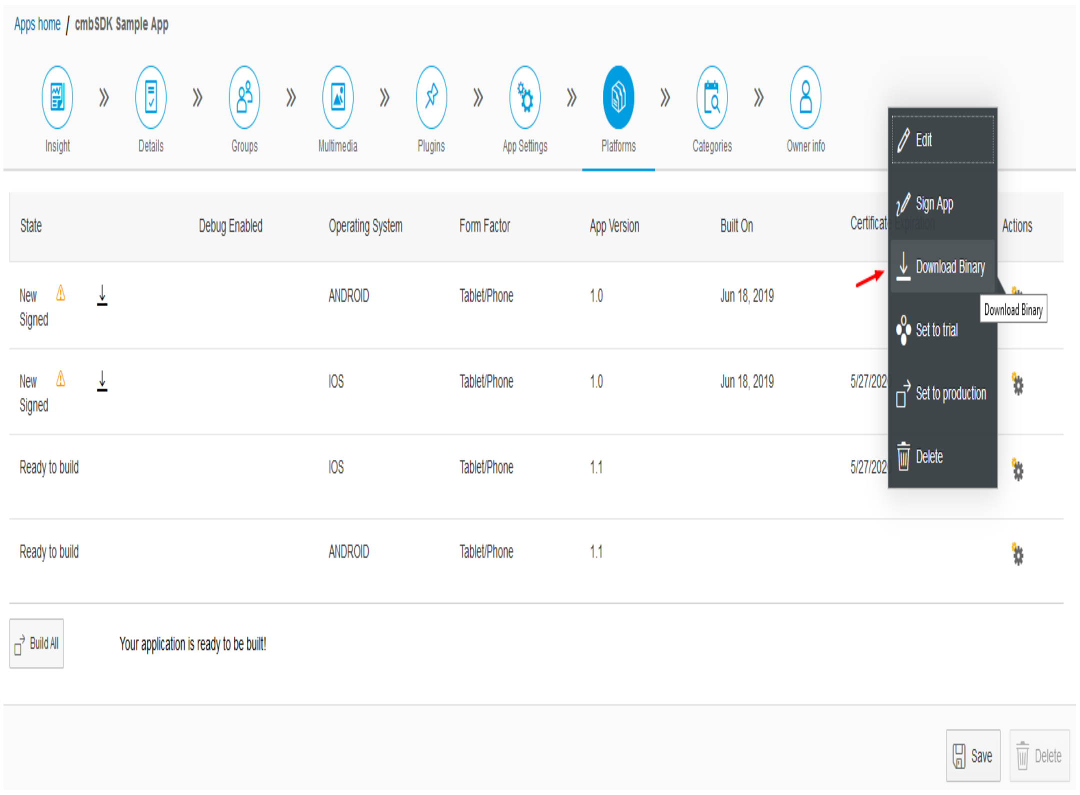
Session Expiration (s): 300 ☐ Enable CGI

Zoom Mode: Auto Screen Scaling ☑ Always Use Data Input Dialog

When we finish with our screens and configurations Verify project.



and Save all screens as HTML for web access



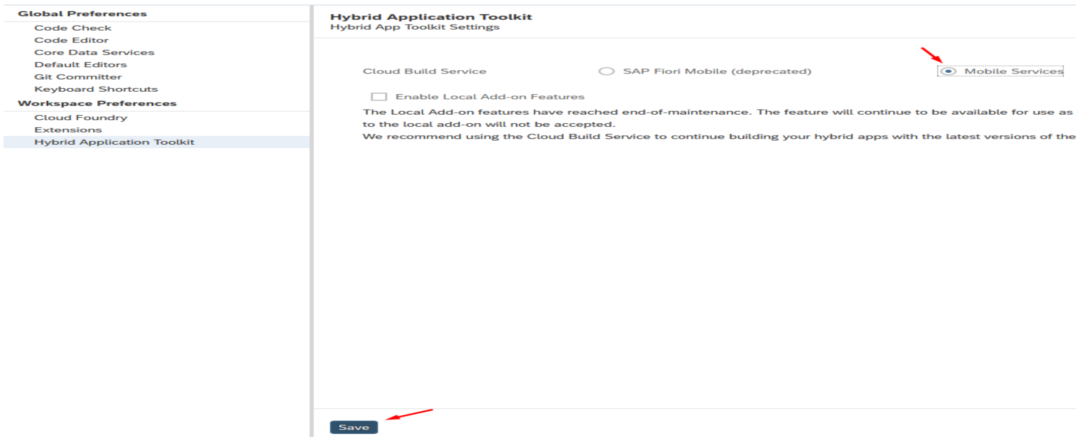
State	Debug Enabled	Operating System	Form Factor	App Version	Built On	Certificat	Actions
New Signed		ANDROID	Tablet/Phone	1.0	Jun 18, 2019		
New Signed		IOS	Tablet/Phone	1.0	Jun 18, 2019	5/27/202	
Ready to build		IOS	Tablet/Phone	1.1		5/27/202	
Ready to build		ANDROID	Tablet/Phone	1.1			

Your application is ready to be built!

With all these steps we finished our ITME project.

Server configuration

Navigate to your ITME web application physical path. If you use local IIS server usually this path is C:\inetpub\wwwroot\ITME81 and copy barcodeReaderServerFiles that you've download.



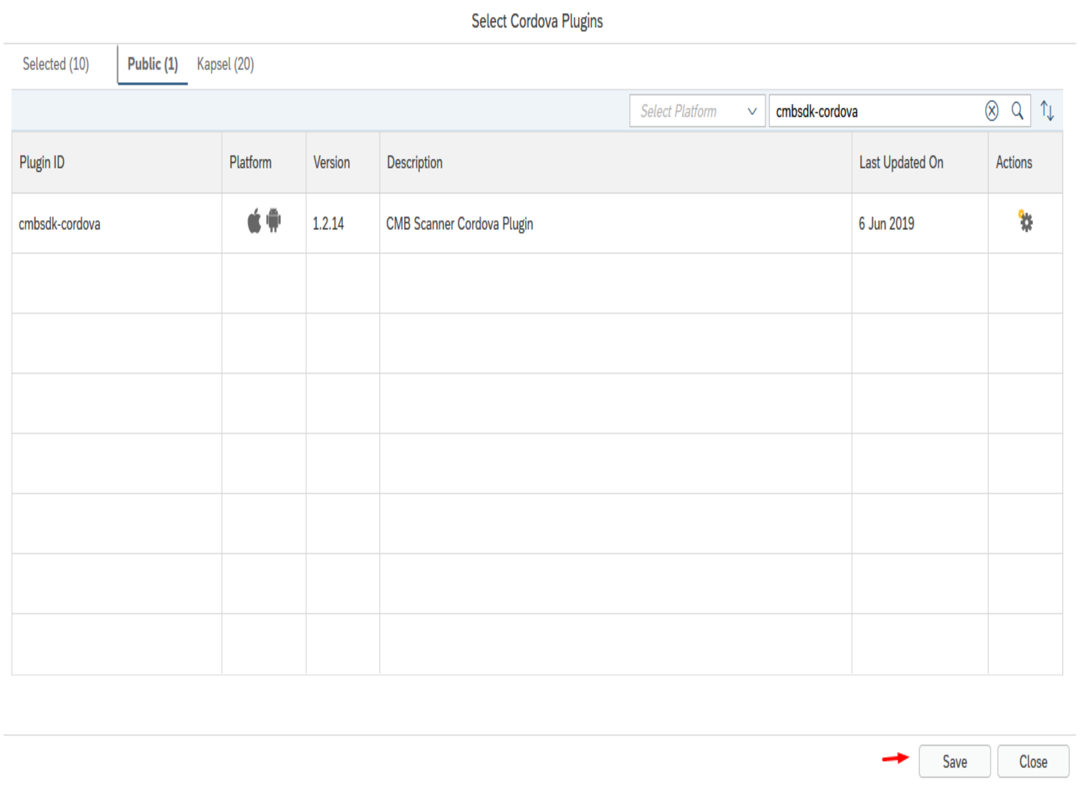
Hybrid Application Toolkit
Hybrid App Toolkit Settings

Cloud Build Service ○ SAP Fiori Mobile (deprecated) [✓ Mobile Services]

☐ Enable Local Add-on Features

The Local Add-on features have reached end-of-maintenance. The feature will continue to be available for use as
to the local add-on will not be accepted.
We recommend using the Cloud Build Service to continue building your hybrid apps with the latest versions of the

These cordova files that will be hosted on server will help us to use native features trough custom widgets. Later open index.html and reference these scripts:


Select Cordova Plugins

Selected (10) | **Public (1)** | Kapsel (20)

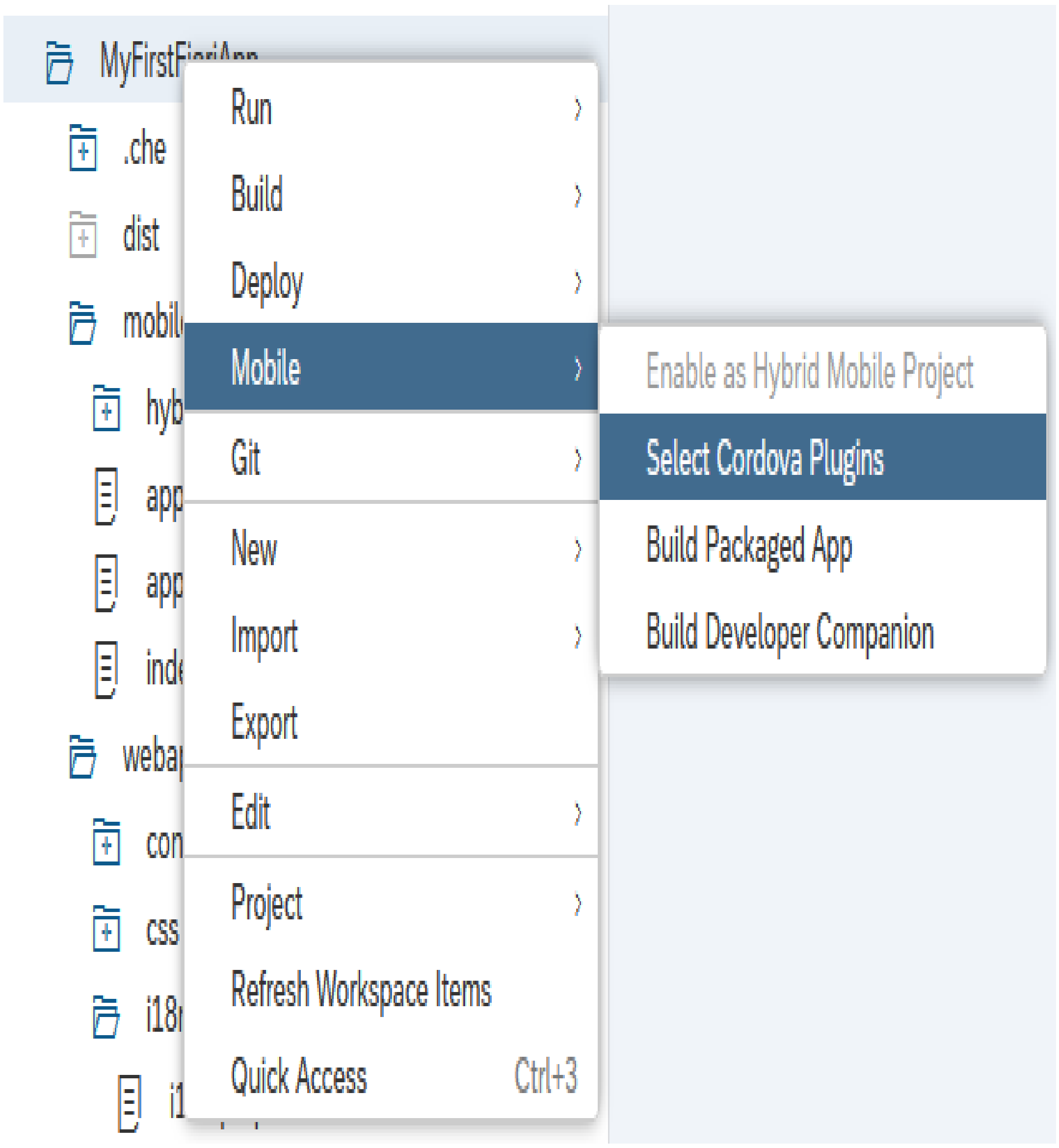
Select Platform | cmbsdk-cordova

Plugin ID	Platform	Version	Description	Last Updated On	Actions
cmbsdk-cordova		1.2.14	CMB Scanner Cordova Plugin	6 Jun 2019	

When we navigate to server URL we add platform that we are using as attribute in query string.

With barcodeReader.js we check this attribute to know which cordova files to load (iOS or android).

Now open IIS manager right click on your ITME81 application and click Add Virtual Directory:



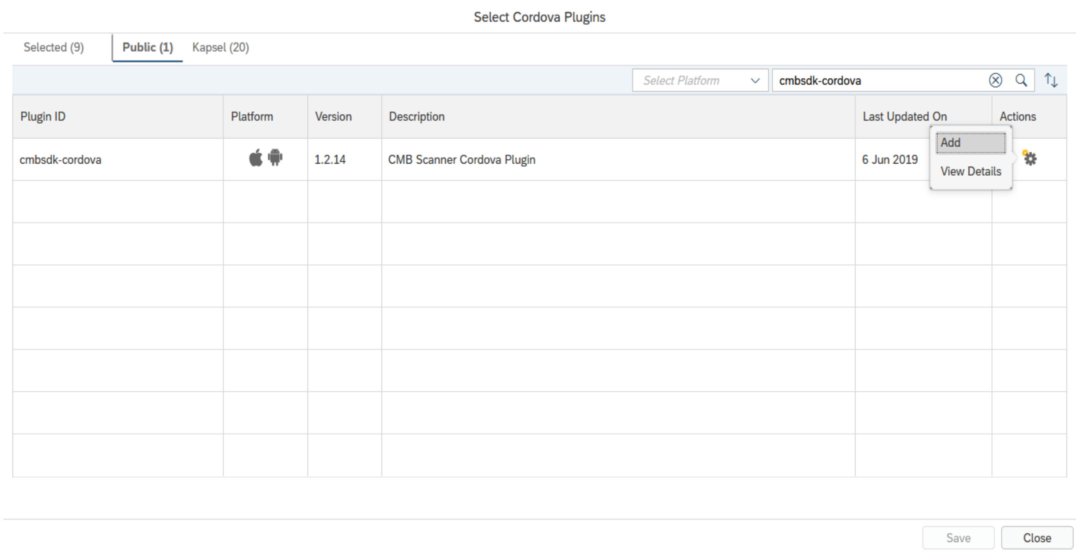
Set “CustomWidget” as Alias and as physical pat set path to your ITME project (C:\Users\Marko\Documents\InTouch Machine Edition v8.1 Projects\cmbSdkSample in our case)

Be sure that you make connection with user that have authorization to selected path.

Cognex Wrapper Application

Run Cognex Wrapper Application on your mobile device. In Server URL input box insert your server URL and in query string send screen and guestuser attributes. Click Navigate button and you will be redirected to your server URL (htttp://192.168.1.103/ITME81?screen=header&guestuser=1 in our case). When you click Navigate button application automatically add platform in query string as attribute.

Note that you must to run your ITME project before you navigate to server URL from Cognex Wrapper App.

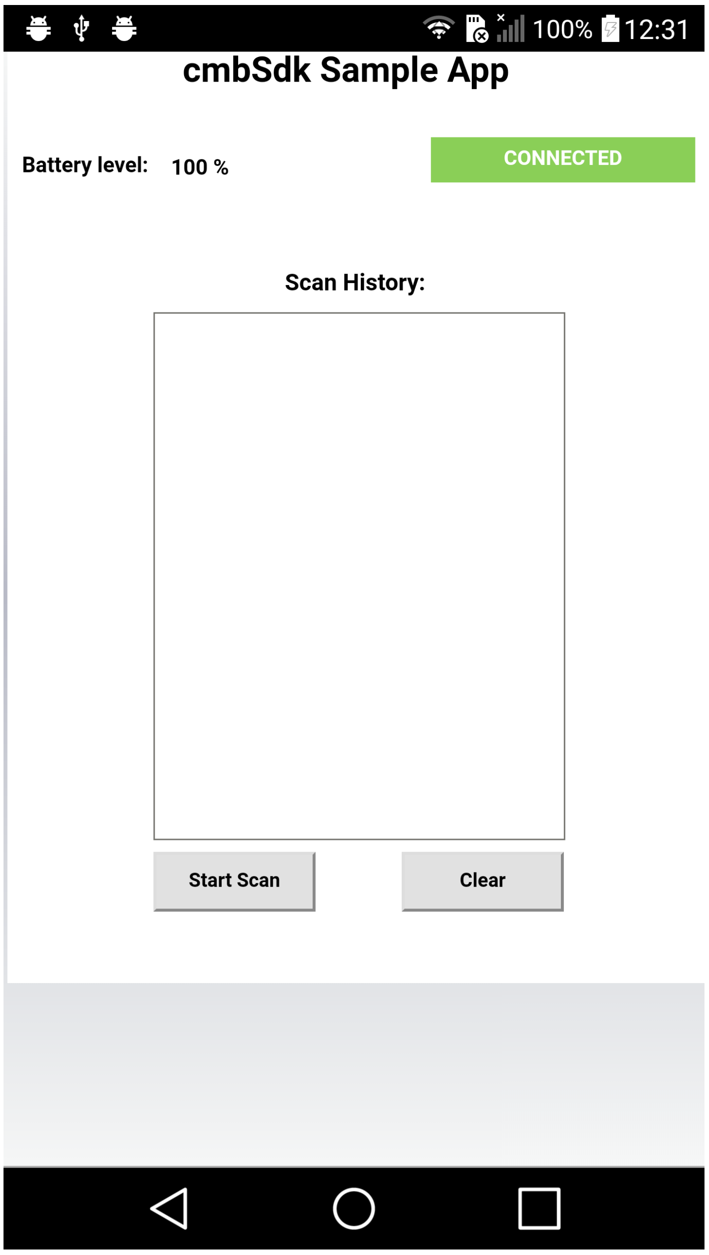


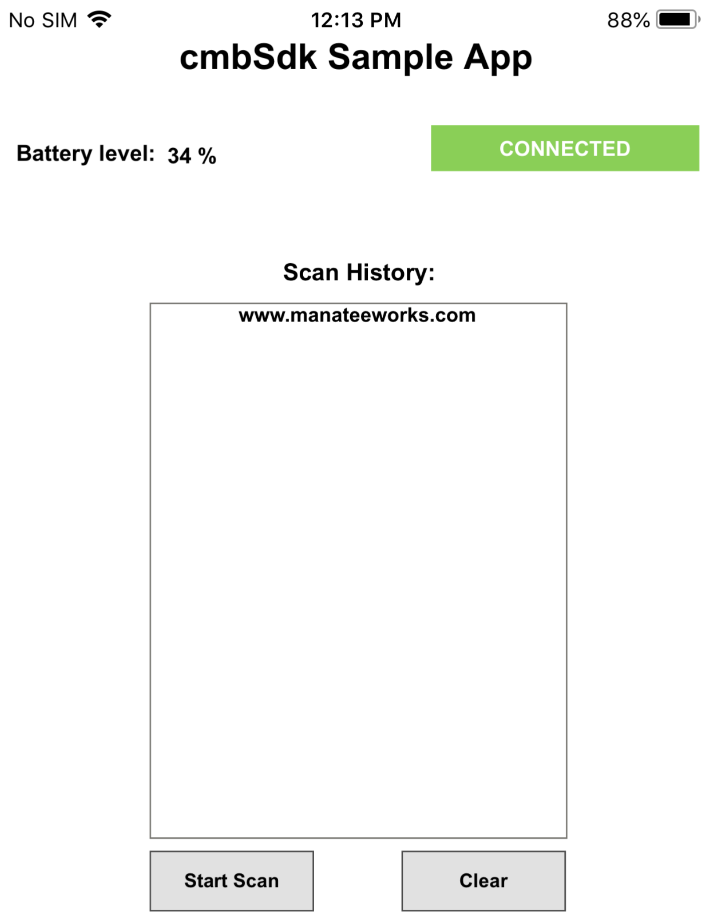
Select Cordova Plugins

Selected (9) | **Public (1)** | Kapsel (20)

Select Platform | cmbsdk-cordova

Plugin ID	Platform	Version	Description	Last Updated On	Actions
cmbsdk-cordova		1.2.14	CMB Scanner Cordova Plugin	6 Jun 2019	Add / View Details





Barcode Widget Properties

loadScannerEventTrigger

```
$loadScannerEventTrigger = "loadScanner(deviceType,sdk_key)"
```

With this property we call loadScanner(deviceType,sdk_key) method which is the first thing we need to do to in order to use reader device

Has two input parameters. First one should be 0 if we want to use MX Device to perform scanning or 1 if we will use Mobile Device for scanning. Second input parameter is sdk_key which is optional. We need to set our sdk license key only if we use Mobile Device for scanning barcodes. Otherwise we will have asterisks in barcode reader result.

Result from this event is returned in loadScannerOutputData property and loadScanner event is called as callback function

Example:

```
'For MX Device
$loadScannerEventTrigger = "loadScanner(0)"
'For Mobile Device
$loadScannerEventTrigger = "loadScanner(1, SDK_KEY)"
```

connectEventTrigger

$connectEventTrigger = "connect()"

With this property we call connect() method to connect with our reader device and should be called after we load reader device.

Result from this event is returned in connectOutputData property and connect event is called as callback function.

Example:

```
$connectEventTrigger = "connect()"
```

disconnectEventTrigger

$disconnectEventTrigger = "disconnect()"

With this property we call disconnect() method to release connection from reader device

Result from this event is returned in disconnectOutputData property and disconnect event is called as callback function.

Example:

```
$disconnectEventTrigger = "disconnect()"
```

setPreviewContainerPositionAndSizeEventTrigger

$setPreviewContainerPositionAndSizeEventTrigger = setPreviewContainerPositionAndSize(startPointX, startPointY, width, height)

With this property we call setPreviewContainerPositionAndSize(startPointX, startPointY, width, height) method which has 4 input parameters. startPointX, startPointY, width and height and they are measured in %.

This should be called before loadScanner method and we use it to place Mobile Device preview container.

Example:

```
'Preview Container positioned on 0,0 (left,top) 100% right and 30% bottom.
$setPreviewContainerPositionAndSizeEventTrigger = "setPreviewContainerPositionAndSize(0,0,100,30)"
```

toggleScannerEventTrigger

$toggleScannerEventTrigger = “toggleScanner()”

With this property we call toggleScanner() method to start/stop scanning process.

Example:

```
$toggleScannerEventTrigger = “toggleScanner()”
```

setSymbologyEnabledEventTrigger

$setSymbologyEnabledEventTrigger = setSymbologyEnabled(p1, p2, p3..)

To enable/disable symbologies we use this property which trigger setSymbologyEnabled(p1, p2, p3..) method. As input parameters we set symbology and status. We can enable/disable one or more symbologies in one call.

List of symbols: UNKNOWN, DATAMATRIX, QR, C128, UPC-EAN, C11, C39, C93, I2O5, CODABAR, EAN-UCC, PHARMACODE, MAXICODE, PDF417, MICROPDF417, DATABAR, POSTNET, PLANET, 4STATE-JAP, 4STATE-AUS, 4STATE-UPU, 4STATE-IMB, VERICODE, RPC, MSI, AZTECCODE, DOTCODE, C25, C39-CONVERT-TO-C32, OCR, 4STATE-RMC.

Result from this event is returned in setSymbologyEnabledOutputData property and setSymbologyEnabled event is called as callback function.

Example:

```
$setSymbologyEnabledEventTrigger = "setSymbologyEnabled(DataMatrix ON, C128 OFF)"
```

setLightsOnEventTrigger

$setLightsOnEventTrigger = “setLightsOn(p1)”

We can set light to be enabled/disabled by default when we start scanning with this property by triggering setLightsOn(p1) method. As input parameter we set ON if we want to enable and OFF if we want to disable light by default.

Result from this event is returned in setLightsOnOutputData property and setLightsOn event is called as callback function.

Example:

```
$setLightsOnEventTrigger = “setLightsOn(ON)”
```

isLightsOnEventTrigger

$isLightsOnEventTrigger = “isLightsOn()”

To check if light is enabled by default we trigger isLightsOn() method

Result from this event is returned in isLightsOnOutputData property and isLightsOn event is called as callback function.

Example:

```
$isLightsOnEventTrigger = “isLightsOn()”
```

sendCommandEventTrigger

```
$sendCommandEventTrigger = “sendCommand (p1, p2, p3…) "
```

With this property we call sendCommand(p1, p2, p3…) method that executes DMC commands which are set as input parameters. We can set one or more dmc commands as input parameters.

Result from this event is returned in sendCommandOutputData property and sendCommand event is called as callback function

Example:

```
$sendCommandEventTrigger = “sendCommand (GET BATTERY.CHARGE)”
```

connectionStateDidChangeOfReaderCallbackOutputData

Integer property that represent current reader connection state. There is four state:

0 - CONNECTION_STATE_DISCONNECTED

1 - CONNECTION_STATE_CONNECTING

2 - CONNECTION_STATE_CONNECTED

3 - CONNECTION_STATE_DISCONNECTING

resultCallbackOutputData

String property that contain last scanned result

availabilityCallbackOutputData

Boolean property that is true if our reader device is available or false if reader device is unavailable.

activeStartScanningCallbackOutputData

Boolean property which will be true when scanning is active or false when scanning is stopped.

loadScannerOutputData

When loadScanner method is executed it return success message or error message if reader device can’t be loaded.

connectOutputData

When connect method is executed it return true if connection is successful or error message if connection can’t be completed

disconnectOutputData

When disconnect method is executed it return success message or error message if there is problem while we execute this method

isSymbologyEnabledOutputData

In this property we return result from isSymbologyEnabled method. Result will be ON if certain symbology is enabled, OFF if is disabled or error message if there is some error thrown while we execute this method. Since isSymbologyEnabled method can have more than one parameter we will return symbology status separated with “,”. For example if we call $isSymbologyEnabledEventTrigger = "isSymbologyEnabled(DataMatrix, C128)" result will be “ON,ON” if both symbologies are enabled.

Note that by default if we use Mobile Device there is no symbologies enabled.

isLightsOnOutputData

Result from isLightsOn method that can be ON if light is enabled, OFF if is disabled or error message if something wrong happened while this command is executed

sendCommandOutputData

String property that represent result from sendCommand() method. If there is more than one DMC commands as input parameters result from every command will be separated with “,”.

For example, if we call $sendCommandEventTrigger = “sendCommand (GET BATTERY.CHARGE, GET LIGHT.INTERNAL-ENABLE)” result will be “50, OFF”

setLightsOnOutputData

Result from setLightsOn method that can be ON if light is enabled, OFF if is disabled or error message if something wrong happened while this command is executed

setSymbologyEnabledOutputData

In this property we return result from setSymbologyEnabled method. Result will be ON if certain symbology is enabled, OFF if is disabled or error message if there is some error thrown while we execute this method. Since setSymbologyEnabled method can have more than one parameter we will return symbology status separated with “,”. For example if we call $setSymbologyEnabledEventTrigger = "setSymbologyEnabled(DataMatrix, C128)" result will be “ON,ON” if both symbologies are enabled.

Barcode Widget Events

sendCommand

This callback event will be executed when sendCommand method is triggered and finished: $sendCommandEventTrigger = "sendCommand(p1, p2, p3 ...)"

isLightsOn

This callback event will be executed when isLightsOn method is triggered and finished: $isLightsOntEventTrigger = "isLightsOn()"

setLightsOn

This callback event will be executed when setLightsOn method is triggered and finished: $setLightsOnEventTrigger = "setLightsOn(p1)"

isSymbologyEnabled

This callback event will be executed when isSymbologyEnabled method is triggered and finished: $isSymbologyEnabledEventTrigger = "isSymbologyEnabled(p1,p2,p3,.....)"

setSymbologyEnabled

This callback event will be executed when setSymbologyEnabled method is triggered and finished: $setSymbologyEnabledTrigger = "setSymbologyEnabled(p1,p2,p3,.....)"

disconnect

This callback event will be executed when disconnect method is triggered and finished: $disconnectEventTrigger = "disconnect()"

connect

This callback event will be executed when connect method is triggered and finished: $connectEventTrigger = "connect()"

loadScanner

This callback event will be executed when loadScanner method is triggered and finished: $loadScannerEventTrigger = "loadScanner(0)"

setActiveStartScanningCallback

'This callback event will be executed when toggleScanner method is triggered and finished: $toggleScannerEventTrigger = "toggleScanner()"

setAvailabilityCallback

This callback event will be executed when availability of MX Device is changed.

setResultCallback

When barcode is scanned this callback event will be executed

setConnectionStateDidChangeOfReaderCallback

This callback event will be executed when connection state of reader device is changed