

MRZScanner Guide for Flutter

Installation guide

Please follow these simple steps to integrate our SDK into your project:

1. Adding our Flutter package

To add the `mrzscanner_flutter` package to your project, you have two options. You can either use the `mrzscanner_flutter` package from pub.dev or from a local path.

To use the package from `pub.dev`, add the package name and version to `pubspec.yaml`:

```
dependencies:
  flutter:
    sdk: flutter
  mrzscanner_flutter: <package version>
```

If you prefer to use a local package, download the package from <https://www.mrzscanner.com> and set the package path in `pubspec.yaml`:

```
dependencies:
  flutter:
    sdk: flutter
  mrzscanner_flutter: <package version>
    path: <package local path>
```

2. Run **flutter pub get** command in terminal to install the specified dependencies in your project

3. In your `main.dart` import the plugin:

```
import 'package:mrzscanner_flutter/mrzscanner_flutter.dart';
```

iOS

1. The MRZScanner plugin requires camera usage so you need to specify the camera usage description in iOS project's `Info.plist`:

```
<key>NSCameraUsageDescription</key>
<string>Camera is required for MRZ scanning </string>
```

API Methods

startScanner

Activates the scanner, with a default **SCANNER_TYPE_MRZ** if not specified.

```
MrzScanner.startScanner()
```

resumeScanning

Resume scanning after the scanner has been paused/stopped. Usually after a successful scan.

```
MrzScanner.resumeScanning()
```

closeScanner

Stop and close the scanner.

```
MrzScanner.closeScanner()
```

setScannerType

Specify which scanner type you want to use. There are two options: "MRZ Scanner" and "Document Image scanner".

- The "MRZ Scanner" option is used to scan for MRZ.
- The "Document image scanner" is used for capturing the front and back image of the ID documents.

@param int [ScannerType.MRZ, ScannerType.DOCUMENT_IMAGE_ID, ScannerType.DOCUMENT_IMAGE_PASSPORT, ScannerType.DRIVING_LICENCE, ScannerType.DOCUMENT_IMAGE_ID_FRONT, ScannerType.ID_SESSION]. The default value is ScannerType.MRZ

```
MrzScanner.setScannerType(ScannerType int)
```

captureIdImage

Returns a base64 string of the scanner id image.

```
MrzScanner.captureIdImage()
```

setDateFormat

- @return the currently set date format in which the parsed dates are formatted.
- Set the date format in which the parsed dates are formatted.
- @param dateFormat the pattern describing the date format. Example: "dd.MM.yyyy"

```
MrzScanner.setDateFormat(String dateFormat)
```

getSdkVersion

- @return the current SDK Version.

```
MrzScanner.sdkVersion(): String
```

setIDActive

Specify whether the scanner should detect and return result for IDs.

- @param isIDActive [true, false]. The default value is true.

```
MrzScanner.setIDActive(isIDActive: Boolean)
```

setPassportActive

Specify whether the scanner should detect and return result for passports.

- @param isPassportActive [true, false]. The default value is true.

```
MrzScanner.setPassportActive(isPassportActive: Boolean)
```

setVisaActive(isVisaActive: Boolean)

Specify whether the scanner should detect and return result for visas.

- @param isVisaActive [true, false]. Default value is true.

```
MrzScanner.setVisaActive(isVisaActive: Boolean)
```

setMaxThreads

Specify the maximum number of CPU threads that the scanner can use during the scanning process.

@param int [MaxThreads.ONE, MaxThreads.TWO, MaxThreads.THREE, MaxThreads.FOUR, MaxThreads.FIVE, MaxThreads.SIX]. The default value is MaxThreads.TWO

```
MrzScanner.setMaxThreads(maxThreads: Int)
```

setEffortLevel

Specify the effort level for the scanner to use during the scanning process.

@param int [EffortLevel.CASUAL, EffortLevel.TRY_HARDER, EffortLevel.SWEATY]. The default value is EffortLevel.TRY_HARDER

```
MrzScanner.setEffortLevel(effortLevel: Int)
```

toggleFlash

Trigger flashlight

```
MrzScanner.toggleFlash(active : Boolean)
```

scanFromGallery

Trigger an image picker, and scan from gallery

```
MrzScanner.scanFromGallery()
```

registerWithLicenceKey

Register with the licence key provided to remove the asterisks (*) from the result.

- @param key the provided licence key.
- @return 0 for success, -1 if registration failed.

```
MrzScanner.registerWithLicenseKey(key: String): Int
```

startScannerWithCustomOverlay

Customize the overlay of the finder in the scanner

@param base64String takes a base64 type string

```
MrzScanner.startScannerWithCustomOverlay(base64String : String)
```

startContinuousScanner

Initiates the scanner view with continuous scanning enabled. After a successful scan, the scanner will not be dismissed or paused. To close the scanner call `closeScanner()`. See `setIgnoreDuplicatesEnabled` for further continuous scanning behavior changes. Ignore duplicate is used to specify whether the scanner should ignore the successful scan if the document number is identical to the previous successful scan.

- @param boolean : ignoreDuplicate [true, false] Default is false.
- @param boolean: activeed [true, false] Default value is false.

```
MrzScanner.setContinuousScanningEnabled(boolean : activated,  
ignoreDuplicate : Bool)
```

setVibrateOnSuccessfulScan

Enables vibrate on successful scan option.

- @param boolean: activated [true, false] Default value is false.

```
MrzScanner.setVibrateOnSuccessfulScan(boolean : activated)
```

setExtractPortraitEnabled

Enable automatically extracting portrait image on successful scan; Default value: false.

@param:activated

```
MrzScanner.setExtractPortraitEnabled(activated);
```

setExtractFullPassportImageEnabled

Enable automatically extracting passport image on successful scan; Default value: false.

@param:activated.

```
MrzScanner.setExtractFullPassportImageEnabled(activated);
```

setExtractIdBackImageEnabled

Enable automatically extracting ID(back) image on successful scan; Default value: false.

@param:activated.

```
MrzScanner.setExtractIdBackImageEnabled(activated);
```

setPartialViewScanner

Set the scanning rectangle to limit the scanning area. The parameters' values are representing percentages of the scanning preview.

- @param x the Default value: top-left 5. point of the scanning rectangle. [0,...,100]
- @param y the Default value: top-left 20. point of the scanning rectangle. [0,...,100]
- @param x the Default value: top-left 5. point of the scanning rectangle. [0,...,100]
- @param x the top-left point of the scanning rectangle. [0,...,100] Default value: 5.

```
MrzScanner.startPartialViewScanner(int : x, int : y, int : width, int : height)
```

If you have any questions or concerns please visit our site at mrzscanner.com where you can [register](#) and send us a support request ticket at our Developer Portal, or feel free to contact us at contact@mrzscanner.com.

You can also find our online documentation [here](#)